



DBeaver User Guide v.24.1.ea

User Guide

Table of contents

[Getting started](#)

[Installation guide](#)

[How to import license](#)

[Application window overview](#)

[Basic operations](#)

Configure connection

[Guide to creating database connections](#)

[Disconnecting from database](#)

[Editing database connection](#)

[Invalidating and reconnecting to database](#)

[Local client configuration](#)

[Connection types](#)

[Multiple datasource connections](#)

[Configure connection initialization settings](#)

[Managing Master password](#)

[Security in DBeaver PRO](#)

Network configuration settings

[SSH configuration](#)

[SSL configuration](#)

[Proxy configuration](#)

[Kubernetes configuration](#)

[AWS SSM configuration](#)

System operations and security

[Shell commands](#)

[Changing current user password](#)

Databases authentication models

[Authentication models overview](#)

[Database native](#)

[DBeaver profile](#)

[Kerberos authentication](#)

[Microsoft Entra ID](#)

[Authentication MongoDB](#)

[PostgreSQL Pgpss](#)

[PostgreSQL SSPI \(Windows SSO\)](#)

[Oracle authentication models](#)

[Authentication Salesforce](#)

[Mysql two-factor authentication](#)

Cloud databases configuration

Cloud Explorer tools

[Cloud Explorer overview](#)

[AWS Cloud Explorer](#)

[Azure Cloud Explorer](#)

Cloud settings in DBeaver

[Managing AWS permissions](#)

[Working with AWS SSO](#)

[Working with Google Cloud Explorer SSO](#)

[AWS credentials](#)

[Google Cloud Explorer credentials](#)

Driver settings

[Database drivers](#)

[How to add additional artifacts to the driver](#)

[ODBC JDBC driver](#)

[Deprecated legacy ODBC driver](#)

[JDBC time zones](#)

[JDBC tracing](#)

Databases support

Classic

[Apache Hive](#)

[Cassandra](#)

[ClickHouse](#)

[Couchbase](#)

[Database driver IBM Db2](#)

[Greenplum](#)

[InfluxDB](#)

[Microsoft SQL Server](#)

[MongoDB](#)

[MongoDB authentication](#)

[MySQL](#)

[Mysql two-factor authentication](#)

[Netezza](#)

[Oracle](#)

[Oracle authentication models](#)

[PostgreSQL](#)

[Arrays](#)

[Extensions](#)

[Pgpass authentication](#)

[SSPI authentication \(Windows SSO\)](#)

[Redis](#)

[Salesforce](#)

[Salesforce authentication](#)

[SQLite](#)

[Teradata](#)

Cloud

AWS

[Athena](#)

[DocumentDB](#)

[DynamoDB](#)

[Keyspaces](#)

[Redshift](#)

[Timestream](#)

Azure

[CosmosDB](#)

[Databricks](#)

Google

[AlloyDB for PostgreSQL](#)

[BigQuery](#)

[Bigtable](#)

[Cloud SQL for MySQL](#)

[Cloud SQL for PostgreSQL](#)

[Cloud SQL for SQL Server](#)

[Firestore](#)

[Spanner](#)

[Snowflake](#)

Database Navigator panel

[Database Navigator overview](#)

[Simple and advanced View](#)

[Configuring Filters](#)

[Filter Database objects](#)

[Bookmarks](#)

Projects workspace

[Projects overview](#)

[Projects View](#)

[Project Explorer](#)

[Project security](#)

Editors in DBeaver

[Editors overview](#)

Data Editor tools

[Data Editor overview](#)

Viewing and editing data

[Data View and Format](#)

[Data Filters](#)

[Data viewing and editing](#)

[Data refresh](#)

[Navigation](#)

Data operations

[Managing Panels](#)

[Managing Data formats](#)

[Managing Charts](#)

Advanced data types

[Working with Spatial Geographic Information System \(GIS\) data](#)

[GIS examples](#)

[Working with XML and JSON](#)

[Virtual column expressions](#)

[Mock Data generation](#)

SQL Editor tools

[SQL Editor overview](#)

Query execution and management tools

[SQL Execution](#)

[Query execution plan](#)

[Script management](#)

Query development and assistance tools

[SQL Assist and Auto Complete](#)

[SQL templates](#)

[Visual Query builder](#)

[SQL Formatting](#)

[Variables panel](#)

[Spelling](#)

[SQL Terminal](#)

[AI Smart Assistance](#)

[Client side scripting](#)

[Export Command](#)

Entity Relation Diagrams

[ER Diagrams overview](#)

[Database Structure ER Diagrams](#)

[Custom ER diagrams](#)

[Edit mode for ER Diagrams](#)

[Properties Editor](#)

[SQL Code Editor](#)

[Dashboards](#)

Work with transactions in DBeaver

[Transaction log](#)

[Pending Transactions](#)

[Auto and manual commit modes](#)

[Query Manager](#)

Data import/export and operations

[Data transfer](#)

[Data import and replace](#)

[Data migration](#)

[Data transfer via email](#)

[Data transfer external storage](#)

[Data compare](#)

[Backup and restore](#)

Schema compare

[Schema compare overview](#)

[Using Liquibase in DBeaver](#)

Search

[Search overview](#)

[Data Search](#)

[Database full text Search](#)

[Database metadata Search](#)

[File Search](#)

Tasks

[Task management](#)

[Task Scheduler](#)

[Composite Tasks](#)

Integrated tools

[GIT integration](#)

[Tableau integration](#)

[Database debugger \(PostgreSQL\)](#)

[Optional extensions](#)

DBEaver configuration

User interface in DBEaver

[Accessibility](#)

[User interface themes](#)

[UI language](#)

[Toolbar customization](#)

Workspace

[Workspace location](#)

[Resetting your workspace](#)

[Keyboard shortcuts](#)

[Certificate management](#)

[Configuration files](#)

DBEaver editions

[Enterprise edition](#)

[Lite edition](#)

[Ultimate edition](#)

[Team edition](#)

[How to migrate to Team edition](#)

Tutorials

[New table creation](#)

[Incorporating triggers](#)

[Creating columns](#)

[Creating Indexes](#)

[Implementing constraints](#)

[Utilizing foreign keys](#)

[How to create a sample database](#)

[How to work with database Partitions](#)

[How to import connections from external tools](#)

Support and development

Admin preferences in DBeaver

[Admin manage preferences](#)

[Admin manage connections](#)

[Admin manage drivers](#)

[Admin preference restrictions](#)

[Admin variables](#)

[Command line](#)

[Session manager](#)

Troubleshooting in DBeaver

[Error Log](#)

[Background Tasks](#)

[Troubleshooting system issues](#)

[Making a thread dump](#)

[Posting issues](#)

[How to set a variable if dbeaver.ini is read only](#)

License administration

[License administration overview](#)

[How to import license](#)

[How to reassign license](#)

[Differences between license types](#)

[Windows Silent Install](#)

[Snap Installation](#)

[Statistics collection](#)

[Release cycles](#)

[Technical support information](#)

Developer tools

[Debug options](#)

[Build from sources](#)

[How to manage DBeaver snap package](#)

[Developing in Eclipse](#)

[3rd party dependencies](#)

[Developing in IDEA](#)

[Eclipse extensions](#)

[Working with extension SVG format](#)

[Extension office for Data Transfer](#)

[Importing CA certificates from your local Java into DBeaver](#)

[Contribute your code](#)

[Localization](#)

[Brazilian Portuguese standardization proposals](#)

[Unit tests](#)

Contacts

For technical support, feature suggestions, and any other questions, please use our [GitHub Issue tracker](#)

Getting started

Table of contents

[Introduction](#)

[First steps](#)

[DBeaver essentials](#)

[Additional resources](#)

Welcome to DBeaver, a powerful and user-friendly database management tool. This guide will help you get started with DBeaver, from installation to basic database operations.

Introduction

DBeaver is an open-source universal database tool that supports a wide variety of databases including SQL, NoSQL, and more. It provides an intuitive interface for database management, development, and analysis.

First steps

To ensure a smooth start with DBeaver, we have outlined the initial steps you need to take. These steps will guide you from download and installation to basic database operations, helping you to become familiar with DBeaver's environment and its powerful features.

- [Download DBeaver](#)

Please download DBeaver from the official website. Choose the appropriate version for your operating system

- [Install DBeaver](#)

Follow the instructions to install DBeaver on your operating system.

- [Import license](#)

If you have a PRO version, learn how to import your license to unlock the full capabilities of DBeaver.

- [Interface overview](#)

Get familiar with DBeaver's user interface to navigate through the tool efficiently.

- [Basic operations](#)

Walk through the basic steps for your first use of DBeaver.

DBeaver essentials

Refer to these resources for comprehensive support in your database operations.

- [Database Navigator](#)

Navigate through your database schemas and objects with ease.

- [Data Editor and Viewer](#)

Manage and manipulate your data directly within DBeaver.

- [Database objects](#)

Create, modify, and manage your database objects.

- [Run queries](#)

Start executing your SQL queries and scripts.

- [Create tables](#)

Define and create new tables within your database.

- [Database diagrams](#)

Visualize and design your database structure.

- [Schema comparison](#) and [Migration](#)

Compare and synchronize database schemas.

- [Export and Import](#)

Move data and objects in and out of your database.

- [Backup and Restore databases](#)

Protect and recover your database data efficiently.

Additional resources

Here are some resources for additional support and information:

- [DBeaver Blog](#)

Read about the latest developments, features, and insights from the DBeaver team on our official blog.

- [YouTube channel](#)

Watch tutorials, feature overviews, and other helpful videos to get the most out of DBeaver.

- [Twitter updates](#)

Follow DBeaver on Twitter for the latest news, updates, and tips.

- [Edition comparison](#)

Compare different DBeaver editions to find the one that fits your needs the best.

Installation guide

Table of contents

[Windows and MacOS Installer](#)

[ZIP Archive](#)

[Debian Package](#)

[RPM Package](#)

[Automatic updates](#)

The installation process depends on the distribution type and your Operational System.

Windows and MacOS Installer

The installer distribution is the recommended way to install DBeaver in Windows and MacOS X. It contains all required dependencies. In addition, the installer automatically upgrades DBeaver to the new version if a previous version has already been installed. To install DBeaver, run the installer executable and follow the instructions on its screens.

Note:

- The installer does not change any system settings or the Java installation.
- The included JDK will be accessible only to DBeaver.

ZIP Archive

When installing DBeaver manually, without using an installer:

1. Extract the contents of the archive.

NOTE: Do not unzip the archive over a previous DBeaver version. If you already have any version of DBeaver

extracted in the same location - remove it before unzipping the new version.

NOTE: All configurations, scripts and other necessary data are stored in a separate location (usually in the user's home directory) so the program deinstallation does not affect them.

2. Run the **dbeaver** executable.

Debian Package

To install DBeaver using a Debian package:

1. Run `sudo dpkg -i dbeaver-<version>.deb`.
2. Execute `dbeaver &`.

RPM Package

To install DBeaver using RPM package:

1. Run `sudo rpm -ivh dbeaver-<version>.rpm`.
2. Execute `dbeaver &`.

NOTE: To upgrade DBeaver to the next version, use `sudo rpm -Uvh dbeaver-<version>.rpm` parameter.

Automatic updates

This feature is available only in Windows and macOS.

From time to time, DBeaver automatically checks for new versions. If the check is not disabled and a new version is available, it will prompt you to decide whether you want to stay or upgrade. If the second option is chosen, the installer will be downloaded and launched upon completion. Note that DBeaver itself will be gracefully closed to avoid data loss.

How to import license

Table of contents

[Import from email](#)

[Import from the personal account](#)

[Insert the License key into the License Manager](#)

[Import of Monthly Subscription license](#)

[License extension](#)

[License Manager](#)

To start using commercial versions of DBeaver you can

- request a [Trial license](#) for 2 weeks;
- request an [Academic license](#) if you are a student or a teacher;
- [buy](#) a monthly or annual subscription.

After purchasing the DBeaver license or getting the Trial/Academic license, you will receive a License text by email. It will also be available in your personal account on our [site](#). This License text will contain your License ID e. g. DB-841MRZHY-ZH54, the start date and license owner's name and company name. It is very important to import your License correctly.

Import from email

You can just copy-paste the License Key to import the license into the License Manager. Please note that you need to copy-paste the full license text (not just the license ID). The license text starts with "-" and ends with "==" characters.

Hello,

Thank you for your interest to DBeaver.

We are glad to see you among DBeaver users. Please, find your license below:

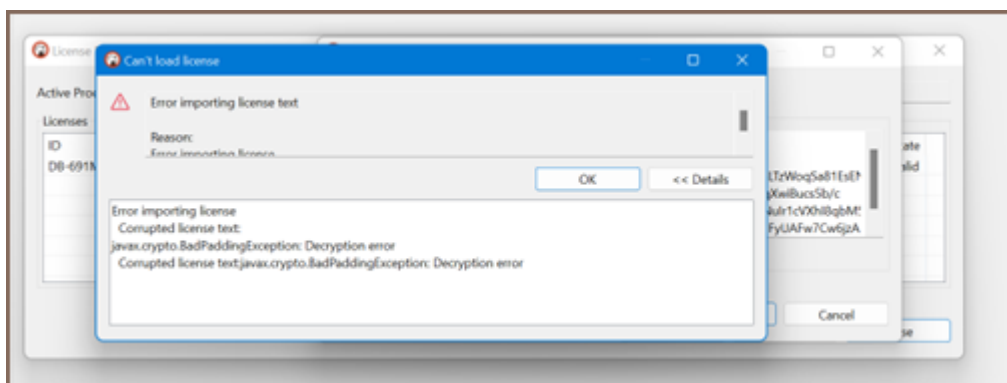
```
-- DBeaver EE LICENSE - DB-841MRZHD-ZH53
-- Starts at Tue Oct 11 10:14:33 UTC 2022 to Louis Anderson //
PK2uJ9eCXU4Zmgd1wvMufQyC2Jk+cSTtePOXewaUoS2/jYvuaX1opFJ0G/H81sLTzWogSa81EsEN
BG3Z6TJ0VFPko5bvXkm2r0hRXjRusa18gM47snPK/NaH/7fg/CEzHhy3+NrstRtRqXwiBucS5b/c
lRgIZm3Z4B5qg8G11V8K2gG99p8K0e690gTB6vWY1hECRfs3vdrz4x5QHnQrNuIr1cVXhI8qbM5
13L9/7e3QgBMQya+XF9yVeGiVRQgDQDP8GNwY1WMoHBzpGKuhm1twBj+jF7FyUAFw7Cw6jzAk0es
gb3Fg2REPkLwvIq2yxangUpnNGDUE1bH7VXoSw==
```

Enjoy your database research with DBeaver and good luck!

Regards,

DBeaver team

Sometimes an email client can corrupt the formatting of the License Key that can cause an error.



Therefore, you need to import your License Key from your personal account on our site <https://dbeaver.com/>.

Import from the personal account

Firstly, you need to [Sign in](#).

Secondly, you should open the **Licenses** tab, where you can find all your licenses.

Ben Johnson

ORDERS

LICENSES

ACCOUNT












TICKETS

Licenses

Assignment

Search licenses...

Rows Per Page: 10 | 20 | 50 | 100

<input type="checkbox"/>	License ID 	Users 	Product 	Type 	End Support Date 	
			All 	All 	All 	
<input type="checkbox"/>	DB-9Q1MSZGF-ZDAM	5	DBeaver Enterprise Edition	Yearly subscription	Dec 6, 2024	
<input type="checkbox"/>	DB-9Q1MSZGC-ZDAL	1	DBeaver Enterprise Edition	Trial	Dec 21, 2023	
<input type="checkbox"/>	DB-9Q1MSZGB-ZDAK	1	DBeaver Enterprise Edition	Monthly subscription	Jan 7, 2024	

Bulk actions:

EXTEND

GET LICENSE KEYS

EXPORT INFO

Export all licenses info as csv

To open License details and copy the license key text click the license ID link. Here you can find your license status, type, maintenance period, and end support date.

Ben Johnson

[ORDERS](#)[LICENSES](#)[ACCOUNT](#)[TICKETS](#)

License ID:	DB-9Q1MSZGF-ZDAM
License Type:	Yearly subscription
Maintenance Period:	365 days
Maximum users:	5
Issue Time:	Dec 7, 2023
Expiration Time:	Dec 6, 2024
License Status:	Valid
License customer:	Ben Johnson, Test

License Keys:

DBeaver EE

-- DBeaver EE LICENSE - DB-9Q1MSZGF-ZDAM
-- Starts at 2023-12-07 09:35:25.0 to Test / Ben Johnson //
DB-9Q1MSZGF-ZDAM-2023-12-07 09:35:25.0 to Test / Ben Johnson //
DB-9Q1MSZGF-ZDAM-2023-12-07 09:35:25.0 to Test / Ben Johnson //
DB-9Q1MSZGF-ZDAM-2023-12-07 09:35:25.0 to Test / Ben Johnson //
DB-9Q1MSZGF-ZDAM-2023-12-07 09:35:25.0 to Test / Ben Johnson //

DOWNLOAD LICENSE

COPY TO CLIPBOARD

At the bottom of the page you can find the License Key required to start using DBeaver. There are two options how to copy your License Key from the personal account:

1) Press the **COPY TO CLIPBOARD** button, then press OK. The license text will be copied to the clipboard.



2) Press the **DOWNLOAD LICENSE** button, then press OK.

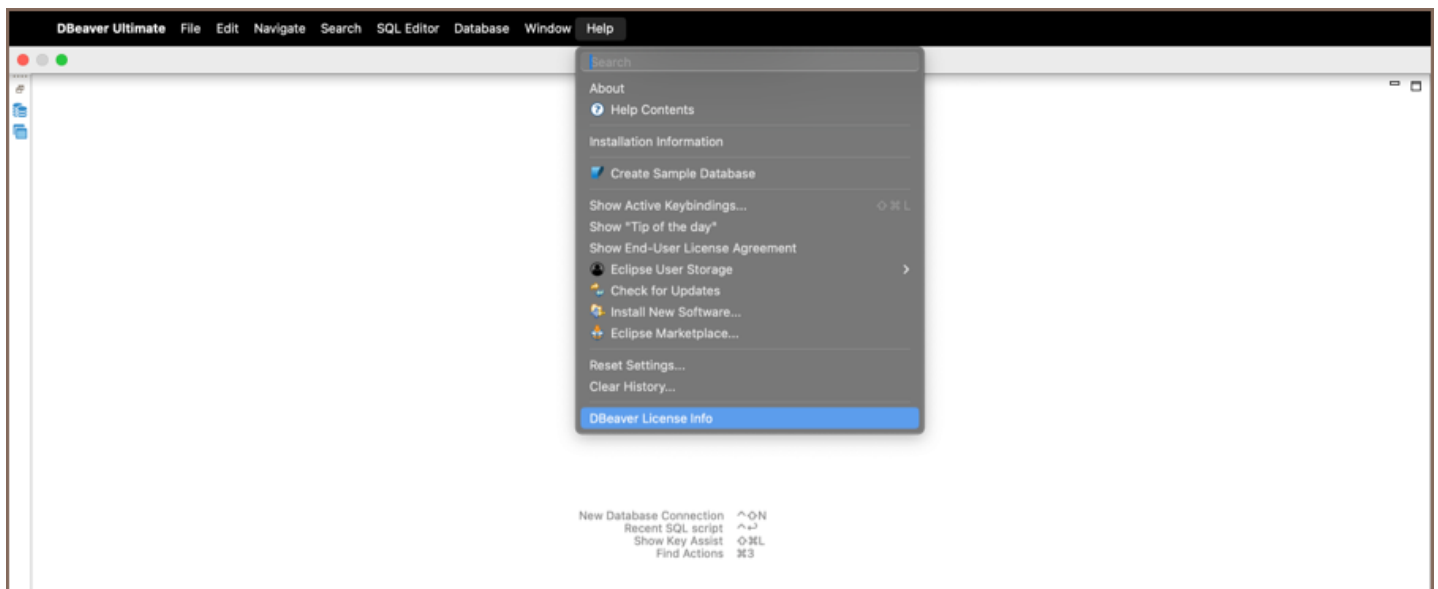
.txt file with your License Key will be downloaded to your download folder. The file name is License ID, e. g. DB-841MRZHY-ZH54.



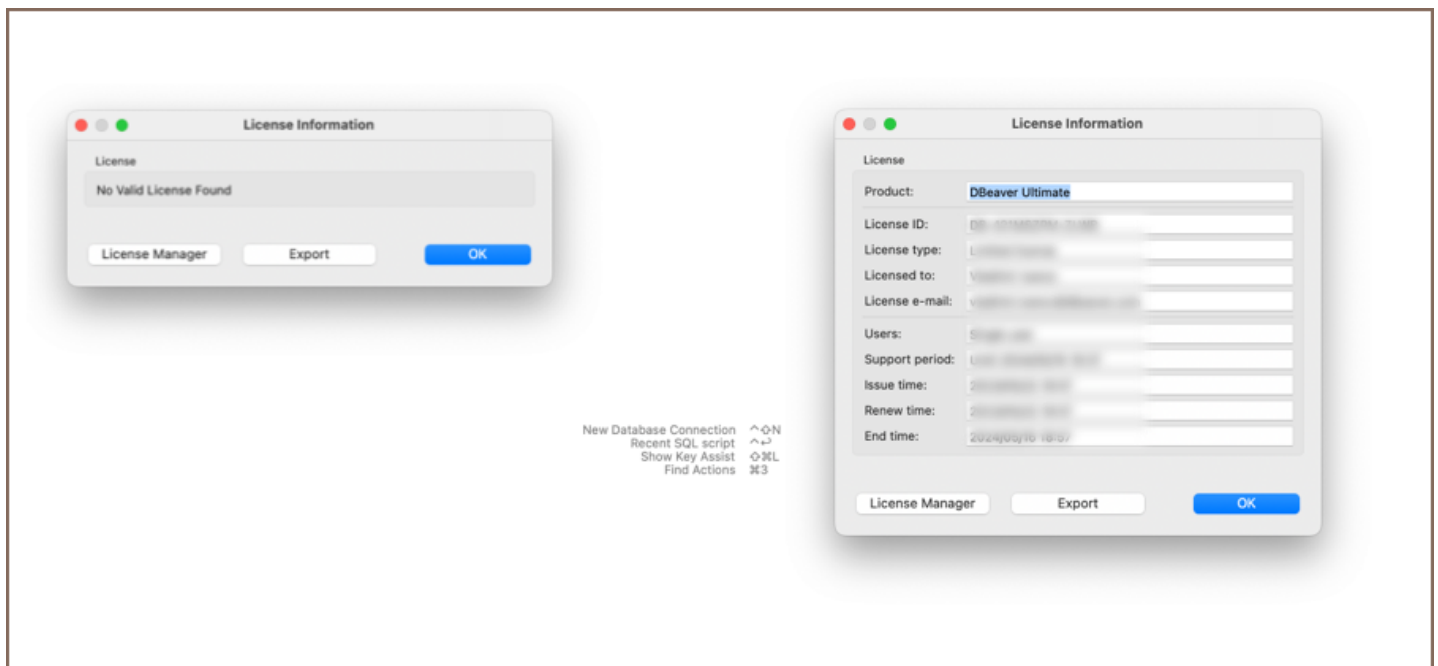
Then you need to insert the copied License Key to License Manager in DBEaver.

Insert the License key into the License Manager

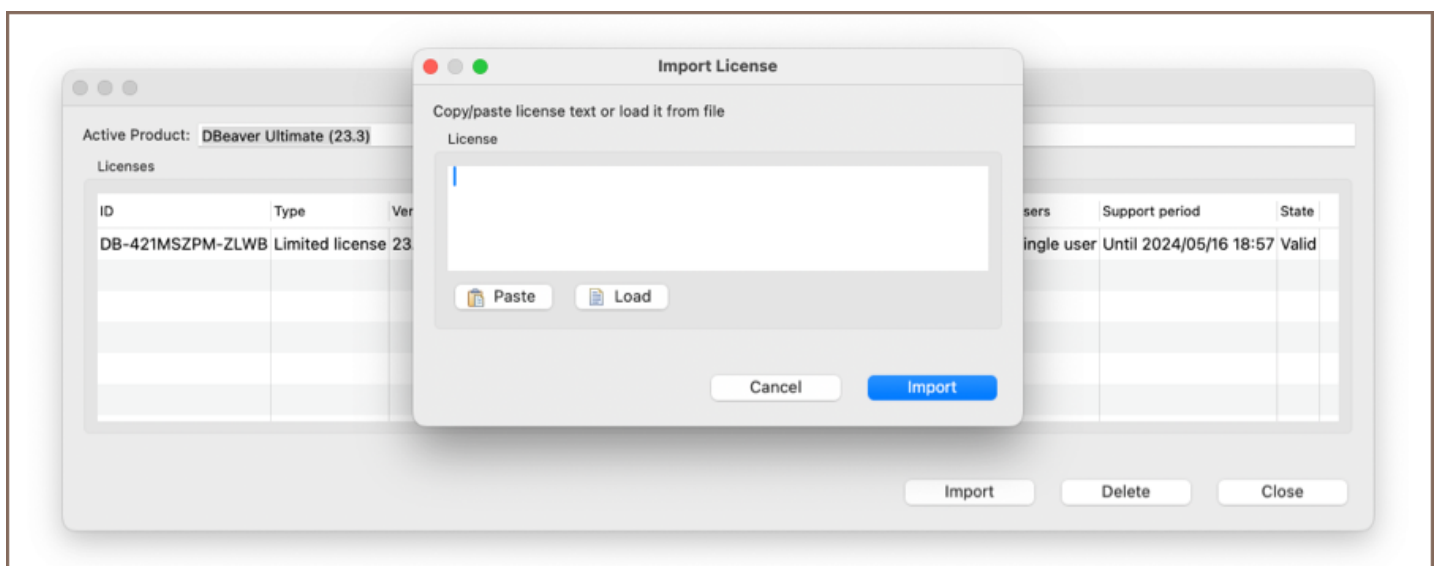
To start using commercial versions of DBEaver with your License Key you need to open License Manager in DBEaver: Help -> DBEaver License Info



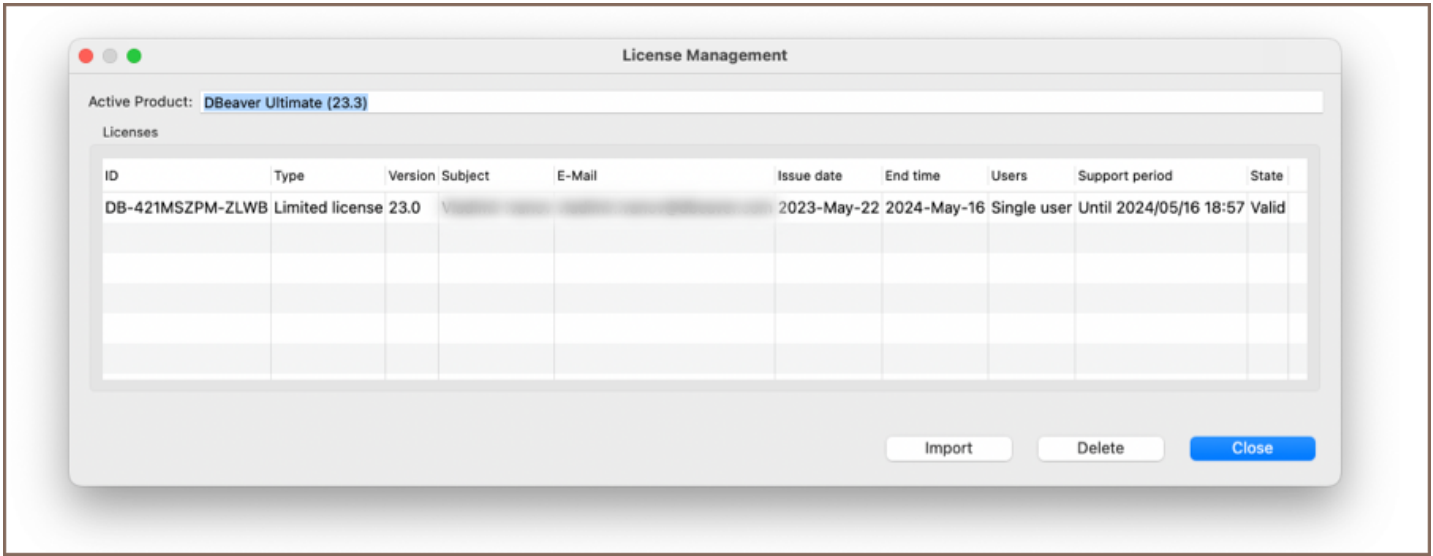
The License information window can look different depending on whether you already have a valid license or not.



Then you open the License Manager and press the Import button to paste your License Key.



If you copied the License Key to the clipboard, press the Paste button and then Import. If you downloaded a .txt file with the License Key, press the Load button and then select the file from Downloads. The License Key will be pasted automatically. Then press the Import button and your license will be added to the License Manager. You have successfully imported your license.



You have successfully imported your license. Now you can close the License Manager and start using DBeaver.

Import of Monthly Subscription license

A monthly subscription requires internet access on the workstation for the first activation and each prolongation.

If you do not have an active internet connection or work behind a corporate firewall while importing the Subscription license, the following error can appear:

```
Invalid subscription
Can't find the subscription information for license 'DB-841MRZHY-ZH54'.
Check your internet connection and/or firewall settings and restart the application.
```

In this case, you need to check that DBeaver has internet access, or you will need to configure your firewall.

License extension

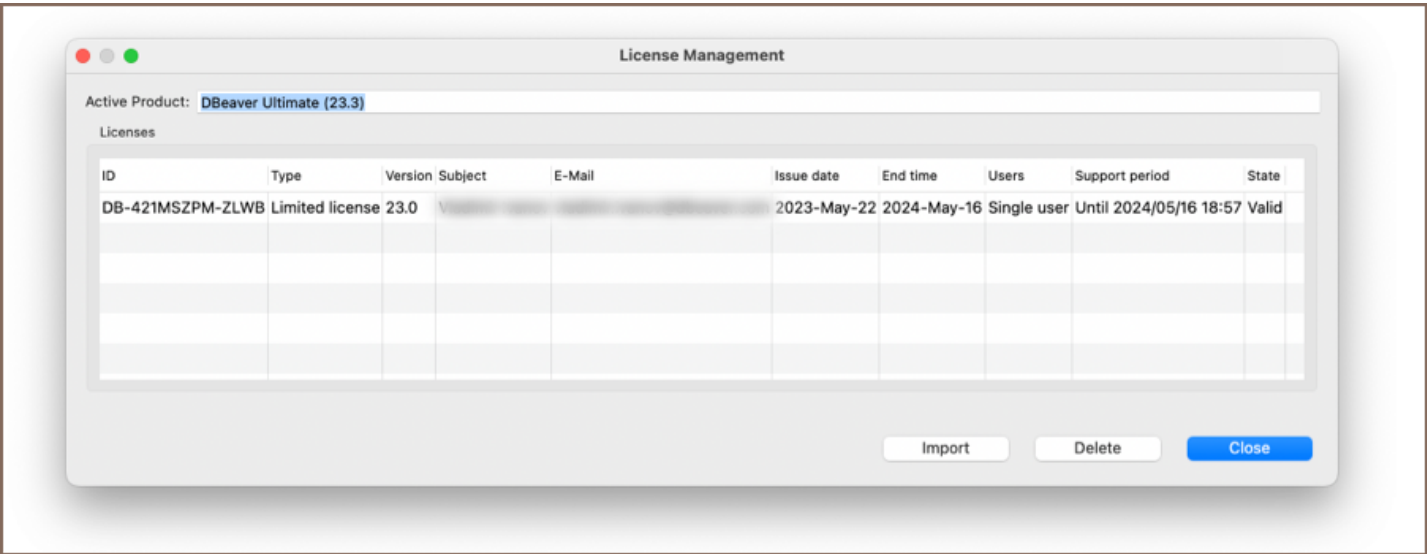
After the perpetual license expires, customers can purchase a license extension on the website.

If the customer buys the DBeaver license extension and DBeaver has internet access, the license in DBeaver will be updated automatically. Otherwise, the customer must again import the license key from the personal account.

License Manager

License Manager provides you with the following information about your licenses:

- License ID e. g. `DB-841MRZHY-ZH54` ;
- License type: Trial/Academic/Subscription/Standard;
- Version;
- License owner's name and company name;
- License owner's email;
- Issue time is the date of the initial license purchase;
- The end time is the date the license expires;
- Number of users: single user or multiuser for group licenses;
- The support period is the period you have access to the internal support system on the site and the possibility to download new product versions;
- State: valid or expired.

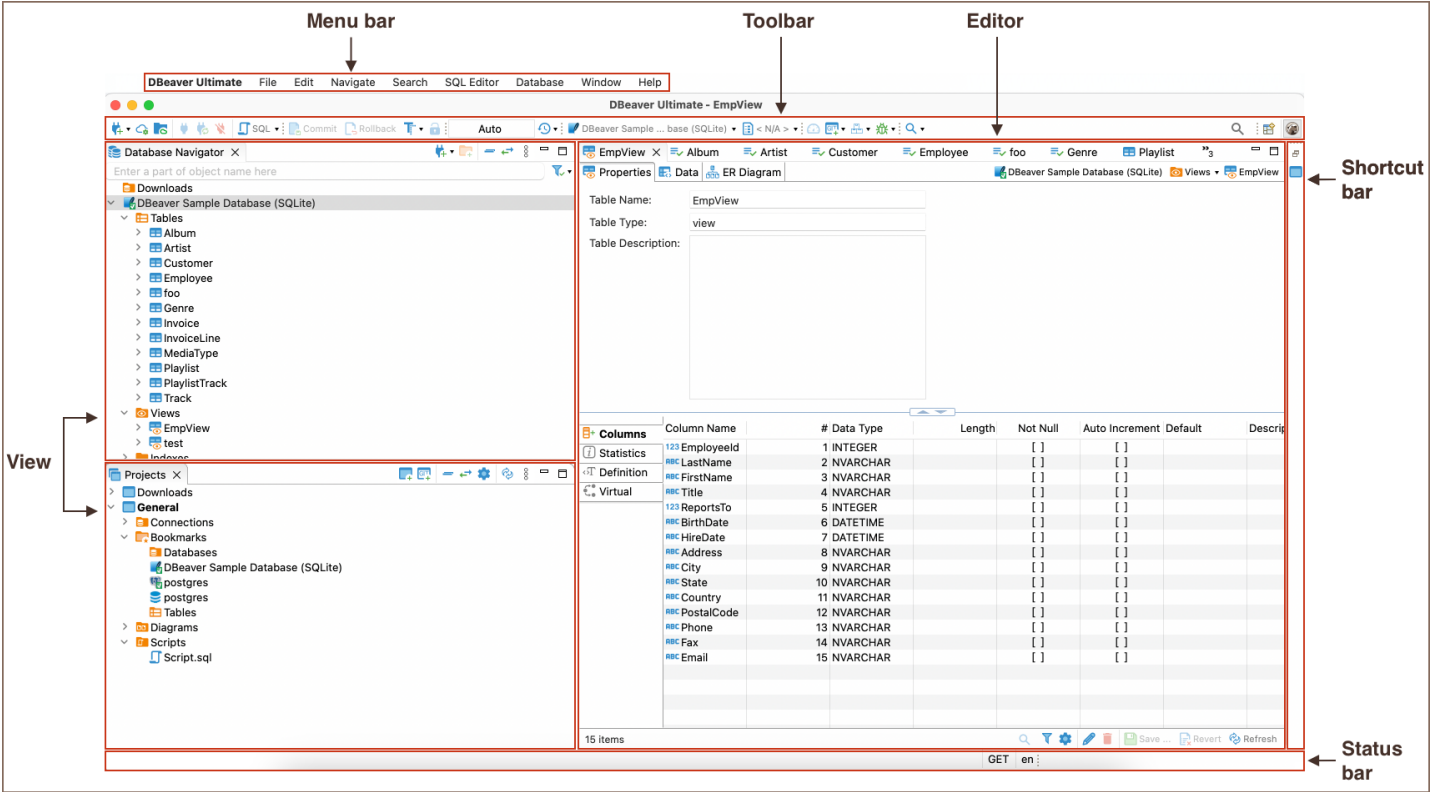


Application window overview

Table of contents

- [Menu Bar](#)
- [Toolbar](#)
- [Shortcut Bar](#)
- [Workspace](#)
 - [Changing Workspace Layout](#)
 - [Managing View and Editors](#)

The DBeaver window contains a menu bar, a toolbar, a workspace with one or more editors and views, and a status bar:



Menu Bar

By default, the menu bar contains the following menus:

- The **File** menu contains menu items for creating files, folders, [projects](#), [database connections](#), database projects, [ER diagrams](#), and Import and Export items.
- The **Edit** menu contains global commands like Cut, Copy, Paste, Delete and [Add Bookmark](#) targeted at the active element.
- The **Navigate** menu allows navigation through scripts and database objects.
- The **Search** menu provides options to [search](#) among files database objects and across data.
- The **SQL Editor** menu is for opening [SQL Editor](#) and managing its appearance.
- The **Database** menu allows the management of database drivers, connections, and transactions and reconnecting to and disconnecting from a database.
- The **Window** menu includes items to manage the look of DBeaver window: show/hide and minimize/maximize views and editors, display bars, split editors, and manage other preferences.
- The **Help** menu contains links to information and help resources, and menu items to check the version number and availability of updates.

You can customize the menu bar and the list of menu items to display; for this, go to **Window -> Customize Perspective -> Menu Visibility** tab.

Toolbar

The toolbar contains buttons for the most basic and frequently used commands:



Some buttons are enabled (colored) and others are disabled (grey). The sets of enabled and disabled buttons change depending on which editor is active in the workspace. Only enabled buttons apply to the active view or editor.

Tip: The toolbar can be customized. For more information, please consult [our customization guide](#).

Shortcut Bar

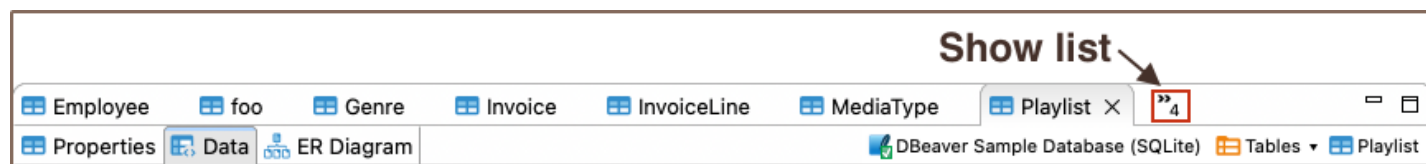
There are two shortcut bars - one on the left and one on the right side of the workspace zone. Shortcut bars host shortcuts of views and editors and appear if at least one view or editor is minimized, otherwise, they are hidden.

Workspace

Views are windows within the workspace that provide presentations and ways to navigate the information. For more information about particular views, see the [Views](#) article.

Editors are windows in which you can interact with the content of files and databases. For more information about particular editors, see the [Editors](#) article.

Both views and editors can appear as separate windows or tabs stacked with other views/editors in a tabbed window. The following image shows the title bar of a tabbed window. They become hidden if tabs do not fit in the title bar of a tabbed window. To see the list of hidden tabs, click the Show List icon that also indicates their number:



Several views and editors can be opened simultaneously in the workspace, but only one can be active at a time.

You can change the layout of the workspace by opening and closing views, docking them in different positions in the workspace, collapsing them to the shortcut bar, or expanding them to occupy the whole workspace and restoring them to the latest docked position.

Changing Workspace Layout

You can move views and editors around the workspace and dock them in different positions:

- As a tab in a tabbed window
- As a separate window with a vertical or horizontal layout in any zone of the workspace

You can also swap locations of two views or editors.

To dock a view to a position in the workspace, press and hold the view's title bar, then drag and drop it onto the desired position.

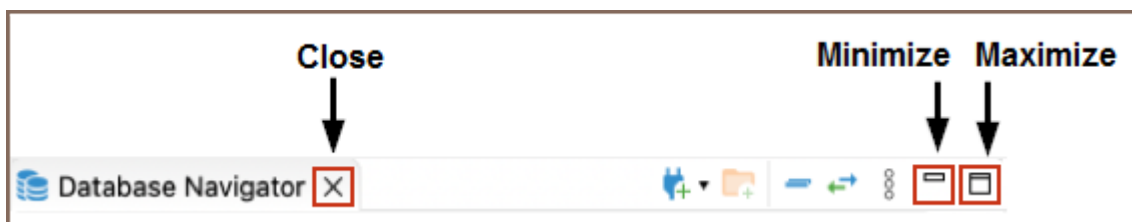
You can resize the view and editor windows. To resize, place the cursor on the window's border until it changes to a double-ended arrow, then click and drag the edge to the needed size.

To close a view or editor, click the Close button, or right-click the title bar of the view / editor, then click one of the options on the context menu (they change depending on the configuration of windows):

- **Close** - to close the active window or tab in a tabbed window.
- **Close Others** (for editors and views that appear as tabs in tabbed windows) - to close all tabs of the current tabbed window except the active tab.
- **Close Tabs to the Right / Left** (also for tabbed windows) - to close all tabs of the current tabbed window that are located to the right / left of the active tab.
- **Close All** - to close all tabs of a tabbed window (close the window).

Managing View and Editors

All views and editors have the **Close**, **Minimize** and **Maximize** buttons:



The Maximize button changes to the Restore button when a view or editor is maximized.

To maximize a view or editor to the size of the whole workspace, do one of the following:

- Click the Maximize button in the upper-right corner of the view.
- Double-click the title bar of the view or editor.
- On the **Window** menu, click **Appearance** -> **Maximize Active View or Editor**.

When one view is maximized, other views and editors appear as shortcuts on the shortcut bar.

To restore a maximized view or editor to its latest docked position, double-click its title bar or the **Restore** button in its upper-right corner.

When you minimize a view, it wraps into a shortcut on the shortcut bar:



The shortcuts of views and editors may appear on the left or the right shortcut bar, depending on the latest docked position of the view or editor.

To minimize a view, do one of the following:

- Click the **Minimize** button in the upper-right corner of the view.
- On the Window menu, click **Appearance** -> **Minimize Active View or Editor**.

To restore a minimized view or editor to its previous position, click the Restore button on its shortcut in the shortcut bar. To restore a minimized view or editor to a new position, click the view / editor name button under the restore button.

Basic operations

Table of contents

[Overview](#)

[First launch](#)

[Database navigator](#)

[Object editor](#)

[Basic operations](#)

[Object creation](#)

[Working with data in DBeaver UI](#)

[Additional Resources](#)

[Executing SQL queries](#)

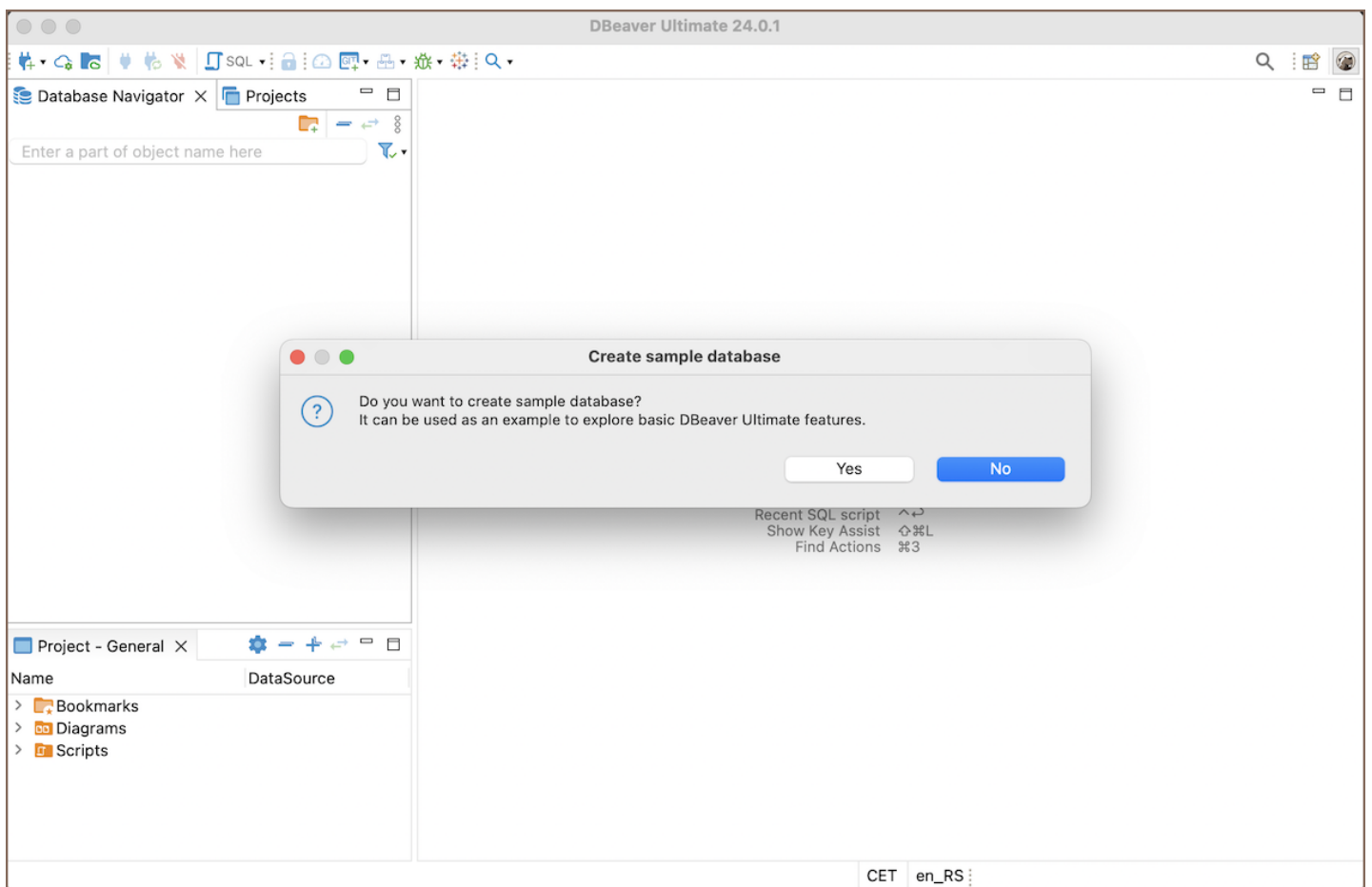
Overview

This guide outlines the essential steps to quickly get started with DBeaver.

To begin working with this guide, it is essential that you install the appropriate version of DBeaver for your operating system. Please download it from the official [DBeaver website](#).

First launch

When you launch DBeaver for the first time, you will encounter several initial setup windows designed to enhance your experience. This section will guide you through what to expect and how to interact with these prompts.



1. **Create Sample Database:** The first window you may see is the **Create Sample Database** prompt. For those new to database management, or if you simply want to explore DBeaver's functionalities without starting from zero, this option is beneficial. It provides a pre-populated database with various data and objects, allowing you to practice and experiment without the need to populate a new database from scratch. For more detailed information, refer to our article on [Sample Database](#).
2. **Tip of the Day:** Another feature you will come across is the **Tip of the Day**. This is a helpful tool for learning new tips and tricks about using DBeaver. If you prefer not to see these tips at startup, you can disable this feature by unchecking the **Show tips on startup** checkbox.
3. **Statistics Collection window:** Lastly, you may see a window related to Statistics Collection. This feature helps you understand how you use DBeaver, which can enhance your user experience. For a more detailed exploration of this feature, please see our article on [Statistics Collection](#).

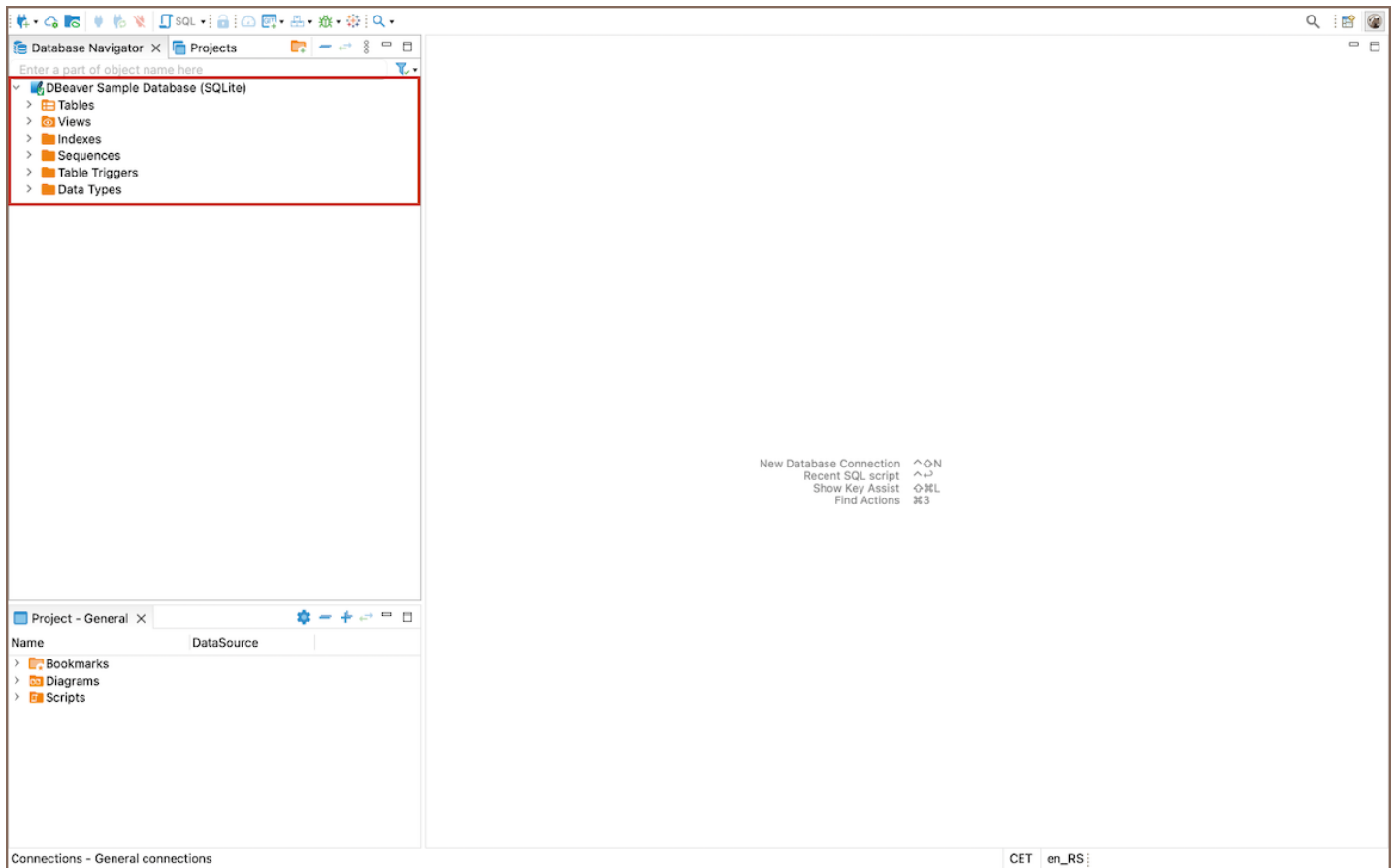
Tip: For a detailed understanding of DBeaver interface, refer to the [Application Window Overview](#) article.

Database navigator

If you opt to create a sample database, you will see it listed in the Database Navigator.

The Database Navigator is the central hub for accessing and managing your databases within DBeaver. It allows you to view and interact with the metadata objects of your databases.

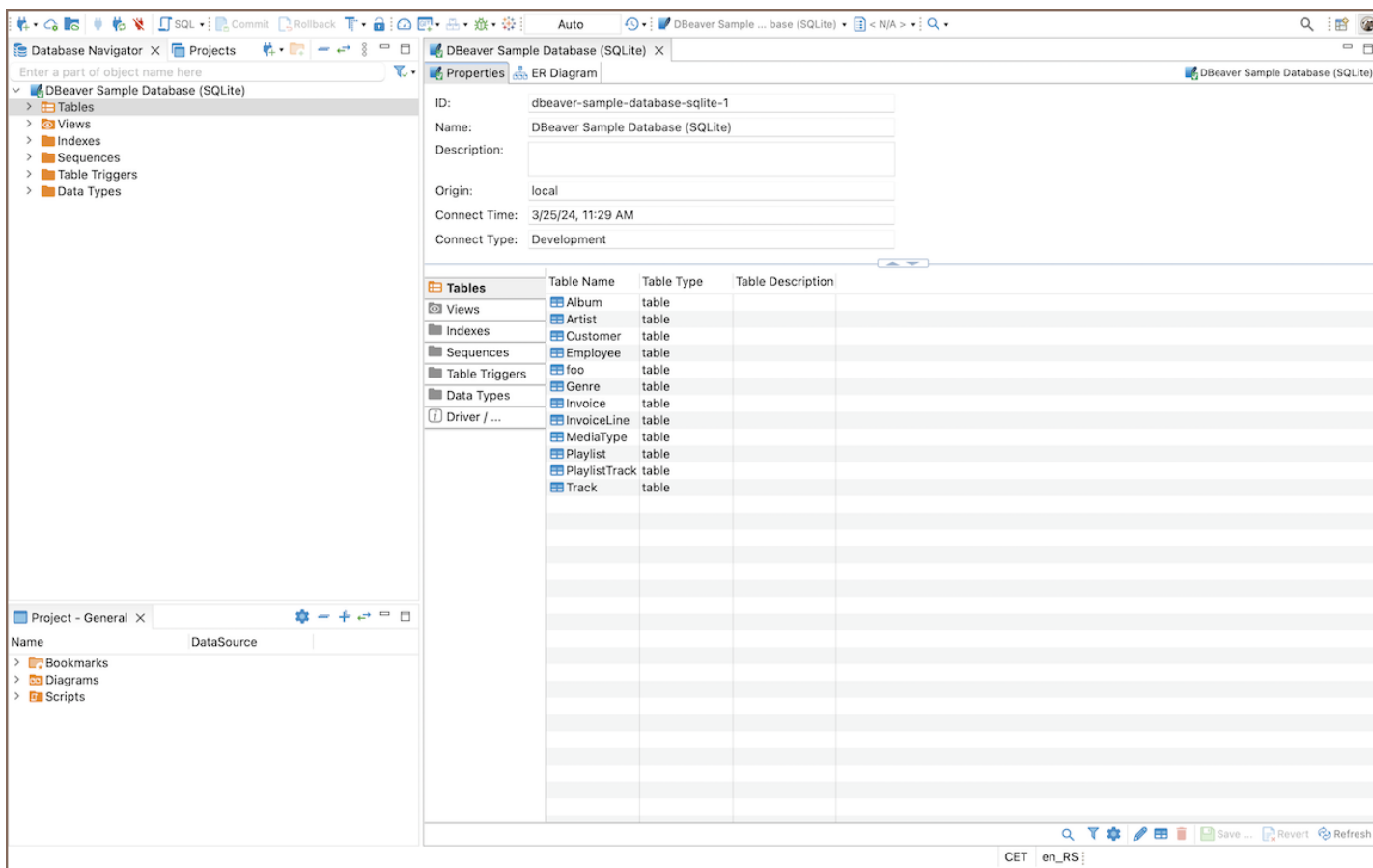
To explore these objects, click on the arrow located next to the database you have created. This action will expand the list of metadata objects available in your database instance.



Tip: For an extensive understanding of the Database Navigator's capabilities, refer to the [Database Navigator](#) article.

Object editor

From the Database Navigator, you can access the Object Editor, where you have the capability to examine database objects in detail. To view the contents of a table, or to delve into any database object, simply double-click on the desired item. This action will open the Object Editor, providing a comprehensive view and interaction options for the selected object.



Tip: You can also access the Editor using the shortcut (F4). For a comprehensive list of shortcuts and their functions, see the article on [Shortcuts](#).

The Object Editor presents a structured environment where you can access different aspects of your database objects through specialized tabs:

Tab	Description	Reference Article
Properties	Displays the properties of the database object and its sub-entities.	Properties Editor
Data	Opens the Data Editor for tables and views, facilitating data management.	Data Editor
ER Diagram	Shows Entity-Relationship Diagrams for tables and schemas, visualizing object relationships.	ER Diagrams

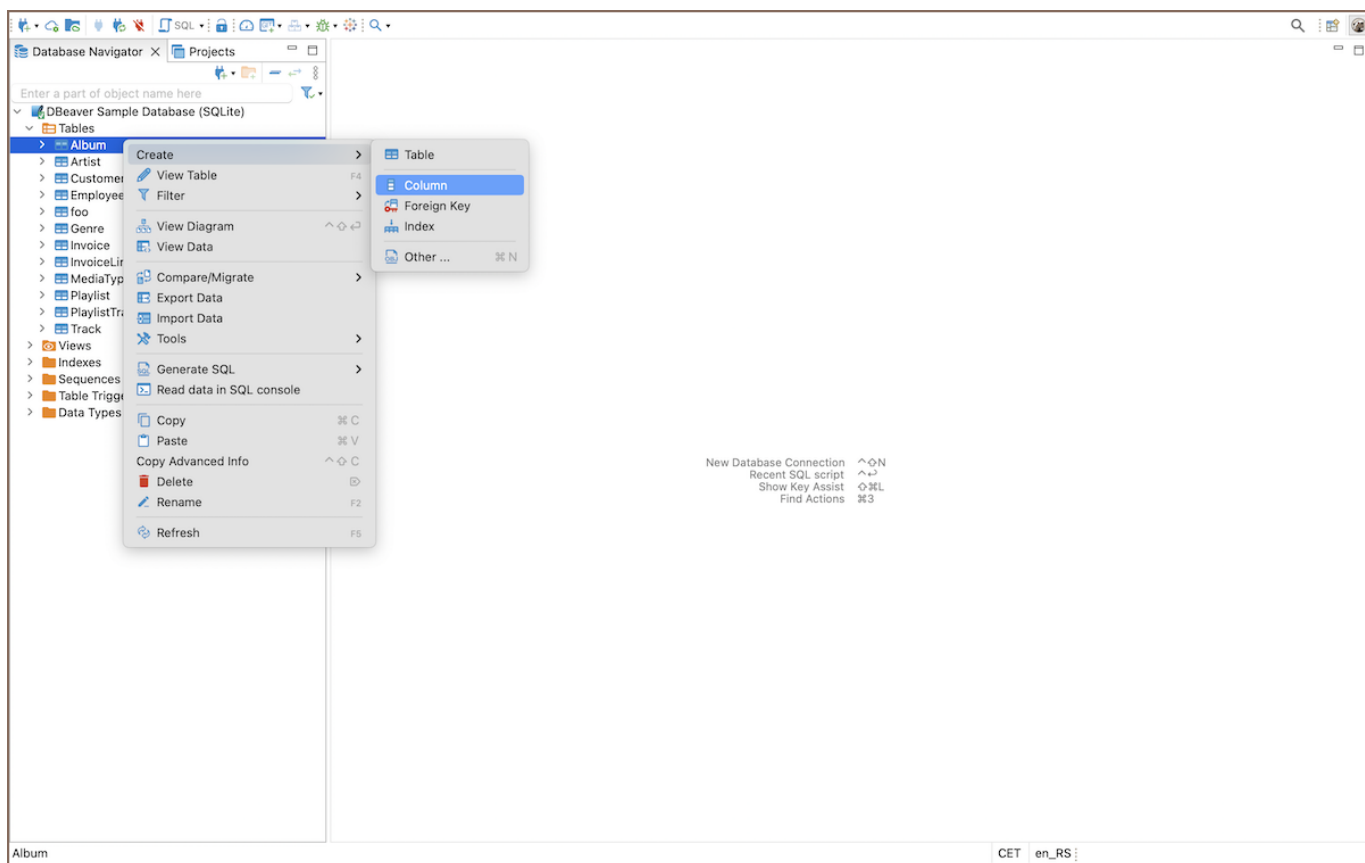
Basic operations

Now, let's explore how to work with data. This section will guide you through the essential processes for working with database objects.

Object creation

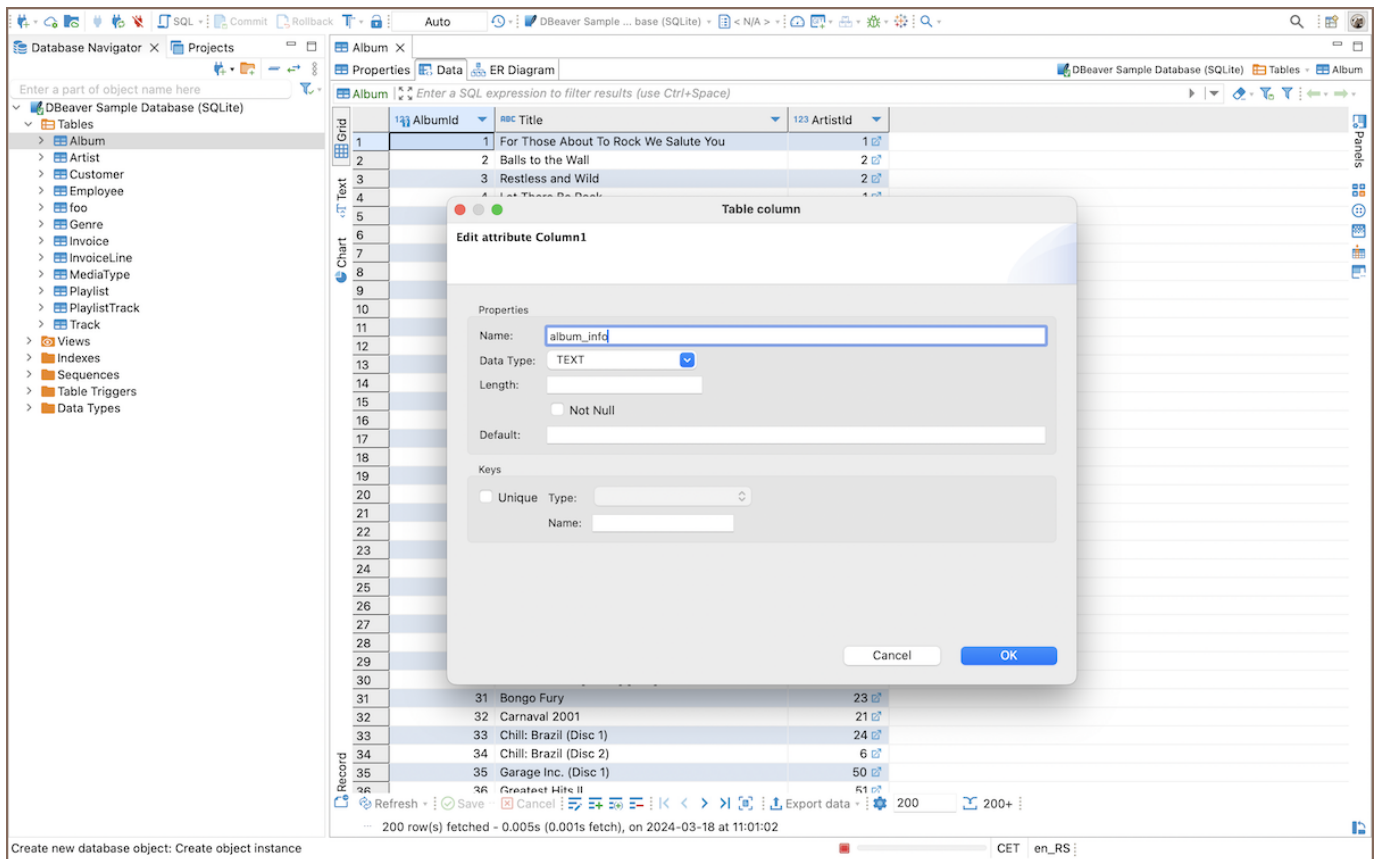
Creating new objects within your database is a fundamental skill in database management. This process allows you to add structures like tables, views, or schemas, which are essential for storing and organizing your data effectively.


1. In the **Database Navigator**, locate your database.
2. **Create new object:** Depending on the type of object you wish to create, you might need to navigate to different levels within your database connection.



- First, in the **Database Navigator**, navigate to the level where your new object should be created. This could be at the connection level (for databases and schemas), within a specific database or schema (for tables, views, etc.), or within a table (for columns, foreign keys, indexes, etc.).
- Right-click on the node (such as a database, schema, table, etc.) where the new object belongs, and select **Create**. A submenu will appear, listing the types of objects you can create at this level.
- From the submenu, choose the type of object you want to create (e.g., Database, Schema, Table, Column, View, Foreign Key, Index, etc.). If necessary, a dialog will appear, prompting you to enter the details for the new object, such as name, data type, constraints, and other relevant properties.

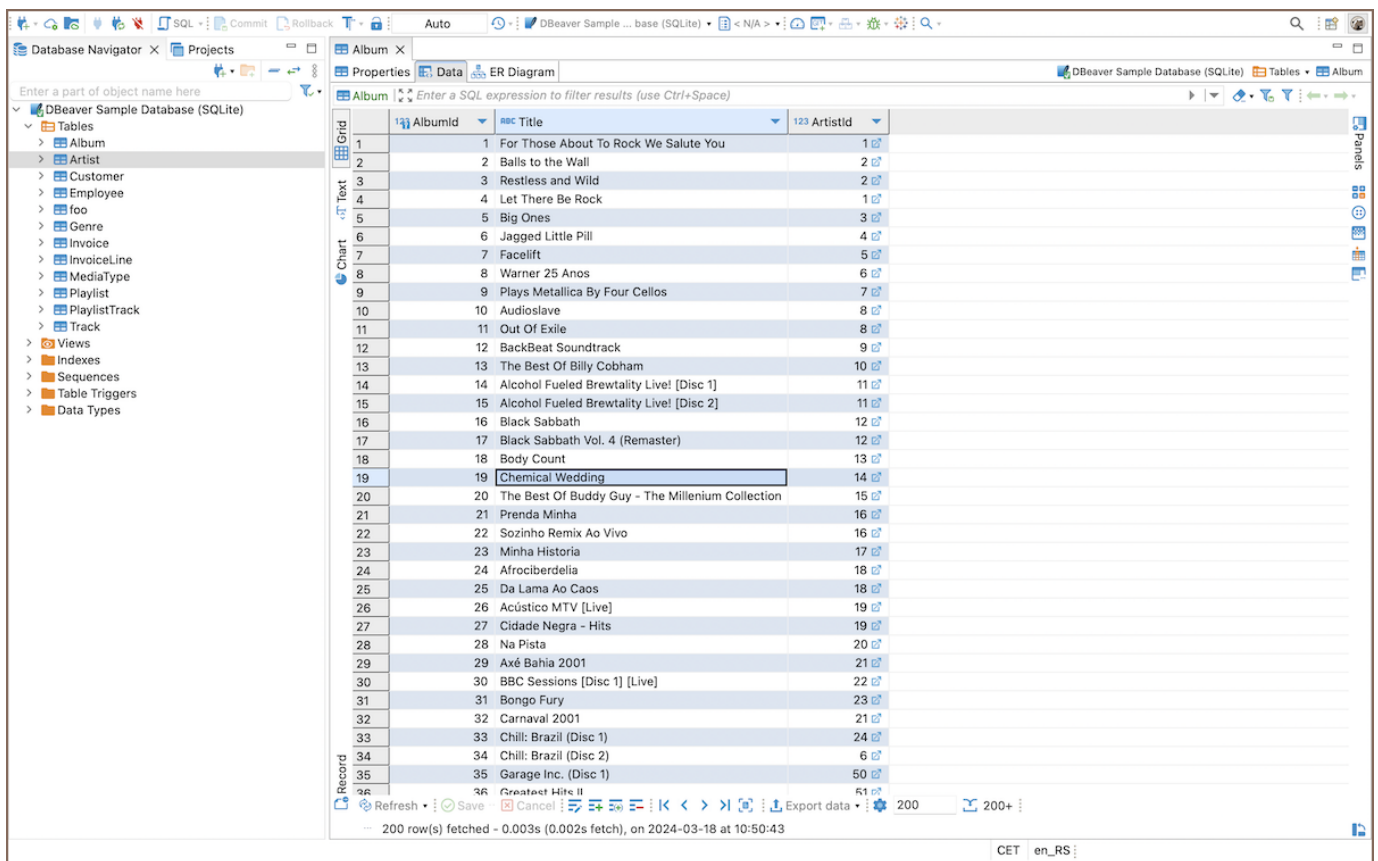
3. **Fill in the details:** Follow the on-screen instructions to complete the creation of the new object. Some objects may require additional steps or information, such as defining columns for a new table or specifying the reference table and columns for a foreign key.




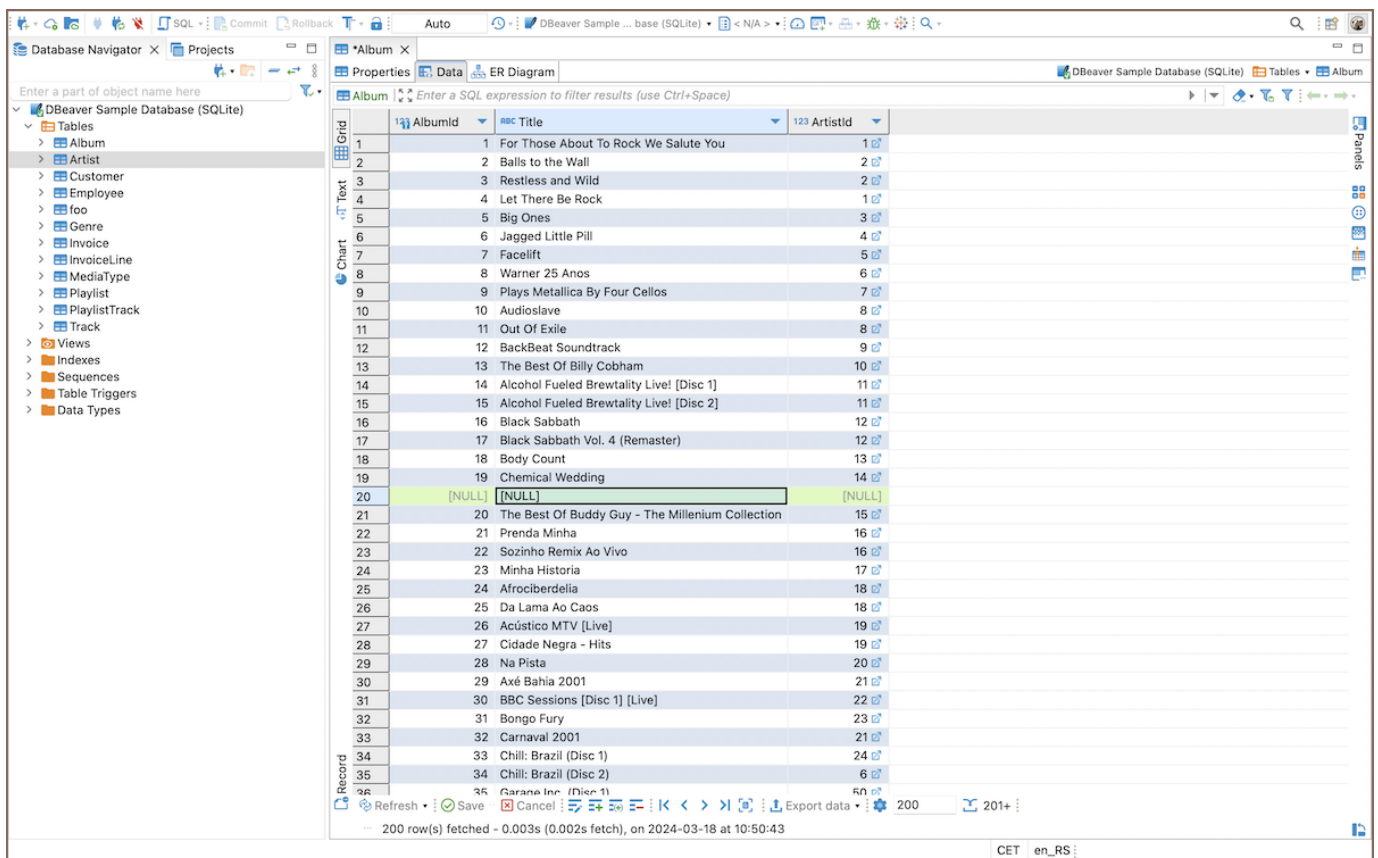
4. **Save:** To complete the creation of the object, click the **Save** button  **Save**.


Working with data in DBeaver UI

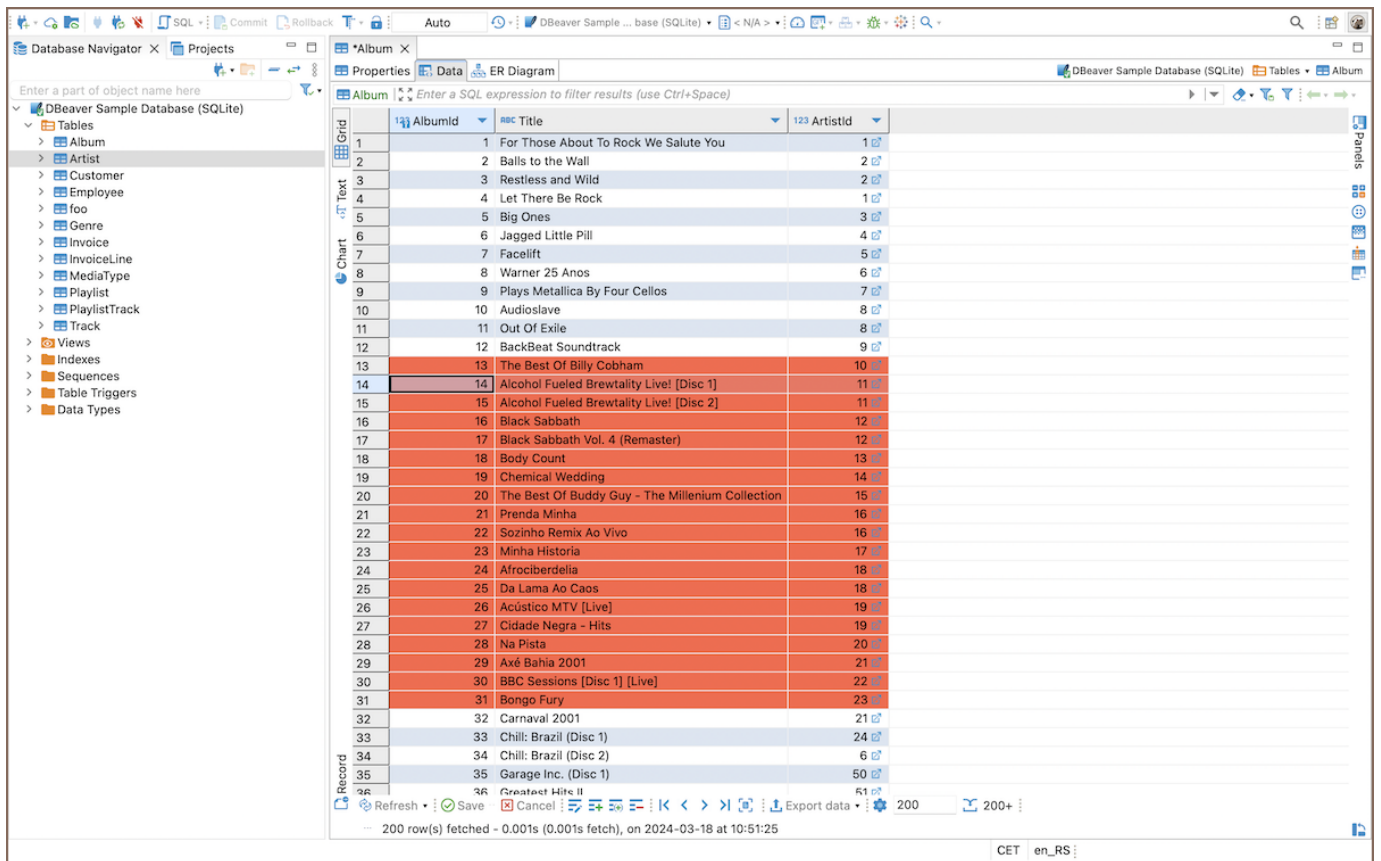
1. **View and Edit Data:** Navigate to your table in the **Database Navigator**, right-click, and select **View Table -> Data** to see the table's contents. Here, you can directly edit the data by typing in new values into the cells, and then save your changes.



2. **Inserting new data:** Use the **Add Row** button  in the table data view to add new records. Fill out the new row with your data and click **Save**.



3. **Deleting data:** Select the rows you want to remove and click the **Delete current row** button . Confirm the deletion and save your changes.



Additional Resources

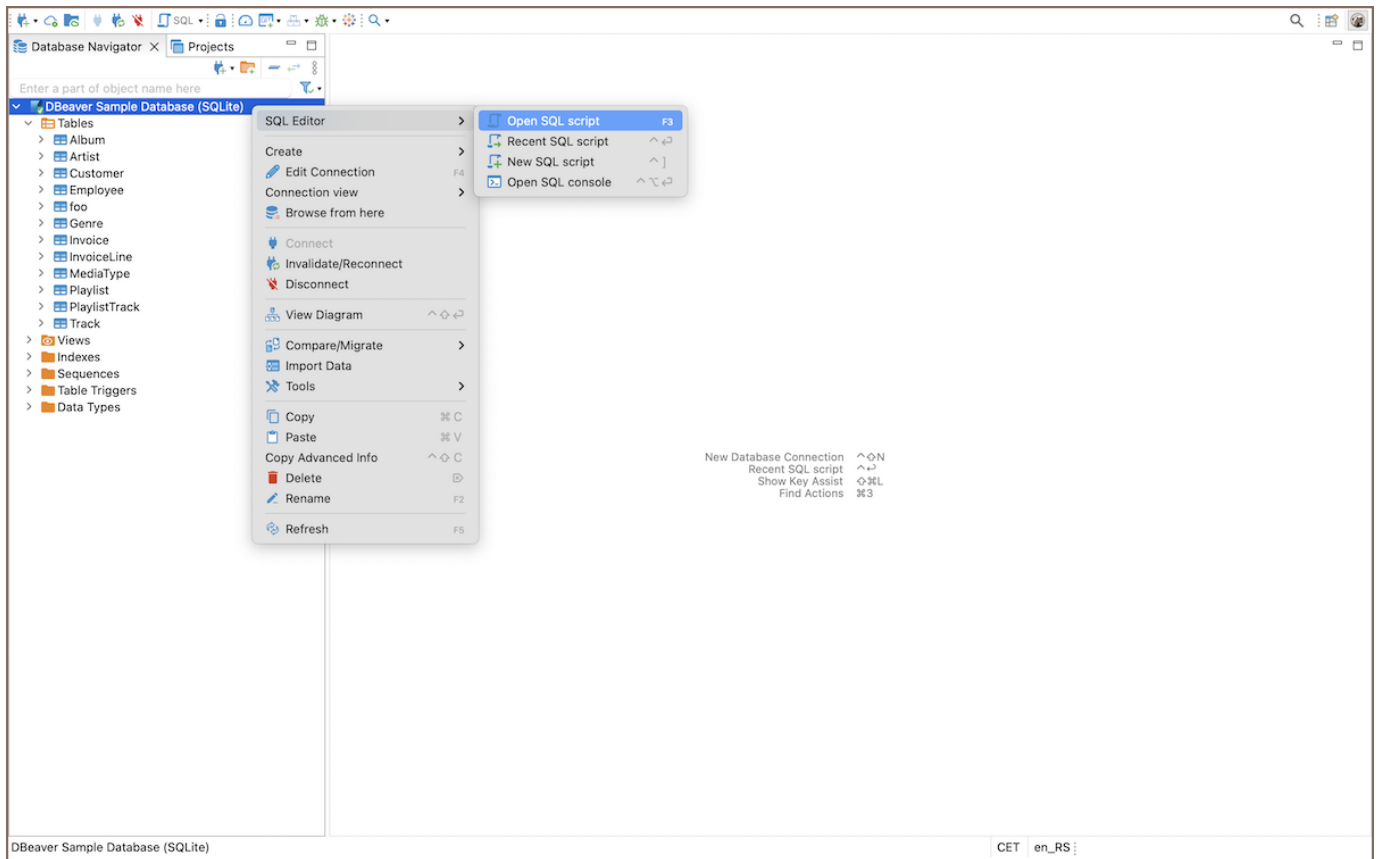
Dive deeper into specific data management topics with these detailed guides:

- [New Table Creation](#)
- [Creating Columns](#)
- [Implementing Constraints](#)
- [Utilizing Foreign Keys](#)
- [Creating Indexes](#)
- [Incorporating Triggers](#)

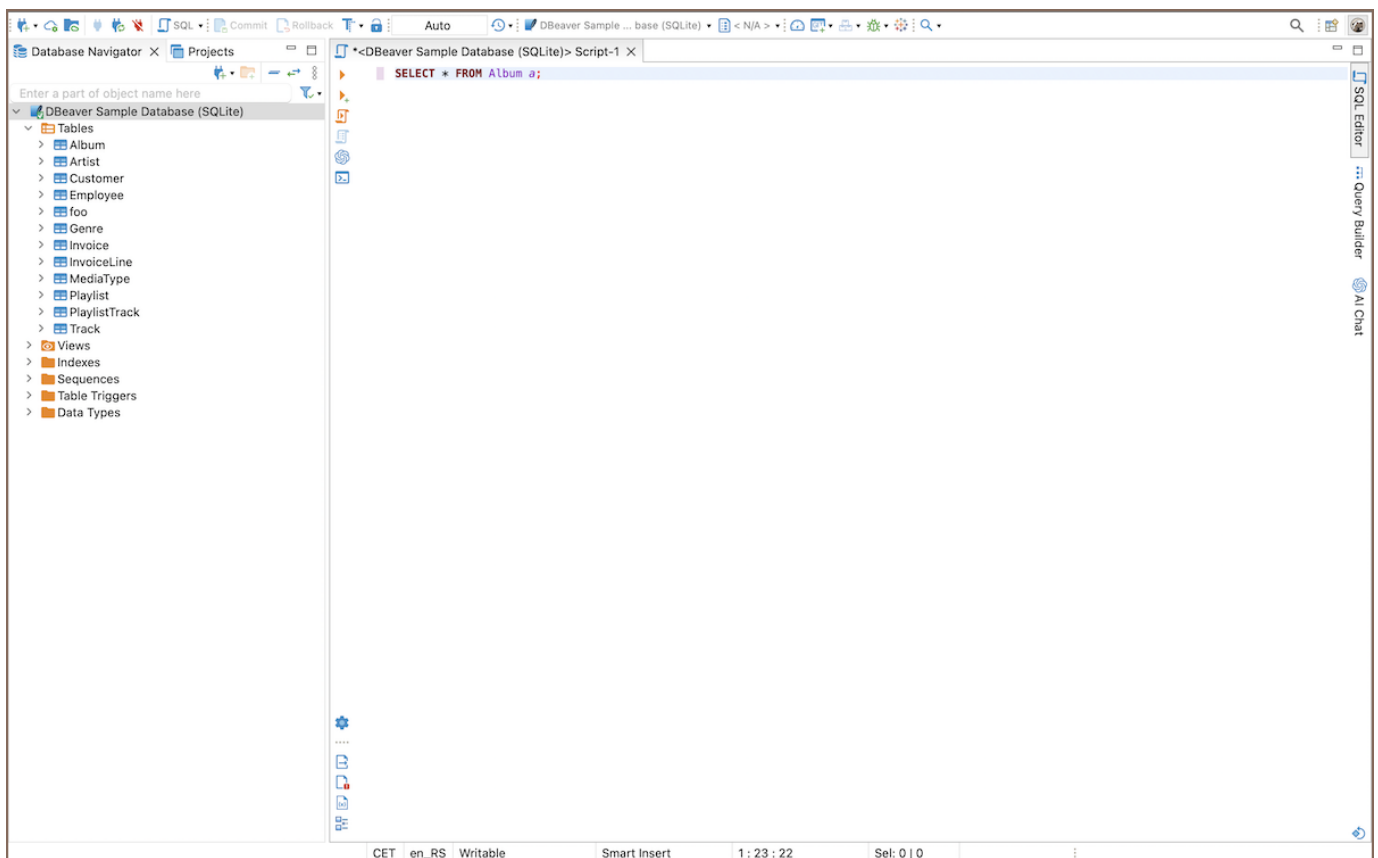
For more on data management, view our tutorials on [YouTube](#).

Executing SQL queries

1. **Open SQL Editor:** Navigate to the database or schema in the [Database Navigator](#), right-click, and select **SQL Editor -> Open SQL Script**.



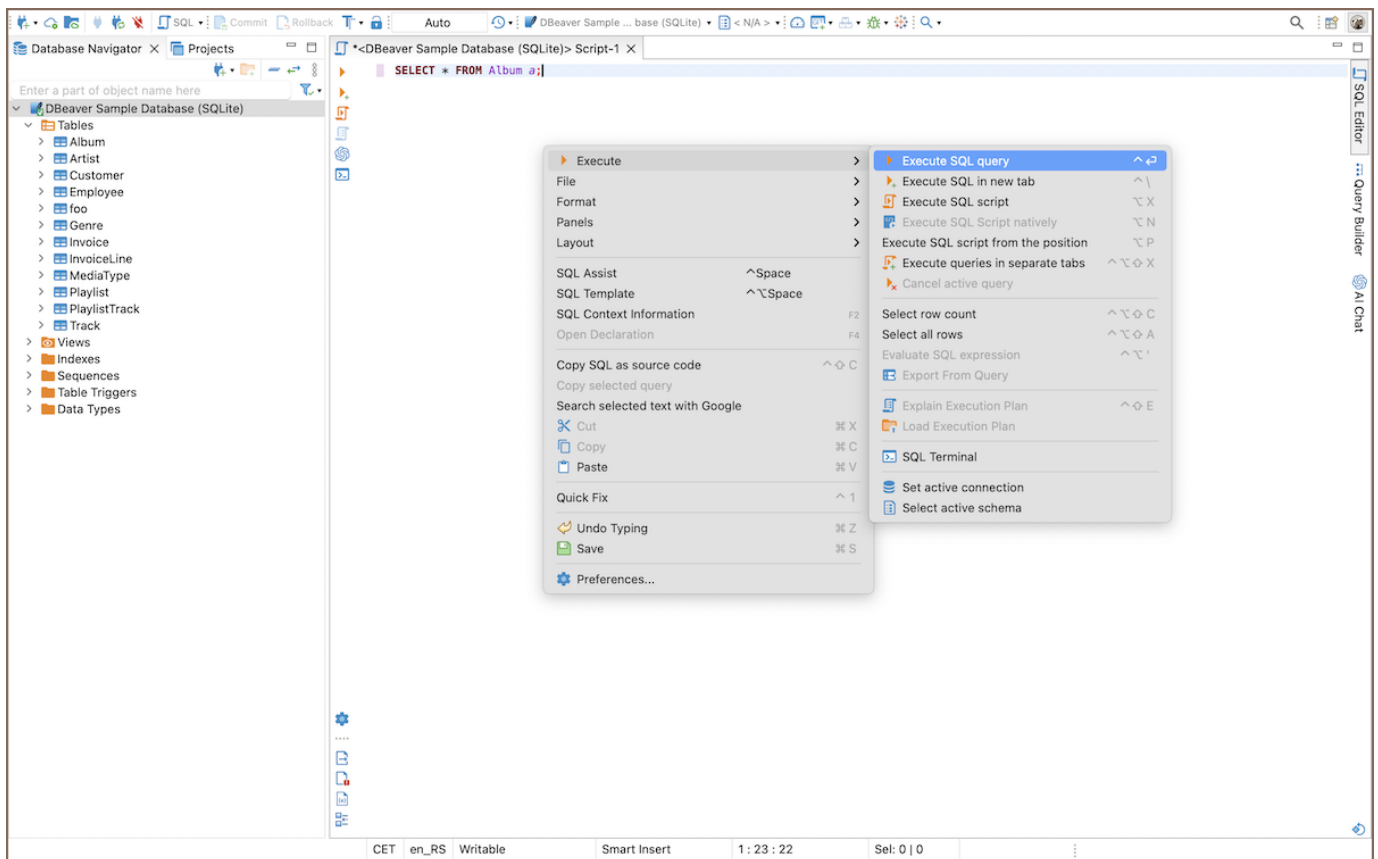
2. **Write your query:** In the [SQL Editor](#), write your SQL query. For example, `SELECT * FROM your_table; .`



Enhance your query writing with these powerful features:

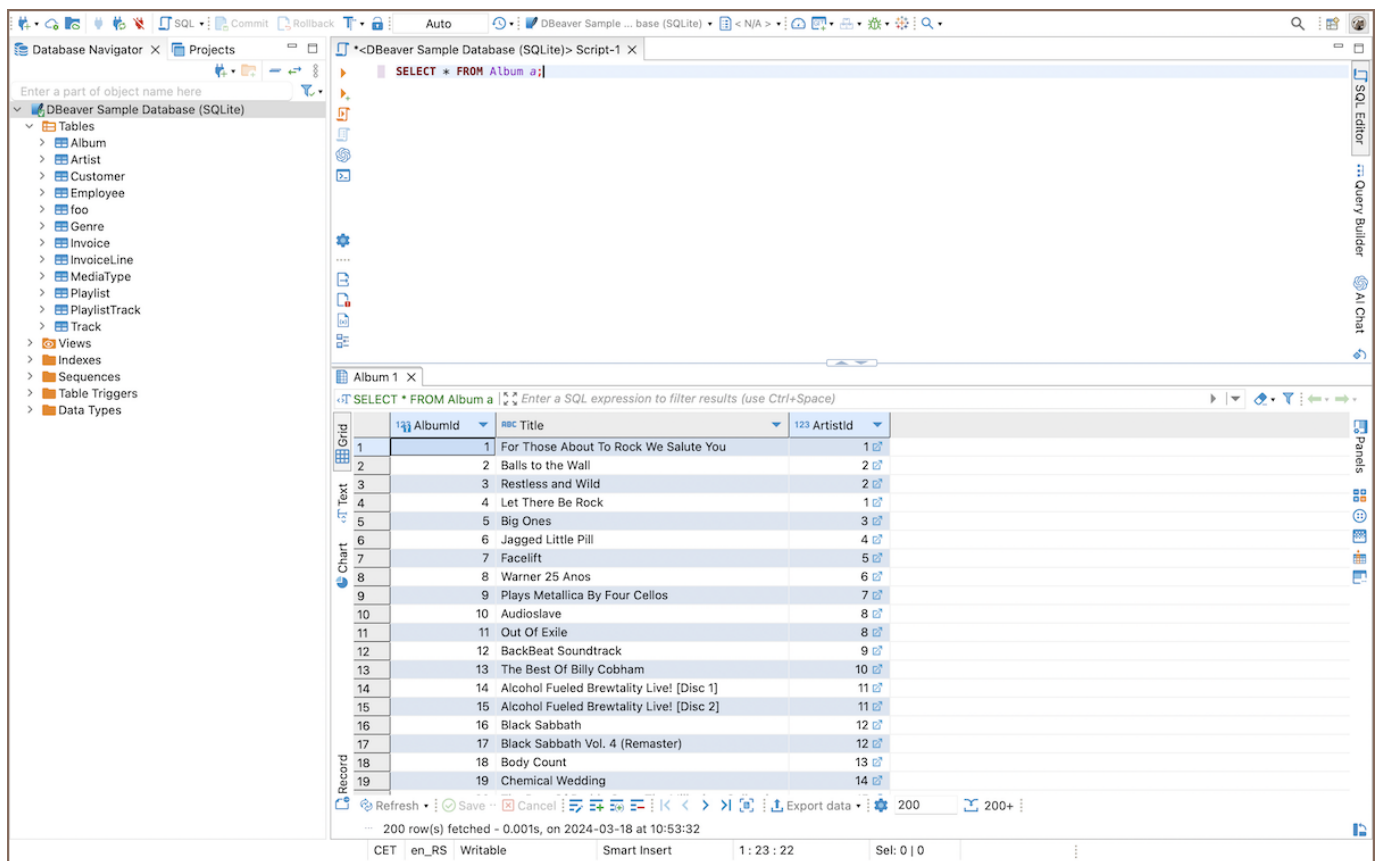
- [AI SQL Assistance](#): Utilize AI to suggest optimizations and best practices for your SQL queries.
- [SQL Code Editor](#): Enhance the behavior of the **Script panel**, focusing on features like displaying problem markers, automatic closing of quotes and brackets, semantic analysis, and more.
- [Spelling](#): Improve the accuracy of your SQL scripts with integrated spelling correction.

3. **Execute query**: To execute a query, press **Ctrl+Enter** or right-click the query and click **Execute -> Execute SQL Statement** on the context menu.



Tip: You can access various execution option, depending on your requirements. For a comprehensive guide on query execution, refer to [SQL Execution](#).

4. **View results**: The results will be displayed in the bottom panel of the **SQL Editor**.



For those who prefer visual learning or seek further instruction on using SQL with DBeaver, we recommend our [YouTube playlist](#). These video guides complement the written documentation and provide practical examples to enhance your understanding and skills in managing databases with DBeaver.

Guide to creating database connections

Table of contents

[Creating a new database connection](#)

[Use the New Connection Wizard](#)

[Use the Menu Bar](#)

[Use the Database Navigator](#)

[Connection states overview](#)

[Tips](#)

[Connection management](#)

[Configuring Connection Settings](#)

[Additional Settings](#)

[Testing and Finalizing the Connection](#)

[Advanced Settings](#)

[Network Settings](#)

[Connection Details](#)


[Additional options](#)

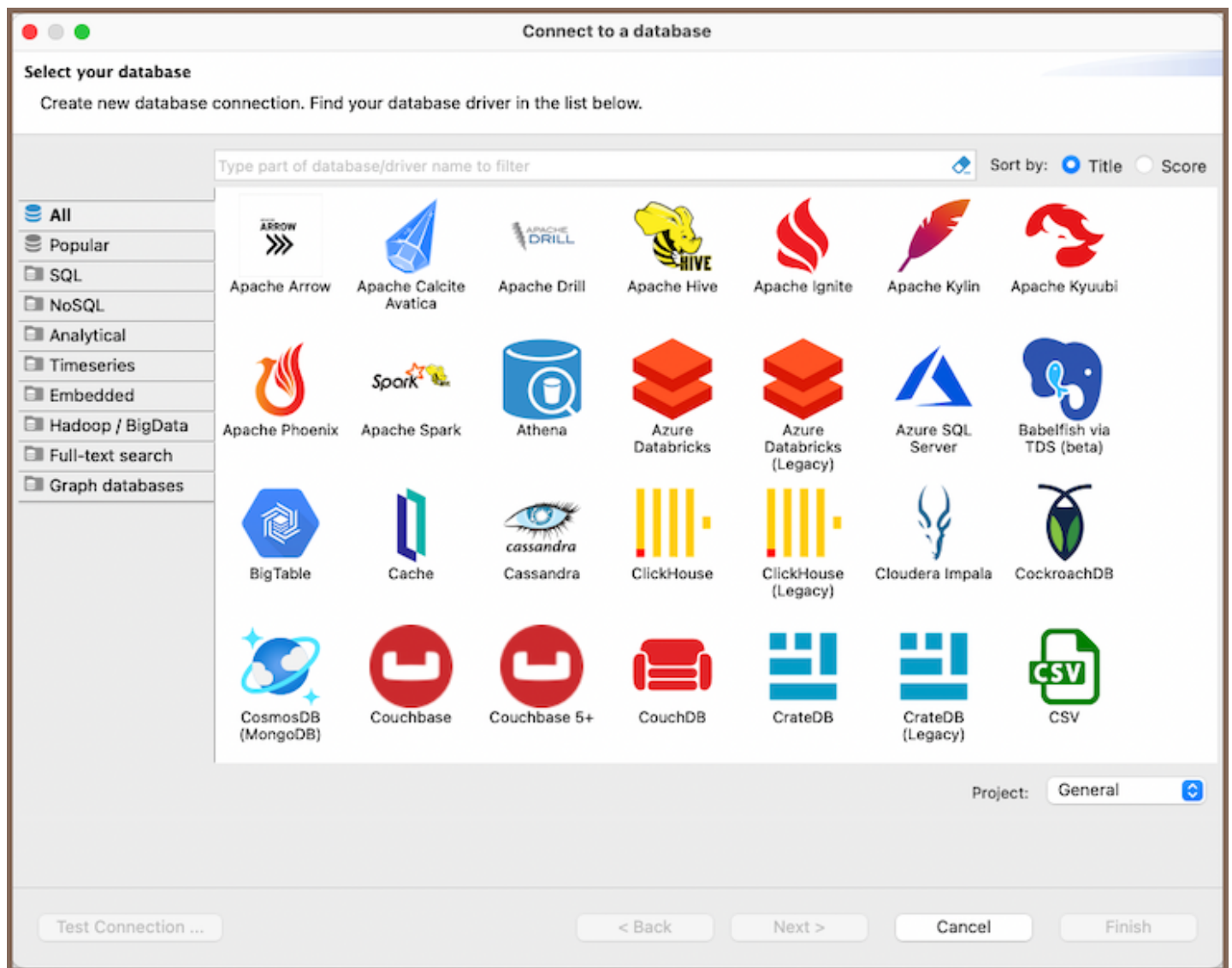
[Driver Properties](#)

Creating a new database connection

DBeaver provides a wizard that guides you through the steps to create a connection. To create a connection, do one of the following:

Use the New Connection Wizard

- Click the **New Connection Wizard** button  in the [Toolbar](#) or in the [Database Navigator](#) view toolbar.



To quickly find the needed driver for your database connection, you can:

- Type a hint in the text field above the list of drivers.
- Browse the database sections to find the required database.
- Search for a section using the **Search** field.
- Sort the options by **Title/Score**.

After selecting the required driver, click **Next**.

Use the Menu Bar

- Click **Database** -> **New Database Connection** in the menu bar.
- Press **Ctrl+N** -> **DBeaver**. Then, in the wizard, click **Database connection** and then click **Next**.





- Click **File** -> **New** -> **DBeaver** and then, in the wizard, click **Database connection** and then click **Next**.

Use the Database Navigator

- Right-click in the **Database Navigator** window area and select **Create** -> **Connection**.

Connection states overview

Database connections in DBeaver might have the following states, each represented by a unique icon:

Icon	Description
	Indicates a non-active connection.
	Shows that network settings (like SSH tunnel) are specified but not connected.
	Marks a successfully established connection.
	Reflects a connection error.

Tips

- If you cannot find a driver for your database, then probably there is no suitable driver, and you need to create one. Please see the [Database Drivers](#) article.
- Within the **New Connection Wizard**, you can select the [project](#) where the database will be saved.
- For additional details on handling multiple connections with MFA authentication, see [Disable multiple connections opening for MFA authentication](#).
- DBeaver supports database connections over IPv6, leveraging its Java-based architecture.
 - For insights into DBeaver's Java-based implementation, refer to the [Building from Sources](#) guide.
 - For more details on IPv6 support in Java, consult the [Java IPv6 Guide](#).

Connection management

In DBeaver, connections are managed within the **Database Navigator** and **Projects** views. Here, you can:

- **Edit Connections:** Refine connection details ([Editing Connections](#)).
- **Rename/Delete:** Manage connections via context menu ([Database Navigator](#)).
- **Connect/Disconnect:** Engage or disengage database connections ([Connect to Database](#) | [Disconnect from Database](#)).

Configuring Connection Settings

In the Connection Settings screen, on the **Main** tab, you need to set all primary connection settings. Here's what you typically need to configure:

The screenshot shows the 'Connect to a database' dialog box in DBeaver. The title bar says 'Connect to a database'. The main heading is 'Connection Settings' with a subtitle 'PostgreSQL connection settings'. The PostgreSQL logo is in the top right. There are tabs: 'Main', 'PostgreSQL', 'Driver properties', 'SSH', and 'SSL'. A '+ Network configurations...' link is on the right. The 'Main' tab is active. It has three sections: 'Server', 'Authentication', and 'Advanced'. In the 'Server' section, 'Connect by:' has 'Host' selected (radio button) and 'URL' (radio button). 'Driver type:' is 'JDBC' with a dropdown arrow. The 'URL:' field contains 'jdbc:postgresql://localhost:5432/postgres'. 'Host:' is 'localhost' and 'Port:' is '5432'. 'Database:' is 'postgres'. There is a 'Show all databases' checkbox. In the 'Authentication' section, 'Authentication:' is 'Database Native' with a dropdown arrow. 'Username:' is 'postgres'. 'Password:' is masked with dots. There is a 'Save password' checkbox (checked) and an eye icon. In the 'Advanced' section, 'Session role:' is empty. 'Local Client:' is '/Library/PostgreSQL/16' with a dropdown arrow. At the bottom, there is a link: 'You can use variables in connection parameters.' and a button: 'Connection details (name, type, ...)'. Below that, 'Driver name:' is 'PostgreSQL'. There are buttons: 'Driver Settings' and 'Driver license'. At the very bottom, there are buttons: 'Test Connection ...', '< Back', 'Next >', 'Cancel', and 'Finish'.

Connect to a database

Connection Settings

PostgreSQL connection settings

PostgreSQL

Main PostgreSQL Driver properties SSH SSL + Network configurations...

Server

Connect by: ☒ Host ☐ URL Driver type: JDBC

URL: jdbc:postgresql://localhost:5432/postgres

Host: localhost Port: 5432

Database: postgres ☐ Show all databases

Authentication

Authentication: Database Native

Username: postgres

Password: ☒ Save password

Advanced

Session role: Local Client: /Library/PostgreSQL/16

[You can use variables in connection parameters.](#) Connection details (name, type, ...)

Driver name: PostgreSQL Driver Settings Driver license

Test Connection ... < Back Next > Cancel Finish

Setting	Description
Host	The address of the database server.
Port	The port number for the connection.
Database Name	The name of the specific database you want to connect to.
Username and Password	The credentials required for authentication. This falls under the Authentication section, where different databases may use various types of authorizations. Common types include Database Native and DBeaver Profile .

However, the number and type of connection properties are very dependent on the driver. For example, embedded drivers (such as SQLite, Derby Embedded, HSQLDB, H2 Embedded), unlike remote ones, require only the path to the database. For more detailed information about the settings specific to each driver, you can refer to the driver documentation articles available in the DBeaver documentation.

Additional Settings

If necessary, you can specify advanced settings. See the [Advanced Settings](#) section below for more details.

Testing and Finalizing the Connection

- **Test the Connection:** Click **Test Connection** to verify if the connection works.
- **Finish:** Click **Finish**. The connection appears in the connections tree in the **Database Navigator**.

By following these instructions, you can configure and establish a connection to your desired database within DBeaver.

Advanced Settings

The Advanced Settings tab in the connection configuration process provides additional options to fine-tune your connection. Here's what you can configure:

Network Settings

Setting	Description
SSH Tunnel	Configure an SSH tunnel if your database requires a secure connection.
SOCKS Proxy	Set up a SOCKS proxy if you need to route the connection through a specific proxy server.
SSL Configuration	Enable and configure SSL if your connection requires encryption.
Kubernetes	Configure Kubernetes settings if your database is hosted within a Kubernetes cluster.

Connection Details

Connect to a database

General

General connection settings.

General

Connection name:

postgres

Description:

Connection type:

Development

Edit connection types

Navigator view:

Custom

Customize ...

Connection folder:

Security

☐ Read-only connection

Edit permissions ...

Filters

Databases / Catalogs

Schemas / Users

Tables

Columns

Virtual model

☒ Show virtual model editor

Reset configuration

[Connection initialization settings](#)

[Shell Commands](#)

Test Connection ...

< Back

Next >

Cancel

Finish

Setting	Description
	Assign a specific name to the connection for easier identification. You can use variables in all connection

Connection Name	<p>parameters and in the driver properties. Variables are system environment variables or one of the following list:</p> <p><code>\${host}</code> : Host name</p> <p><code>\${port}</code> : Port number</p> <p><code>\${database}</code> : Database name</p> <p><code>\${server}</code> : Server name</p> <p><code>\${url}</code> : Connection URL</p> <p><code>\${user}</code> : User name</p> <p><code>\${connection.type}</code> : Connection type (e.g., production, development)</p> <p><code>\${project.name}</code> : Project name</p> <p><code>\${date}</code> : Current date</p> <p><code>\${host_or_database}</code> : Host name or database name</p>
Connection Type	Choose the type of connection, such as production, development, or test.
Descriptions	Add descriptions for your connection for better understanding.
Navigator View	Customize the navigator view settings.
Connection Folder	Organize your connections by placing them in specific folders.

Note: `${password}` property is deprecated due to security concerns since 23.2.0.

Additional options

Setting	Description
Security	See the Security restrictions article for more details.
Filters	Configure filters for your connection. See the Filters article for more details.
Virtual Model Editor	<p>Reset virtual model settings, including:</p> <ul style="list-style-type: none"> - All virtual constraints and foreign keys. - All column transformers. - All table row colorings.
Connection Initialization Settings	Customize the initialization settings for your connection. See our article for more details.

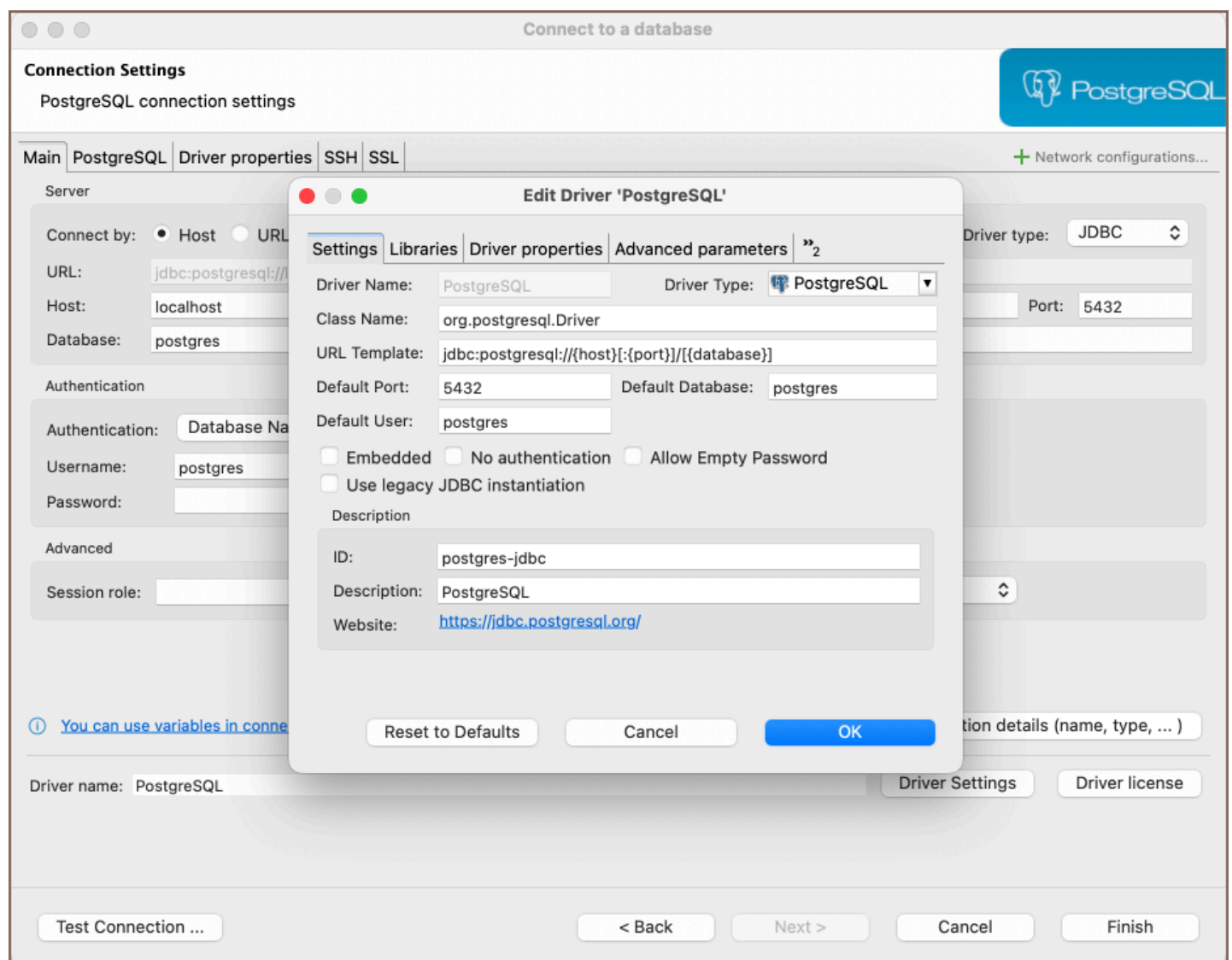
Shell Commands

Configure shell commands for your connection. See the [Shell Commands article](#) for more details.

These advanced settings allow you to tailor the connection to fit specific requirements and preferences. Be sure to consult your database documentation or system administrator if you need guidance on the appropriate settings for your particular database and use case.

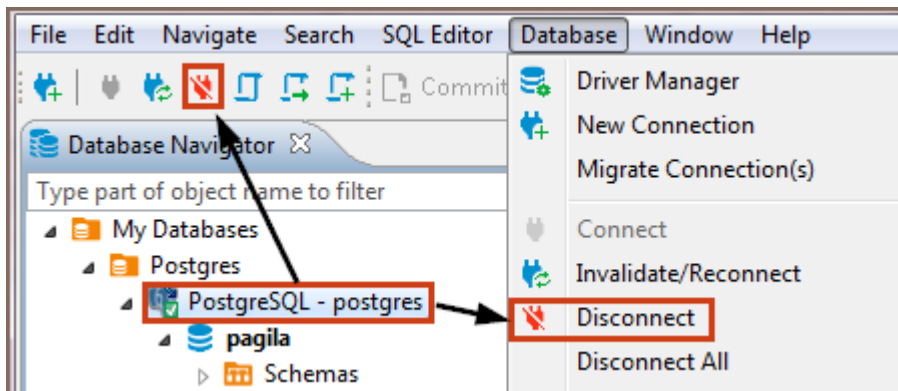
Driver Properties

Each driver has its own set of additional properties. Refer to the driver documentation to get information about available properties and their values. Understanding and configuring these properties can help you optimize the connection according to the specific requirements and behavior of the database driver you are using.





Disconnecting from database

You might need to disconnect from a database to free up resources or close transactions. To disconnect from a database, click the connection in the [Database Navigator](#) or [Projects](#) view, and then click the Disconnect button in the toolbar or click **Database -> Disconnect** on the main menu:



You can also right-click the connection and click **Disconnect** on the context menu.

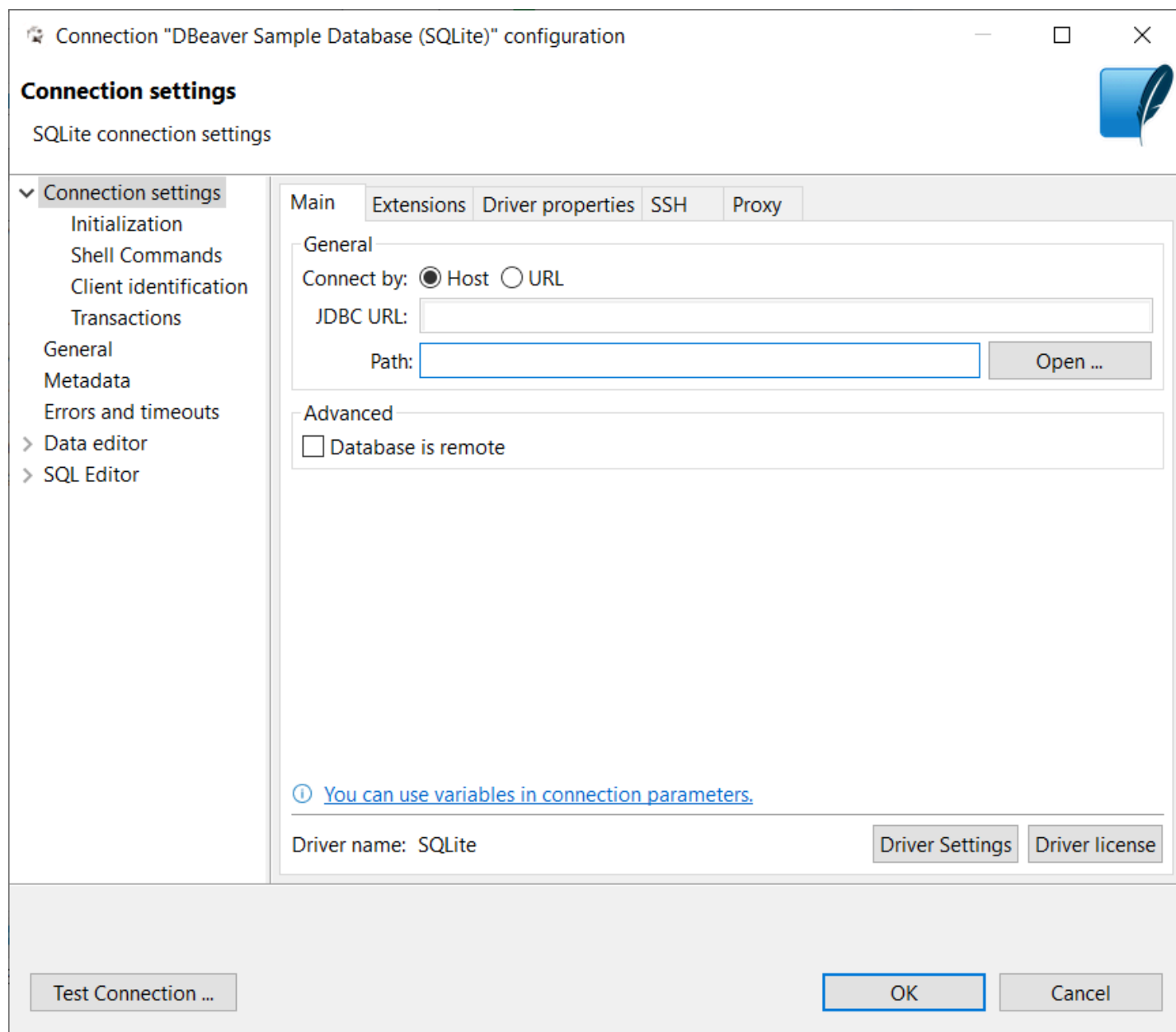
NOTE: The Disconnect button and menu items are available only for those connections that are activated, that is, marked with the connected sign: .

When DBeaver disconnects from a database, its icon changes to its original state (not connected), for example,  for PostgreSQL database.

To disconnect from all active connections, click **Database -> Disconnect All** on the main menu.

Editing database connection

To edit the configuration settings of a database connection, in the [Database Navigator](#) or in the [Projects](#) view, right-click the connection and click **Edit Connection** on the context menu. The Connection configuration window will open:



The screenshot shows the 'Connection "DBEaver Sample Database (SQLite)" configuration' dialog box. The title bar includes standard window controls and a blue feather icon. The main title is 'Connection settings', with a subtitle 'SQLite connection settings'. On the left is a navigation pane with a tree view containing: 'Connection settings' (expanded), 'Initialization', 'Shell Commands', 'Client identification', 'Transactions', 'General', 'Metadata', 'Errors and timeouts', 'Data editor', and 'SQL Editor'. The main area has tabs for 'Main', 'Extensions', 'Driver properties', 'SSH', and 'Proxy'. The 'Main' tab is active, showing a 'General' section with 'Connect by:' options (radio buttons for 'Host' and 'URL', with 'Host' selected), a 'JDBC URL:' text field, and a 'Path:' text field with an 'Open ...' button. Below this is an 'Advanced' section with a checkbox for 'Database is remote'. At the bottom of the main area, there is a link: 'You can use variables in connection parameters.' and a label 'Driver name: SQLite' with 'Driver Settings' and 'Driver license' buttons. The bottom of the dialog features a 'Test Connection ...' button on the left and 'OK' and 'Cancel' buttons on the right.

The navigation pane on the left displays the configuration sections, most of which are the same as those in the Create new connection wizard, see [Connect to Database](#). There are additional configuration sections as well, such as **Result Sets** and **SQL Editor**. Click the section name to open the configuration settings for editing.

You can test if your connection works with modified settings - click **Test Connection**. When you finish editing your connection, click **OK** to save the changes or **Cancel** to discard them.

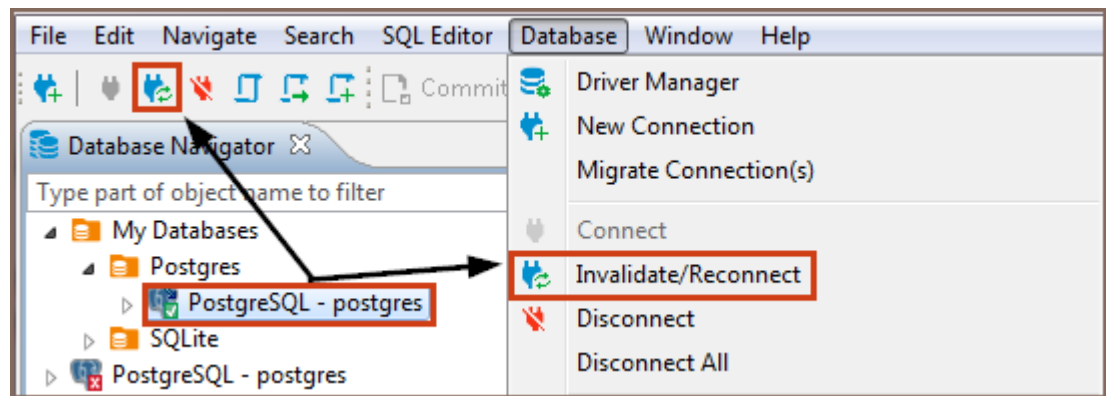
Driver settings:

- In development

[JDBC Time Zones](#)

Invalidating and reconnecting to database

You might need to invalidate a database connection and then reconnect to it again in cases such as where the connection to the server is temporarily lost, etc. To invalidate a database connection and then reconnect to the database, click the database connection in the Database Navigator or Projects view, and then click the **Invalidate/Reconnect** button in the toolbar or **Database -> Invalidate/Reconnect** on the main menu:



You can also right-click the connection and click **Invalidate/Reconnect** on the context menu.

Local client configuration

Table of contents

[Overview](#)

[Local Client configuration for Windows](#)

[Local Client configuration for Mac and Linux](#)

[Find a database installation path](#)

[Configure a local client in DBeaver on Mac and Linux](#)

Overview

This article provides instructions on how to configure a local database client for use with DBeaver.

A local client refers to the database client software installed on your machine, allowing DBeaver to interact with your databases. This local installation is necessary for specific features like native [backup and restore](#) operations.

Key points to consider:

- In the **Community Edition** of DBeaver on Windows, local clients are downloaded from the internet when a remote client is selected in the **Local Client** options, typically during the first usage of database-specific tools such as backup, or restore functions.
- Windows users with the [Enterprise](#), [Ultimate](#), or [Team](#) editions enjoy a simplified process, as many local clients come pre-included with the software.
- Mac and Linux users may need to perform additional steps to configure the local client. The exact procedure will vary depending on the system setup.

Local Client configuration for Windows

Upon initial configuration, DBeaver will attempt to automatically identify the paths to local database clients in the Windows environment. If this process is unsuccessful, manual path configuration may be necessary.

Local Client configuration for Mac and Linux

On macOS, DBeaver actively tries to find local database tools automatically, covering a range of installations without requiring specific configurations from the user.

Tips:

- **For PostgreSQL on macOS:** Installing the `libpq` package may help DBeaver automatically detect the client. Use `brew install libpq` in your terminal.
- **For MySQL Users on macOS:** To facilitate automatic client detection by DBeaver, consider adding the `mysql-client` package. This can be done using the command `brew install mysql-client` in the terminal.

If you encounter any issues with automatic detection, or if you're on a Linux system, follow the subsequent manual configuration steps below.

Find a database installation path

To find the installation path for the local client of a database in DBeaver, follow these general guidelines:

- **For MySQL:** Use `mysqldump` as a keyword in your search command. This utility is often located in the `bin` directory of the MySQL installation.
- **For PostgreSQL:** Substitute `mysqldump` with `pg_dump` when conducting your search. Like MySQL, this command should be in the `bin` directory of the PostgreSQL installation path.

These instructions are applicable to the most commonly used databases, MySQL and PostgreSQL.

1. Open your terminal.
2. Use the `find` command to locate the MySQL installation:

```
find / -name "mysqldump" 2>/dev/null
```

3. Upon successful execution, the terminal will output the path to your MySQL installation. For example: -

```
/usr/local/mysql-8.1.0-macos13-arm64/bin/mysqldump
```

4. Copy this path for use in the next steps of the configuration process.

Configure a local client in DBeaver on Mac and Linux

1. Open DBeaver.
2. Right-click on the respective database and navigate to **Edit Connection Settings**.
3. Find the **Local Client** section and click on the **Browse** button that appears next to it.
4. Click the **Add Home** button to add a new client path.
5. For Mac users, press `Command+Shift+G` to open the prompt for path input. For Linux users, enter the path directly into the file selection dialog.
6. Input the MySQL path location found earlier and confirm. Make sure to input the path up to the database version folder, omitting `/bin/mysql`.

For example, if the path found is `/usr/local/mysql-8.1.0-macos13-arm64/bin/mysqldump`, input `/usr/local/mysql-8.1.0-macos13-arm64/`.

7. After inputting the corrected path, click the **Open** button to select the path.
8. Click the **OK** button to finalize the local client configuration.

Connection types

Table of contents

[Change Connection Type](#)

[Create Connection Type](#)

[Edit Connection Type](#)

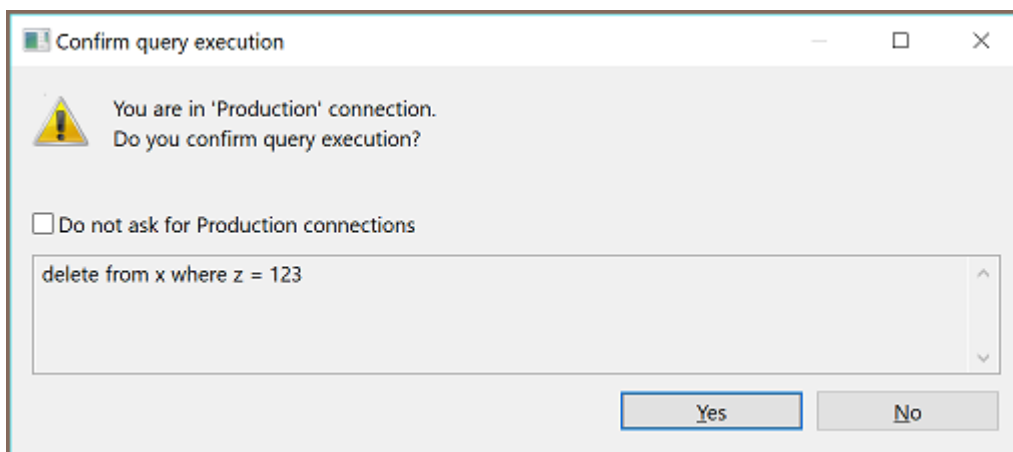
[Delete Connection Type](#)

Connection types define how DBeaver behaves regarding:

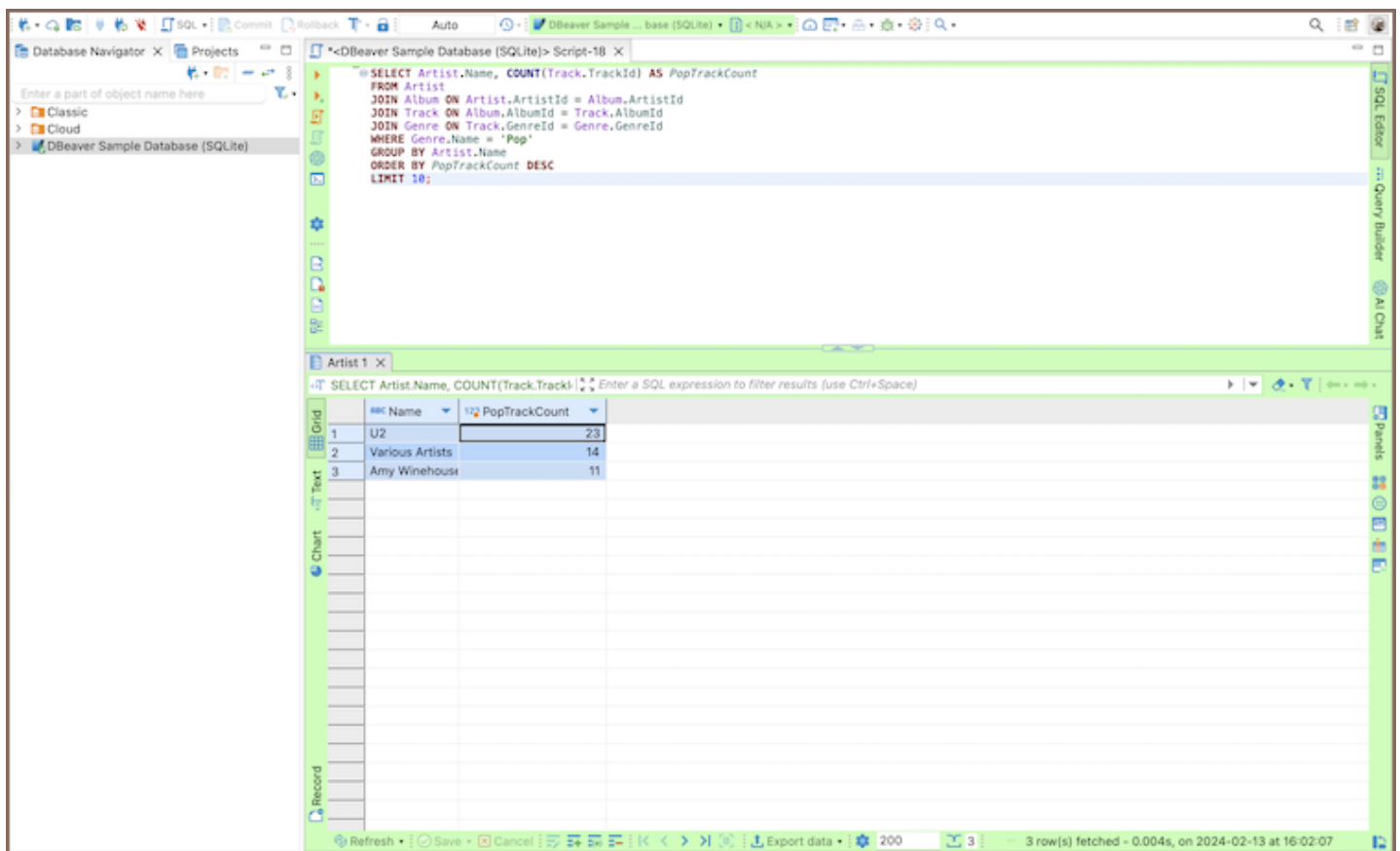
- Default transactions commit mode - with or without automatic commit of changes to the database.

NOTE: You can override the default commit behavior during your work with connections by changing the commit mode, see [Auto and Manual Commit Modes](#).

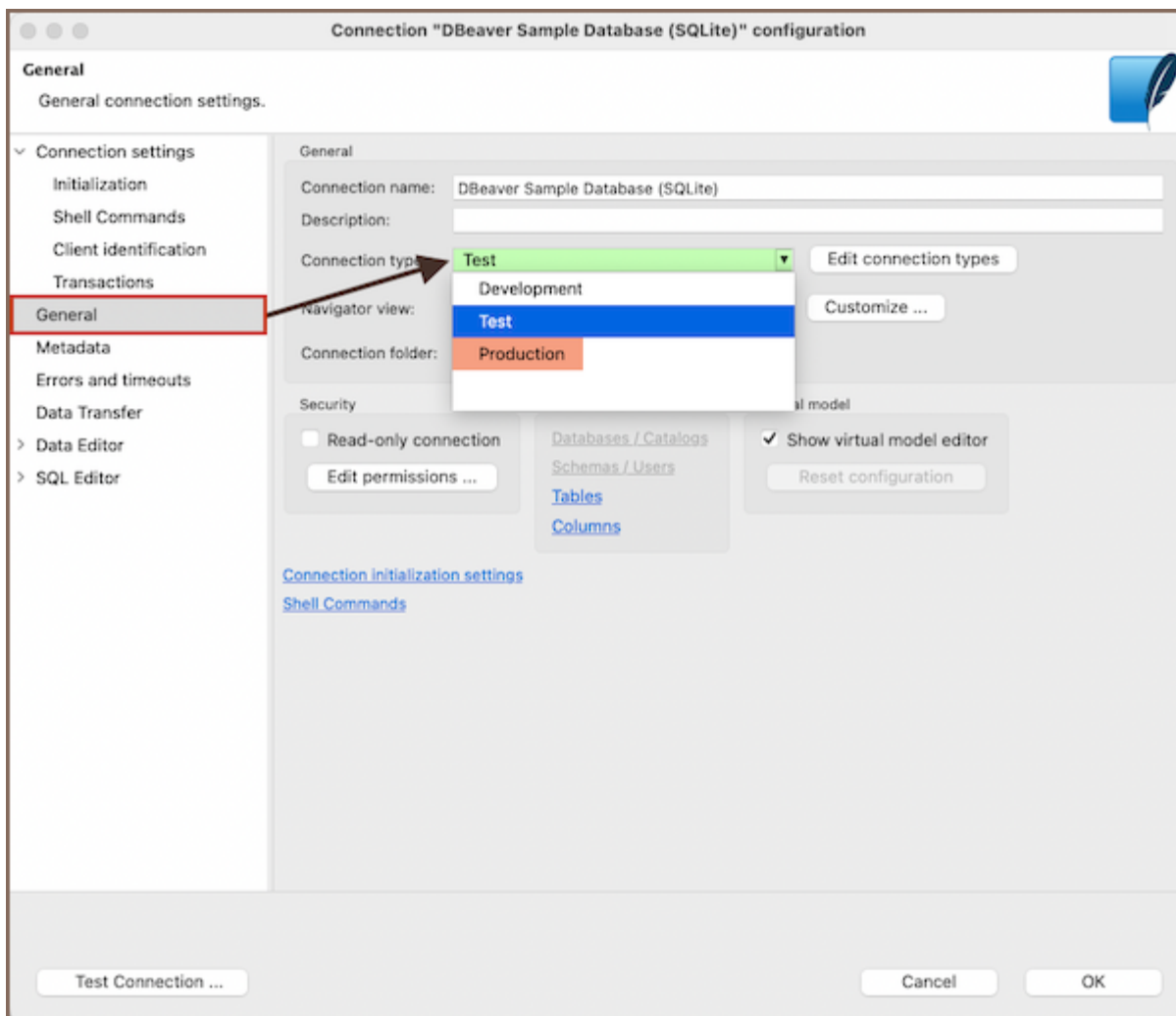
- SQL statements execution (with or without user confirmation): If set to the required user confirmation for SQL execution, DBeaver shows a confirmation message every time you attempt to execute a 'transaction' type of query (INSERT/DELETE/UPDATE, etc.):



For your convenience, DBeaver supports color-coding of connection types so that you know at once which behavior to expect when you use a certain connection. The screenshot below visualizes how the color coordinated database connections are used in [Database Navigator](#) and [Projects](#) views as well as editors related to these connections:



To manage connection types for a database connection, in the Database Navigator or Projects view, click the connection to set the focus on it and then press **F4** to open the connection properties window. Then, in the properties window, in the navigation pane on the left, click **General** to see the general settings. You can see **Connection Type** field among the settings:



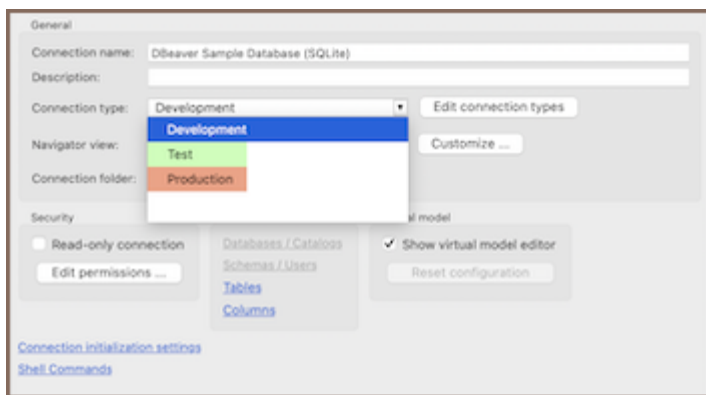
There are three default connection types – **Development**, **Test**, and **Production**. You can change the connection type for your database connection as well as you can create a new connection type, edit or delete an existing one.

Change Connection Type

By default, the **Development** connection is preset for all database connections. You can change the connection type to one of the default connection types or to a custom type, if there are any.

To change the connection type:

1. In the connection properties window, on the **General** page, click the **Connection type** field and then click the connection type in the dropdown list:

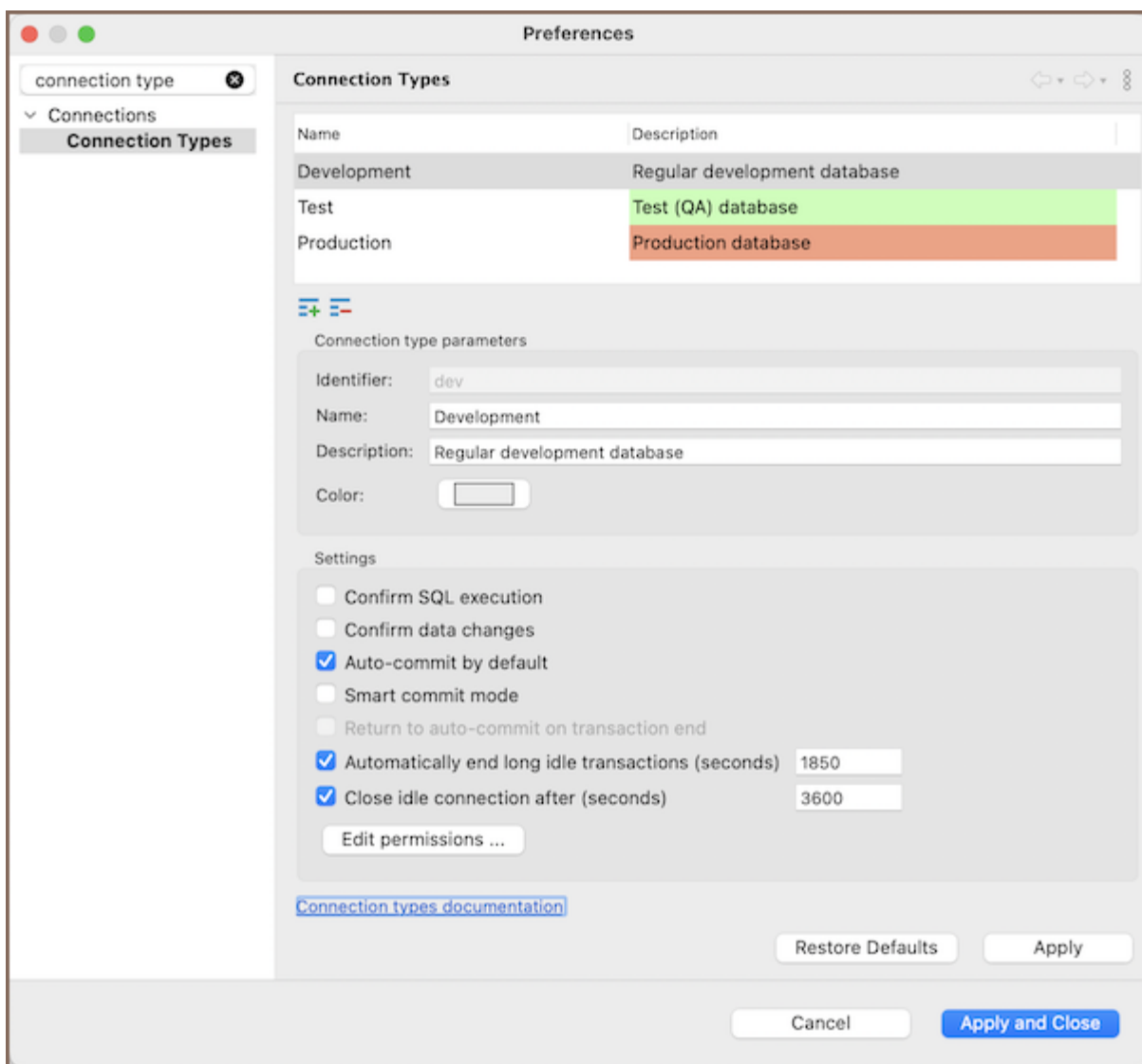


2. To test the connection, click **Test Connection**. To confirm the change, click **OK**.

Create Connection Type

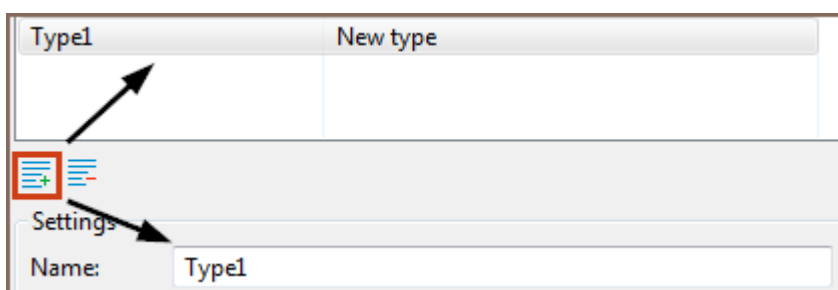
To create a connection type:

1. In the connection properties window (F4 on a connection), on the **General** page, click **Edit** next to the **Connection type** field. The Properties for connection types window opens:



The window displays existing connection types and their settings.

2. Click the new connection type button. A new connection type appears in the list:



3. Now you can specify the settings for the new connection type:

Setting	Description
Confirm SQL execution	Displays a confirmation dialog for any non-SELECT SQL operations.
Confirm data changes	Displays a confirmation dialog before persisting changes in the database.

Auto-commit by default	Sets SQL operations to commit automatically, making it the default transactional mode.
Smart commit mode	Initiates a transaction before any data modifying queries (e.g., <code>INSERT</code> , <code>UPDATE</code>), allowing configuration of additional keywords for transaction activation.
Return to auto-commit on transaction end	Switches back to auto-commit mode after the transaction is completed (explicit commit or rollback), only active with Smart commit mode .
Automatically end long idle transactions (seconds)	Closes transactions after a specified idle period with no user activity, performing a rollback before transaction ends.
Close idle connection after (seconds)	Closes connections with no activity after a specified number of seconds. A value of zero and unchecking this option allows the connections with the current type to remain active indefinitely.

Tip: For embedded databases, the setting **Close idle connection after (seconds)** is always ignored. This means they will never automatically disconnect, regardless of the configured value.

- Click **Apply** to apply the changes and keep the window open or click **Apply and Close** to apply the changes and close the window. To discard all changes and return to the previous state, click **Restore Defaults**.


Edit Connection Type

To edit a connection type:

- In the connection properties window (`F4` on a connection), on the **General** page, click **Edit** next to the **Connection type** field.
- Specify the settings for the new connection type the same way as when you create a connection type, see 'Create Connection Types' section above.
- When you finish editing the connection types, click **Apply** to apply the changes and keep the window open or click **Apply and Close** to apply the changes and close the window. To discard all changes and return to the previous state, click **Restore Defaults**.

Delete Connection Type

To delete a connection type:

1. In the connection properties window (**F4** on a connection), on the **General** page, click **Edit** next to the **Connection type** field. The Properties for connection types window opens.
2. In the Properties window, in the list of connection types, click the connection type to set the focus on it and then click the delete button under the list: . The delete button is represented by a small icon with three horizontal lines and a red minus sign.
3. Click **Yes** in the confirmation dialog box to confirm the deletion. Otherwise, click **No**.
4. Click **Apply** to apply the changes and keep the window open or click **Apply and Close** to apply the changes and close the window.

Multiple datasource connections

Table of contents

[Metadata](#)

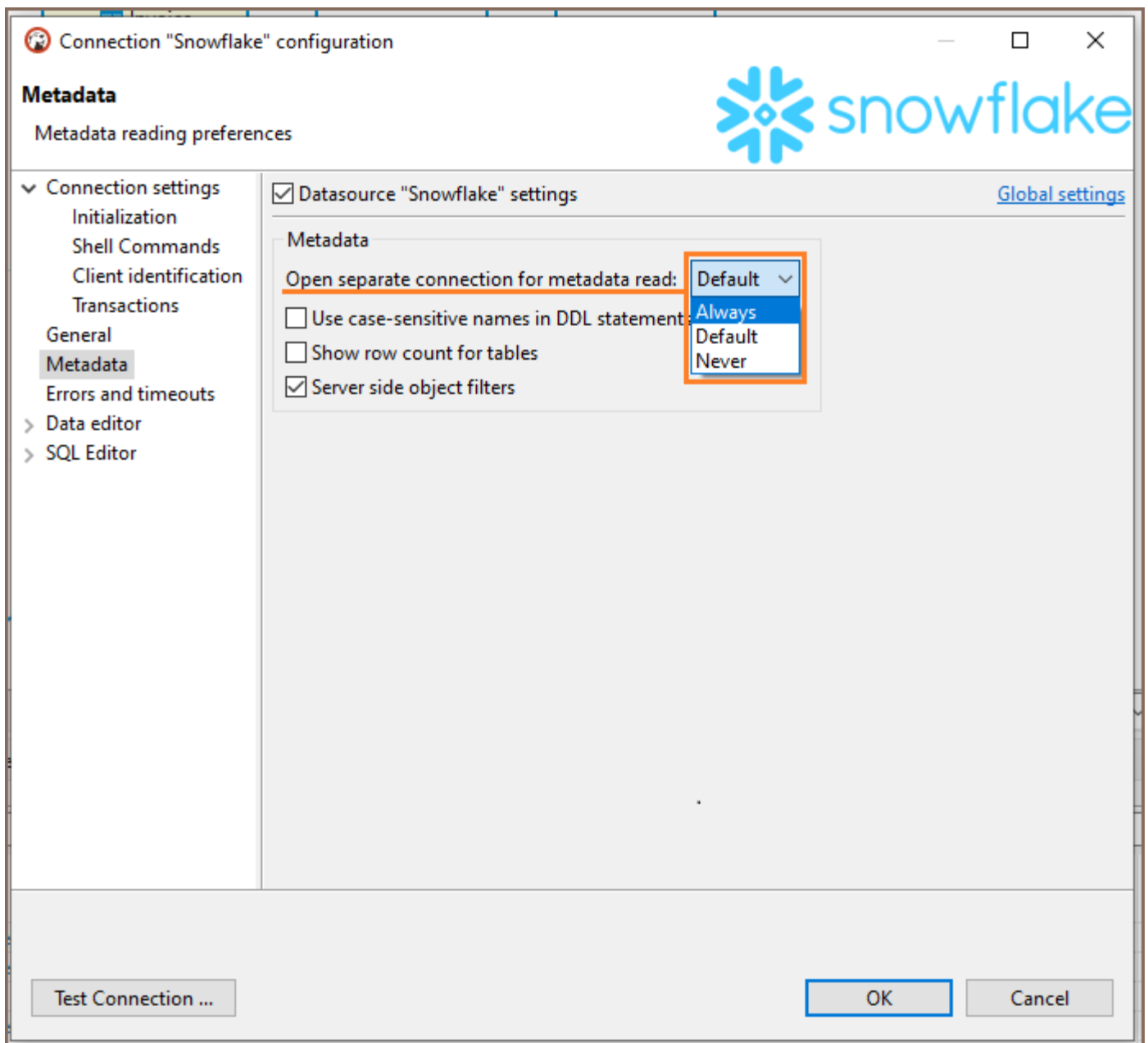
[SQL Editor](#)

[Options](#)

Metadata

Opening separate metadata connection may increase performance because there will be no UI locks during query execution.

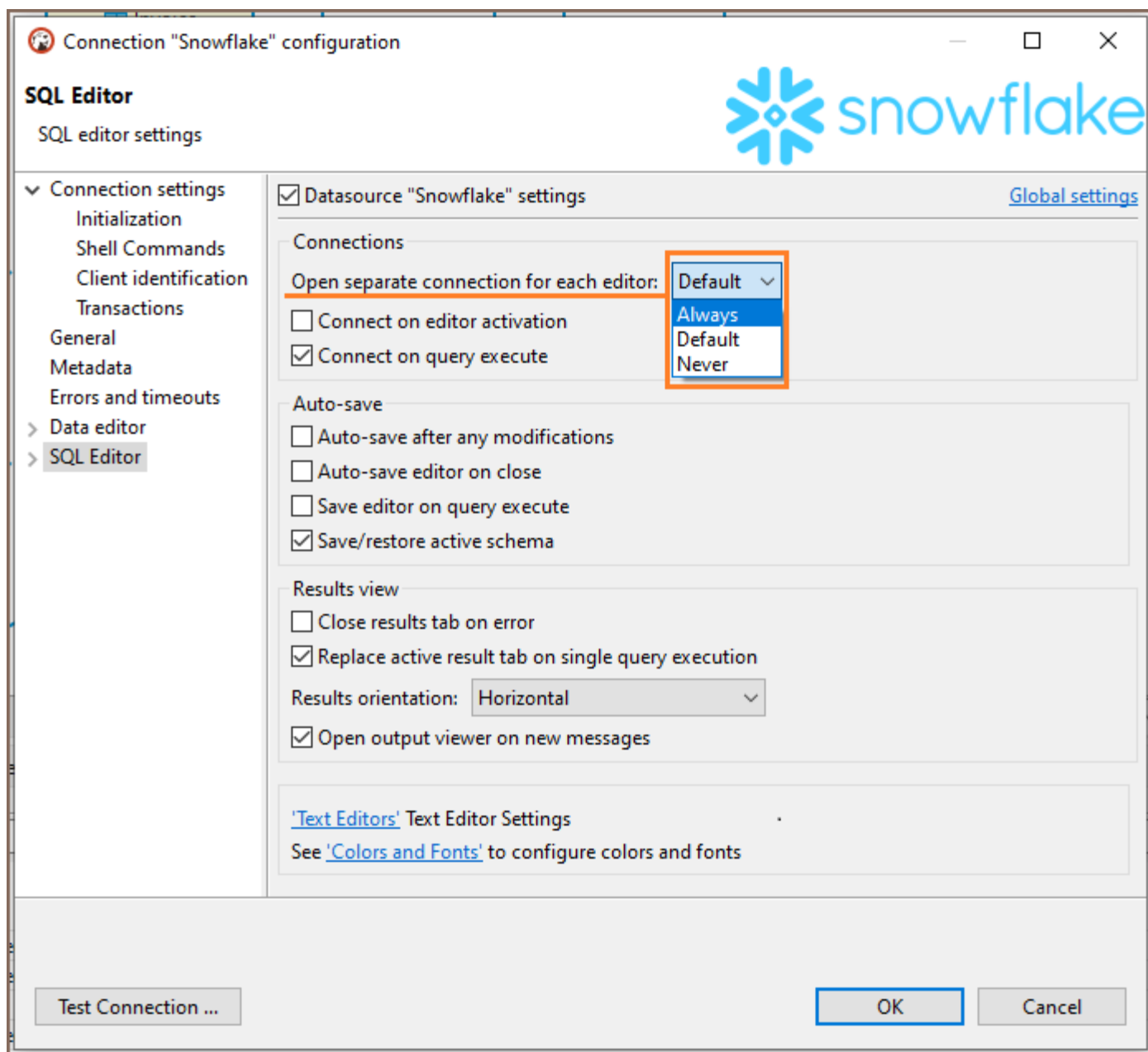
You can set up opening separate connection for metadata read globally in Preferences->Connections->Metadata or per each connection in Connection configuration->Metadata.



SQL Editor

Opening separate connection for SQL Editor may allow you to have different execution context for each script.

You can set up opening separate connection for each editor in Preferences->Editors->SQL Editor and Connection configuration->SQL Editor and Connection configuration->SQL Editor.



Options

Always option means that DBeaver will create separate connections for SQL Editor and/or metadata read.

If you choose **Never** option, then DBeaver will use single connection - no new connections will be opened.

Default option means that DBeaver will create separate connections for SQL Editor and/or metadata read for most databases except these:

- SQLServer Azure Entra ID MFA
- BigQuery

- BigTable
- Spanner
- Apache Hadoop, Apache Drill, Apache Kyuubi, Apache Spark, Apache Hive, SnappyData, Gemfire XD, Apache Phoenix
- Redshift
- Snowflake

Configure connection initialization settings

Table of contents

[Overview](#)

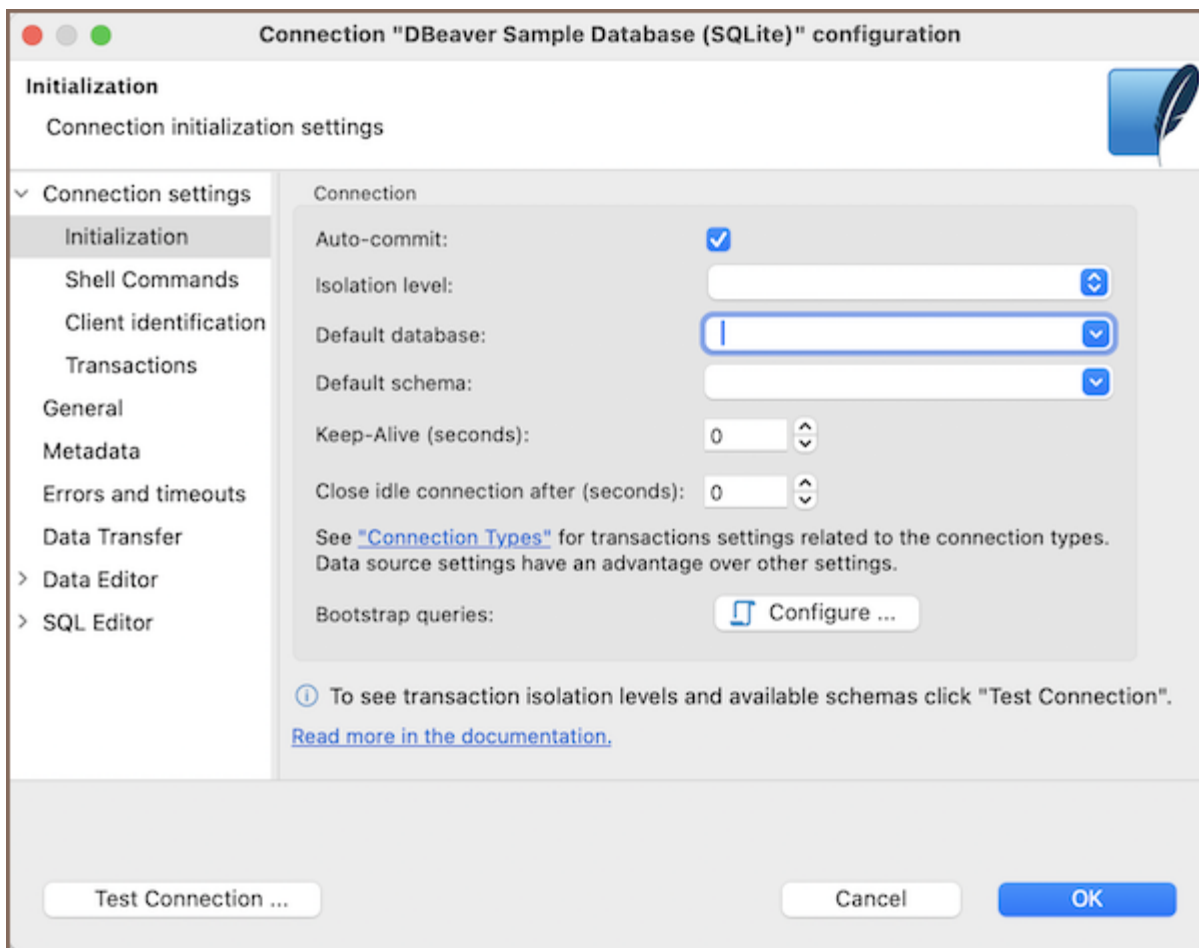
[Accessing Connection Initialization Settings](#)

[Settings](#)

Overview

During the initialization of a new database session, the Connection Initialization Settings provide a customizable set of parameters. These settings ensure that every time you connect to the database, certain predefined behaviors are automatically initialized. This feature allows you to set the:

- Transaction management options such as auto-commit status and transaction isolation levels.
- Defaults for the initial database and schema selection upon connection establishment.
- Network handling features like keep-alive settings and idle connection timeout.
- Bootstrap queries to execute predetermined SQL commands at the start of every session.



Accessing Connection Initialization Settings

To configure the Initialization Settings for your database connections, follow these steps:

For a new connection:

1. [Create a new connection](#).
2. In the connection creation wizard, navigate to **Connection details** -> **Connection Initialization Settings** to specify your preferences.

For an existing connection:

1. Go to the [Database Navigator](#), find the connection you want to configure.
2. Right-click on it and select **Edit Connection**.
3. In the edit menu, navigate to the **Initialization** tab where you can adjust the settings.

Settings

Setting	Description
Auto-commit	Determines if each SQL statement is committed automatically upon execution.
Isolation level	Sets the transaction isolation level which defines the degree of visibility of transactions to each other.
	Read uncommitted: Allows a transaction to read data that has not yet been committed by other transactions.
	Read committed: Ensures a transaction only reads data that has been committed at the time the transaction began.
	Repeatable read: Guarantees if a row is read twice in the same transaction, the result will be the same.
	Serializable: Ensures transactions occur in a completely isolated fashion, equivalent to serial transaction execution.
Default database	Specifies the default database to be selected once the connection is established.
Default schema	Indicates the default schema to be used within the selected database.
Keep-Alive	Interval (seconds) for sending a signal to maintain the connection open during user inactivity.
Close idle connections after	<p>The duration in seconds after which idle connection (with no activity) will be closed. Unchecking this option allows the connection to remain active indefinitely.</p> <p>If the field is set to zero, the system will apply the values configured in the Connection Types settings. If a value other than zero is specified, that value will override the default settings from the Connection Types.</p>
Bootstrap queries	Allows configuration of specific SQL queries to run at the start of the session.

Note: The transaction isolation levels and available schemas can only be viewed by clicking on **Test Connection** after the connection has been established.

Managing Master password

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Enabling secure password storage](#)

[Administrator configuration of Master password policy](#)

[Configuring Master password](#)

[Changing the Master password](#)

[Password recovery setup](#)

[Setting a password hint](#)

[Deleting the Master password](#)

[Viewing secured resources](#)

[Advanced encryption settings](#)

Overview

The Master password feature strengthens the security of DBeaver by encrypting credentials and sensitive data using a secure storage system.

When you use the Master password feature:

- **Isolated Security:** Connection details cannot be shared with other users because credentials are securely stored in an encrypted, user-specific location.
- **Individual Protection:** Each set of credentials is safeguarded by the local user's own the Master password, ensuring personalized security.
- **Project Specificity:** Connections with passwords can't be shared across users in projects because of the secure, user-specific storage. For more details, see [Project Security](#) article.

Secure storage is designed with security prioritization, which has certain trade-offs:

- **Non-Portability:** The approach does not support the portability or sharing of configurations among a team of developers, which is mainly due to the OS-specific nature of password storage.
- **User-Specific Encryption:** The integration of the Master password further emphasizes individual security by tying the encrypted data to the user's local environment.

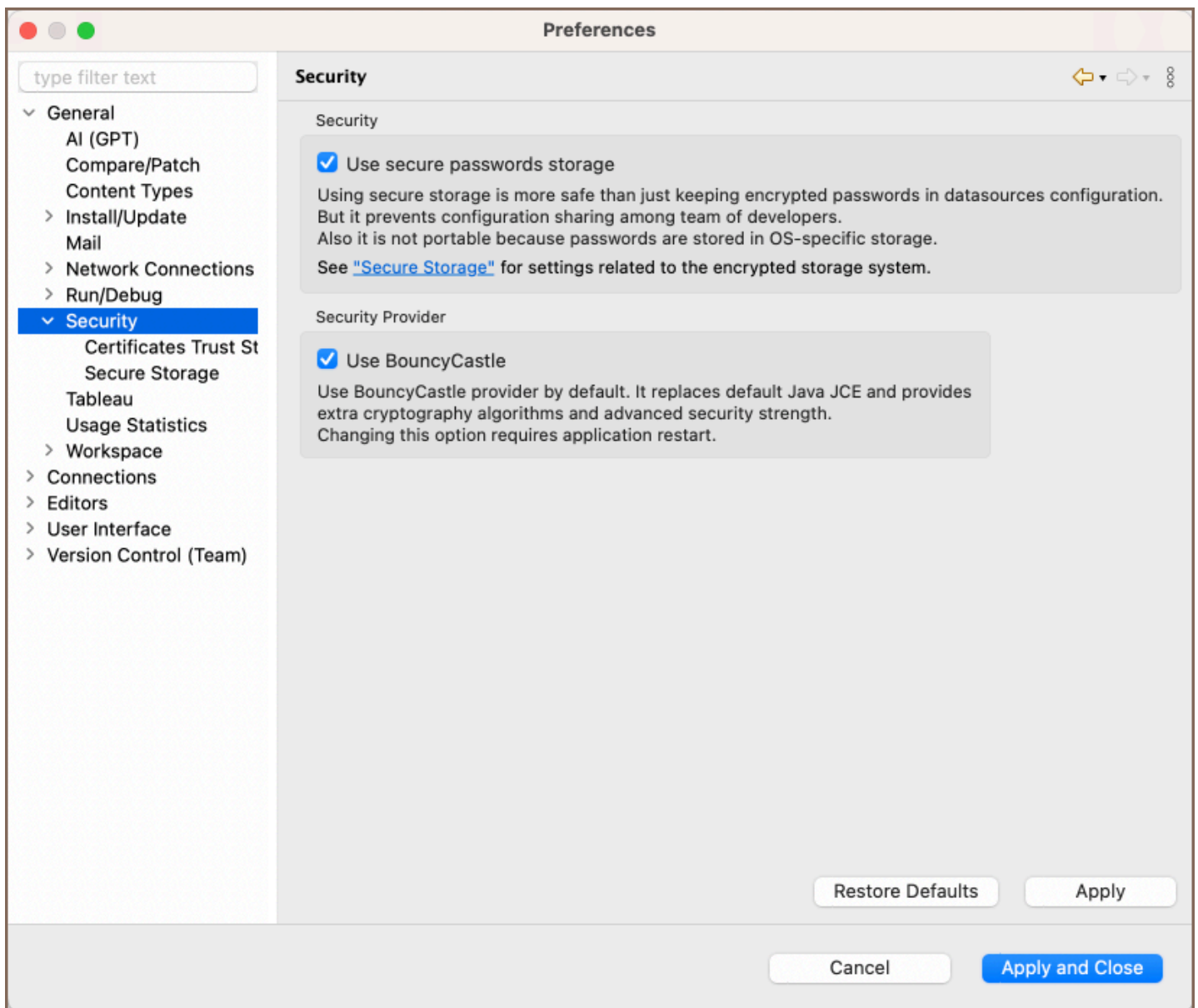
Tip: When you first attempt to save secure data (for example, accessing [Cloud Explorer](#), performing [Git operations](#), or configuring [email profile settings](#)), DBeaver will prompt you to set up a Master password if you haven't already established one. You will have two options:

- **Set up a Master password (recommended):** Choose this to create a new Master password. This is the preferred option for securing sensitive data.
- **Use integrated security:** Utilize your OS' security system, involving a password automatically stored in your local user's secure storage. For more information, refer to the [Managing Master password](#) section.

Enabling secure password storage

To enable the Master password functionality, follow these steps:

1. Navigate to **Window -> Preferences -> General -> Security**.
2. Select the **Use secure password storage** option.



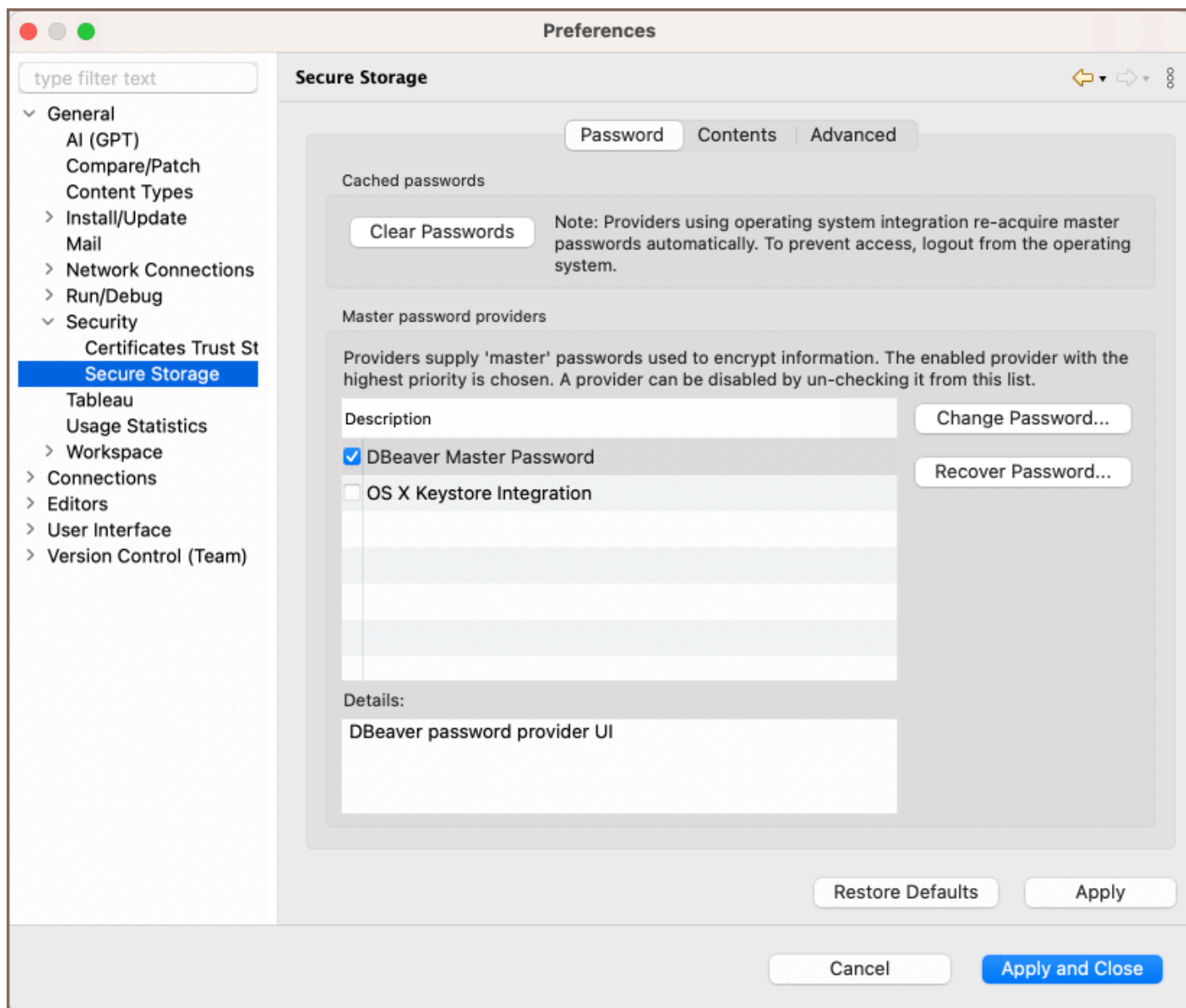
For further configuration, proceed to the **Window -> Preferences -> General -> Security -> Secure Storage**.

Administrator configuration of Master password policy

Administrators can customize various aspects of the Master password policy. This includes setting the minimum length of the password, the minimum count of letters, digits, and symbols required, and the enforcement of mixed case (uppercase and lowercase letters) in the Master password. For detailed instructions on configuring these settings, please refer to the [Admin Preference Restrictions](#) article.

Note: Currently, this functionality is limited to Windows users and is only accessible through the Windows Registry.

Configuring Master password



Under the **Secure Storage** section, accessible via the path **Window -> Preferences -> General -> Security -> Secure Storage**, in the **Passwords** tab, you have the following options:

Field	Description
Clear passwords	Clears the password cache, requiring re-entry of passwords upon next access to confidential information. This action prevents unauthorized use of stored credentials.
Choose a Master password provider	Select from various providers. Options include:
	OS X Keystore Integration for macOS users.
	Windows Integration for Windows users.
	DBeaver Master Password provider (default), which is available to all PRO users regardless of the

	<p>operating system.</p> <p>When using the DBeaver Master Password provider, you will be prompted to specify a master password.</p>
Change Passwords	<p>If a Master password has not been set, this option initiates the process of creating one. If a Master password is already in place, it allows for changing to a new master password.</p>
Recover password	<p>Available after setting a Master password, this feature uses password hints and recovery questions to help recover a forgotten master password.</p>

Important: DBeaver does not store the Master Password. If both the password and [password hints](#) are forgotten, the Master Password cannot be recovered or reset.

Tip: Choosing OS Integration options, like **OS X Keystore Integration** or **Windows Integration**, allows DBeaver to utilize your operating system's native security features. This process involves using a password that is automatically generated and stored in your local user's secure storage, as seen in Windows environments. This method simplifies the process as there's no need to remember a Master password, but be aware it may offer less security if others have access to your OS user account.

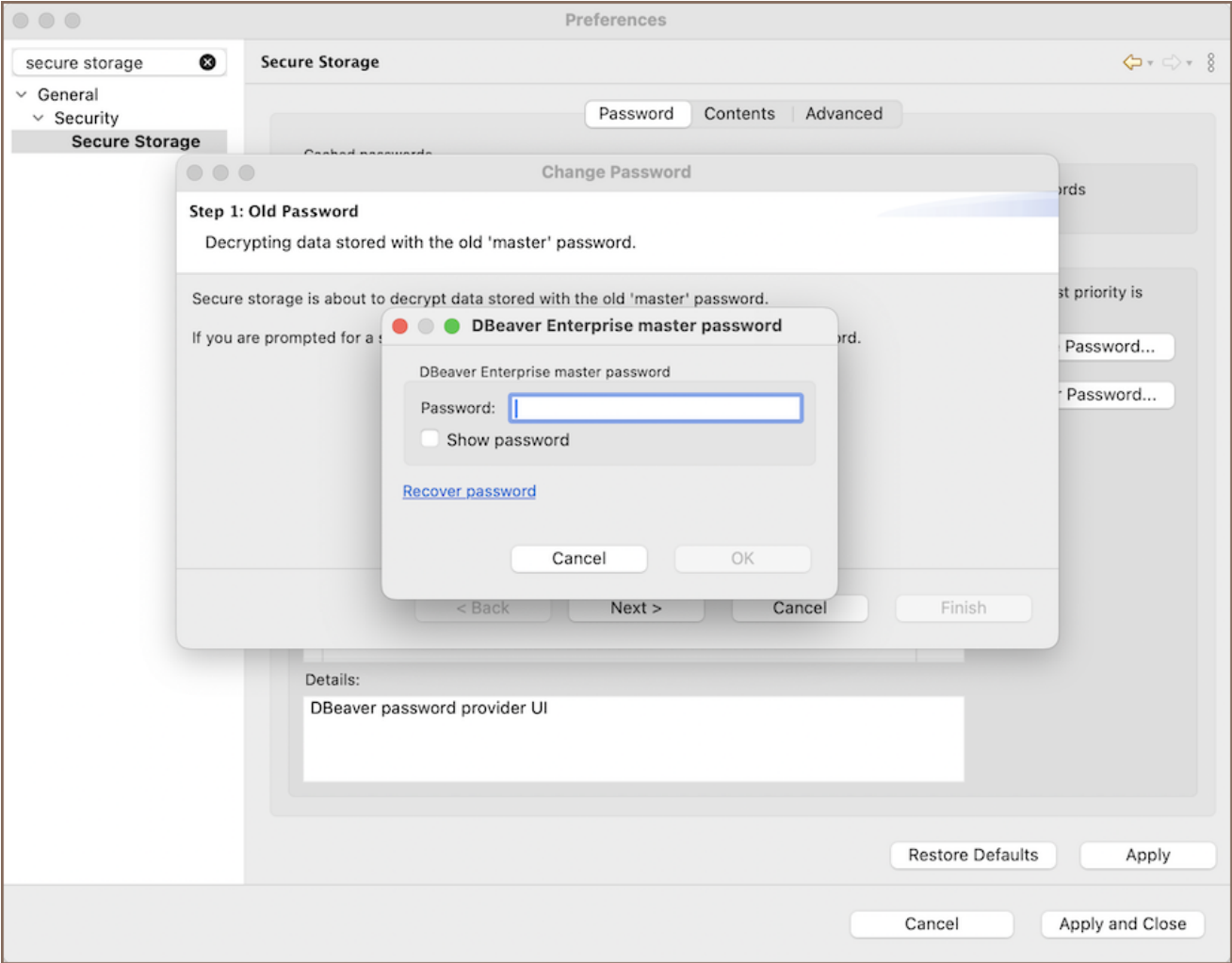
Changing the Master password

Important: When you change your Master password, the old one cannot be viewed or retrieved after recovery. Ensure you remember your new password or write it down in a secure place.

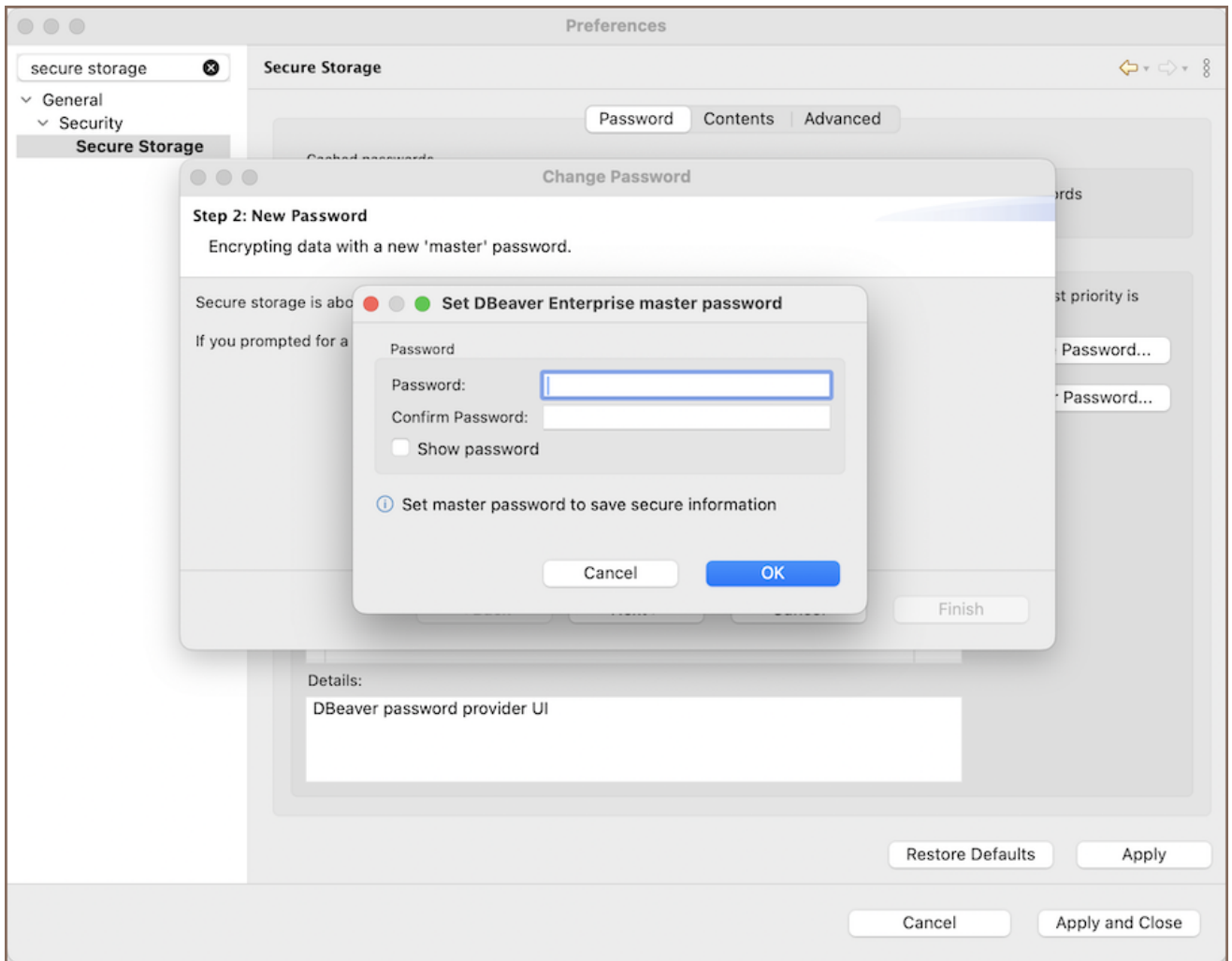
To change the Master password, follow these steps:

1. Go to **Window -> Preferences -> General -> Security -> Secure Storage**
2. Click on the **Change Password** button.

1. The **Step 1: Old Password** window will appear. Enter your current Master password and click **Next**.



2. The **Step 2: New Password** window will prompt you to enter a new Master password and confirm it. Fill in the fields and click **Next**.



3. After setting your new password, you will be asked to provide a [password hint](#). You may skip this step and press **Finish** to conclude the process.

Password recovery setup

To set up password recovery:

1. Go to **Window -> Preferences -> General -> Security -> Secure Storage -> Password Recovery**.
2. Specify security questions and provide answers for them. This step is essential for recovering a lost Master password.

Note:

- The questions and answers are case-sensitive.

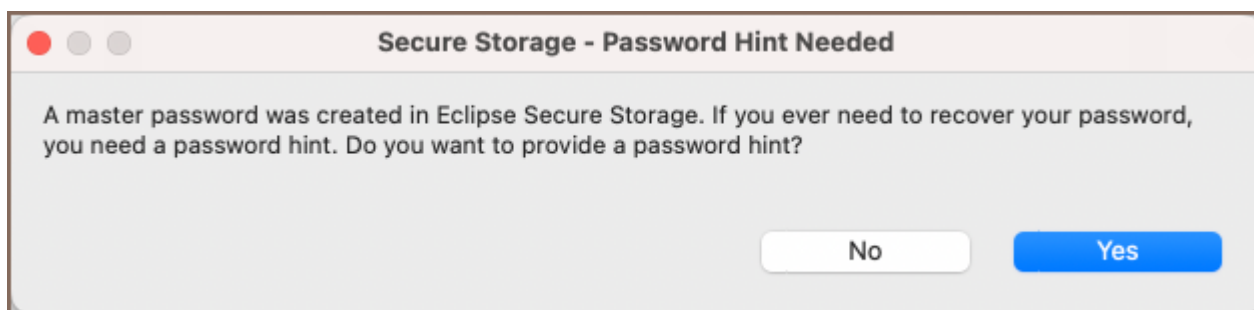
- Treat the answers as secondary passwords; they should be kept confidential and secure.

Setting a password hint

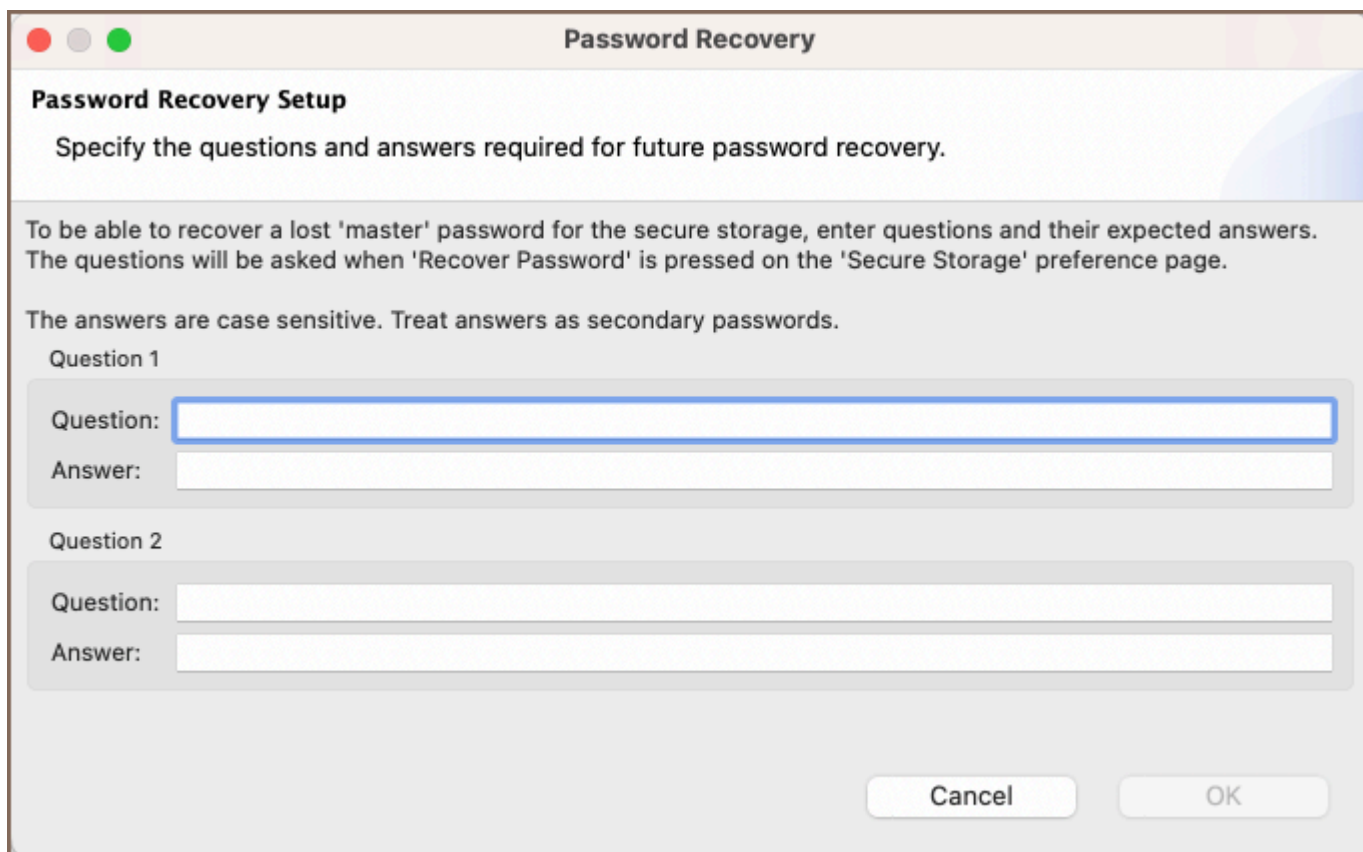
When you create a Master password in DBeaver's [Secure Storage](#), it is recommended to set up a password hint.

This hint will assist you if you need to recover your password.

1. When prompted, select **Yes** to provide a password hint after creating a Master password.



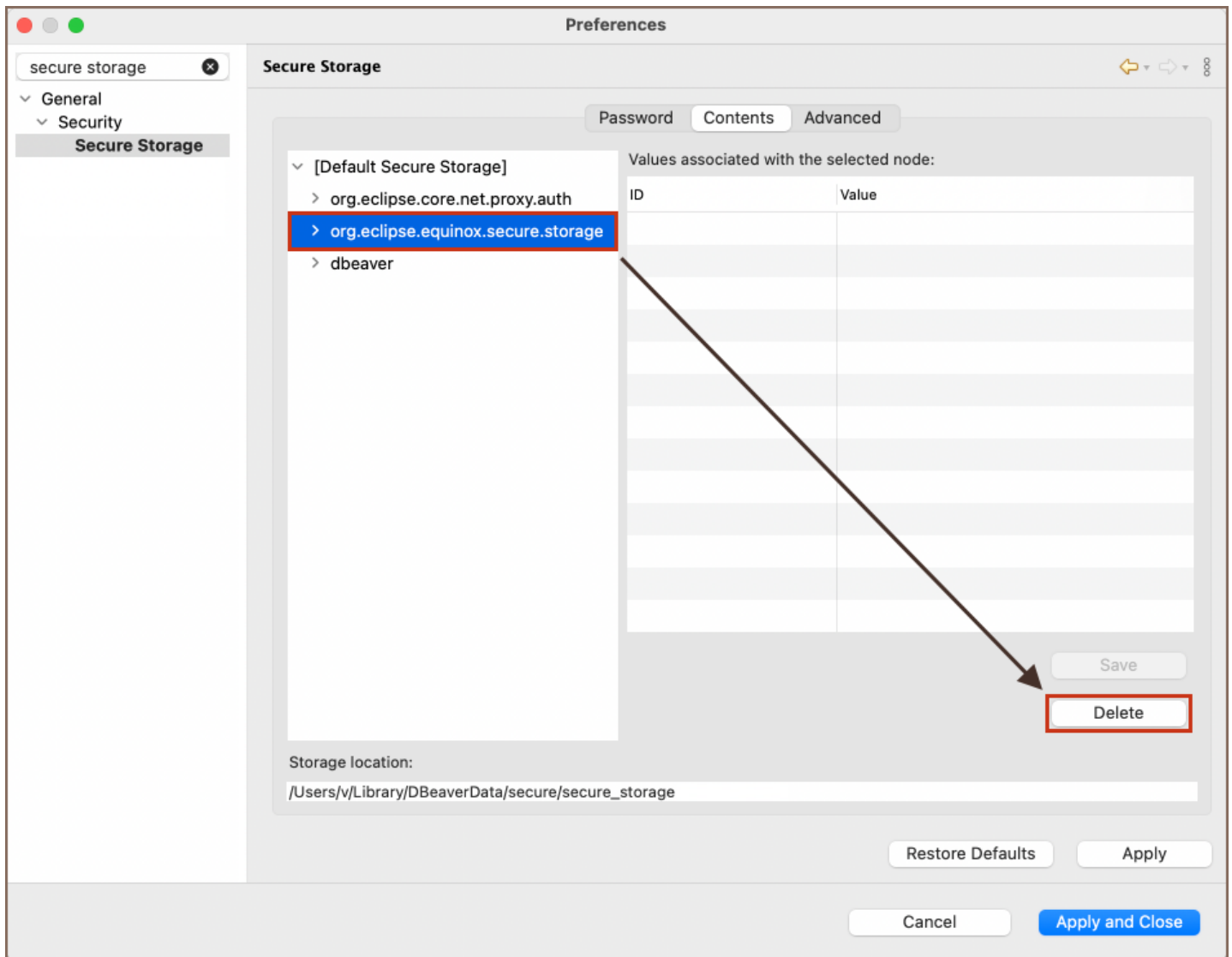
2. Enter a hint that will help you remember the password without revealing it directly.



Deleting the Master password

To remove the Master password, follow these steps:

1. Navigate to **Window -> Preferences -> General -> Security -> Secure Storage**.
2. Click on the **Contents** tab.
3. Locate and select `[Default Secure Storage]` -> `org.eclipse.equinox.secure.storage`.
4. Click on the **Delete** button.



Warning: Be aware that deleting the Master password will result in the loss of all saved credentials.

Viewing secured resources

Under the **Secure Storage** section, accessible via the path **Window -> Preferences -> General -> Security -> Secure Storage**, in the **Contents** tab, you may:

- Review the resources that are secured. This includes resources protected by the Master Password and those secured through integrated operating system (OS) security mechanisms.
- Delete entries if necessary to remove security constraints from the resources.

Advanced encryption settings

In the **Advanced** tab, located within the **Secure Storage** section, which can be found by navigating through **Window** -> **Preferences** -> **General** -> **Security** -> **Secure Storage**, you have the option to set the encryption algorithm to use for new storages to maintain data integrity and confidentiality.

Security in DBeaver PRO

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Master password](#)

[Changing database password](#)

[Password protection for Projects](#)

[Master password for all your Projects](#)

[Password for one project](#)

[Secure authentication](#)

[Kerberos support](#)

[SSO authentication](#)

[Predefined connections](#)

[Connections import](#)

[Read-only connections](#)

[Users roles and permissions](#)

[Configuring preferences](#)

[Roles in Team Edition](#)

[Centralized automatic updates](#)

[License management](#)

The level of security is one of the key questions for enterprises, and the DBeaver team pays a lot of attention to it. One of the best reasons to use PRO versions is to take advantage of its security tools and features, such as password protection, SSO authentication, teams and roles in Team Edition.

This article briefly describes the most important security options available in DBeaver PRO.

Master password

You can enhance security in DBeaver with the Master Password, which encrypts credentials and sensitive data using a secure storage system. This feature ensures that each user's credentials are individually protected by their

own Master Password, tying the encryption specifically to their local environment. For more details, refer to the [Managing Master Password](#) article.

Changing database password

Users can change the current database password directly in DBeaver in the following databases:

- Cockroach
- Exasol
- Greenplum
- Netezza
- Oracle
- PostgreSQL
- Redshift
- Snowflake
- SQL Server
- Vertica.

Oracle, PostgreSQL, and Netezza allow changing an expired password in DBeaver as well.

- [How to change the user password](#)

Password protection for Projects

Master password for all your Projects

You can protect all [Projects](#) in your local workspace with a Master password.

You can set this password and store it in DBeaver password provider or use a generated password from your local password provider (for instance, OS X Keystore Integration or Windows integration provider).

[Learn more about the Master password for Projects](#)

Password for one project

You can specify a password for any project to protect all the project's configurations.

[Learn more about project password](#)

Secure authentication

You can connect to databases using secure authentication via Kerberos or GCP, AWS, and Azure cloud services.

Kerberos support

Kerberos is an authentication protocol, the default authentication technology used in Microsoft Windows.

You can connect via Kerberos using keytab, kinit, or a password. Open the connection settings, choose one of the [supported databases](#) and select Kerberos as the authentication method.

[Learn more about authentication via Kerberos](#)

SSO authentication

Users can connect to all company services using only one login and password. This is possible if you use SSO - Single Sign-On authentication service.

You do not need to manage, store, and transfer user credentials. When a user connects to the database, DBeaver opens a web browser with SSO authentication.

DBeaver supports the following SSO authentication services:

- [AWS SSO](#)
- [GCP SSO](#)
- [Azure](#)

Predefined connections

Connections import

You can describe all available database connections in [configuration files](#) (in JSON format) or [import from CSV or XML](#) DBeaver.

Read-only connections

If you want to restrict users from editing connection parameters, you can [protect them with passwords](#).

Users roles and permissions

Configuring preferences

You can customize users preferences before they run DBeaver. For example, you can set the default [simple mode](#) for all connections (to show only schemas and tables and hide all system and service objects).

[How to manage preferences](#)

Roles in Team Edition

The best way to manage user access, restrictions, and permissions is to use [Team Edition](#).

Team Edition allows you to create users and assign them appropriate roles with predefined capabilities.

You can add Viewers and Editors to work with prepared data, Managers to prepare data for them, Developers to work with scripts and connections, and administrators to manage everything.

[Learn more about Team Edition](#)

Centralized automatic updates

If your team works on Microsoft Windows, you can organize DBeaver mass updates in silent mode, without user input, using the Windows Installer command line options.

[Learn more about silent install](#)

License management

You can place the license file in the user's workspace or store it elsewhere, and specify the license path on the command line or in the DBeaver configuration file.

- [Learn more about license management automation](#)
- [How to specify license path in config](#)

SSH configuration

Table of contents

[Overview](#)

[Configuring SSH Tunnels](#)

[Configuring Jump servers](#)

[Practical use](#)

[Advanced options](#)

[Port Forwarding](#)

[SSH and URL based connections](#)

[SSH shared tunnels](#)

[SSH tunnel sharing criteria](#)

[Configuring Cloud SSH Tunnels](#)

Overview

Sometimes the database cannot be accessed directly, in such cases you can use the SSH tunnel. The SSH or Secure Shell is a network communication protocol that allows two computers to communicate and encrypts the shared data.

DBeaver supports the following SSH authentication methods:

- Username/password
- Public key authentication
- Agent authentication

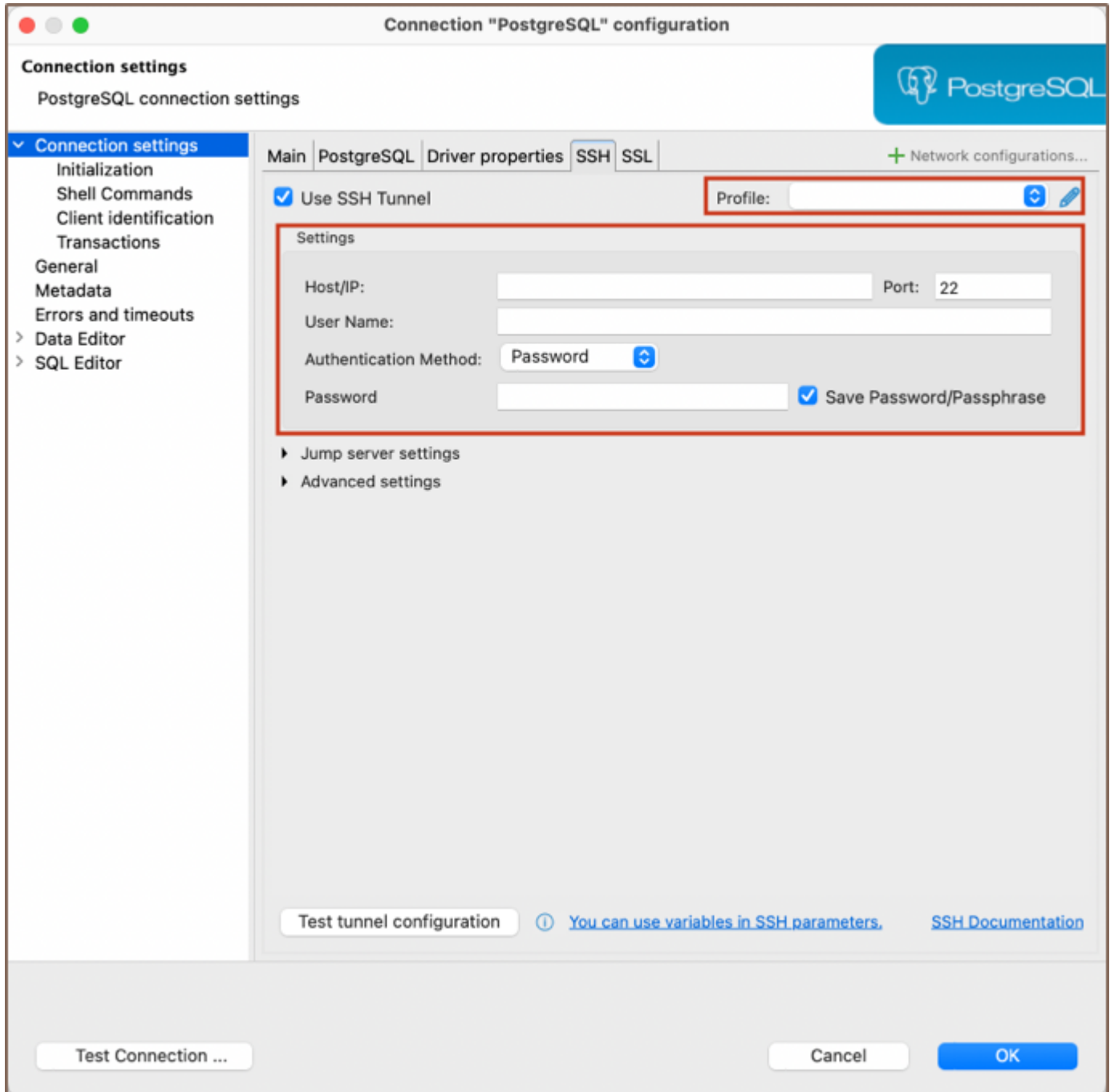
Available agent authentication implementations:

- [pageant](#) (*Windows only*)
- [ssh-agent](#)

Note: You have to run the agent first before connecting to the database in DBeaver.

Configuring SSH Tunnels

SSH can either be configured individually for each connection in **Connection settings** -> **SSH** or as a network profile which can be selected from the drop-down menu on the same page mentioned earlier:



In order to use **SSH**, you have to enable it first. After that, configure it for your needs, then click on **Test tunnel configuration** to test whether all parameters are valid or not.

It is often required to set the hostname to `localhost` in **Connection settings** -> **Main**. SSH establishes a connection between two machines, authenticating each side to the other, and passing commands and output back and forth. After connecting to remote machine, all commands you execute are executed on that remote machine, so by having the host set to `localhost` you are opening a connection to the database on this remote machine, not on your local machine, and just redirect I/O to the latter.

Configuring Jump servers

DBeaver allows the use of Jump servers, also known as Gateway hosts. This feature is essential when the direct access to a target machine is not possible from your local machine, and a gateway server is required for connection.

Connect to a database

Connection Settings
PostgreSQL connection settings

Main | PostgreSQL | Driver properties | **SSH** | SSL

☒ Use SSH Tunnel Profile:

Settings

Host/IP: Port:

User Name:

Authentication Method:

Password: ☒ Save credentials

Jump servers


Order	Host	User	Authentication
Jump #1	host.com:22	user_1	Password
Target	example.com:22	user_name	Password

Advanced settings

[You can use variables in SSH parameters.](#) [SSH Documentation](#)

To add a jump server:

1. Ensure **Use SSH Tunnel** is checked.

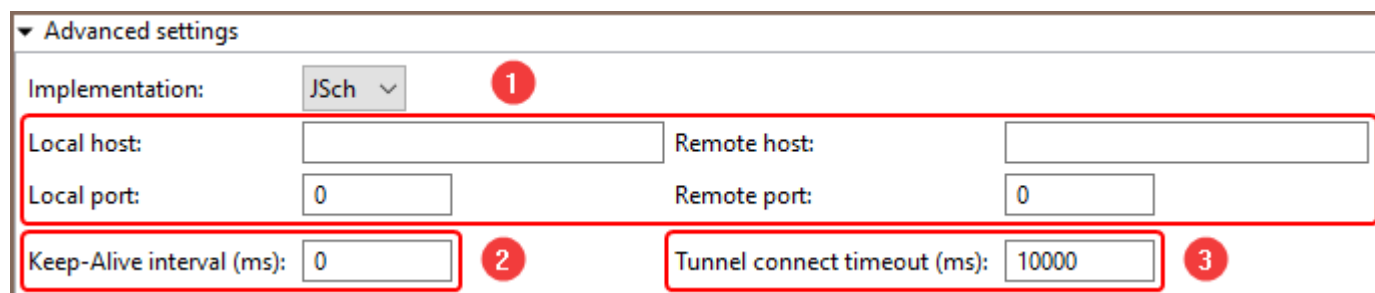
2. Expand the **Jump servers** group.
3. Specify your target in the **Settings** section.
4. Click on the **Create new jump host** button  to add your first jump server configuration.

Repeat step 3 for each additional jump host you need to add, configuring each one within the **Settings** section.

Practical use

Imagine you have the following situation: your database is located on remote machine with IP address `100.100.100.100` and which is accessible through SSH. You cannot access this IP directly from your machine because of your network settings or firewall, but you can access other machines with the IP address `200.200.200.200` through SSH and which **is able** to access the desired remote machine. You will need to specify `100.100.100.100` as a host in regular SSH configuration and `200.200.200.200` as a host in a gateway configuration to achieve such a "connection order": `localhost` → `200.200.200.200` → `100.100.100.100`.

Advanced options



Advanced settings

Implementation: JSch 1

Local host: Remote host:

Local port: Remote port:

Keep-Alive interval (ms): 2 Tunnel connect timeout (ms): 3

1. Optional port forwarding configuration, see information at [below](#).
2. Keep-alive interval. Can be used to "ping" a remote machine to make sure that the SSH connection is still alive.
3. Connection timeout. If the destination SSH machine is quite slow and the connection takes ages to complete, you can adjust the value of this parameter.

Port Forwarding

Port forwarding is used to expose some socket sitting on a remote port to your local port, allowing you to communicate with it. After the SSH connection to the remote machine is established, you now need to connect to

the database on that machine. Now you need to pass the port on which the database is listening to from that remote machine to your local machine. This is when port forwarding becomes handy.

By default, it randomly chooses the local port and forwards it to the destination port set in `Connection setting`
⇒ `Main`

It is not usually required to change this configuration, but it can be useful e.g. when you want to use a particular local port.

SSH and URL based connections

By default DBeaver opens random local port and forwards it to a target database port. It then implicitly replaces port configured in the connection with this random port.

So technically DBeaver will connect to something like `localhost:45678` (random port on localhost) instead of `database.server.domain:5432` (default PostgreSQL server port).

But if you use URL instead of manual connection configuration then you connect to

`jdbc:postgresql://database.server.domain/postgresql`. DBeaver cannot replace port number automatically because you can use any type of sophisticated URLs. DBeaver just don't know how to change this URL.

If you use URL connection then you must configure port forwarding manually. In the Advanced Settings set both Local Port and Remote Port to 5432 (database server port, 5432 is the default value for PostgreSQL). It will work in most cases.

But it will fail if you have PostgreSQL server on your local machine because port 5432 is already occupied. In this case you can use any random port number for Local Port (e.g. 15432). Use the same 5432 for Remote Port because it is real port number for remote PostgreSQL server.

SSH shared tunnels

By default, DBeaver shares SSH tunnels across connections when the same SSH settings are used. This is particularly useful when connecting to several databases on the same remote server, as it avoids the need to establish a separate SSH tunnel for each connection.

To view the active SSH tunnels and the databases utilizing a shared tunnel, use the **SSH tunnel explorer**. This feature provides a comprehensive overview of your SSH tunnel usage, including detailed information on each tunnel and the databases it serves. Here is how to access **SSH tunnel explorer**:

1. Navigate to **Window -> Show View -> Other...**
2. In the window that appears, expand the **Database** category.
3. Select **SSH tunnel explorer** and click **Open**.

In the **SSH tunnel explorer** view, you will find information about each active tunnel. This includes:

Field	Description
Destination	The remote server to which the tunnel connects.
Used by	A list of databases that are currently using the tunnel.
Port Forwards	The local and remote ports involved in the tunneling process.

SSH tunnel sharing criteria

DBeaver shares SSH tunnels across connections if SSH settings are identical. The application compares the following parameters to determine if the SSH settings match:




- **Hostname:** The address of the SSH server.
- **Host port:** The port number used for the SSH connection.
- **Username:** The username required for SSH authentication.

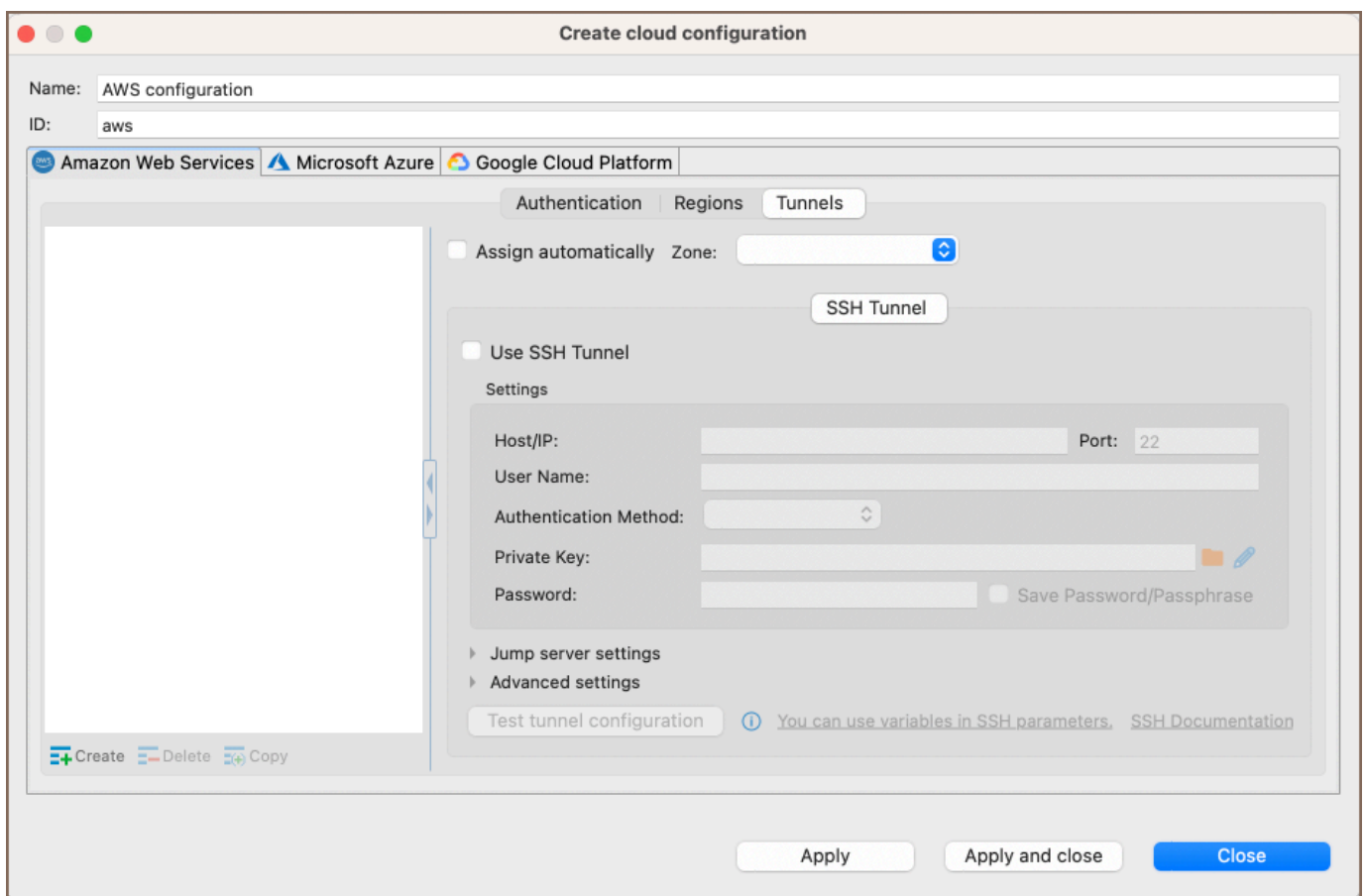
Depending on the authentication method selected, DBeaver also compares:

- For password authentication:
 - **Password:** The SSH account password.
- For public key authentication:
 - **Private Key:** The file path to the SSH private key.
 - **Passphrase:** The passphrase for decrypting the private key, if set.

- For agent authentication:
 - **Agents:** DBeaver treats all agent authentication configurations as identical, ignoring specific agent settings and focusing only on the method's selection.

Configuring Cloud SSH Tunnels

- To configure Cloud SSH Tunnels, start by navigating to the **Cloud Explorer** button  in the Toolbar.
- Select **Edit** button  for an existing connection or **New** button  to create a new connection.
- In the opened window, go to the **Tunnels** tab.



- This tab allows you to set up SSH tunnels specifically for cloud connections.
- The configuration here is similar to the standard SSH setup, but it's integrated within your cloud service provider's

SSL configuration

Table of contents

[Overview](#)

[Setting up SSL configuration](#)

[Setting up SSL configuration via Driver properties](#)

[Setting up SSL configuration for Oracle connections](#)

[Troubleshooting SSL issues](#)

Overview

DBeaver supports the ability to establish connections using Secure Socket Layer (SSL), a protocol for creating authenticated and encrypted links between networked computers.

Note: SSL parameters are optional and should only be specified if required by your server configuration.

Setting up SSL configuration

To set up SSL, you need to configure the following settings:

1. Navigate to the **SSL** tab within the **Connection settings**.
2. Enable the **Use SSL** option to activate SSL for your connection.

Connect to a database

Connection Settings

PostgreSQL connection settings

PostgreSQL

Main

PostgreSQL

Driver properties

SSH

SSL

+ Network configurations...

☒ Use SSL

Profile:

All SSL parameters are optional.
You must specify SSL certificates if they are required by your server configuration.

Parameters

CA Certificate:

Client Certificate:

Client Private Key:

Advanced

SSL mode:

SSL Factory:

Test Connection ...

< Back

Next >

Cancel

Finish

3. Enter the necessary SSL configuration details (optional):

Parameter	Description
CA Certificate	Path to the Certificate Authority (CA) certificate.
Client Certificate	Path to the client's public key certificate.
Client Private Key	Path to the client's private key.
Keystore	Path to the keystore file.
Keystore Password	Password for the keystore.
Certificate hostname	Expected hostname in the certificate.
SSL Mode	Desired security level for SSL.
SSL Factory	Provide SSLSocketFactory factory class name.
Cipher suites (optional)	Specific cipher suites to use.
Skip host name validation	Whether to bypass hostname validation.
Use self signed certificate (non-secure)	Acceptance of self-signed certificates.
Force TLS 1.2	Enforce using TLS version 1.2.

Verify server certificate	Ensures the server certificate is valid.
Allow public key retrieval	Allows the retrieval of the server's public key.

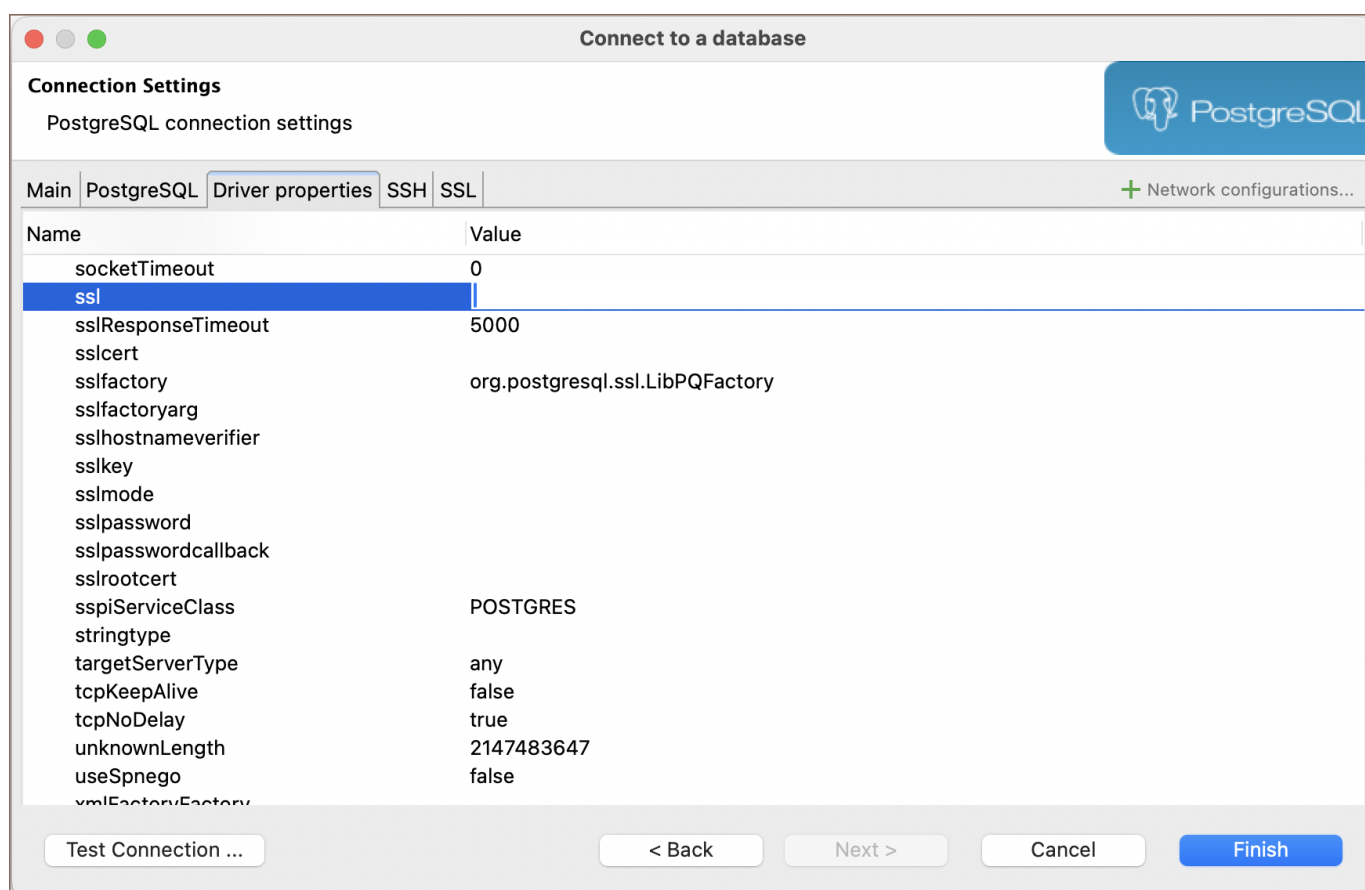
Note: The availability of these parameters can vary depending on the database. Check the SSL tab for your database within the **Connection settings** for any specific options.

- If all other necessary connection settings are configured, complete the setup by clicking **Finish**, or test the configuration by clicking **Test Connection** to ensure that SSL is functioning correctly.

Setting up SSL configuration via Driver properties

For some databases, an alternative method to configure SSL settings is available through the **Driver properties** tab. This approach allows access to additional or driver-specific SSL options that may not be present in the main SSL tab:

- Select the **Driver properties** tab within the **Connection settings**.



- Look for properties prefixed with **ssl**, such as **sslResponseTime**, **sslfactoryarg**, **sslhostnameverifier**, etc. Configure the necessary properties according to your database and server requirements.

3. If all other necessary connection settings are configured, complete the setup by clicking **Finish**, or test the configuration by clicking **Test Connection** to ensure that SSL is functioning correctly.

Setting up SSL configuration for Oracle connections

Prerequisites:

- **Oracle Client** installed.
- An **Oracle Database Server** that supports SSL.
- **Access rights** to create directories and files.
- A valid **SSL certificate** from a trusted CA.
- **Oracle Wallet Manager** for wallet management.

To set up SSL configuration for Oracle connections, follow these steps:

1) Create a Wallet Folder:

First, you need to create a directory that will serve as a wallet folder. This wallet stores your security credentials.

Execute the following command to create this folder:

```
mkdir /opt/oracle/wallet
```

2) Generate Key:

Use the **orapki** utility to create a new wallet and generate a self-signed certificate. Create the wallet:

```
$ORACLE_HOME/bin/orapki wallet create -wallet /opt/oracle/wallet -pwd YourWalletPassword -auto_login
```

Replace **YourWalletPassword** with a secure password for accessing the wallet.

3) Add a self-signed certificate to the wallet:

When setting up your Oracle wallet, you may need to add a certificate. There are two main scenarios for this:

1. **Creating and adding a self-signed certificate:** If you do not have a certificate, you can create a self-signed certificate and add it to your wallet using the following command:

```
$ORACLE_HOME/bin/orapki wallet add -wallet /opt/oracle/wallet -pwd YourWalletPassword -dn "CN=example.com"
```


2. **Adding a certificate signed by a certificate authority (CA):** If you use a certificate signed by a Certificate Authority (CA), you need to add the Root Certificate Authority (RootCA) certificate to your wallet as well. This is done using the following command:

```
$ORACLE_HOME/bin/orapki wallet add -wallet /opt/oracle/wallet -pwd YourWalletPassword -trusted_cert -cert
```

Replace the following placeholders:

YourWalletPassword : Use the same wallet password as before.

CN=example.com : Replace this with your server's fully qualified domain name (FQDN) or the hostname.

path_to_root_ca_cert : This is the actual path to your **RootCA** certificate file:

4) Generate JKS File:

Convert the wallet to Java Keystore format.

```
$ORACLE_HOME/bin/orapki wallet pkcs12_to_jks -wallet /opt/oracle/wallet -pwd YourWalletPassword -jksKeySto
```

Replace the following placeholders:

YourWalletPassword : Use the same wallet password as before. **YourJKSPassword** : This is a new password for the Java Keystore files. Choose a secure password.

5) Modify configuration files for SSL in Oracle

listener.ora : Configure the listener to recognize SSL connections. For instance:

```
SSL_CLIENT_AUTHENTICATION = FALSE

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /opt/oracle/wallet)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION=
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
      (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))
    )
    (DESCRIPTION=
      (ADDRESS = (PROTOCOL = TCPS)(HOST = 0.0.0.0)(PORT = 2484))
    )
  )

DEDICATED_THROUGH_BROKER_LISTENER=ON
DIAG_ADR_ENABLED = off
```

sqlnet.ora : Configure SSL parameters and wallet location. Example:

```

WALLET_LOCATION =
  (SOURCE =
    (METHOD = FILE)
    (METHOD_DATA =
      (DIRECTORY = /opt/oracle/wallet)
    )
  )

SQLNET.AUTHENTICATION_SERVICES = (TCPS,NTS,BEQ)
SSL_CLIENT_AUTHENTICATION = FALSE
SSL_CIPHER_SUITES = (SSL_RSA_WITH_AES_256_CBC_SHA, SSL_RSA_WITH_3DES_EDE_CBC_SHA)

```

`tnsnames.ora` : Defines connection aliases. Example:

```

SSL=
(DESCRIPTION=
  (ADDRESS = (PROTOCOL = TCPS)(HOST = 0.0.0.0)(PORT = 2484))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)

XE=
(DESCRIPTION=
  (ADDRESS = (PROTOCOL = TCP)(HOST = 0.0.0.0)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)

```

Replace placeholders like `0.0.0.0` , `2484` , and `XE` with your actual server `IP` , `SSL` port, and `service name` .

6) Restart listener To apply the changes made to the `listener.ora` file, you need to restart the Oracle listener.

Use the following commands:

```

$ORACLE_HOME/bin/lsnrctl stop
$ORACLE_HOME/bin/lsnrctl start

```

7) Restart Oracle database.

After updating the SSL configuration, restart the Oracle Database to ensure all settings are applied:

```

$ORACLE_HOME/bin/sqlplus / as sysdba
shutdown
startup

```

This sequence shuts down the database and then starts it up again.

Tip: If your Oracle Database runs in a Docker container, you can restart the container to apply the configuration changes - `docker restart oracle_container_name` . Replace `oracle_container_name` with the name of your Oracle Docker container.

8) Open DBeaver and initiate the process to create a new database connection. If unsure how to start, refer to the article on [creating a connection](#).

9) Specify the **Custom JDBC URL**:

Connect to a database

Oracle Connection Settings
Oracle connection settings

Main | Oracle properties | Driver properties | SSH | [+ Network configurations...](#)

Connection Type:

Basic | TNS | **Custom**

JDBC URL Template: jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=0.0.0.0)(PORT=2484))(CONNECT_DATA=(SERVICE_NAME=XE)))

Authentication

Authentication: Oracle Database Native

Username: system Role: Normal

Password: ☒ Save password

Client: <not present>

[You can use variables in connection parameters.](#) Connection details (name, type, ...)

Driver name: Oracle Driver Settings Driver license

Test Connection ... < Back Next > Cancel Finish

In the **Main** tab, construct the JDBC URL using the SSL configuration.

Example:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=0.0.0.0)(PORT=2484))(CONNECT_DATA=(SERVICE_NAME=XE)))
```

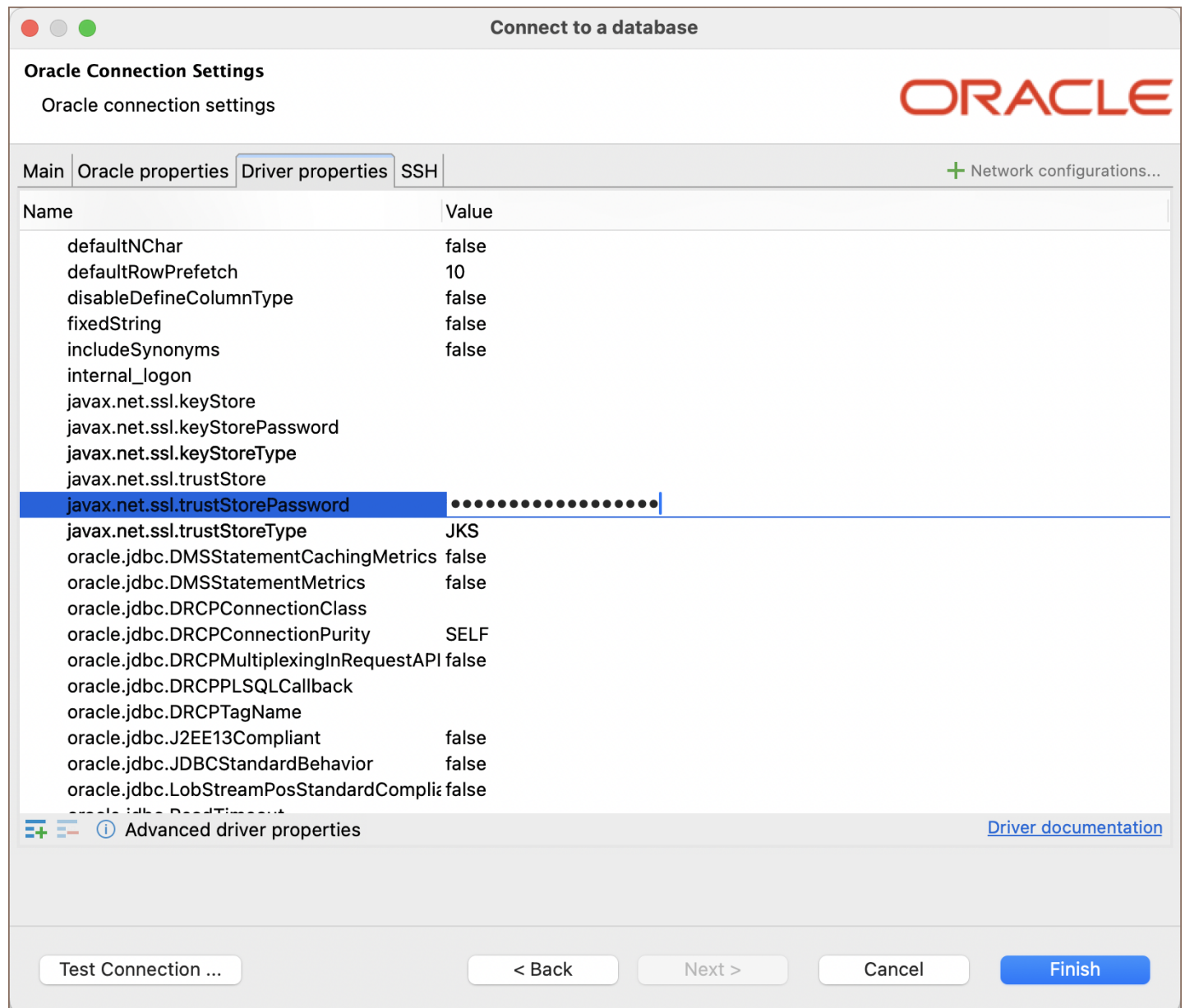
Replace **0.0.0.0** , **2484** , and **XE** with your actual server's hostname/IP, SSL port, and service name as specified in your **tnsnames.ora** .

10) Move to the **Driver properties** tab within the connection setup and fill in the following properties:

javax.net.ssl.trustStorePassword : Enter the password for your truststore. This property secures access to the truststore.

`javax.net.ssl.trustStoreType` : Set this to `JKS` to indicate the type of your truststore.

`oracle.net.wallet.location` : Provide the location of your Oracle wallet. This is the directory path where your wallet is stored, as specified in your configuration files.



11) After configuring all these properties, you can test the connection to verify that the SSL setup is correct and that you can successfully connect to your Oracle database over SSL.

Troubleshooting SSL issues

If you encounter a `Pkix path building failed` error, it may be due to incorrect SSL configurations. A common resolution is to consider ignoring host name validation, an option available for some databases in the **SSL** tab of the Connection settings, which can help bypass common SSL errors.

Proxy configuration

Table of contents

[External resources access](#)

[How to configure a proxy for drivers download](#)

[How to configure network for license activation](#)

[How to configure a proxy for external databases access](#)

External resources access

Sometimes DBeaver needs to access external internet resources for tasks such as:

- 3rd party JDBC drivers download
- Information about a new DBeaver version
- Connect to remote databases outside of your corporate network
- Subscription license activation (commercial version)
- License information update (commercial version)

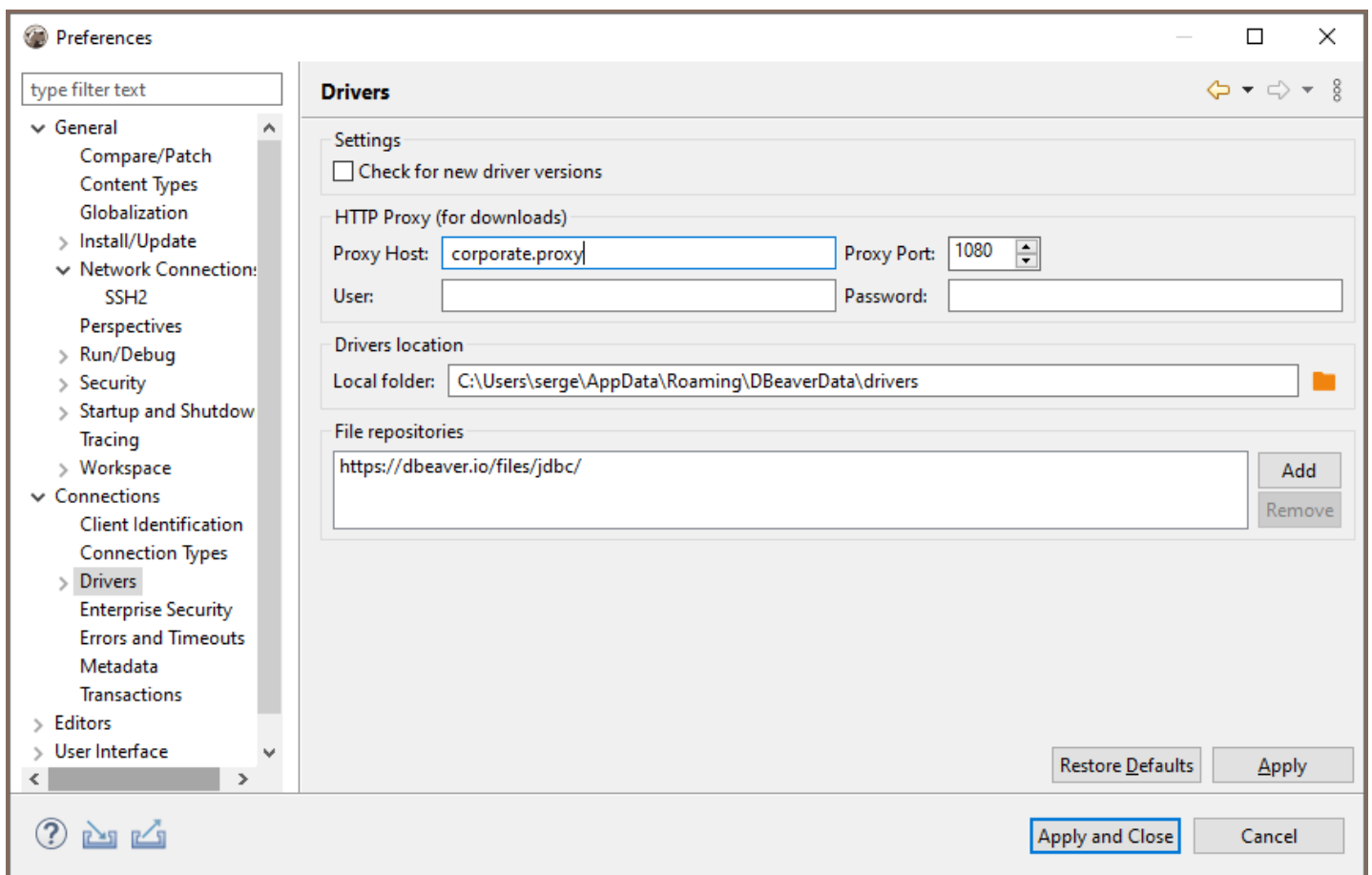
If you are behind some corporate firewall which restricts access to external internet resources then it may become a real problem.

Sometimes corporate firewalls allow access to external resources using a web browser but restricts this for all other applications.

How to configure a proxy for drivers download

You need to ask your network administrator about proxy parameters.

Then go to Preferences->Connections->Drivers.



You can enter a proxy host/port and (optionally) user/password here. It will be used for drivers download only.

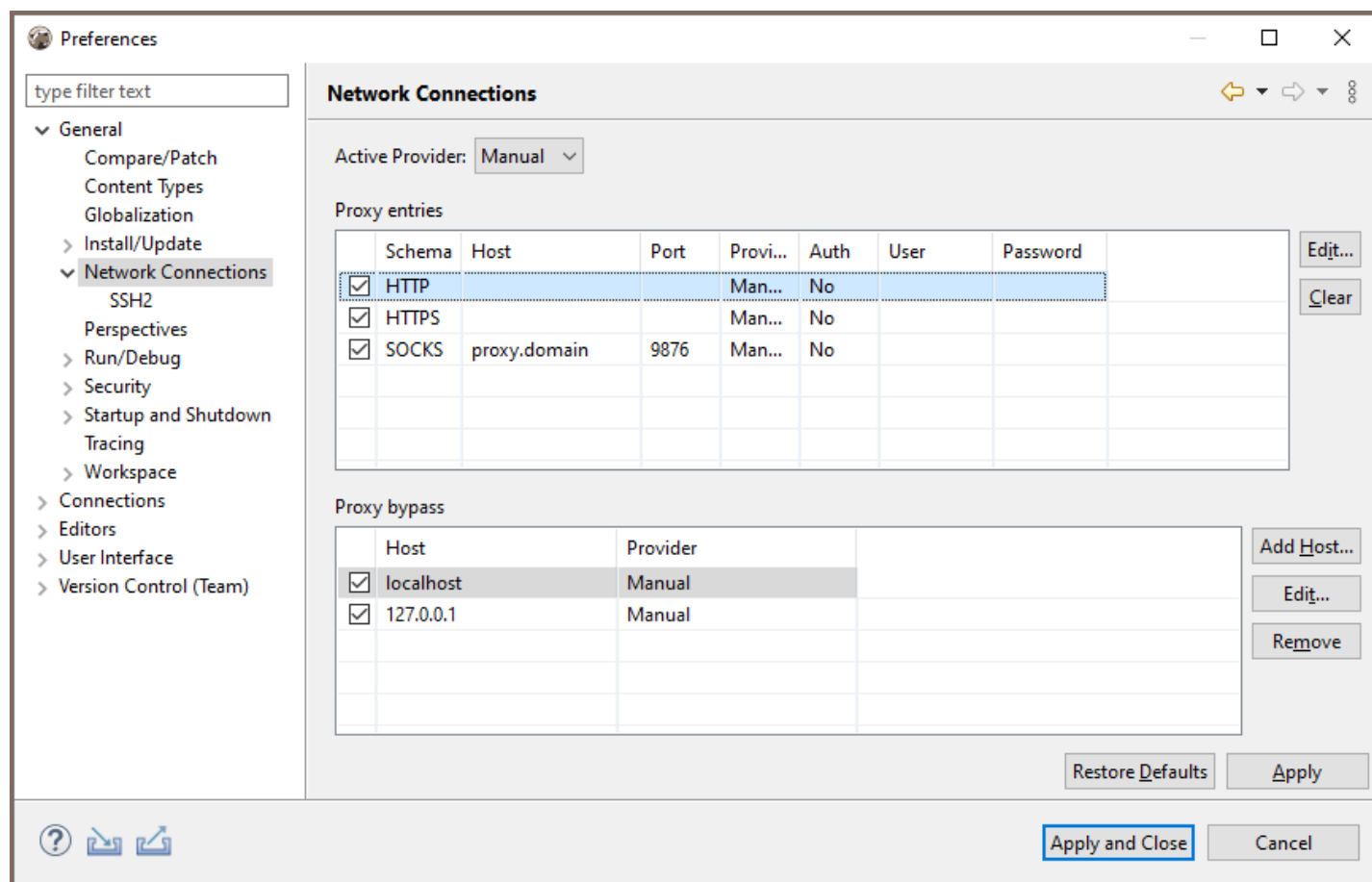
Drivers are usually downloaded from maven.org web site. You may also ask your network admin to add `maven.org` to the list of allowed external domains.

How to configure network for license activation

You need to configure a global proxy server.

If you cannot activate your subscription license then you first need to use a trial version to start DBeaver and configure a proxy.

Go to Preferences->General->Network Connections:



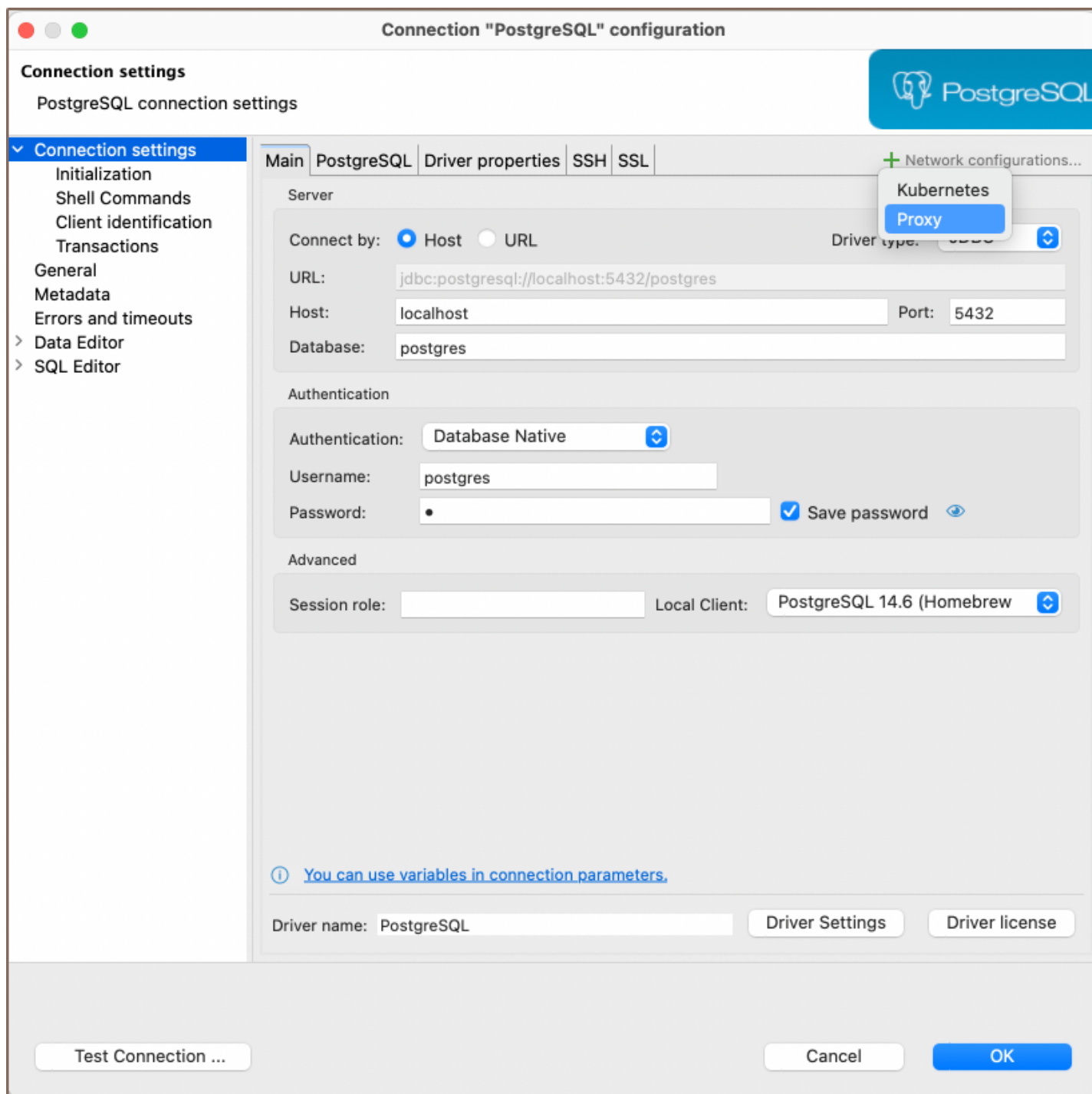
Switch to Manual or Native proxy (native proxy settings will use an active web browser proxy configuration).

Note: in order to activate/update a license DBeaver only needs to access the website dbeaver.com. You may ask your network administrator to add dbeaver.com to the white list.

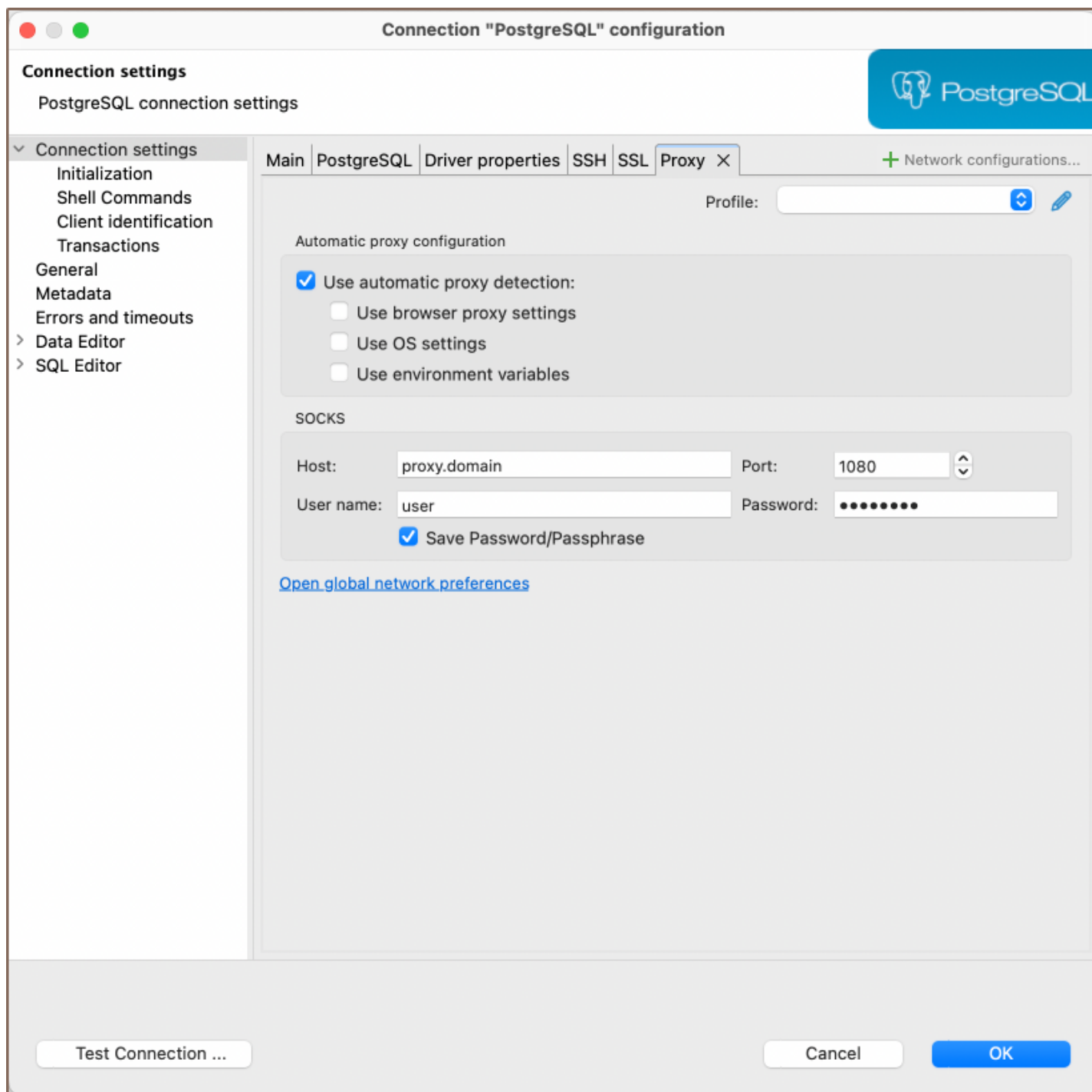
How to configure a proxy for external databases access

You can configure proxy settings for an individual connection.

1) **Add a Network Handler - Proxy:** To do this, click on the **Network configurations...** button and select **Proxy**.



2) **Set the Proxy Settings:** After selecting **Proxy**, a **Proxy** tab will appear in the menu at the top. You may set the proxy settings manually or use the active OS/web browser settings from this tab.



Note: Automatic proxy configuration is available in [Lite](#), [Enterprise](#), and [Ultimate](#) editions only.

Kubernetes configuration

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Setting up Kubernetes](#)

[Kubectl settings](#)

[Testing port forwarding](#)

In some scenarios, you may need to interact with databases that are part of a Kubernetes cluster. DBeaver offers a way to configure Kubernetes settings to facilitate this. Before proceeding, ensure that you meet the following prerequisites:

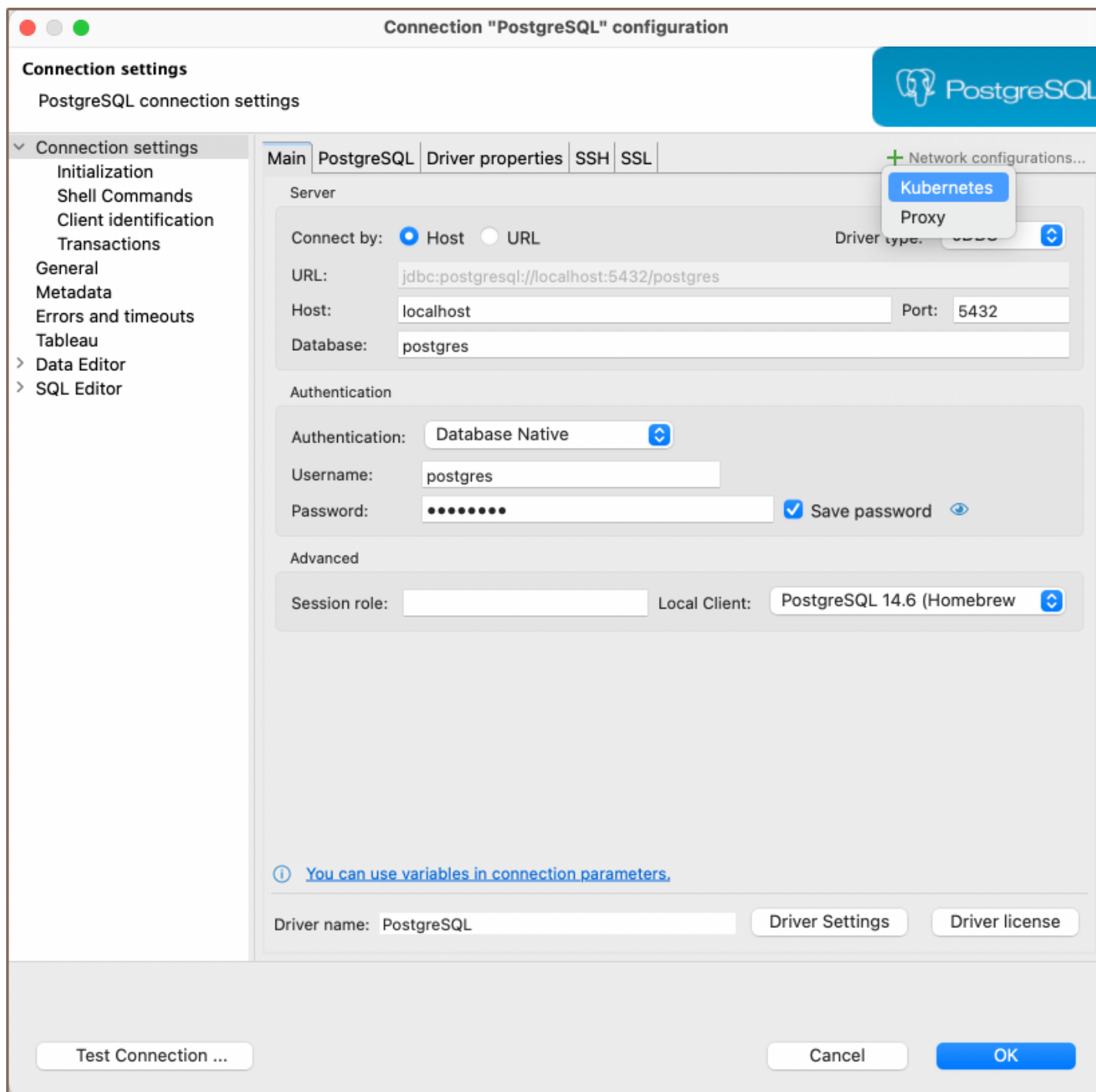
You must have the [kubectl](#) tool installed on your local machine.

You need a Kubernetes [configuration file](#). Use the local one or consult your administrator for access.

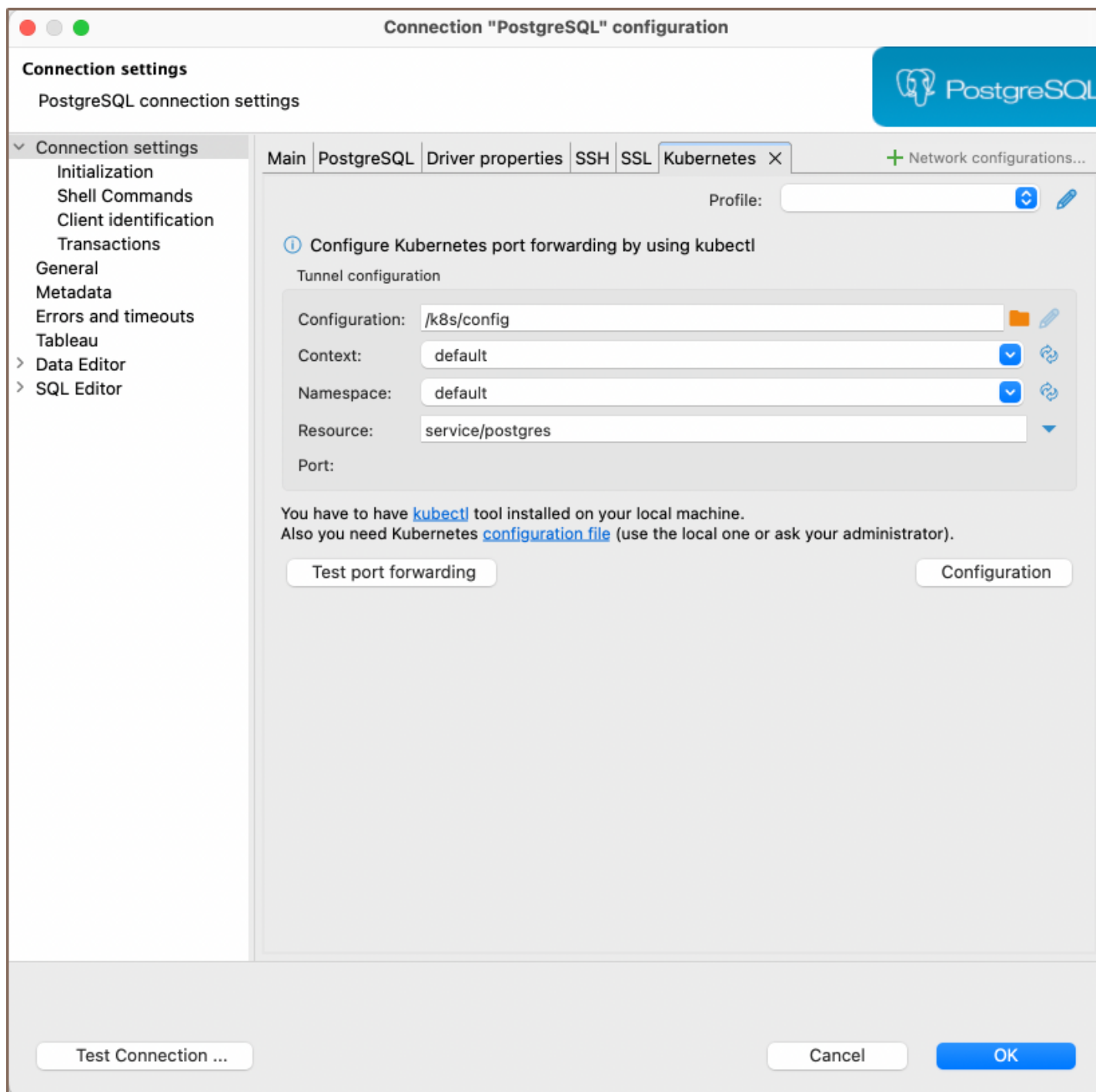
Setting up Kubernetes

To set up a tunnel to a Kubernetes cluster in DBeaver, you need to configure the following settings:

1) First, you need to add a Network Handler for Kubernetes. To do this, click on the **Network configurations...** button and select **Kubernetes**.




2) After making your selection, a new tab will appear in the menu at the top, where you can proceed to configure the various settings.



Here you need to fill out the following fields:

Field	Description
Configuration	Specify the path to your Kubernetes configuration file.
Context	Manually select a kube-context if your configuration offers multiple contexts for different clusters. The default setting is <code>default</code> .
Namespace	The default value is <code>default</code> , but this may differ in managed Kubernetes environments. Obtain the correct namespace from your <code>kubectl</code> config.

Resource	By clicking the Find Resource button  next to the field, a list of available services to connect to will open. From this list, you can select the service you wish to connect to.
Port	Configure Kubernetes port forwarding by using <code>kubectl</code> .

Tip: When clicking the dropdown menus in **Context** and **Namespace**, you can select settings directly from your `kubectl` config.

Kubectl settings

The Kubernetes Command-Line Tool, `kubectl` , is essential for interacting with Kubernetes clusters and is integral to the functioning of DBeaver's Kubernetes features. Learn more in the [official Kubernetes documentation](#).

To configure `kubectl` settings, you have two options:

- 1) Click on **Configuration** button on the Kubernetes connection settings page to access the `kubectl` settings.
- 2) Alternatively, navigate to **Window** -> **Preferences** -> **Connections** -> **Command Line Tools** -> **Kubernetes** to specify the path to your `kubectl` executable.

Testing port forwarding

The **Test port forwarding** button utilizes the `kubectl port-forward` feature to enable secure access to applications running inside Pods from your local machine. This provides a secure way to interact with the application without exposing it to the outside world. Learn more in the [official Kubernetes documentation](#).

To test port forwarding:

- 1) Fill out the necessary Kubernetes settings as described in the previous section.
- 2) Click on the **Test port forwarding** button to initiate the port forwarding process.

If the test is successful, this confirms that you can securely access the targeted Kubernetes resource from your local machine using the configured port settings.

AWS SSM configuration

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Configuring AWS CLI and Session Manager Plugin paths](#)

[Setting up AWS SSM](#)

Overview

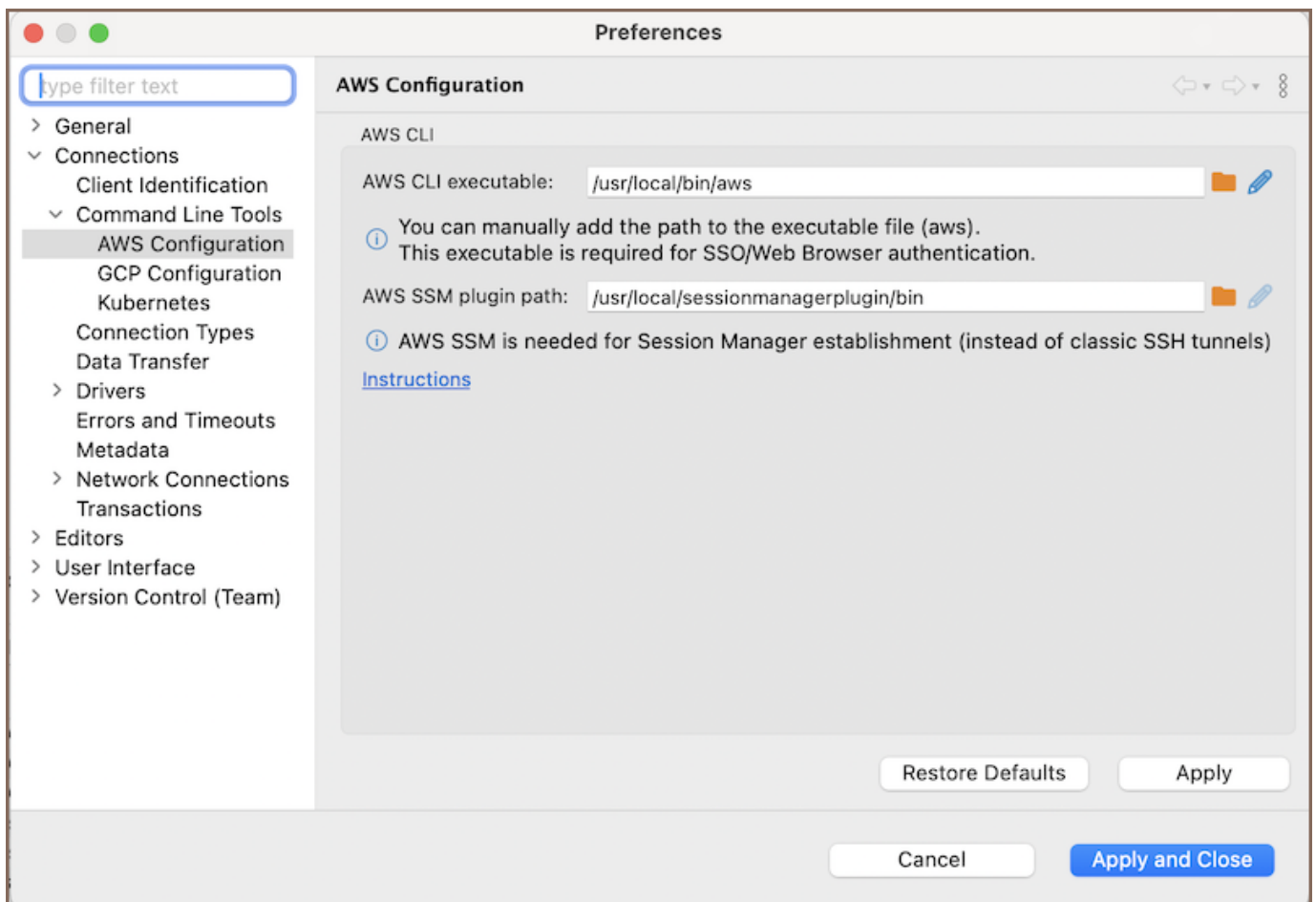
In some scenarios, you may need to interact with AWS resources through Systems Manager (SSM) to manage and automate operations. DBeaver offers a way to configure AWS SSM settings to facilitate this.

Before proceeding, ensure that you have the [AWS CLI](#) and [Session Manager Plugin for the AWS CLI](#) installed on your local machine.

Configuring AWS CLI and Session Manager Plugin paths

Before you can use AWS SSM with DBeaver, you need to specify the paths for the AWS CLI executable and the Session Manager Plugin. Follow these steps to configure the paths:

1. Navigate to **Window -> Preferences -> Connections -> AWS Configuration**.
2. You will see fields for the AWS CLI executable and the AWS SSM plugin path.



3. Enter the appropriate paths based on your operating system.

Path	Windows	MacOS/Unix
AWS CLI executable	<code>C:\Program Files\Amazon\AWSCLI\bin\aws.exe</code>	<code>/usr/local/bin/aws</code>
AWS SSM plugin path	<code>C:\Program Files\Amazon\SessionManagerPlugin\bin\session-manager-plugin.exe</code>	<code>/usr/local/bin/sessionmar</code>

4. After entering the paths, click the **Apply and Close** to apply the changes.

Setting up AWS SSM

To configure AWS SSM, follow these steps:

1) Start setting up a [database connection](#).

2) Add a Network Handler for **AWS SSM**. To do this, click on the **Network configurations...** button and select **AWS SSM** tab.

3) After making your selection, a new tab will appear in the menu at the top, where you can proceed to configure the various settings.

Here you need to fill out the following fields:

Field	Description
Instance ID	Enter the ID of the EC2 instance you want to connect to.
Region	Specify the AWS region where your EC2 instance is located.
Credentials	Select the type of credentials you want to use. You can choose from Default credentials , AWS profile , or Access/secret keys . Read more about AWS Credentials
Role Name (Optional)	Enter the name or ARN of the AWS IAM role if you want to assume a role for this connection.
Enable SSO	Check this box to enable Single Sign-On (SSO) authentication. Learn more about AWS SSO
3rd party account	Check this box if you are using a third-party AWS account for this connection.

Shell commands

Table of contents

[Overview](#)

[How to Work with Shell Commands](#)

[Events](#)

[Command Window](#)

[Settings](#)

[Examples](#)

[Example 1 - Backup PostgreSQL Database Before Disconnect](#)

[Example 2 - Run a Script After Connect](#)

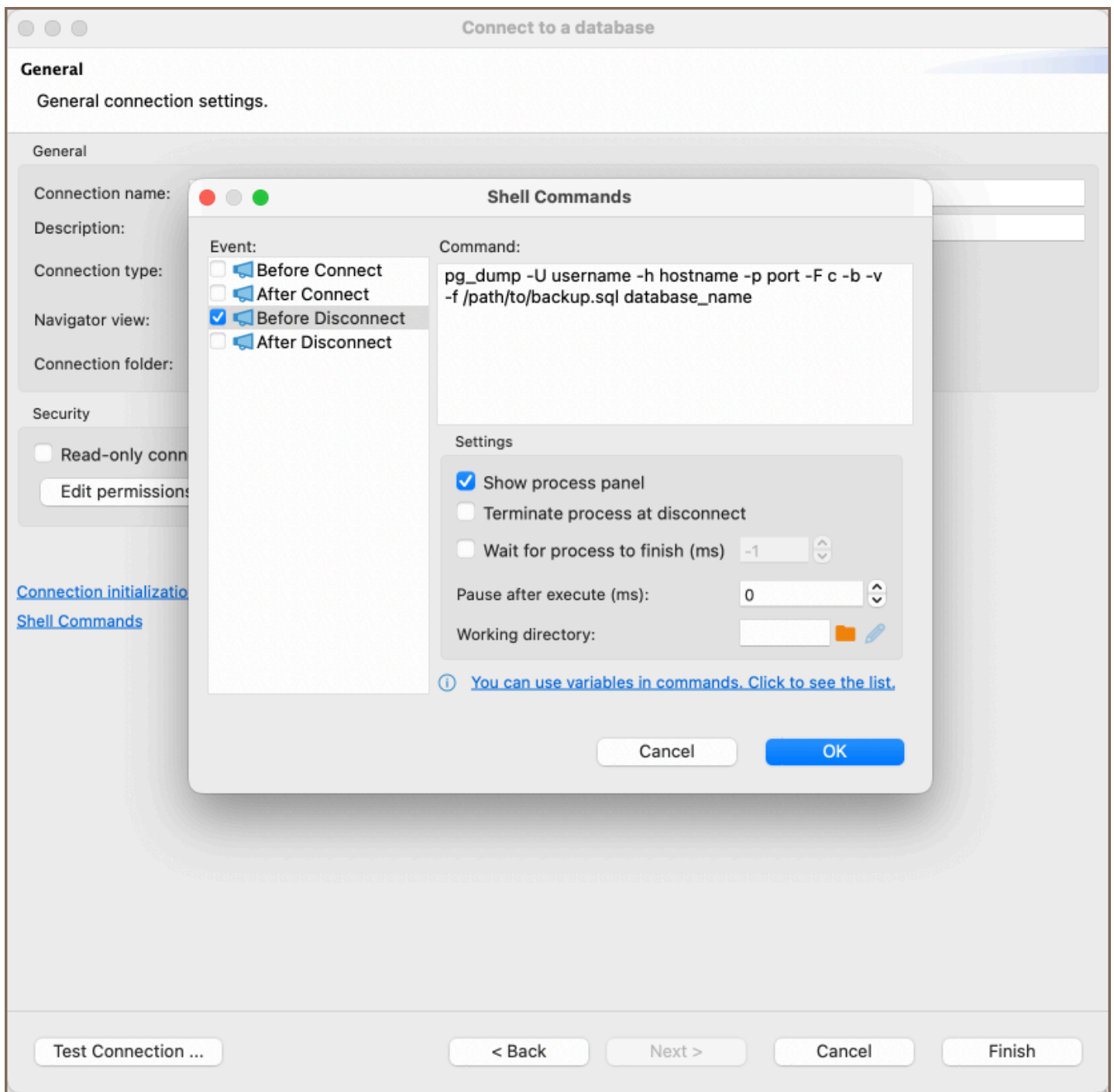
Overview

DBeaver offers the ability to execute shell commands. This feature allows you to automate various database tasks, such as running scripts or [backing up data](#), based on specific events. This article provides an insight into how to work with shell commands in DBeaver, including the events that can trigger these commands, the command window, settings, and examples.

How to Work with Shell Commands

To configure and use shell commands in DBeaver, follow these steps:

- 1) Right-click on the database connection and select **Edit Connection**.
- 2) In the **Connection details** window, navigate to the **Shell commands** section.



From here, you can set up shell commands to be executed at different stages of the connection process, such as before connecting, after connecting, before disconnecting, and after disconnecting. You can also configure various settings to control the behavior of these commands.

Events

Shell commands in DBeaver can be triggered by specific events. These events include:

Event	Description

Before Connect	Executed before a connection is established.
After Connect	Executed after a connection has been made.
Before Disconnect	Executed before disconnecting from the database.
After Disconnect	Executed after the disconnection.

Command Window

The command window is where you can write the shell commands that you want to execute. It provides an interface to input the necessary commands, which will be executed upon the occurrence of one of the specified events.

Settings

The settings for shell commands in DBeaver include several options that allow you to customize the behavior of the commands:

Setting	Description
Show Process Panel	Enables a panel that displays the ongoing process.
Terminate Process at Disconnect	Ends the process when the connection is terminated.
Wait for Process to Finish (ms)	Specifies the time in milliseconds to wait for the process to complete.
Pause After Execute (ms)	Defines a pause time in milliseconds after the execution of a command.
Working Directory	Allows you to set the directory where the commands will be executed.

For more information on the variables that can be used in commands, click on the informational tip within the settings.

Examples

Here are some examples of how you can utilize **Shell commands** in DBeaver:

Example 1 - Backup PostgreSQL Database Before Disconnect

1) **Event:** Select **Before Disconnect**.

2) **Command Window:** Enter the following command:

```
pg_dump -U username -h hostname -p port -F c -b -v -f /path/to/backup.sql database_name
```

Replace `username` , `hostname` , `port` , `database_name` , and `/path/to/backup.sql` with your PostgreSQL credentials and the desired backup file path.

3) **Settings:** You may want to check **Show Process Panel** to see the progress of the backup.

This command will create a custom-format dump of the specified PostgreSQL database. By setting this command to run before disconnecting, you can ensure that your database is backed up automatically each time you disconnect, providing an extra layer of security for your data.

Example 2 - Run a Script After Connect

You can execute a specific script right after connecting to the database to automate routine tasks. Here's an example:

1) **Event:** Select "After Connect."

2) **Command Window:** Enter the following command:

```
sh /path/to/your_script.sh
```

Replace `/path/to/your_script.sh` with the actual path to the script you want to run.

3) **Settings:** You may want to set **Wait for Process to Finish (ms)** to ensure that the script completes before proceeding with other tasks.

This command will execute the specified shell script immediately after connecting to the database. By setting this command to run after connecting, you can automate routine tasks, such as initializing specific settings or running maintenance scripts, each time you establish a connection. This automation enhances efficiency and ensures consistency in your database management process.

Changing current user password

Table of contents

[Changing expired user password](#)

[How to change the user password](#)

DBeaver can change credentials for the current database user.

Databases that support this feature:

- PostgreSQL
- Greenplum
- Cockroach
- Redshift ★
- Netezza ★
- SQL Server
- Oracle
- Exasol
- Snowflake ★
- Vertica

Changing expired user password

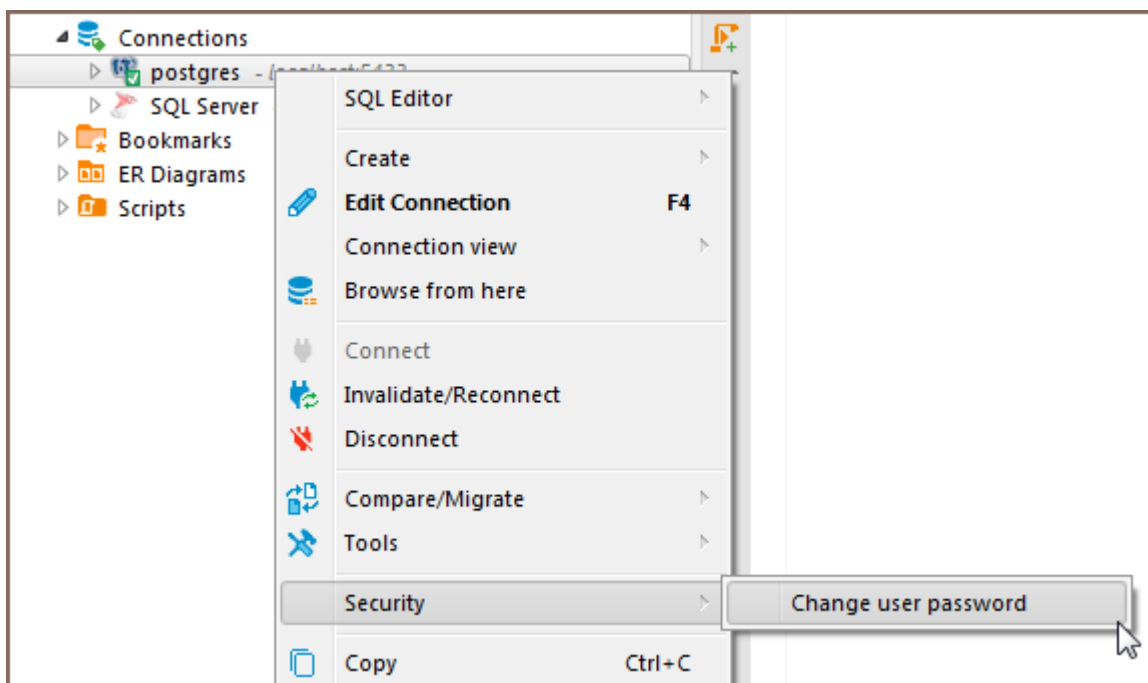
Different databases perform password management for users in different ways. For some of them, you can change the user password after the expiration date. It works for Oracle, PostgreSQL, and Netezza databases.

For the rest, database users must change the password before the expiration. Otherwise, it will be impossible to do it in DBeaver.

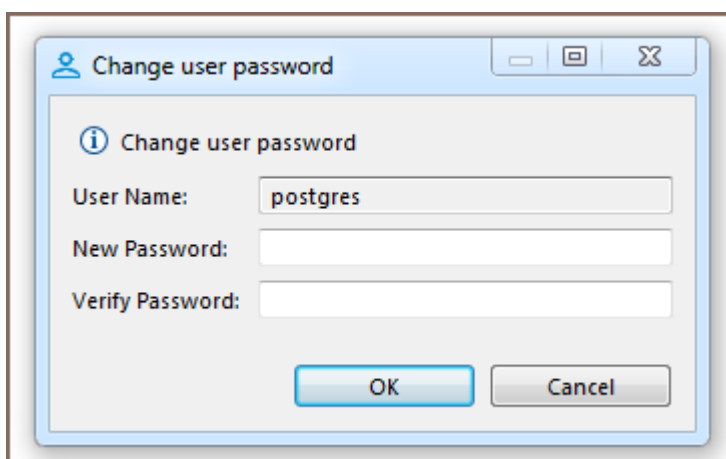
How to change the user password

You can change the current user password in the Navigation Tree according to the instructions below:

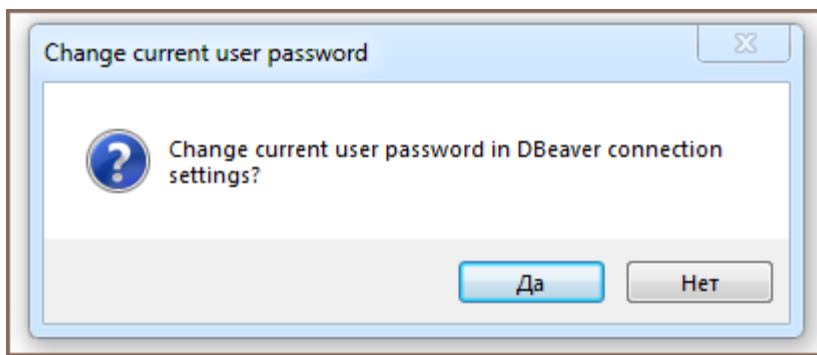
1. Connect to the database.
2. Open the context menu by right-clicking on the connection in the Navigation tree.
3. Select a "Security" point and click on a "Change user password" point in a sub-menu.



4. When a new password input dialog opens, enter and confirm the new password.



5. Confirm the password changes. (This dialog will not appear if the entered password is incorrect).



The password has been changed.

Authentication models overview

Table of contents

[PgPass](#)

[Kerberos](#)

[Oracle Wallet](#)

[Oracle OS auth](#)

[Oracle native database auth](#)

[AWS IAM auth](#)

DBeaver supports many different authentication models for different databases. The default authentication model is **native** - it needs a user name and password which are used to authenticate at the remote database server.

Some database drivers support other database-specific authentications.

PgPass

PostgreSQL specific model. Uses local file PGPASS to obtain user name and/or password.

It does not have any additional parameters.

Parameter	Description
Model ID	postgres_pgpass

Kerberos

Parameter	Description
Model ID	krb5, postgresql_krb5, prestodb_krb5, prestosql_krb5, oracle_krb5
krb5.realm	Kerberos realm/AD domain
krb5.kdc_server	KDC server IP/host name
krb5.user	Kerberos user name

krb5.use_keytab	Force keytab file usage (true/false)
krb5.keytab_path	Keytab file path
krb5.use_kinit	Force kinit utility usage
krb5. kdc_over_tcp	Force TCP/IP protocol usage (UDP is used by default)

Oracle Wallet

Oracle Wallet authentication

Parameter	Description
Model ID	oracle_wallet
oracle.net. wallet_location	Wallet file location

Oracle OS auth

Oracle native OS authentication

Parameter	Description
Model ID	oracle_os

Oracle native database auth

Oracle native database authentication

Parameter	Description
Model ID	oracle_native

AWS IAM auth

AWS IAM authentication

Parameter	Description
Model ID	iam
iam.region	AWS region ID
iam.aws_access_key	AWS Access Key
iam.aws_secret_key	AWS Secret Key
iam. use_default_aws	Load access/secret key from local AWS configuration

Database native

Table of contents

[Overview](#)

[Settings](#)

Overview

DBeaver includes support for Database Native authentication, enabling secure connections to your databases.

Connection "postgres" configuration

PostgreSQL connection settings

Connection settings

- Initialization
- Shell Commands
- Client identification
- Transactions
- General
- Metadata
- Errors and timeouts
- > Data Editor
- > SQL Editor

Main | PostgreSQL | Driver properties | SSH | SSL | + Network configurations...

Server

Connect by: ☒ Host ☐ URL Driver type: JDBC

URL: jdbc:postgresql://localhost:5432/postgres

Host: localhost Port: 5432

Database: postgres

Authentication

Authentication: Database Native

Username: postgres

Password: [masked] ☒ Save password

Advanced

Session role: Local Client: PostgreSQL 14.6 (Homebrew)

[You can use variables in connection parameters.](#)

Driver name: PostgreSQL Driver Settings Driver license

Test Connection ... Cancel OK

Settings

Here are the steps to configure Database Native authentication:

1) From the **Authentication** dropdown menu, select the **Database Native** method. 2) Enter the username and password that are associated with your database in the **Username** and **Password** fields respectively.

Tip: If you check the **Save credentials locally** option, you will not be asked to input your credentials every time you connect. 3) Once you have input your credentials, click the **Test Connection** button to verify your settings. If

everything is configured correctly, DBeaver should now be able to connect to your database using the Database Native authentication method.

DBeaver profile

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

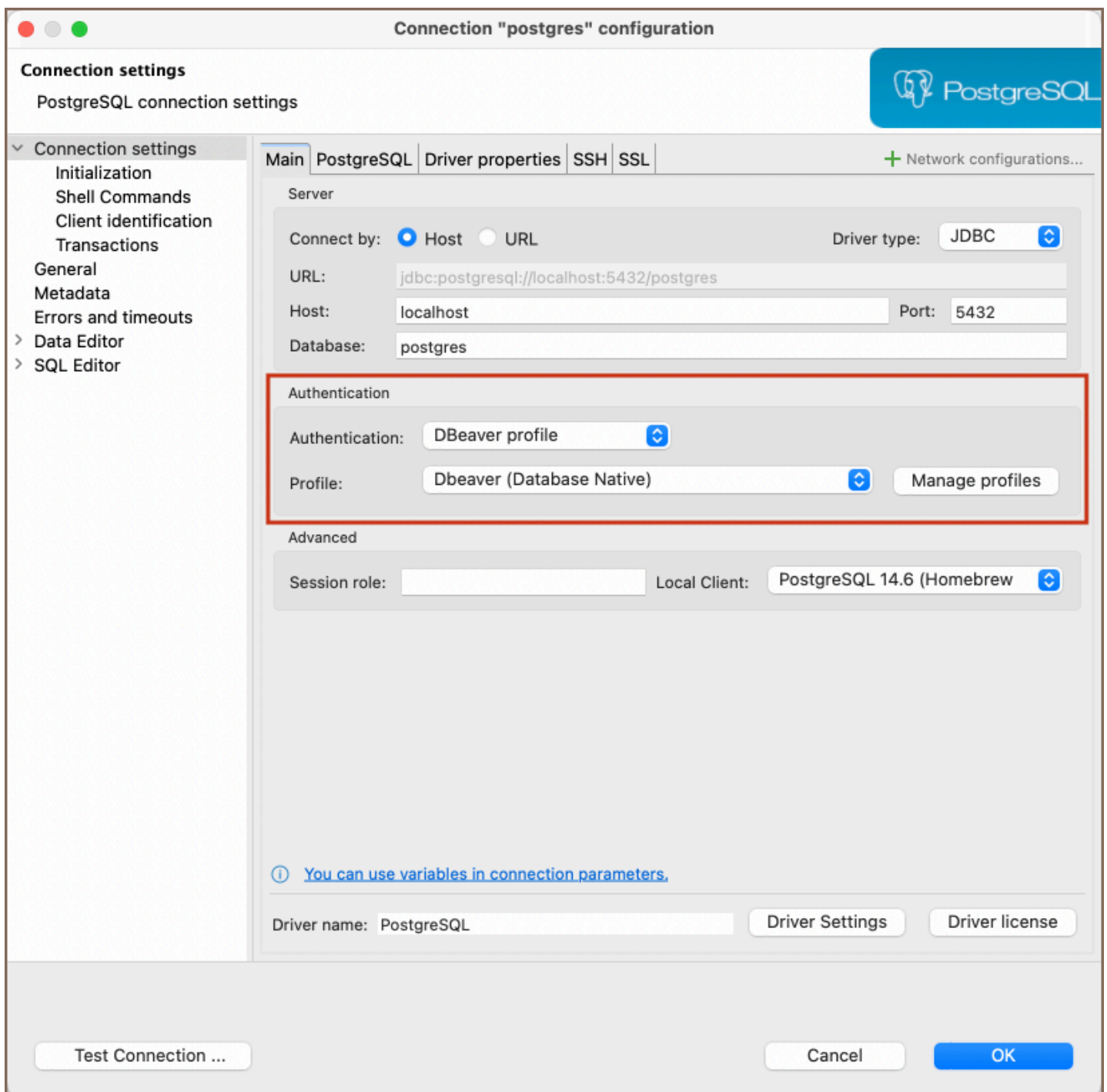
Table of contents

[Overview](#)

[Settings](#)

Overview

DBeaver offers a feature for setting up authentication profiles. This DBeaver Profile authentication type streamlines database management by allowing the creation of unique profiles for each authentication method. It effectively reduces the complexity and increases the efficiency of managing database connections.



Settings

1) From the **Authentication** dropdown menu, select the **DBeaver profile** method. 2) If you already have a profile set up, select it from the **Profile** dropdown. 3) If you don't have a profile yet, click the **Manage profiles** button to start creating a new one. 4) In the window that opens, click the **Create button** and name your profile as per your preference.

5) Select the type of authentication from the available options and fill out the required fields.

Tip: If you check the **Save credentials locally** option, you will not be asked to input your credentials every time you connect. 6) After filling in all the necessary fields, click **OK** to save your profile settings. 7) Back in the main connection settings window, click on the **Test Connection** button to verify if your settings are correct. 8) If everything is configured correctly, DBeaver should now be able to connect to your database using the **DBeaver profile** authentication method.

Kerberos authentication

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Supported databases](#)

[Settings](#)

[Extra configuration](#)

[Using the kinit](#)

[Using the keytab file](#)

[Windows specifics](#)

[Local Kerberos configuration file](#)

[Additional settings](#)

[Troubleshooting](#)

[Oracle JDBC driver and Kerberos authentication](#)

Overview

DBeaver includes support for Kerberos authentication, enabling secure connections to your databases.

Connect to a database

Connection Settings

PostgreSQL connection settings

PostgreSQL

Driver properties

SSH

SSL

+ Network configurations...

Server

Connect by: ☒ Host ☐ URL

Driver type: JDBC

URL: jdbc:postgresql://kerb.example.com:5432/postgres

Host: kerb.example.com Port: 5432

Database: postgres ☐ Show all databases

Authentication

Authentication: PostgreSQL Kerberos

Username: postgres [Instructions](#)

Kerberos user: admin

Realm: EXAMPLE.COM

KDC server: kdc.example.com

Password:

••••••••••

☒ Save password

Extra configuration

ⓘ Kerberos settings might be cached between sessions. If you change them and cannot connect, please try to restart the application.

Advanced

Session role:

Local Client: /Library/PostgreSQL/16

ⓘ You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: PostgreSQL

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Kerberos authentication is a secure method for verifying user identities over non-secure networks. It is widely used in various environments, especially in database management systems, to ensure that communication between the client and server remains encrypted and authenticated.

This guide is designed to help you configure Kerberos authentication on the client machine, where DBeaver is installed. It assumes that the Kerberos Key Distribution Center (KDC) and the necessary server configurations have already been completed.

Supported databases

- [AlloyDB](#)
- Cockroach
- EnterpriseDB (EDB)
- Fujitsu
- [Greenplum](#)
- HANA
- [MariaDB](#)
- [MySQL](#)
- [Oracle](#)
- [PostgreSQL](#)
- PrestoDB
- PrestoSQL
- [Redshift](#)
- [Teradata](#)
- TimescaleDB
- Trino
- Yellowbrick
- YugabyteDB

Settings

The table below lists the basic settings required for Kerberos authentication in DBeaver. These settings are essential for establishing a secure connection to your database using Kerberos, providing user identification, authentication, and access control.

Setting	Description
Username	Specifies the name of the user or role within the database. This is the identity under which you will connect to your database.
Kerberos user	A unique identity in the Kerberos system to which Kerberos can assign tickets, enabling access to services that are Kerberos-aware.
Realm	The domain over which a Kerberos authentication server can authenticate users, hosts, or services. It's often the uppercase version of the DNS domain name it oversees.
KDC Server	The hostname of your Kerberos Key Distribution Center (KDC), which is a service that provides session tickets and keys within an Active Directory domain.
Password	The password associated with your Kerberos user. This method involves directly entering your password. It's a simple, manual process suitable for environments where automated authentication methods are not preferred or available.

Below are additional Kerberos authentication settings for advanced configuration:

Extra configuration	Description
Use kinit	When selected, it indicates the use of the <code>kinit</code> tool on your machine, which obtains and caches an initial ticket-granting ticket. Generally, you only need to provide your Kerberos username when this option is selected. Selecting this option makes the Password field inactive because authentication is managed through the ticket obtained by <code>kinit</code> .
Use keytab	A checkbox option for those who prefer to use a keytab file instead of manually entering a password. Selecting this option also makes the Password field inactive because the keytab file contains encrypted keys that the Kerberos system uses for authentication.
Custom krb5.conf	Allows you to specify the path to your local Kerberos configuration file. This file contains settings that define the Kerberos realms, KDCs, and other parameters for your environment.
Debug Kerberos	A checkbox that, when selected, enables the logging of detailed Kerberos connection information in your log .

Note: Some settings may be stored in a session cache. If you have updated the settings and still encounter connection issues, restarting DBeaver might be necessary to apply the changes effectively.

Extra configuration

The **Extra configuration** section within DBeaver's connection settings offers advanced options for customizing the Kerberos. This section allows users to specify detailed settings, such as the use of a keytab file, `kinit` commands, or a custom `krb5.conf` / `krb5.ini` file.

Using the kinit

The `kinit` command is a utility Kerberos provides to obtain and cache an initial ticket-granting ticket (TGT).

In the context of using DBeaver, the `kinit` command streamlines the authentication process by obtaining and caching an initial ticket-granting ticket (TGT). This mechanism enables users to authenticate against Kerberos-enabled services, reducing the need for continuous password inputs during the session.

To configure Kerberos authentication using `kinit` command, follow these steps:

1. Make sure you have a valid Kerberos ticket-granting ticket (TGT), which can be obtained using the `kinit` command. For a detailed guide, refer to the [official documentation](#) on `kinit` command.
2. Specify the path to your `kinit` tool in the **Extra Configuration** section in the **Use kinit** field.
3. Try connecting to your database. With these settings, DBeaver will use the `kinit` command for authentication.

The screenshot shows the 'Authentication' tab in the DBeaver configuration window. The 'Authentication' dropdown is set to 'PostgreSQL Kerberos'. The 'Username' field contains 'postgres' with an 'Instructions' link. The 'Kerberos user' field contains 'postgres/admin@example.com'. The 'Realm' field contains 'EXAMPLE.COM'. The 'KDC server' field contains 'kdc.example.com'. The 'Password' field is masked with dots, and there are 'Save password' and 'toggle visibility' options. Under the 'Extra configuration' section, 'Use kinit' is checked with the path '/usr/bin/kinit'. 'Use keytab', 'Custom krb5.conf', and 'Debug Kerberos connection' are unchecked. A note at the bottom states: 'Kerberos settings might be cached between sessions. If you change them and cannot connect, please try to restart the application.'

Using the keytab file

A keytab is a file containing pairs of Kerberos principals and encrypted keys derived from the Kerberos password. It enables authentication to various remote systems using Kerberos without entering a password.

In the context of using DBeaver, the keytab file simplifies the authentication process by requiring the user to provide all necessary credentials except the password.

To configure Kerberos authentication using a keytab file, follow these steps:

1. Utilize the `ktutil` command-line utility to generate a keytab file. This involves adding entries for your principals along with their encrypted keys. For comprehensive instructions on creating a keytab file, consult the [official documentation](#) on creating a Kerberos keytab file.
2. Specify the path to your keytab file in the **Extra Configuration** section in the **Use keytab** field.
3. Attempt to connect to your database. DBeaver will utilize the specified keytab file for authentication.

Authentication

Authentication:

PostgreSQL Kerberos

Username:

postgres

[Instructions](#)

Kerberos user:

postgres/admin@example.com

Realm:

EXAMPLE.COM

KDC server:

kdc.example.com

Password:

☒ Save password

▼ Extra configuration

☐ Use kinit

☒ Use keytab

/home/username/krb5.keytab

☐ Custom krb5.conf

☐ Debug Kerberos connection

Kerberos settings might be cached between sessions. If you change them and cannot connect, please try to restart the application.

Windows specifics

On Linux and macOS, `kinit` and keytab files are supported natively. However, Windows does not support these functionalities natively. To enable these features, Windows users need to install additional software.

A key tool for Windows users is [MIT Kerberos](#) for Windows. It allows the use of `kinit` and provides utilities for managing keytab files, bridging the gap between Windows and Kerberos authentication standards.

Local Kerberos configuration file

If you have a custom Kerberos configuration file (`krb5.conf` / `krb5.ini`), input the complete file path in the **Custom krb5.conf** field.

The configuration file includes necessary configuration details such as KDCs and admin server locations for Kerberos realms, default values for the current Realm, and hostname to Kerberos realm mappings. If a custom configuration file path is provided, it is not necessary to fill in the standard authentication fields, as the configuration file contains all the required information.

Additional settings

When configuring PrestoSQL and Trino connections, especially in environments that utilize Kerberos authentication and SSL, certain additional settings might be necessary for connection:

Setting	Description
---------	-------------

Service name	Add the Kerberos service name of the remote coordinator.
Use SSL from JKS	Check this setting and manually add the file path to your <code>.jks</code> file if you need an SSL certificate from a JKS file. This is useful when the server's SSL certificate is not automatically trusted. If your JKS file is password-protected, add the password to the SSL JKS Password field to allow database to access the keystone.
SSL JKS Password	For additional security, the JKS file is password-protected. Add the password to the "SSL JKS" field to allow database to access the keystone.

Troubleshooting

Oracle JDBC driver and Kerberos authentication

When configuring Kerberos authentication for Oracle databases, it is important to be aware of compatibility issues with certain JDBC driver versions. Specifically, Oracle JDBC driver version 21 has been known to cause issues with Kerberos authentication, often not working with older configurations.

To ensure Kerberos authentication functions properly with Oracle databases, it is recommended to use an older version of the Oracle JDBC driver. Versions 12.x or 19.x are known to be compatible and should be used for Kerberos authentication. By selecting one of these older driver versions, you can avoid the authentication problems introduced in version 21.

Microsoft Entra ID

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Settings](#)

Overview

DBeaver comes with Microsoft Entra ID (formerly Azure AD) authentication support, allowing secure access to your databases. This article assumes that you are already registered with Microsoft Azure.

You can find detailed information about Entra's features and capabilities in the official [Entra documentation](#).

Connect to a database

Connection Settings
PostgreSQL connection settings

PostgreSQL

Main PostgreSQL Driver properties SSH SSL + Network configurations...

Server

Connect by: ☒ Host ☐ URL Driver type: JDBC

URL: jdbc:postgresql://localhost:5432/postgres

Host: localhost Port: 5432

Database: postgres ☐ Show all databases

Authentication

Authentication: Azure AD for PostgreSQL

Credentials: Enterprise application [Details](#)

Client ID: Identifier of the Azure AD Application used for authentication

Tenant ID: Tenant ID, required in single-tenant applications

AD Group name: Group is required if access was granted on a group level

Advanced

Session role: Local Client: /Library/PostgreSQL/16

[You can use variables in connection parameters.](#) Connection details (name, type, ...)

Driver name: PostgreSQL Driver Settings Driver license

Test Connection ... < Back Next > Cancel Finish

Settings

To properly configure Azure Entra ID authentication in DBeaver, fill out the credential fields detailed in [our guide](#).

Note: If you belong to an Azure Entra ID Group, you must add the **AD Group** name to your Connection configuration. This step, possibly requiring admin assistance, involves right-clicking on your database in the **Database Navigator** and selecting **Edit Connection**. In the ensuing dialog, navigate to the **Connection Settings** tab and input your **AD Group** name in the designated field.

Authentication MongoDB

Table of contents

[Overview](#)

[Settings](#)

Overview

DBeaver is fully compatible with the MongoDB authentication method, offering you a secure way to access your [MongoDB](#) databases.

Connection "test" configuration

Connection settings
MongoDB connection settings

Connection settings
Initialization
Shell Commands
Client identification
Transactions
General
Metadata
Errors and timeouts
Tableau
> Data Editor
> SQL Editor

Main Driver properties SSH SSL + Network configurations...

Address

Type: ☒ Manual ☐ URL

Host: localhost Port: 27017

Database: test

Replica Set:

Authentication

Authentication: MongoDB authentication

Mechanism: SCRAM-SHA-1

Source: admin

Username: dbeaver

Password: ☒ Save password

[You can use variables in connection parameters.](#)

Driver name: MongoDB Driver Settings Driver license

Test Connection ... Cancel OK

Settings

Here are the steps to configure MongoDB authentication:

1. From the **Authentication** dropdown menu, select the **MongoDB authentication** method.
2. From the **Mechanism** dropdown menu, select an option. The following table outlines the available authentication mechanisms:

--	--

Mechanism	Description
SCRAM-SHA-1	A password-based authentication mechanism using SHA-1 hashing algorithm. Suitable for securing data transmissions.
SCRAM-SHA-256	An upgraded version of <code>SCRAM-SHA-1</code> , using the stronger <code>SHA-256</code> hashing algorithm. Offers enhanced security.
MONGODB-CR	A protocol specific to MongoDB, utilizing a challenge-response mechanism for authentication.
PLAIN	A simple, unencrypted password-based authentication mechanism. Not recommended for production environments due to security vulnerabilities.
GSSAPI	Based on the Kerberos protocol, this mechanism offers secure password authentication.
MONGODB-X509	A certificate-based authentication mechanism that validates the client's <code>X.509</code> certificates.
NONE	No authentication is required. This method should only be used in secure, isolated environments.

You can read more about these mechanisms on the [MongoDB documentation](#) page.

3. Fill in the **Source** field if necessary.

4. Enter the username and password in the **Username** and **Password** fields respectively.

Tip: If you check the **Save credentials locally** option, you will not be asked to input your credentials every time you connect.

5. Once you have input your credentials, click the **Test Connection** button to verify your settings. If everything is configured correctly, DBeaver should now be able to connect to your database using the MongoDB authentication method.

PostgreSQL Pgpas

Table of contents

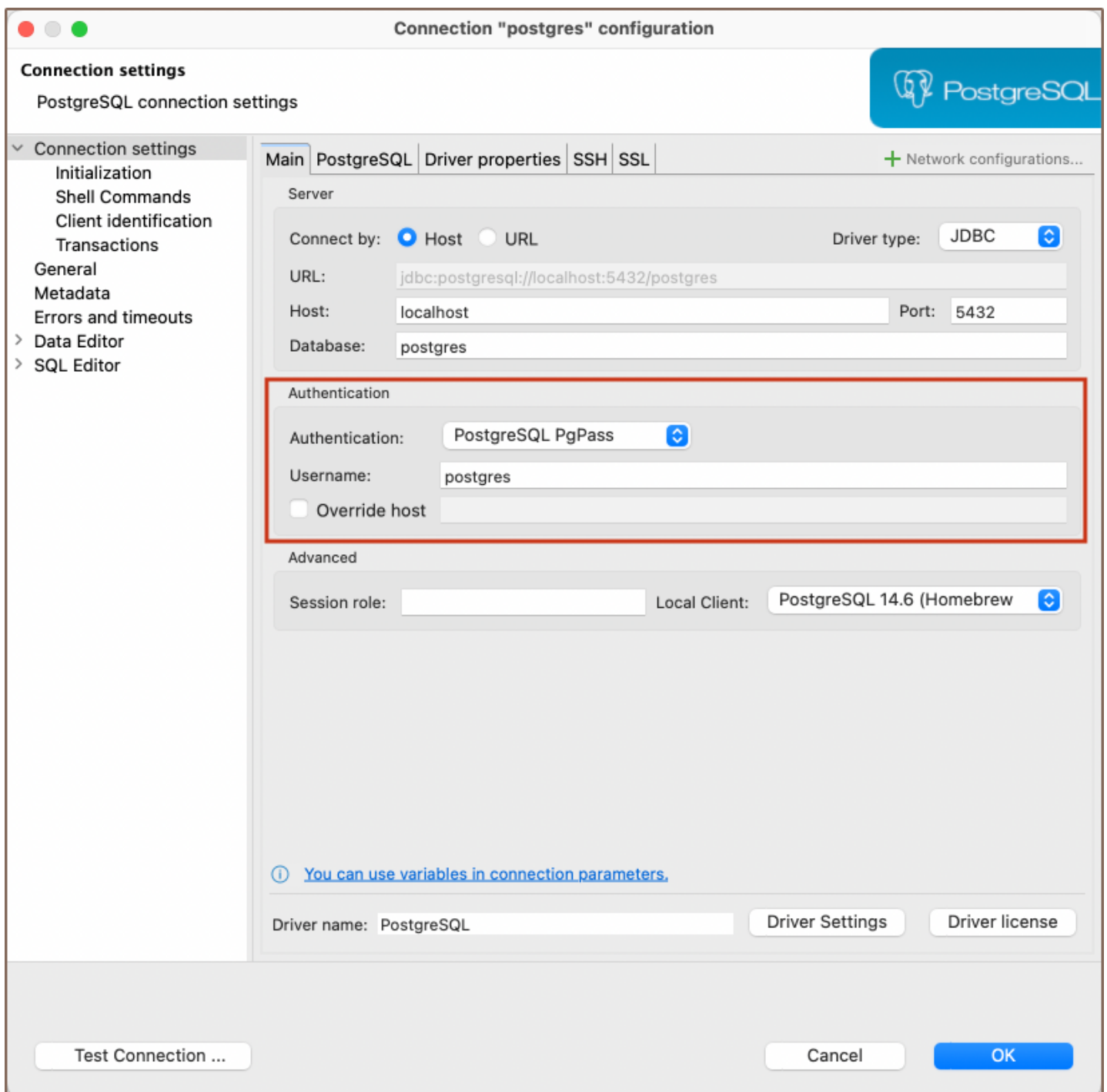
[Overview](#)

[Settings](#)

[Override host](#)

Overview

DBeaver is fully compatible with the PgPass authentication method, offering you a secure way to access your [PostgreSQL](#) databases.



Settings

PostgreSQL PgPass method allows you to authenticate using credentials stored in a `.pgpass` file. Here are the steps to configure this:

- 1) From the **Authentication** dropdown menu, select the **PostgreSQL Pgpass** method.
- 2) Enter the username that is associated with the database in the **Username** field.

3) DBeaver will use the `.pgpass` file for authentication. If you haven't created this file yet, follow the official [PostgreSQL documentation](#) to create it, format it correctly, and understand where it should be located on your system.

Note: Ensure that the permissions for your `.pgpass` file are set correctly. The file should only be readable and writable by the file owner. Improper permissions can lead to security issues.

4) Once your `.pgpass` file is set up as per the PostgreSQL documentation, you can test the connection from within the **Test Connection** button. If everything is configured correctly, DBeaver should now be able to connect to your PostgreSQL database using the PgPass authentication method.

Override host

DBeaver gives you the option to override the default host. If you wish to do this, select the checkbox next to **Override host** and then input the new host information in the field that appears.

Tip: This is useful in situations where you need to connect to a different server without changing the `.pgpass` file, or if the `.pgpass` file is shared among several users or applications with different host requirements.

Authentication PostgreSQL SSPI

Table of contents

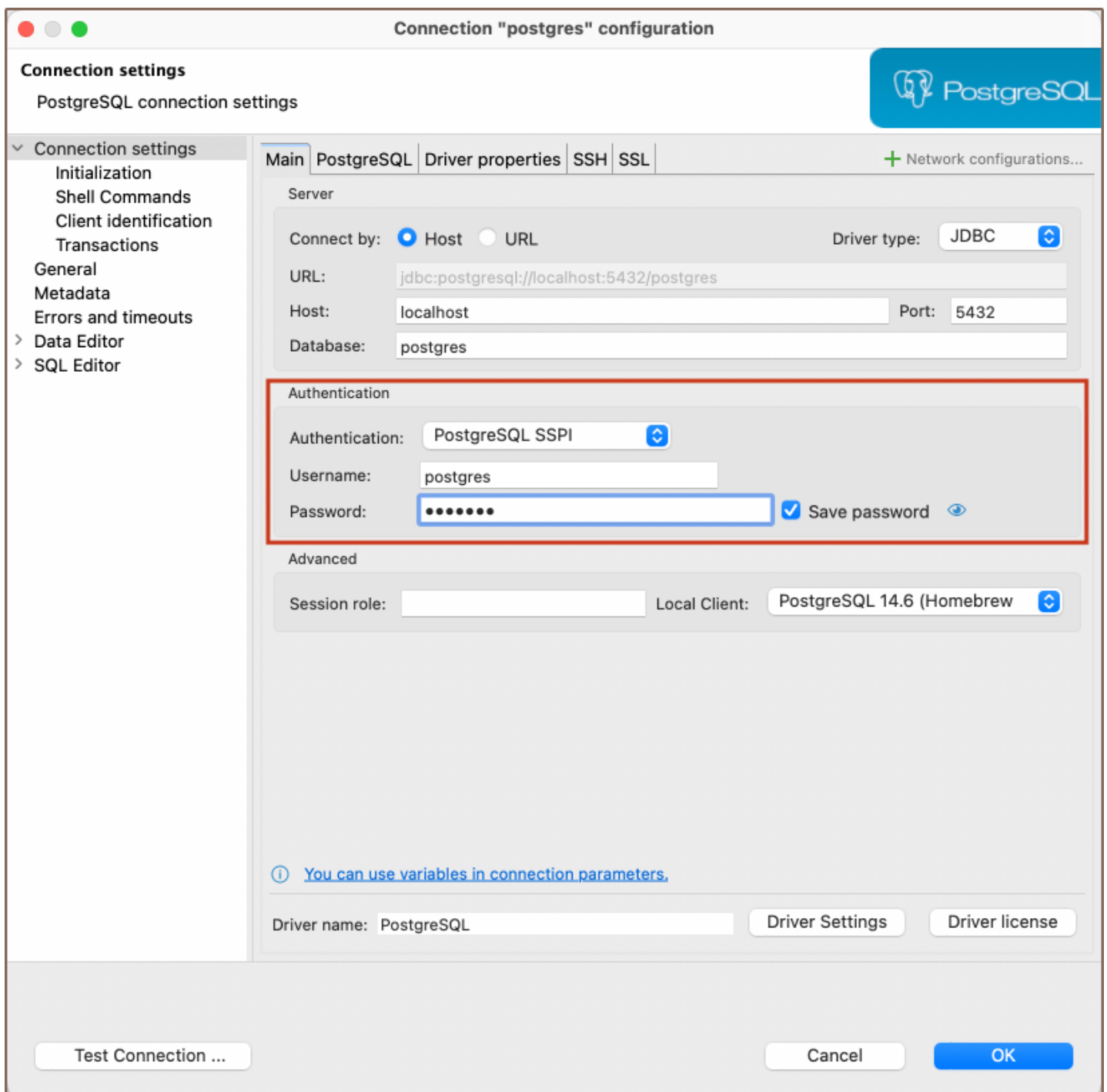
[Overview](#)

[Settings](#)

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Overview

DBeaver comes with SSPI (Security Support Provider Interface) authentication support, allowing secure access to [PostgreSQL](#) databases. This authentication type enables Windows Single Sign-On (SSO), allowing users to authenticate using their existing Windows account credentials.



Settings

To configure SSPI authentication in DBeaver, follow these steps:

1) From the **Authentication** dropdown menu, select the **PostgreSQL SSPI** method. 2) For the authentication process, your system will utilize the relevant credentials automatically. On Windows systems, your current login credentials are used, and on non-Windows platforms that support GSSAPI, the integrated authentication system will apply.

Note: Your system (whether it's Windows or a non-Windows platform with GSSAPI support), and PostgreSQL server must be correctly configured for SSPI authentication. If you encounter any issues during this process, please follow the official [PostgreSQL documentation](#). 3) Click **Test Connection** to confirm your settings. If the test is successful, your setup is complete.

Oracle authentication models

Table of contents

[Configuration types](#)

[Basic connections](#)

[TNS](#)

[Custom URL](#)

[Authentication](#)

[Database](#)

[OS authentication](#)

[Oracle Wallet](#)

[Kerberos](#)

[Oracle Cloud connections](#)

[Plain URL connection](#)

[Oracle Wallet connection](#)

There are several ways to configure a database connection and several ways to perform an authentication.

Connect to a database

Oracle Connection Settings
Oracle connection settings

Main | Oracle properties | Driver properties | SSH | [+ Network configurations...](#)

Connection Type: Basic | TNS | Custom

Host: localhost Port: 1521
Database: ORCL Service Name

Authentication

Authentication: Oracle Database Native
Username: system Role: Normal
Password: Save password

Client: <not present>

[You can use variables in connection parameters.](#) Connection details (name, type, ...)

Driver name: Oracle Driver Settings Driver license

Test Connection ... < Back Next > Cancel Finish

Configuration types

Basic connections

Host/port based configuration

Parameter	Description	Example
Host	Server host name	192.168.1.25
Post number	Server listener port	1521 (default)
Database	Service or SID name	ORCL
Service/SID	It depends on the server configuration. SID must be selected for some servers and the Service Name for	SID

	others	
--	--------	--

TNS

TNS configuration is the simplest, but it requires you to have the `tnsnames.ora` file somewhere on your disk. `tnsnames.ora` contains information about all accessible Oracle server connections. DBeaver can determine the default location of this file, but sometimes you will need to manually specify it.

Parameter	Description	Example
Network Alias	Name of configuration from <code>tnsnames.ora</code>	ORCL1
TNS names path	Path to <code>tnsnames.ora</code> file. By default, it is got from the <code>TNS_ADMIN</code> environment variable or from Windows' registry	C: \oracle\network\admin

Custom URL

For a more sophisticated configuration, you can specify the full JDBC URL manually (see [Data Sources and URLs](#)).

Sample URL (Oracle Cloud):

```
jdbc:oracle:thin:@(description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1522)(host=ac
```

Authentication

Database

Parameter	Description	Example
User name	Database user name	SYS
Password	Database user password	
Role	Role for connection. Roles SYSDBA and SYSOPER are needed for some administrative	Normal

	operations	
Save password	Saves the user/password information in the local DBeaver configuration	SID

OS authentication

The Oracle driver gets user information from the current OS user.

You do not need to explicitly specify any credentials.

Oracle Wallet

A more secure way to connect is to use the Oracle Wallet. Wallet is a directory with security keys and some other optional connection information. Wallets are usually distributed as ZIP archives. You need to extract the ZIP archive to a folder on a disk and specify this folder in the `Wallet location` field.

Wallet may contain information about a database user. This, however, is optional. You will sometimes need to specify the user too.

Wallet may also contain a TNS configuration. If it does, you can use the TNS connection configuration easily by setting the `TNS path` to the same value as the `Wallet location`.

Parameter	Description	Example
User name, Password, Role	See Database authentication	
Wallet location	Oracle wallet directory	C:\oracle\network\wallet\example
Wallet password	Optional. Some wallets are password-protected	

Kerberos

Kerberos is the most complicated authentication in Oracle.

Parameter	Description	Example
-----------	-------------	---------

Username	Database user name	c##testuser
Kerberos user	Kerberos / Active directory user name	testuser@THE-REALM
Realm	Kerberos realm	THE-REALM
KDC server	KDC server address	krb5.your-domain.com
Password	Kerberos user password	

Oracle Cloud connections

DBeaver supports Oracle Cloud Autonomous databases connectivity.

There are two ways to authenticate:

Plain URL connection

- To use a plain URL connection you must enable the `Access control list` for the Oracle autonomous database.
- Then add your IP address to the IP list.
- Use the Custom connection configuration ([URL](#)). You can copy the URL from the Oracle Cloud database page (link "DB Connection").

Oracle Wallet connection

It is the default authentication type for the Oracle Cloud.

- Download Wallet from the Oracle Cloud website
- Expand the wallet archive to a folder
- Set [TNS](#) configuration type
- Set the `TNS path` to the wallet location directory

- Choose the proper `Network Alias` from the drop-down menu
- Set `Authentication` to the Oracle Wallet
- Set the database username and password (you can get them from the Oracle Cloud database information page)
- Set the `Wallet location` to the wallet location directory

Authentication Salesforce

Table of contents

[Overview](#)

[Settings](#)

Overview

DBeaver is fully compatible with the Salesforce authentication method, offering you a secure way to access your [Salesforce](#) databases.

Connection "login.salesforce.com" configuration

Connection settings

Salesforce connection settings

Connection settings

- Initialization
- Shell Commands
- Client identification
- Transactions
- General
- Metadata
- Errors and timeouts
- Tableau
- > Data Editor
- > SQL Editor

Main Driver properties SSH

+ Network configurations...

General

Connect by: ☒ Host ☐ URL Driver type: JDBC

JDBC URL: jdbc:dbeaver:salesforce:login.salesforce.com

Host: login.salesforce.com Port:

Authentication

Authentication: Salesforce authentication

Username:

Password: ☒ Save password

Security token:

[You can use variables in connection parameters.](#)

Driver name: Salesforce Driver Settings

Test Connection ... Cancel OK

Settings

Note: Having a security token is necessary for securely connecting to Salesforce from DBeaver. This token can be generated in your Salesforce account settings.

Here are the steps to configure Salesforce authentication:

- 1) From the **Authentication** dropdown menu, select the **Salesforce authentication** method.

2) Enter the username, password and security token that are associated with your Salesforce database in the **Username**, **Password** and **Security token** fields respectively.

Tip: If you check the **Save credentials locally** option, you will not be asked to input your credentials every time you connect.

3) Once you have input your credentials, click the **Test Connection** button to verify your settings. If everything is configured correctly, DBeaver should now be able to connect to your database using the Salesforce authentication method.

Mysql two-factor authentication

Table of contents

[Overview](#)

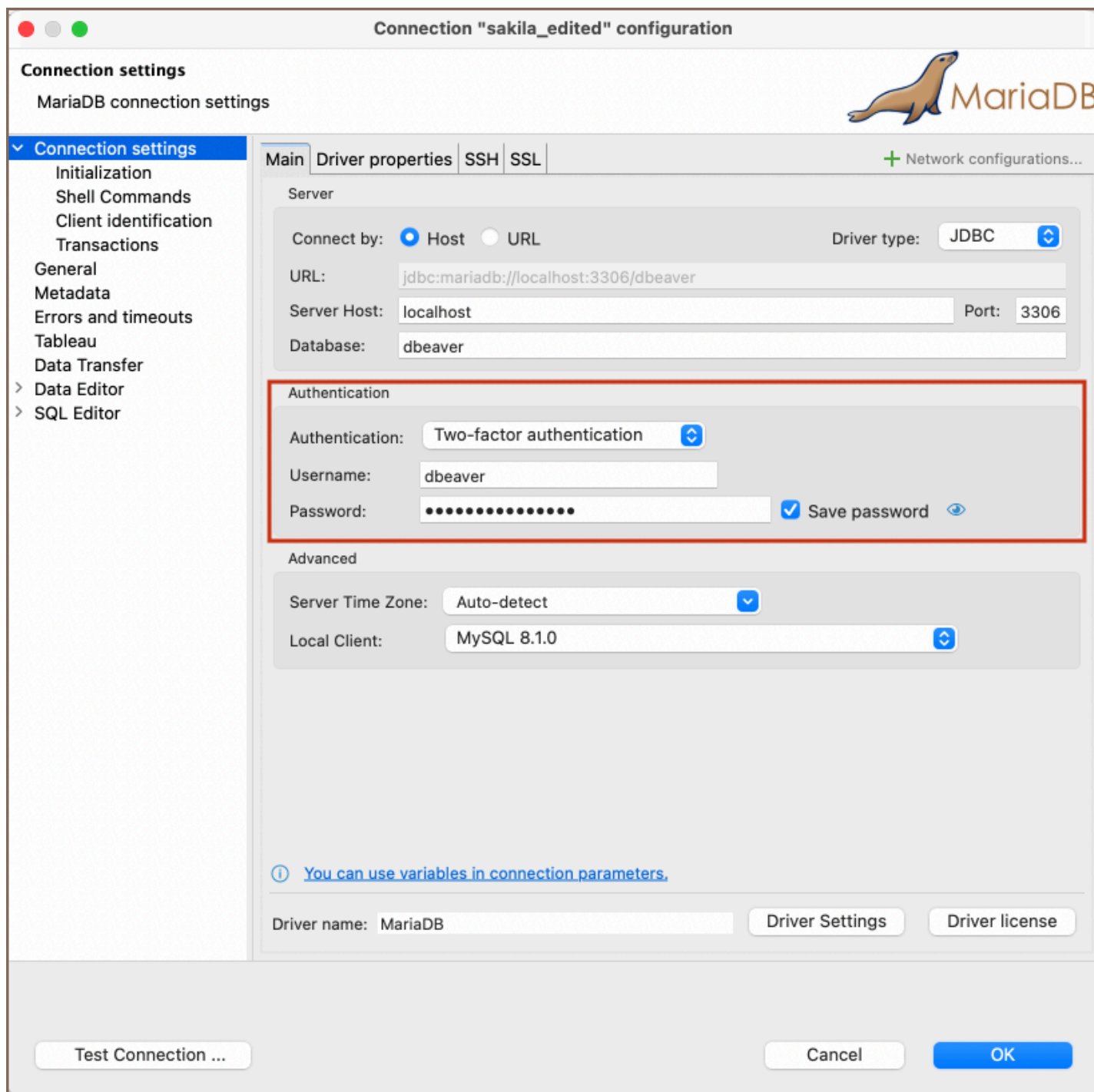
[Settings](#)

[How to Use](#)

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Overview

In addition to Database Native authentication, DBeaver also provides an option for Two-Factor Authentication (2FA). This feature requires a secondary password, offering enhanced security for database connections.



Settings

To set up Two-Factor Authentication with a secondary password, please refer to the official MySQL documentation: [Multi-Factor Authentication](#).

How to Use

When Two-Factor Authentication is enabled, you need to enter the secondary password along with your primary credentials:

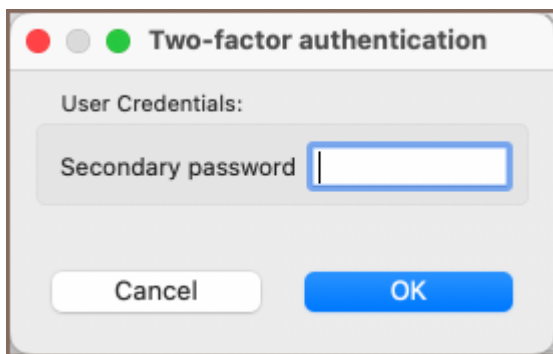
1) In the **Authentication** dropdown menu, choose the **Two-factor authentication** method.

Tip: If you check the **Save credentials locally** option, you will not be asked to input your credentials every time you connect.

2) Input your **Username** and **Password** associated with your database.

3) Click the **Test Connection** button or attempt to connect to the database.

4) A window for the **Secondary Password** will appear. Enter your secondary password.



5) If all details are correct, DBeaver will establish a connection using both your primary and secondary passwords.

Cloud Explorer overview

Note: This feature is available in [Ultimate](#) and [Team](#) editions, as well as in [CloudBeaver Enterprise](#).

Table of contents

[Overview](#)

[Supported Databases](#)

[Cloud configuration](#)

[Amazon Web Services](#)

[Google Cloud Platform](#)

[Microsoft Azure](#)

[Configuring SSH Tunnels](#)

[Explorer](#)

[Database cloud information](#)

Overview

Cloud Explorer offers deep integration with popular cloud service providers. As of the latest version, it supports Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure.

With **Cloud Explorer**, you can set up your cloud access once and then easily browse, connect, and manage your cloud databases. It saves you from manually configuring each database connection, as it reads all database endpoint information directly from the cloud provider.

Authentication is centralized. You use your cloud account to access your cloud databases. Before you start using **Cloud Explorer**, you need to set up your cloud provider access. This setup includes access credentials, availability zones for database search, and other cloud-specific settings.


Supported Databases

The **Cloud Explorer** allows you to easily connect with and manage various databases. Here is the list of databases it supports:


Providers	Databases
Amazon Web Services	PostgreSQL
	MySQL
	Oracle
	Amazon Redshift
	Amazon Athena
	Amazon DocumentDB
	Amazon DynamoDB
	Amazon Keyspaces
	Amazon ElastiCache (Redis)
	Amazon Timestream
	Amazon Neptune
Google Cloud Platform	AlloyDB for PostgreSQL
	Microsoft SQL Server
	PostgreSQL
	MySQL
	Spanner
	Firestore
	BigQuery
Microsoft Azure	Bigtable
	Microsoft SQL Server
	PostgreSQL
	MySQL

	CosmosDB (MongoDB)
	CosmosDB (Cassandra)

Cloud configuration

To begin the configuration process, click the **Cloud Explorer** button  located in the [Toolbar](#). From there, you can start setting up your cloud provider access according to your needs. This involves configuring your access credentials, specifying the availability zones for database search, and adjusting other cloud-specific settings. The setup process varies depending on the cloud service provider you're using.

If this is your first time setting up, the **Create Cloud Configuration** dialog will open, guiding you through the setup process. Once you've set up your first cloud configuration, clicking the **Cloud Explorer** button will open a dialog displaying your cloud databases.

If you need to set up a new cloud configuration, you can do so by clicking the **New** button  in the **Cloud Explorer** dialog. You can start setting up your new cloud provider access according to your needs.

Cloud configuration is different for each cloud service provider.

Amazon Web Services

Name	Description	Additional article
Default Credentials	Allows AWS to determine credentials using the standard credential providers.	AWS Default credentials
AWS Profile	Allows you to choose which credentials profile you want to use.	AWS Profile
Access /Secret Key	Allows you to choose which credentials profile you want to use. If you have an AWS Secret set up, enter it in the Connection settings .	AWS Access /Secret Key
Single Sign-On (SSO)	Allows access to AWS resources without explicitly specifying user credentials in DBeaver connections configuration.	AWS SSO guide

Tip: About permissions, read [AWS managed policies](#) article.

Google Cloud Platform

Name	Description	Additional article
Default Credentials	Allows GCP to determine credentials using the standard credential providers.	GCP Default credentials
Web browser or SSO	Allows you to authorize your connection through your Google account without needing a service or user-based key file.	GCP SSO
Access key file	Allows you to provide the path to your service or user credentials files.	GCP Access key

Microsoft Azure


Name	Description
Default credentials	Utilizes Azure identity to authenticate using the identity assigned to the service or local development environment.
Enterprise application	Represents a service principal identity in Azure Active Directory.
Client secret	A confidential string created in the Azure portal to authenticate your application to Azure services.
Client certificate	Uses a certificate to authenticate an application with Azure Active Directory.

[Read more](#) about available Microsoft Azure credentials types.

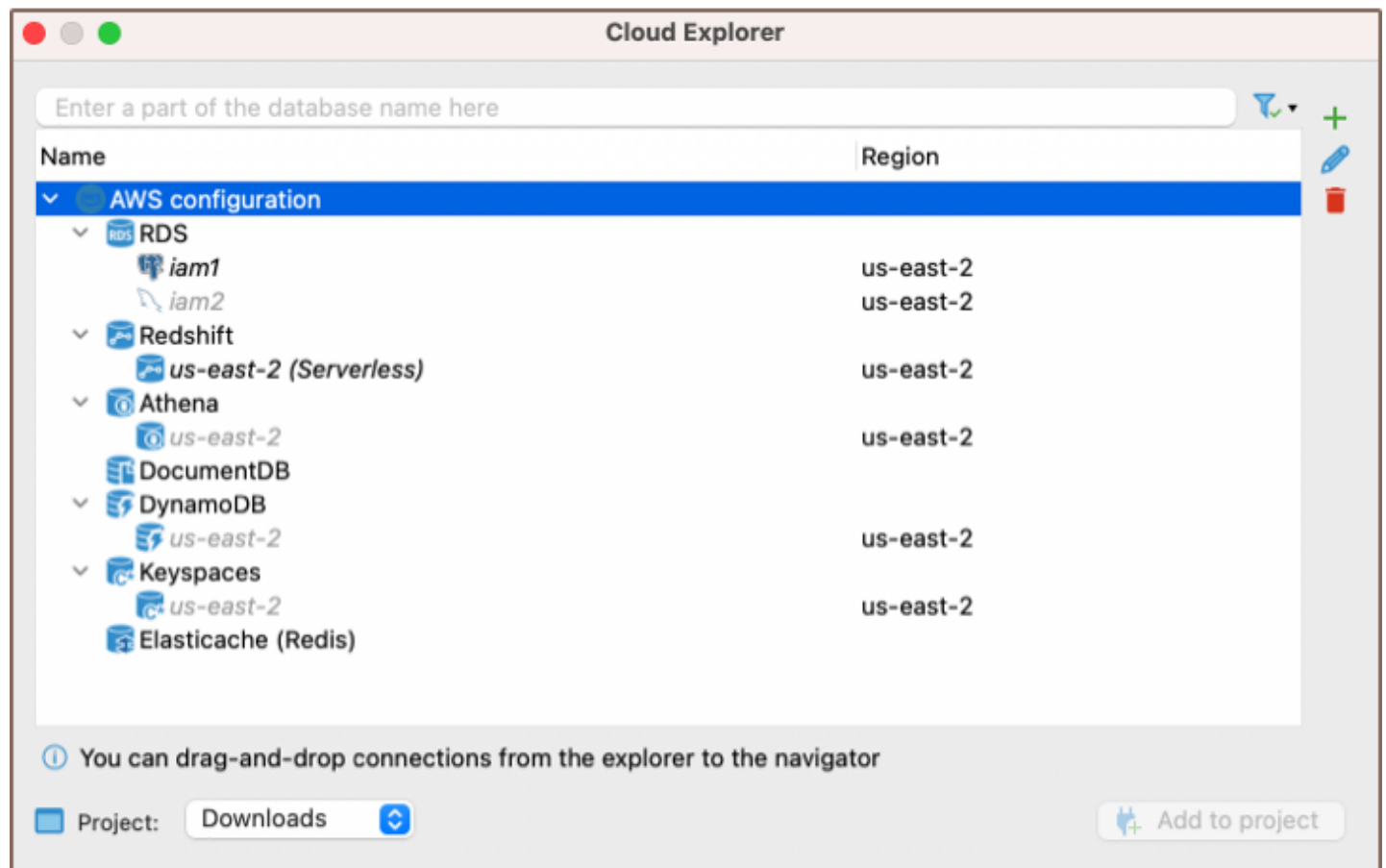
Configuring SSH Tunnels

DBeaver supports SSH tunnel connections for enhanced security when accessing cloud databases. Detailed instructions for setting up SSH tunnels in DBeaver are available in the [SSH Tunnel Configuration](#) article.

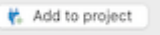
Explorer

Once you've configured your cloud provider access, open the **Cloud Explorer** dialog and add database connections. In the top drop-down of the explorer dialog, you can select the active cloud configuration or click **Edit** button  to change the cloud configuration.

You can see cloud databases in the center of the dialog in a hierarchical view. All databases are grouped by database/service type. When you expand one of the top elements, DBeaver will search cloud databases in configured availability zones/regions.



If you have a large number of databases in your cloud, you can search or filter them using filter text above the cloud navigator.

You can drag-and-drop cloud databases directly to [database navigator view](#) or [projects view](#). You can also add any number of databases to your project directly from the **Cloud Explorer**. To do this, click on the database you want to add. Once selected, click on the **Add to Project** button  located in the bottom right corner of the **Cloud Explorer** dialog, or right-click on the database and select **Add to Project** from the context menu. This will add the selected database to your project, making it easier to manage and navigate your databases within specific projects.

Database cloud information

You can access your cloud database configuration directly from the **Database Navigator**. To do this, right-click on the database you are interested in and select **Edit Connection**. This action opens a special tab in the connection settings dialog. The information displayed in this tab is specific to the cloud and database type of the selected database.

AWS Cloud Explorer

Table of contents

[Overview](#)

[Requirements for AWS connection](#)

[Setting up AWS permissions](#)

[Connection setup](#)

Note: This feature is available in [Ultimate](#) and [Team](#) editions only.

Overview

DBeaver facilitates advanced functionalities for efficiently managing and connecting to Amazon Web Services (AWS) databases through its user interface. The process for setting up and altering connections is detailed in the [Cloud Explorer article](#). This article outlines the key steps and prerequisites for connecting to an AWS database using DBeaver.

Requirements for AWS connection

Before initiating a connection to an AWS database using Cloud Explorer, ensure that you meet the following prerequisites:

- **Existing AWS account:** An active AWS account is required. Ensure it is [properly configured](#) following the setup guide provided by AWS. If you do not have an account, sign up at the [Amazon website](#).
- **Security credentials:** Ensure you have at least one of the following security credentials for your AWS account:
 - AWS Access/Secret keys.
 - Default credentials.
 - AWS profile.


Setting up AWS permissions

Before you initiate a connection, ensure your AWS account is configured with the correct roles and permissions.

- **IAM User Permissions:** Your IAM (Identity and Access Management) user needs to have policies attached that grant the necessary permissions for accessing the database.
- **Role Assignment:** In case you are assuming a role for database access, ensure that the role has the required permissions.
- **Security Group Configuration:** The security group associated with your database instance should be configured to allow inbound traffic from your IP address on the database port.

For detailed instructions on roles and permissions setup, consult the official [AWS roles guide](#).

Connection setup

1. Click on the Cloud Explorer button  or navigate to **Database** -> **Cloud Explorer** in the menu.
2. Navigate to the Amazon Web Services section.

Create cloud configuration

Name:

ID:

☒ Amazon Web Services
 ☐ Microsoft Azure
 ☐ Google Cloud Platform

☒ Authentication
 ☐ Regions
 ☐ Tunnels

Credentials: [Details](#)

Access key:

Secret key:

☒ Save credentials

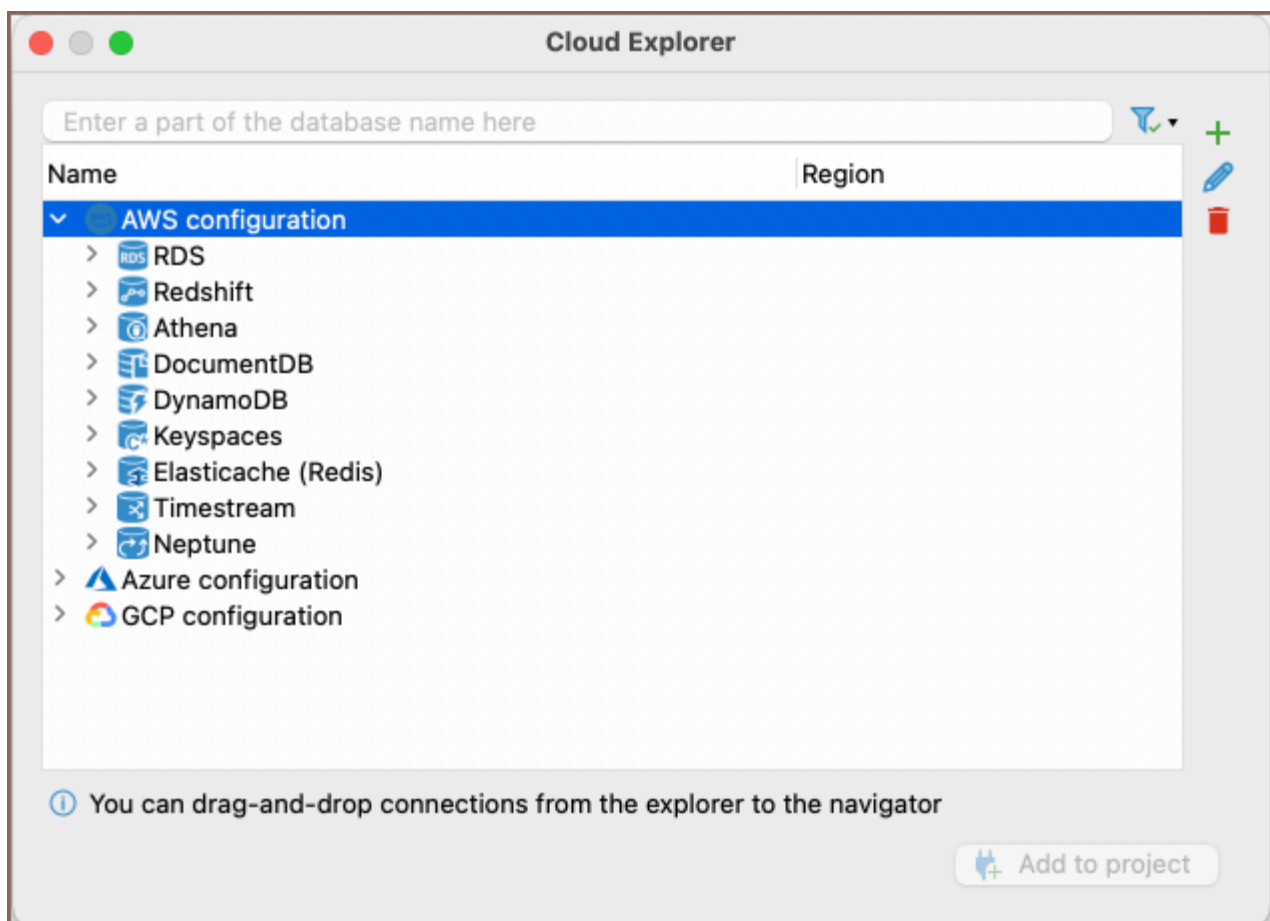
Role Name:

☐ 3rd party account

3. Select one of the available authentication methods in DBeaver.

Tip: For detailed information about the authentication methods supported by DBeaver for AWS connections, see the [Cloud Explorer documentation](#). If your connection requires SSH, fill in the necessary credentials in the [Tunnels](#) tab.

4. Verify the correctness of your credentials by clicking on the **Test Connection** button. Then click on **Apply and Close** to save your settings and close the settings window.
5. After completing the steps, the **AWS configuration** will appear in your configuration list.
6. Select the desired database from the drop-down menu to connect via DBeaver.



Azure Cloud Explorer

Table of contents

[Overview](#)

[Pre-requisites for Connecting to Azure](#)

[Obtaining credentials](#)

[Setting up permissions](#)

[Connection setup](#)

[Authentication types in Azure Cloud Explorer](#)

[Troubleshooting](#)

[Set Environment on macOS](#)

Note: This feature is available in [Ultimate](#) and [Team](#) editions only.

Overview

DBeaver offers enhanced capabilities for managing and connecting to Azure databases directly from the DBeaver interface. The processes for creating and modifying connections are comprehensively detailed in the [Cloud Explorer article](#). This guide explains the essential steps and requirements to connect to an Azure database using DBeaver.

Pre-requisites for Connecting to Azure


Obtaining credentials

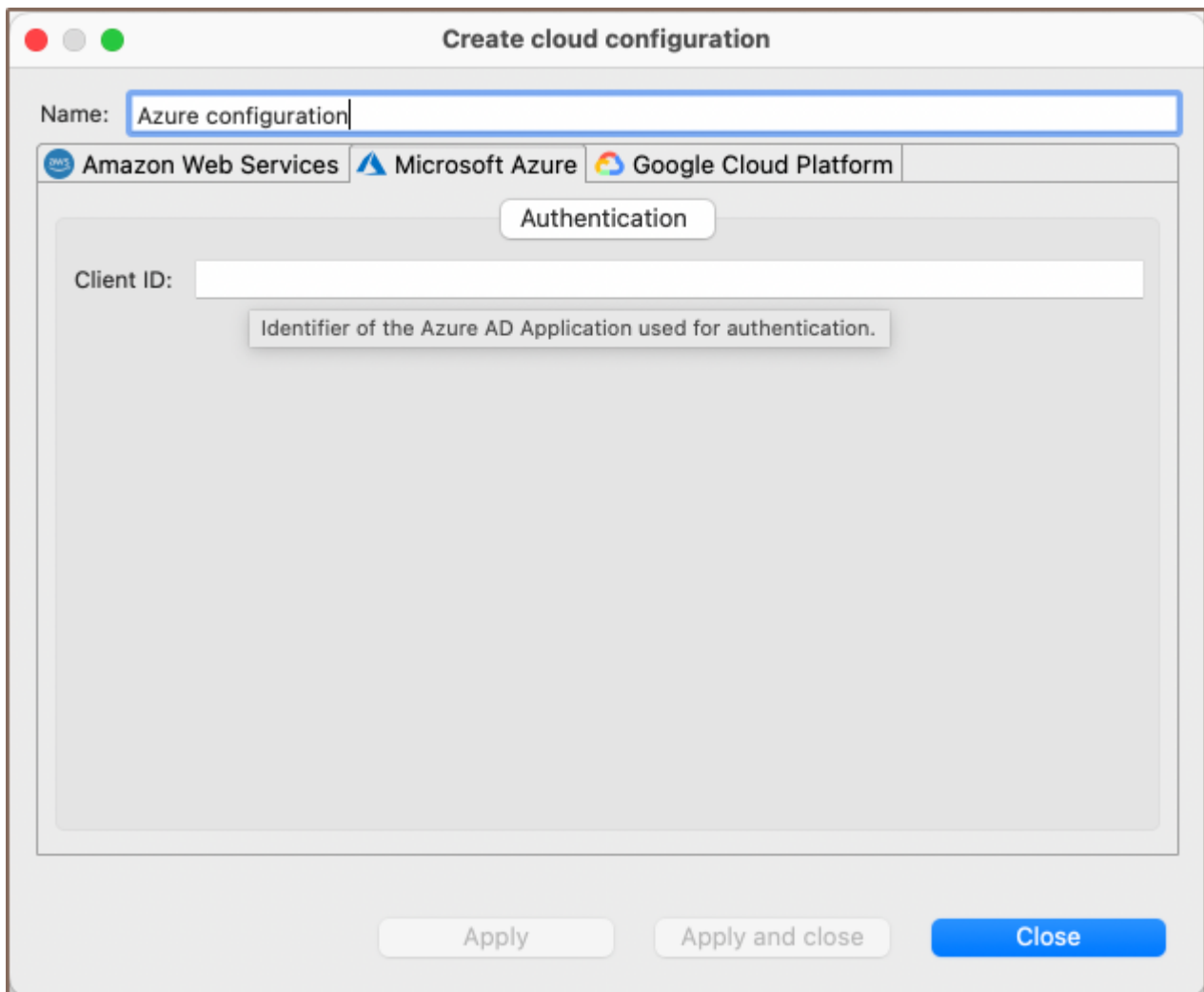
To initiate a connection with Azure, you need to obtain your application credentials. For a comprehensive guide on registering your application with Azure, please refer to the [Microsoft documentation](#).

Setting up permissions

Before you connect, ensure your Azure account is assigned the right roles and permissions. Properly setting up permissions on the Azure portal can help avoid potential access issues during the connection process. For more information on configuring roles and permissions, please consult the [Azure roles guide](#).

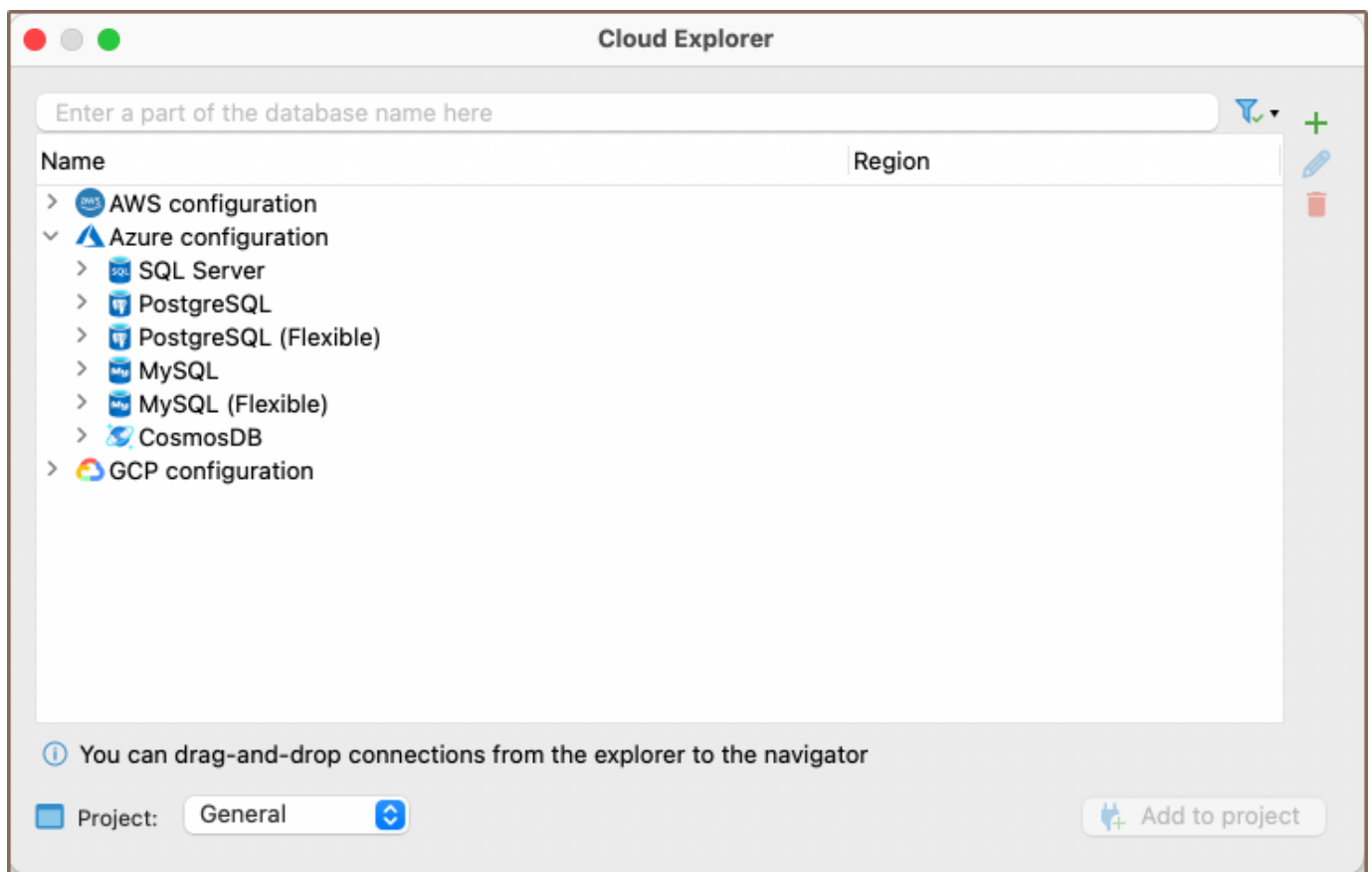
Connection setup

1. Click on the Cloud Explorer button  or navigate to **Database** -> **Cloud Explorer** in the menu.
2. Navigate to the **Microsoft Azure** section, choose the authentication type and enter your credentials.



[Read more](#) about available credentials types.

3. Click on **Apply and Close**. This action will save your input and close the settings window.
4. Once you have completed the previous step, you will notice **Azure configuration** appear in your configurations. This means you have successfully set up your **Client_id**, and Azure recognizes your application.



5. Select the database you want to connect to within the drop-down menu. This action initiates the process of connecting your selected database to DBeaver.
6. You will be redirected to the Microsoft authentication page. Here, you need to enter your login details.
7. After successful authentication, return to DBeaver to continue working with your connected database.

Authentication types in Azure Cloud Explorer

There are several credential types available in Azure Cloud Explorer.

Authentication Type	Description	Required credentials
Default credentials	Attempt to authenticate through several methods.	Read more about Default credentials
Enterprise application	Use credentials from your Azure Application.	- Client ID - Tenant ID
Client secret	Provide specific credentials from your Azure Application.	- Client ID - Client Secret

		- Tenant ID
Client certificate	Utilize certificate-based authentication with your Azure Application.	- Client ID - Tenant ID - Upload the relevant certificate

Troubleshooting

Set Environment on macOS

Issue - Launching DBeaver from Finder or Launchpad may result in incorrect environment variables.

Solution - Set the environment variables globally using `launchctl` :

1. Open Terminal.

2. Run:

```
launchctl setenv AZURE_TENANT_ID VARIABLE_VALUE
launchctl setenv AZURE_CLIENT_ID VARIABLE_VALUE
launchctl setenv AZURE_CLIENT_SECRET VARIABLE_VALUE
```

3. Restart DBeaver.

Google Cloud Explorer

Note: This feature is available in [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Requirements for Google Cloud Platform connection](#)

[Connection setup](#)

Overview

DBeaver offers enhanced capabilities for managing and connecting to Google Cloud Platform (GCP) databases directly from the DBeaver interface. The processes for creating and modifying connections are comprehensively detailed in the [Cloud Explorer article](#). This guide explains the essential steps and requirements to connect to a GCP databases using DBeaver.

Requirements for Google Cloud Platform connection

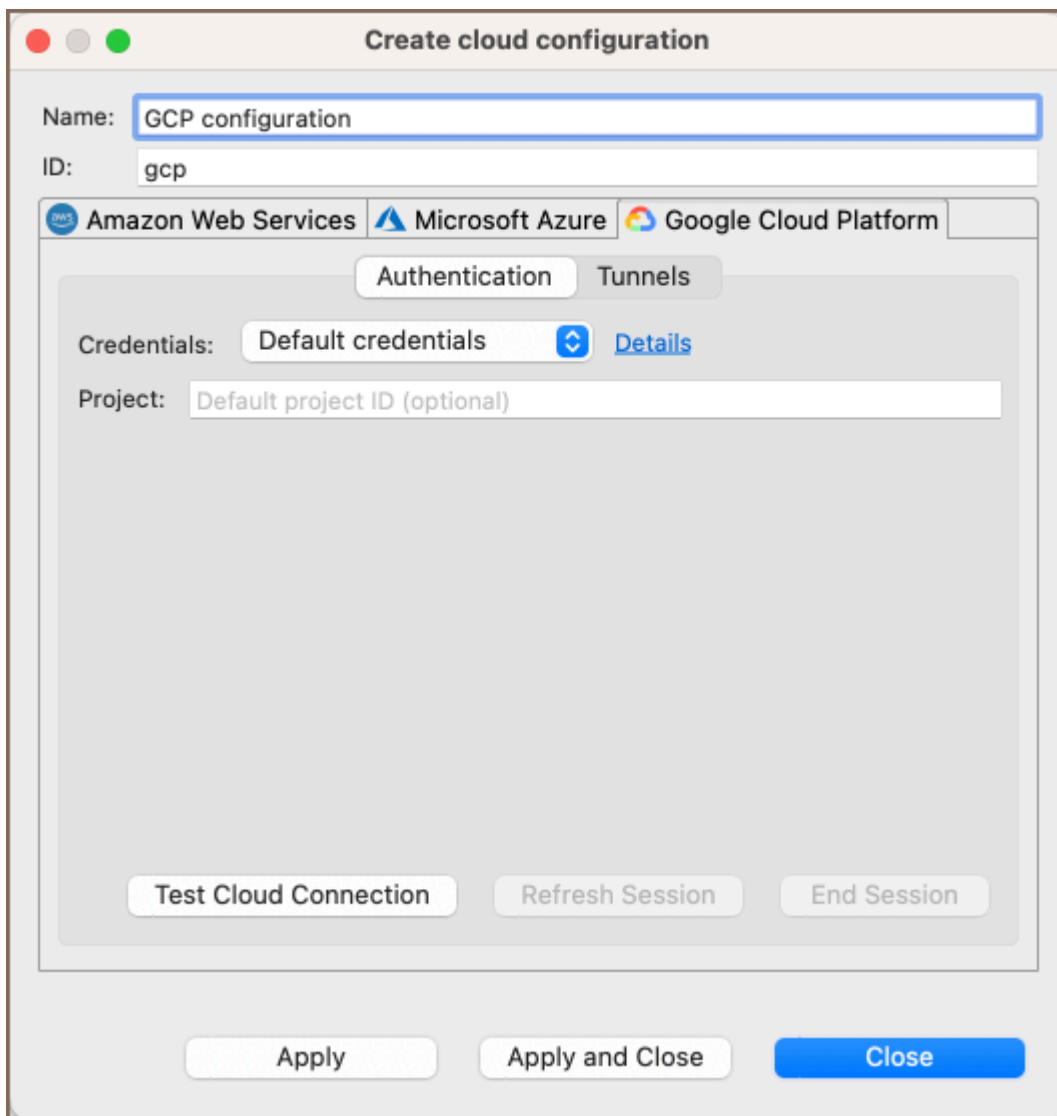
Before initiating a connection to a Google Cloud Platform (GCP) database using DBeaver, ensure that you meet the following pre-requisites:

- **Existing GCP account:** You must have an active and properly configured GCP account. If you do not have an account, you can create one by visiting the [GCP website](#).
- **Authentication set-up:** Proper authentication is crucial for a secure and successful connection to your GCP database. For detailed instructions on setting up authentication, refer to Google Cloud's [authentication documentation](#).

Connection setup

1. Click on the Cloud Explorer button  or navigate to **Database -> Cloud Explorer** in the menu.

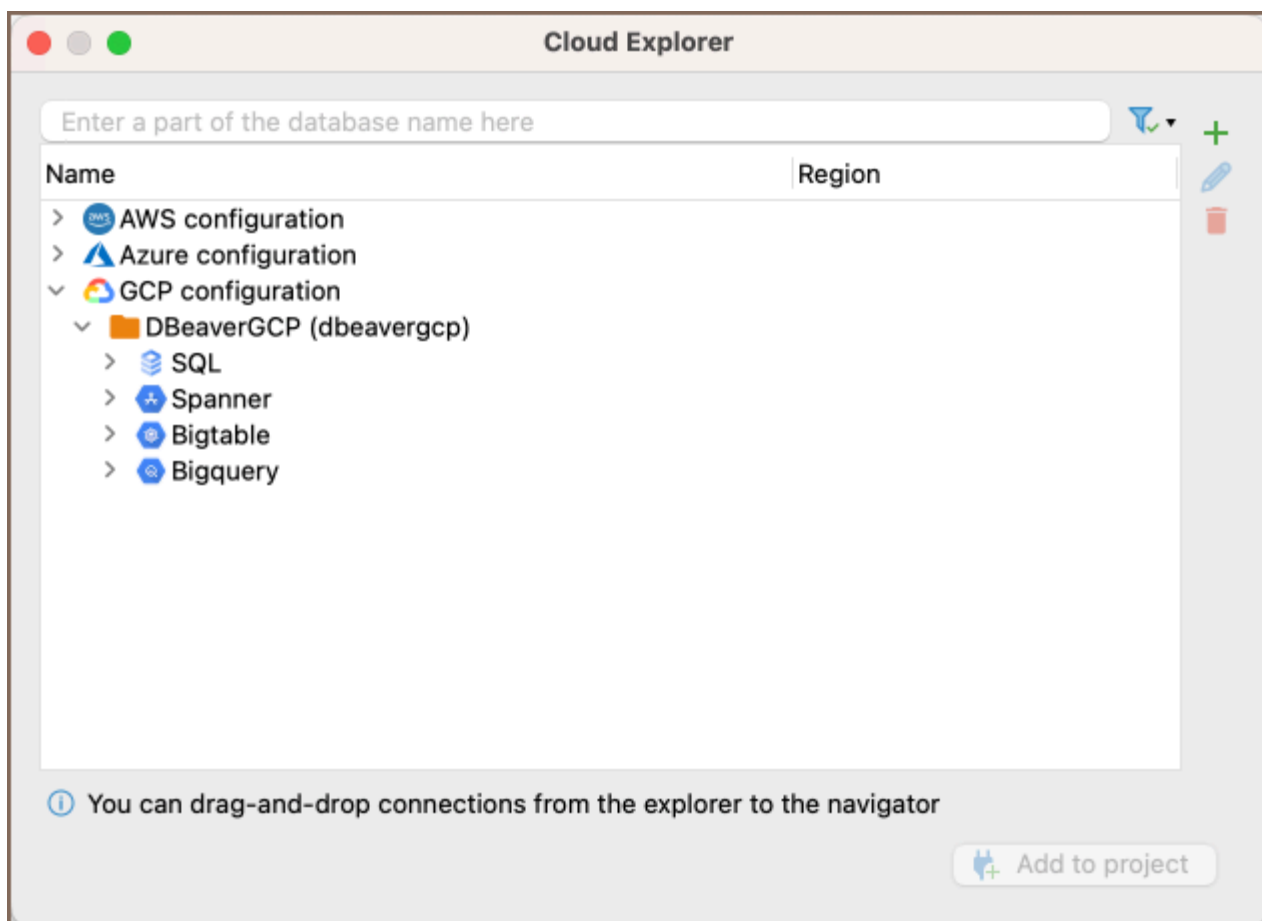
2. Proceed to the **Google Cloud Platform** section within Cloud Explorer.



3. Select one of the available authentication methods in DBeaver.

Tip: For detailed information about the authentication methods supported by DBeaver for GCP connections, see the [Cloud Explorer documentation](#). If your connection requires SSH, fill in the necessary credentials in the [Tunnels](#) tab.

4. Verify the correctness of your credentials by clicking on the **Test Connection** button. Then click on **Apply and Close** to save your settings and close the settings window.
5. After completing the steps, the **GCP configuration** will appear in your configuration list.
6. Select the desired database from the drop-down menu to start connecting it to DBeaver.



Cloud Storage

Note: This feature is available in [Ultimate](#) and [Team](#) editions, as well as in [CloudBeaver Enterprise](#).

Table of contents

[Overview](#)

[Cloud Storage](#)

[Capabilities in Cloud Storage](#)

[Direct capabilities in Cloud Storage](#)

[Additional interactions via other features](#)

Overview

Cloud Storage extends the functionality of [Cloud Explorer](#) to enable file management within cloud environments of Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure. This feature simplifies the process of interacting with cloud-stored files. To use **Cloud Storage**, you must have either an existing cloud provider access connection or configure a new one.


The following are the specific cloud storage services that **Cloud Storage** interacts with in each cloud environment:

- AWS: [Amazon Simple Storage Service \(S3\)](#)
- GCP: [Google Cloud Storage](#)

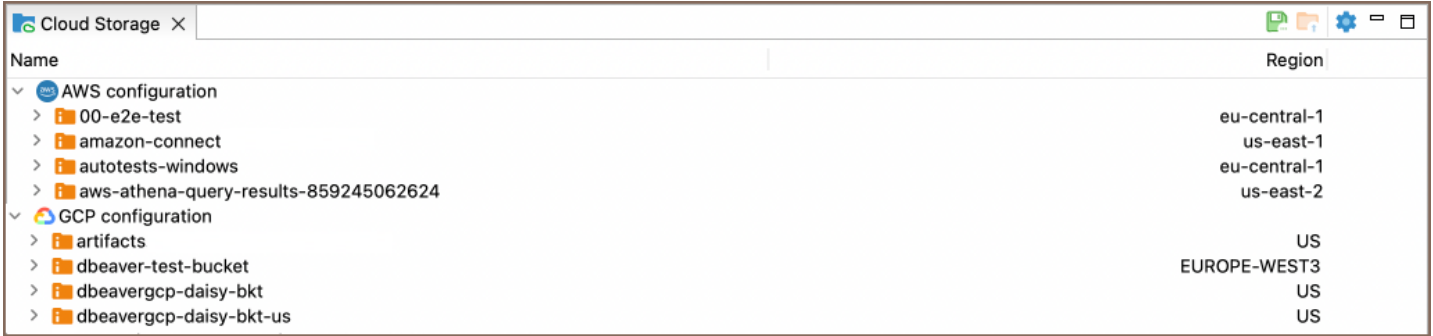
Tip: For detailed setup instructions, refer to the [Cloud configuration](#) section in the **Cloud Explorer** article.

Cloud Storage

After configuring your cloud provider access, you can start using **Cloud Storage**. To access it, follow these steps:

- Click the **Cloud Storage** button  on the toolbar.
- Alternatively, go to the menu and select **Window -> Cloud Storage**.

This action will launch a window displaying the cloud services for which credentials were provided during setup. In this window, use the dropdown menu to navigate through folders and view files stored in the cloud.







You can filter and sort your folders and files by their name or region using the interface's **Name** and **Region** columns. Click on the respective column header to organize the displayed folders, according to these criteria, especially useful when managing a large number of folders.


Capabilities in Cloud Storage

The **Cloud Storage** offers a range of functionalities for managing files within the cloud. Here are the actions you can perform:

Direct capabilities in Cloud Storage

Some functionalities are directly accessible within the **Cloud Storage** interface:

Capability	Description
Open Resource	<p>Open a folder in a new window to view details like Name , Path , Size , Location , and Last Modified .</p> <p>Double-click on the folder, or right-click on the folder and select Open resource folder name from the context menu.</p> <p>Use the Configure columns button  to customize which columns are displayed in the window, tailoring the view to your specific needs.</p> <p>In the opened window, use the Search item button  to conduct searches within the folder.</p>
Create new folder	<p>Create new folders within the cloud environment. To do this, right-click where you want the new folder and select Create New Folder button .</p>
Save resource	<p>Save selected resource(s) from the cloud to the local disk. Select the file(s) or folder(s) in Cloud Storage you wish to download, then choose the Save resource(s) to local disk option  available both in the top toolbar of Cloud</p>


(s) to local disk	Storage and the context menu.
Load resource (s) from local disk	Upload resource(s) from the local disk to the cloud. Select the target folder in Cloud Storage where you want to upload files, then choose the Load resource(s) from local disk option  which can be found in the top toolbar of Cloud Storage as well as the context menu.

Tip: In addition to the **Save to Local Disk** and **Load from Local Disk** features, you can also copy files using the drag-and-drop mechanism.

Additional interactions via other features

Some functionalities of **Cloud Storage** are accessible through the integration with other features of DBeaver:

Capability	Description
Export /Import files	Export to or import files from folders within the Cloud Storage . Details are available in Data transfer article.
Backup /Restore	Perform backup and restore operations using files from or saving outputs to the Cloud Storage , as detailed in Backup and Restore .
Data compare	After comparing files or folders, save the results directly to Cloud Storage . Details are available in Data Comparison .
Schema Compare	Compare database schemas and optionally save comparison results to Cloud Storage . Further information is found in Schema Comparison .

Tip: To ensure you see the most updated information after performing file-related actions in **Cloud Storage**, it's important to refresh the folder manually. Right-click on the folder and select the refresh option  from the context menu.

Managing AWS permissions

Service	Read Policy	Write Policy
Login	-	
Cloud Explorer	-	
S3	AmazonS3ReadOnlyAccess	AmazonS3FullAccess
Redshift	AmazonRedshiftReadOnlyAccess	AmazonRedshiftFullAccess
RDS	AmazonRDSReadOnlyAccess	AmazonRDSFullAccess
DynamoDB	AmazonDynamoDBReadOnlyAccess	AmazonDynamoDBFullAccess
DocumentDB	AmazonDocDBReadOnlyAccess	AmazonDocDBFullAccess
Keyspaces	AmazonKeyspacesReadOnlyAccess	AmazonKeyspacesFullAccess
Athena	"lakeformation:Describe", "lakeformation:Get", "lakeformation:List", "sns:ListTopics", "sns:GetTopicAttributes", "s3:Describe", "s3:Get", "s3:List", "s3:Put", "glue:Get", "glue:List", "glue:BatchGet", "athena:List", "athena:Get", "athena:Batch*", "athena:StartQueryExecution", "athena:StopQueryExecution"	AmazonAthenaFullAccess
Redis	AmazonElastiCacheReadOnlyAccess	AmazonElastiCacheFullAccess

Working with AWS SSO

Table of contents

[AWS CLI](#)

[AWS SSO configuration](#)

[Connection configuration](#)

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

[AWS Single Sign-On](#) is a cloud-based single sign-on (SSO) service that makes it easy to centrally manage SSO access to AWS resources.

You do not need to specify any user credentials explicitly in DBeaver connections configuration. All authorization is performed in a web browser in a 3rd party SSO provider, e.g. Google workspace, Microsoft AD portal, Facebook, etc.

Tip: DBeaver also supports AWS Systems Manager (SSM) for accessing databases. For detailed instructions on setting up AWS SSM, refer to the [AWS SSM setup guide](#).

AWS CLI

You need to install AWS CLI (Command Line Interface) utilities to enable SSO authorization.

[AWS CLI installation](#)

AWS CLI version 2.2 is recommended.

AWS SSO configuration

If you are in a corporate environment where all AWS configurations are provided by system administrators then you do not need to configure SSO parameters. Otherwise, you need to open the command shell (win+R), enter `aws configure sso` , press enter, and provide the required parameters. Read [configuration instructions](#) for the details.

Restart DBeaver after the AWS CLI SSO configuration will be finished.

Connection configuration

In the DBeaver database connection dialog you need to:

- Set Authentication to `AWS IAM`.
- Set Credentials to `AWS Profile`.
- Choose the profile which was configured with AWS SSO (see the previous chapter).
- Click on the `Enable SSO` check.

Now you can connect. DBeaver will open a web browser with SSO authorization.

Working with Google Cloud Explorer SSO

Note: This feature is available in [Ultimate](#) and [Team](#) editions only.

Table of contents

[GCP CLI](#)

[GCP web browser-based authentication configuration](#)

[Connection configuration](#)

GCP web browser-based authentication allows you not to use a service or user-based key file, or other default authorization methods. You can provide access to your connection by authorization through the Google account.

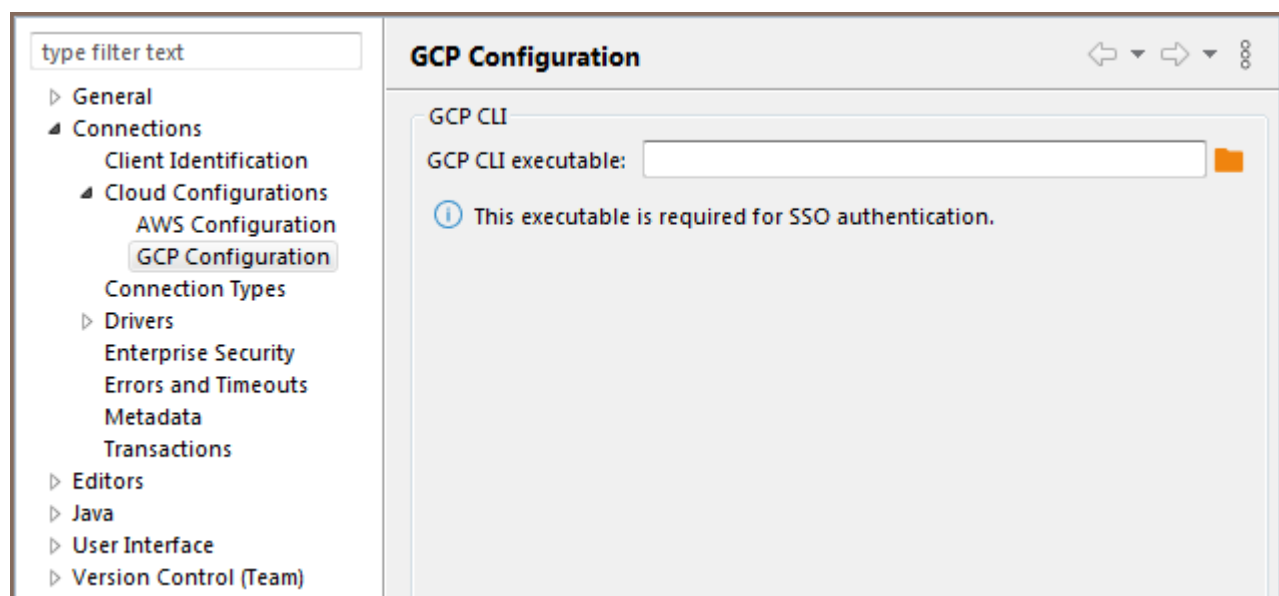
GCP CLI

To enable SSO authorization, you need to install GCP CLI (Command Line Interface) utilities.

[GCP CLI installation](#)

If DBeaver didn't find your executable CLI file (error message: "Authentication error: Error running GCP CLI. Is it installed on the local machine?"), you can add the path for your executable file manually in the Preferences ->

Connections -> Cloud Configurations -> GCP Configuration



GCP web browser-based authentication configuration

First `gcloud auth print-access-token` request to your CLI will be used. If token will be not empty, then this token will be used for the authentication. If token will be empty, then command `gcloud auth login` will be executed, which will open the web-browser and offer to choose your Google account.

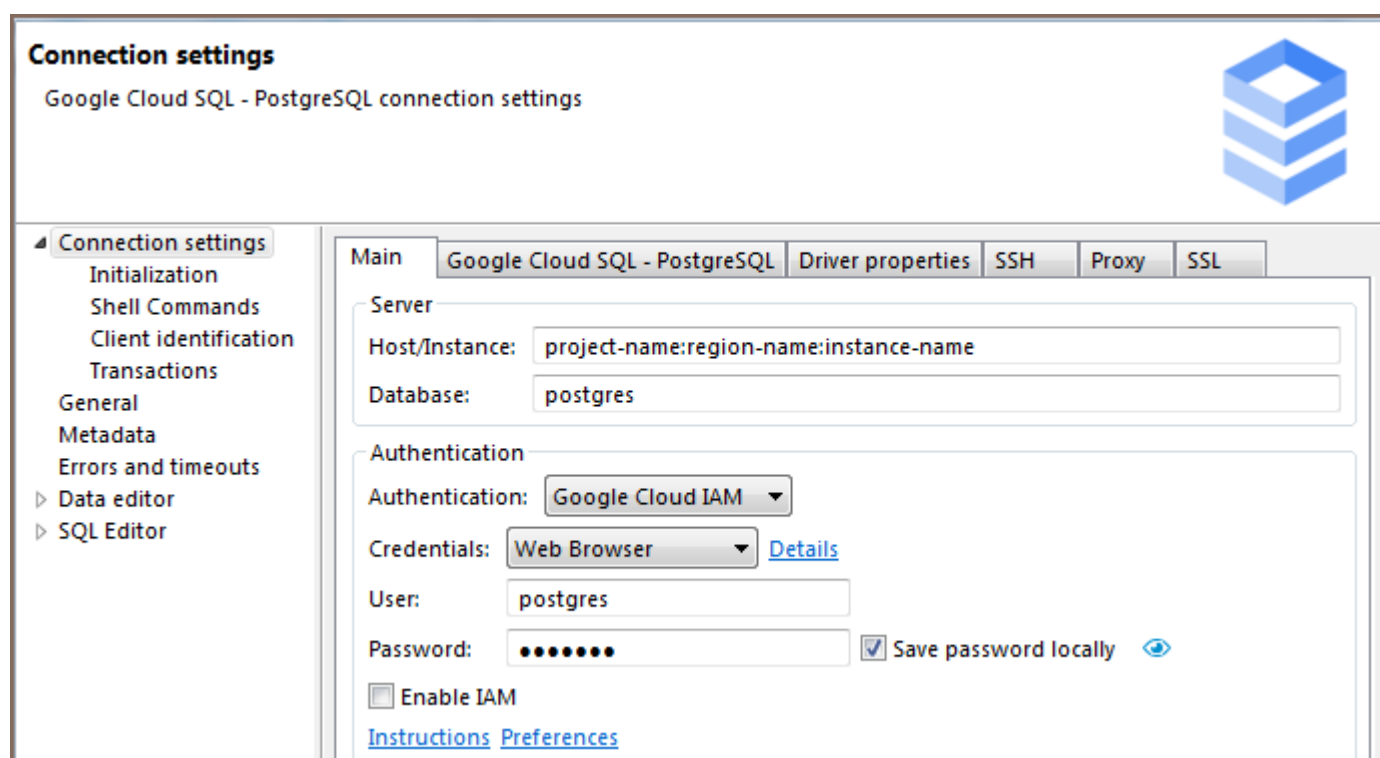
Connection configuration

In the DBeaver database connection dialog, you need to:

- Set Authentication to `Google Cloud IAM`.
- Set Credentials to `Web Browser`.
- Add user and password info if they needed

Now you can connect. DBeaver will open a web browser with SSO authorization. Choose your account with Goggle projects on the web page.

As example:



AWS credentials

Table of contents

[Overview](#)

[Settings](#)

[Credentials](#)

[Default credentials](#)

[Access keys](#)

[AWS Profiles](#)

[SSO](#)

[AWS Secrets Manager](#)

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Overview

DBeaver is integrated with AWS RDS IAM authentication, providing you with the ability to authenticate in AWS to access your cloud databases. There are numerous ways to authorize and authenticate users in DBeaver AWS IAM, and DBeaver supports all the basic ones.

The instructions provided here are intended for the client machine where DBeaver is installed. It is assumed that the necessary configuration on the server side, including the setup of AWS RDS IAM, has already been completed.

Connect to a database

Connection Settings

PostgreSQL connection settings

PostgreSQL

Main
PostgreSQL
Driver properties
SSH
SSL

+ Network configurations...

Server

Connect by:
☒ Host
☐ URL

Driver type:
JDBC

URL:

Host:
Port: 5432

Database: postgres

Authentication

Authentication:
AWS RDS IAM

Credentials:
Access/secret keys
Details

User: postgres
Region: us-east-2

Access key: AWS access key
Secret key:

☒ Save credentials

Role Name: Optional AWS role (ARN or simple name)

☐ 3rd party account
☐ Use AWS Secrets Manager

Advanced

Session role:

Local Client: PostgreSQL 14.6 (Homebrew)

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: PostgreSQL

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Settings

To use IAM authentication in DBeaver, you need to select **AWS RDS IAM** as your authentication method in the connection settings.

Setting	Description
Credentials	Choose between Access/secret keys , AWS profile , or Default credentials for the type of IAM credentials configuration.

User	Input your username.
Region	Optional field to specify the AWS region. If not specified, the global region will be used.
Access key	AWS Access key.
Secret key	AWS Secret key.
Role name	Input the name of the role you want to use.
Save credentials locally	Check this box if you want to save your password locally.
3rd party account	Check this box if you want to access using a 3rd party role-based account.
Use AWS Secrets Manager	Check this box if you want to connect to AWS Secrets Manager to access the database credentials. When selected, a field Secret Name will appear where you can input the name of your AWS Secret .

Tip: DBeaver also supports AWS Systems Manager (SSM) for accessing databases. For detailed instructions on setting up AWS SSM, refer to the [AWS SSM setup guide](#).

Credentials

Choose the type of credentials by selecting the appropriate option from the **Credentials** dropdown menu.

Default credentials

When you use Default Credentials, AWS will then try to determine credentials by using the standard credential providers chain:

1. Java system properties
2. Environment variables
3. Web identity token from AWS STS

4. The shared credentials and config files
5. Amazon ECS container credentials
6. Amazon EC2 instance profile credentials
7. Amazon SSO credentials

Using default credentials is the easiest way to integrate with various Single Sign-On (SSO) and web identity providers, as these providers typically supply credentials through configuration files.

For a more detailed explanation, please refer to the [AWS credentials documentation](#).

To use Default credentials, simply enter your username in the **User** field and select your AWS region.

Access keys

Using the IAM user access key and secret key is the simplest way to authenticate. You just need to input these two keys. You have the option to save them locally for convenience, or, for better security, you can choose to enter them each time you connect to a database.

As with the Default configuration, you need to enter your username and select the AWS region. If you've checked the **Save credentials locally** box, you'll need to fill in the **Access key** and **Secret key** fields. If you haven't checked this box, you'll be prompted to fill in these fields each time you connect to the database.

For more detailed instructions on managing access keys for IAM users, you can refer to the [official AWS guide](#).

AWS Profiles

Just like with default credentials, you also have the option to select a specific credentials profile.

To do this, first select the profile you've already configured. If you need information on how to configure a profile, you can find it below. After selecting a profile, fill in the **User** field and select your AWS region, just like you would with the default credentials.

For more detailed instructions, you can refer to the official AWS guide on [credentials config files](#).

SSO

If you've set up an [SSO](#) portal on your AWS account, you can use it for web-based SSO authorization. This SSO support can be activated for both Default and Profile-based AWS authorization types. To use this feature, you need to enable the **Enable SSO** option.

AWS Secrets Manager

If you've set up an **AWS Secret**, you can use it to access your database. This method can be used for both RDS databases and Redshift. You can find instructions on how to create an **AWS Secret** in [official guide](#). Remember, you'll need to fill in the **Password** field.

To use this feature, check the **Use AWS Secrets Manager** box and then fill in the **Secret Name** field.

Note: Make sure that the secret is in the same region as the database.

Google Cloud Explorer credentials

Table of contents

[Default credentials](#)

[Access key file](#)

[Web browser or Single Sign On](#)

DBeaver is integrated with Google Cloud IAM authentication.

Thus it provides the possibility to authenticate in GCP to access your cloud databases.

Default credentials

When you use Default Credentials, Google Cloud will then try to determine credentials by using the standard credential providers chain:

1. Environment variables (GOOGLE_APPLICATION_CREDENTIALS)
2. Identity token from GCP CLI
3. The shared user or service credentials and config files (usually application_default_credentials.json in AppData)
4. Google Compute Engine

Using default credentials is essentially the simplest way to integrate with various SSO providers and web identity providers as they usually provide credentials through config files.

Please read the [GCP authentication documentation](#) for a detailed explanation.

Access key file

You can provide the path to your service credentials or user credentials file in the "Configuration" field.

You can read more about User and Service authentication [here](#)

Web browser or Single Sign On

Google Cloud Shell will be used for authentication. If your GCP account has a configured SSO portal, you can use a web-based SSO authorization.

Database drivers

Table of contents

[Obtaining JDBC driver](#)

[Adding driver configuration in DBeaver](#)

[Open driver manager dialog](#)

[Add a new driver](#)

[Main parameters](#)

[Libraries](#)

[Maven artifacts](#)

[Saving driver, adding connection](#)

[URL Templates](#)

[Advanced settings](#)

[Main parameters](#)

[Queries -Custom driver queries](#)

[DDL - generation options](#)

[Formatting - SQL values formats](#)

[Summary](#)

You can use a pre-configured database driver or create a new driver.

DBeaver has a lot of pre-configured drivers including SQL, NoSQL, key-value databases, graph databases, search engines, etc. But sometimes you need to connect to a database which was not configured in DBeaver yet.

All you need is a JDBC driver of your database. The rest is easy.

Obtaining JDBC driver

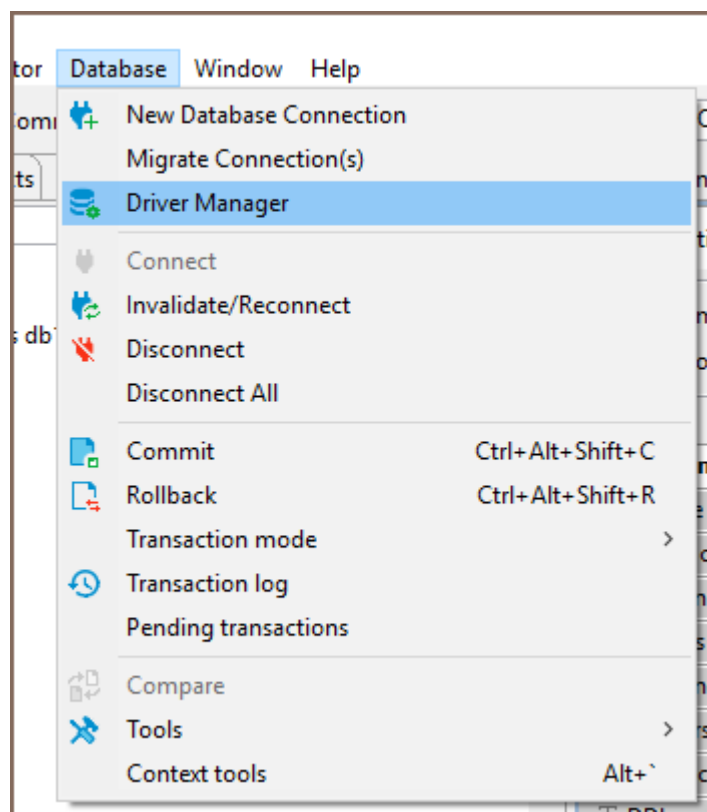
JDBC driver is a program (in Java) which can connect and operate with some local or remote database server. It usually provides all needed functionality to cover 100% of database functionality. The JDBC driver is usually provided by database vendors to allow customers to work with their databases.

The JDBC driver consists of one or multiple `jar` files. The Jar file is a library which contains program code and some other files. You need to download the driver's jar files before adding them to DBeaver. Sometimes the jar files are included in the database server distribution - in that case you need to refer to your database documentation or ask your DBA.

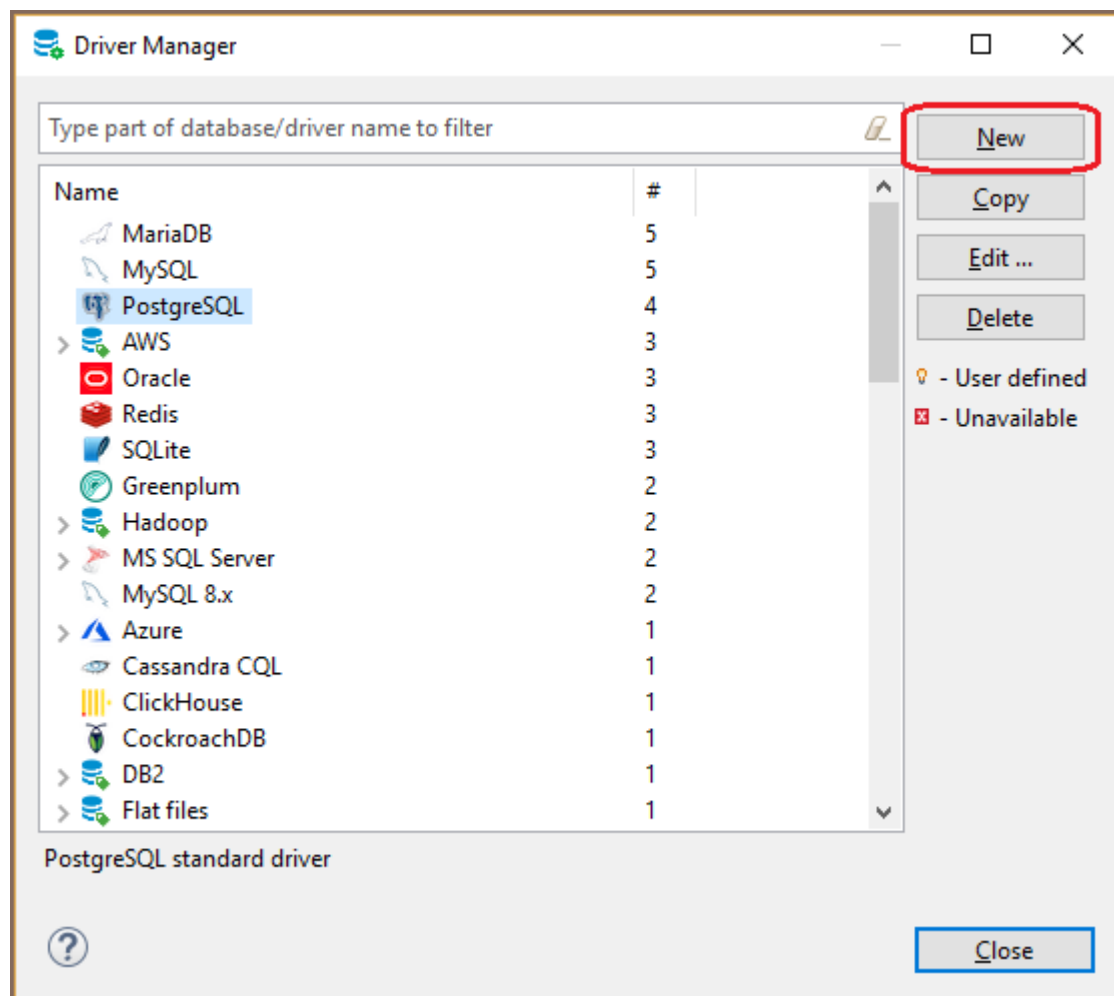
Adding driver configuration in DBeaver

Open driver manager dialog

You can open the driver manager from the main menu:



or from Database Navigator drop-down menu.



Add a new driver

Just click the button New and create a new driver. On the driver edit dialog you need to enter all required information:

Main parameters

Edit Driver 'PostgreSQL Custom'

Settings

Driver Name*: PostgreSQL Custom

Driver Type: Generic

Class Name: org.postgresql.Driver

URL Template: jdbc:postgresql://{host}:{port}/database

Default Port: 5432

☐ Embedded

☐ No authentication

Description

Category:

ID: 0290D03E-D1F1-342D-9F25-543C57C28DE0

Description: Custom PG driver for inhouse development

Libraries

Connection properties

Adv. parameters

JAR

C:\java\pgjdbc\postgresql-42.2.5.jar

Add File

Add Folder

Add Artifact

Download/Update

Information

Delete

Classpath

Driver class:

Find Class

?

Reset to Defaults

OK

Cancel

Parameter	Description
Driver Name	Name of your driver. It can be any name you like
Driver Type	Driver provider. In 99% cases you will need a generic driver (JDBC provider)
	JDBC driver class name. You can get it from the documentation or find it in the jar files (see "Find Class")

Class Name	button description)
URL Template	Template of driver URL. You can leave it empty. But in this case you will be ready to set JDBC URL for each your connection. It is better to define a valid template, which will greatly simplify the connections creation. See "URL Templates" for a detailed description
Default Port	Default database port. You can get it from the documentation or leave it empty
Embedded	Enable it for server-less databases. This flag affects a few config options related to the network/connections management
No Authentication	This means that driver does not require authentication (no user/password fields will be shown)
Category	Driver category, deprecated
ID	Driver unique ID, ignore it
Description	Driver description, it is shown in some dialogs/wizards as a hint

Libraries

This is the list of jar files, binary libraries (dll or so) and any other files required by the driver. In most cases you only need the jar files.

Click "Add File" to add a single jar file, "Add Folder" to add to the folder with Java classes/resources and "Add Artifact" to add the Maven artifact (see below).

After you add the jar files you will be able to find all JDBC driver classes which are found in these jars. Just click on the "Find Class" button and DBeaver will show all of them. In most cases there is just one driver class in the driver. If there are many of them, you need to refer to the driver's documentation.

Maven artifacts

DBeaver can download driver jars directly from the Maven repository (it is a global public repository of Java libraries, usually an open-source). If your database driver is published on some public repository you can use this feature. Maven artifacts are better than plain jar files because you can see all existing driver versions and can change the driver version in runtime without any driver properties reconfiguration.

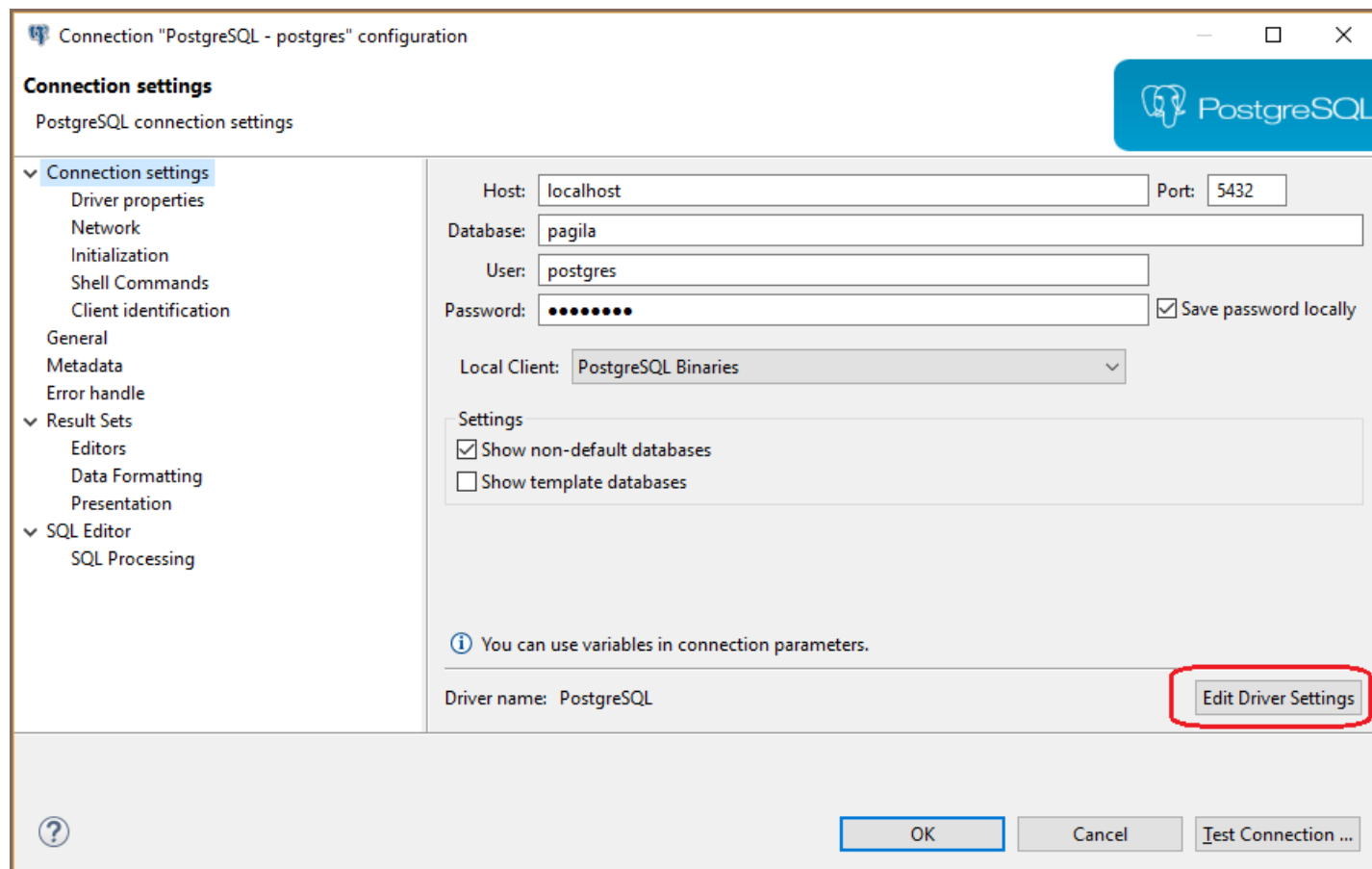
for additional information look [How to add additional artifacts to the driver](#)

Saving driver, adding connection

After you have finished configuring your driver, just press the Ok button.

Now you can [create connection](#).

If you need to change some driver properties later you can access them directly from connection properties dialog:



URL Templates

JDBC drivers use URLs to identify remote servers - strings similar to classic web URLs. Usually, URL has form `jdbc:vendor:host:port/database`, for example `'jdbc:postgresql:localhost:5432/postgres'`. It is not very convenient to edit such a long and an unobvious string. DBeaver can construct this URL from connection parameters (like host, port, etc).

For example above the URL template is: `jdbc:postgresql://{host}:{port}/{database}`

Host, port and database are parameters which you will need to enter on the connection configuration page.

Supported URL variables:

Parameter	Description
{host}	Database server host name
{port}	Database server port number
{database}	Target database name
{server}	Target server name (rarely used)
{folder}	Folder path (on the local file system). Used for embedded drivers
{file}	File path (on the local file system). Used for embedded drivers

Advanced settings

For most drivers you do not need to change any advanced properties. But in some cases you can use this as driver tuning, e.g. for better performance or for structure fixing.

Edit Driver 'PostgreSQL Custom'

Settings

Driver Name*:
Driver Type: Generic

Class Name:

URL Template:

Default Port:
☐ Embedded
☐ No authentication

Description

Category:
ID:

Description:

Libraries

Connection properties

Adv. parameters

Name	Value
Parameters	
Driver supports indexes	<input checked="" type="checkbox"/>
Driver supports stored code	<input checked="" type="checkbox"/>
Driver supports references	<input checked="" type="checkbox"/>
Driver supports SELECT count(*) clause	<input checked="" type="checkbox"/>
Split procedures and functions	<input type="checkbox"/>
Script delimiter	;
Script delimiter redefiner	;
Use script delimiter after query	<input type="checkbox"/>
Use script delimiter after SQL block	<input type="checkbox"/>
String escape character	
Meta model type	

?

Reset to Defaults

OK

Cancel

Main parameters

Parameter	Description
Driver supports indexes	Driver supports table indexes
Driver supports stored code	Whether this driver supports stored code (procedures, functions, packages, etc)
Driver supports references	Driver supports table references (foreign keys)
Driver supports SELECT count() clause / Driver supports SELECT count() clause	
Driver supports views	Driver supports table views
Split procedures and functions	Show procedures and functions in different folders

Script delimiter	Literal for SQL queries separation in scripts
Script delimiter redefiner	SQL clause which redefines script delimiter value
Use script delimiter after query	Keep SQL script delimiter after each SQL query
Use script delimiter after SQL block	Keep SQL script delimiter after SQL script blocks (BEGIN/END)
String escape character	Character used to escape special symbols in strings
Meta model type	Type of metadata reading model - standard or indexed
All Objects Pattern	SQL pattern for all metadata objects
Omit catalog(s)	Do not read and use catalog (aka database) information
Omit single catalog	Hide catalog (database) if there is only one catalog on server
Omit schema(s)	Do not read and use schemas information
Omit single schema	Hide schema if there is only one schema on the server
Use schema filters	Use JDBC schema filters when the database does not support catalogs. Otherwise just read all database schemas and filter on client-side
Omit type cache	Do not use data types provided by driver
Shutdown parameter	Database shutdown URL parameter
Create database parameter	Database create URL parameter
Driver supports multiple results	Driver supports multiple results for a single query
Driver supports result set limit	Driver supports multiple result set limit (max rows)
Driver supports structure cache	Driver supports structure cache reading. Enables schema columns, keys, etc
Driver supports TRUNCATE operation	Driver supports TRUNCATE command. It is much faster than DELETE without criteria

Queries -Custom driver queries

Parameter	Description
Get active database	Query to obtain active database name
Set active database	Query to change active database

Shutdown database	Query to shutdown active database connection. Used for some embedded databases
PING query	Query to check connection state
Dual table name	Name of dummy 'DUAL' table which is used for expressions evaluation
Active object type	Type of selectable object (schema, catalog)
Driver supports results scrolling	Driver supports resultset scrolling
Quote reserved words	Quote columns/table names if they conflicts with reserved SQL keywords
Escape LIKE masks in search queries	Use to access JDBC metadata API. Enabled by default but should be disabled for some (broken) drivers

DDL - generation options

Parameter	Description
Drop column short syntax	Use 'ALTER TABLE DROP column-name' instead of standard syntax
Drop column - use brackets	Use 'ALTER TABLE DROP (column-name)' instead of standard syntax
Use legacy SQL dialect for DDL	Use legacy SQL dialect for DDL
Add COLUMN keyword in alter table query	Add COLUMN keyword after keyword ADD and before column name in alter table query

Formatting - SQL values formats

Parameter	Description
Timestamp format	Format pattern for timestamp columns
Date format	Format pattern for date columns
Time format	Format pattern for time columns

Summary

If you have configured some driver, it works well and you think that it makes sense to have this driver configuration in standard DBeaver, please send your configuration to us. Just create a feature request issue on GitHub and copy/paste driver description to the ticket (in any suitable form).

Thank you.

How to add additional artifacts to the driver

Table of contents

[Introduction](#)

[How to open](#)

[How to edit](#)

[How to add](#)

[How to use](#)

[Dependency Declaration](#)

[Supported formats](#)

[Declare Artifacts Manually](#)

Introduction

DBeaver has the functionality to add and edit artifacts in existing drivers.

How to open

Database -> Driver Manager -> **Select driver** -> Libraries

How to edit

Double click on the library

How to add

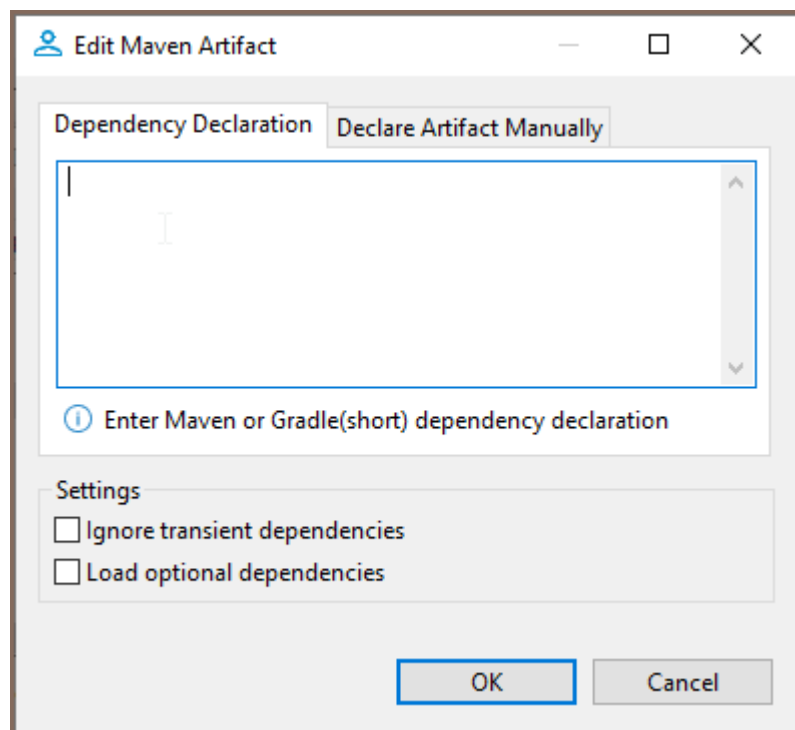
Click 'Add Artifact'

How to use

For add mode, **DBeaver** supports two modes **Dependency Declaration** and **Declare Artifacts Manually**. The editing mode only supports the second **Declare Artifacts Manually**.

Dependency Declaration

Mode parses the inputted text and creates artifacts out of the results of parsing. It supports the multiple input formats written below.



Supported formats

- Short Gradle

```
group:artifact_name:version
```

- Maven

- For a single artifact

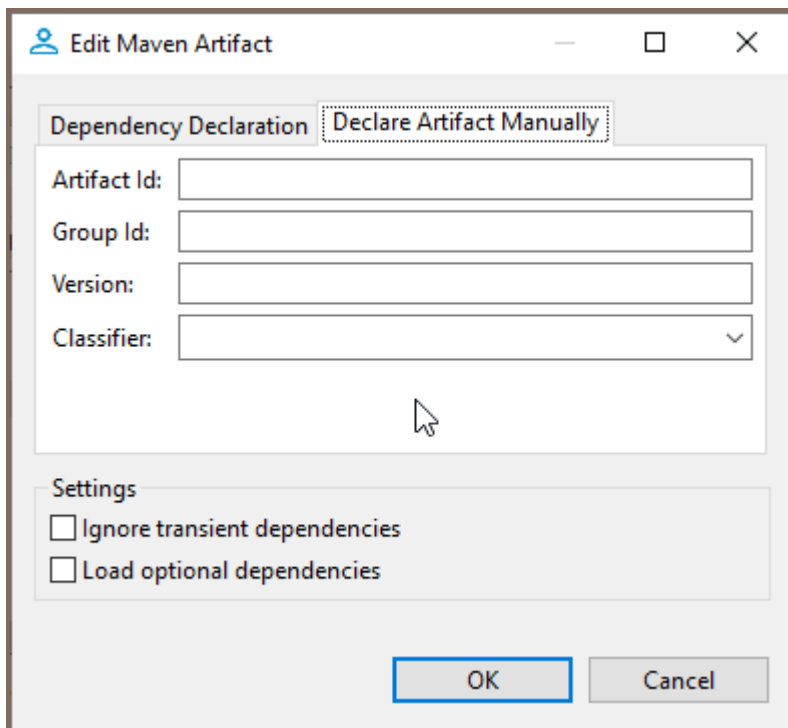
```
<dependency>  
  <groupId>group</groupId>  
  <artifactId>artifact</artifactId>  
  <version>version</version>  
</dependency>
```

- For multiple artifacts

```
<dependencies>
  <dependency>
    <groupId>group</groupId>
    <artifactId>artifact</artifactId>
    <version>version</version>
  </dependency>
  <dependency>
    <groupId>group</groupId>
    <artifactId>artifact</artifactId>
    <version>version</version>
  </dependency>
  <dependency>
    <groupId>group</groupId>
    <artifactId>artifact</artifactId>
    <version>version</version>
  </dependency>
</dependencies>
```

Declare Artifacts Manually

It allows you to manually set up parameters. It allows you to add only one item per dialog.



The screenshot shows a dialog box titled "Edit Maven Artifact". It has two tabs: "Dependency Declaration" and "Declare Artifact Manually", with the latter being selected. The "Declare Artifact Manually" tab contains four input fields: "Artifact Id:", "Group Id:", "Version:", and "Classifier:". The "Classifier:" field has a dropdown arrow. Below these fields is a "Settings" section with two checkboxes: "Ignore transient dependencies" and "Load optional dependencies". At the bottom of the dialog are "OK" and "Cancel" buttons.

ODBC JDBC driver

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Installing ODBC driver manager](#)

[Installing ODBC drivers](#)

[Configuring ODBC data sources](#)

[Creating ODBC connections in DBeaver](#)

[Making existing DBeaver connections use ODBC driver](#)

ODBC allows you to create a connection to almost any database in DBeaver by using native ODBC drivers. You can use it as an alternative to DBeaver drivers or if DBeaver does not have a driver for your database.

As of now, we aim to support any ODBC drivers. However, the following database drivers have been verified for compatibility:

- Amazon Redshift
- BigQuery
- DB2 for LUW
- DB2 for IBM i
- MariaDB
- MySQL
- Microsoft Access
- Oracle
- PostgreSQL
- SQL Server
- SQLite

- Snowflake

Installing ODBC driver manager

DBeaver works with ODBC drivers using ODBC driver manager. A driver manager is a component is installed in your operating system that provides access to drivers written in native programming languages.

- On **Windows**, a driver manager is already installed by default.

You can access it by pressing `WIN+R` and typing `odbcad32`.

- On **Linux** and **macOS**, you need to install this component separately.

We recommend the [iODBC driver manager](#) as it's confirmed to work flawlessly with DBeaver against Linux and macOS. To install, follow the provided instructions.

You can access it by typing `iodbcadm-gtk` in the terminal.

Installing ODBC drivers

- On **Windows**, it's done by installing a driver using Windows Installer (`.msi`). The installation is the same as for any other software.

- On **Linux** and **macOS**, a driver can be either installed from a package manager or by building it manually.

Once a driver is installed, you need to add it to the driver manager. Open the driver manager, go to **ODBC Drivers** tab, press **Add a driver** and select required files.

Configuring ODBC data sources

Note: The process is mostly the same for all operating systems.

1. Open the driver manager
2. Go to **User DSN** or **System DSN** tab
 1. Choose **User DSN** if you want to create a data source accessible only by the current user

2. Choose **System DSN** if you want to create a data source accessible by all users
3. Click **Add** and choose the appropriate driver
4. Configure data source settings and click **OK**

Creating ODBC connections in DBeaver

Open the **New Database Connection** dialog and select **ODBC** (Not **ODBC - Legacy**).

If you want to use a **configured ODBC data source**, switch to **Data Source** connection type and choose it from a drop-down menu:

Data Source	Data Source (drop-c
	

If you want to use an **ODBC driver directly** and manually configure connection parameters, switch to **Manual** connection type and choose an appropriate driver from a drop-down menu and fill rest of the fields according to your configuration:


Manual	Manual (drop-down
	

Making existing DBeaver connections use ODBC driver

DBeaver allows you to replace the underlying database driver of a connection with our ODBC driver. This is not the same as creating a new ODBC connection as it keeps all database-specific functionality untouched.


Note: In this mode, you can't use database-specific authentication methods and settings.

To do so, open existing connection and choose **ODBC** as the driver type from the drop-down menu:

 Connect to a database

SQLite Connection Settings

SQLite connection settings



Main

Extensions

Driver properties

SSH

Proxy

General

Connect by: ☒ Host ☐ URL

Driver type: JDBC


JDBC URL: jdbc:sqlite:{file}

Path:

Open ...

Advanced

☐ Database is remote

 You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: SQLite

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Finish

Cancel

Deprecated legacy ODBC driver

Table of contents

[How to connect via ODBC](#)

[Use ODBC driver in PRO versions](#)

[Download Legacy ODBC from GitHub](#)

[Use a third-party ODBC driver](#)

[How to transfer connections from the old ODBC to the new one in the Community version](#)

The Legacy ODBC driver was removed in DBeaver Community Edition **23.1**.

Why was it removed?

- DBeaver users have encountered many issues with the outdated ODBC driver related to old drivers, which cannot be fixed by the DBeaver team.
- This ODBC driver is based on the obsolete Java version 7.
- It does not work on Linux and macOS, and that cannot be improved.

How to connect via ODBC

If you want to connect to the databases via ODBC, you have the following options:

Use ODBC driver in PRO versions

The new ODBC JDBC driver developed by the DBeaver team supports most new database drivers, works on Linux and macOS, and will be supported by the DBeaver team.

[Learn more about the PRO ODBC driver](#)

Download Legacy ODBC from GitHub

This driver is deprecated and not recommended for use because of many issues. It is based on Java 7, which prevents us from improving it. The DBeaver team does not support this driver and is not responsible for its work, but you can use it at your own risk.

[Download the legacy ODBC driver](#)

Use a third-party ODBC driver

You can look for another ODBC driver that matches your database and try to use it. Follow this instruction to add a driver in DBeaver and configure it.

[How to add a driver in DBeaver](#)

How to transfer connections from the old ODBC to the new one in the Community version

If you have been using the legacy ODBC driver in DBeaver Community and have some connections associated with it, you need to do the following:

1. Install the new ODBC driver that you want to use instead of the legacy one ([legacy ODBC from GitHub](#) or [third-party ODBC](#)).
2. Re-associate the old connections: select such connections, go to Database -> Change Connection(s) Driver, and find the newly installed ODBC.

Another option is to remove the old connections created with the legacy ODBC driver and create new connections associated with the newly installed ODBC.

JDBC time zones

ID	Display Name	GMT shift
Africa/Abidjan	Greenwich Mean Time	0.0
Africa/Accra	Ghana Mean Time	0.0
Africa/Addis_Ababa	Eastern African Time	3.0
Africa/Algiers	Central European Time	1.0
Africa/Asmara	Eastern African Time	3.0
Africa/Asmera	Eastern African Time	3.0
Africa/Bamako	Greenwich Mean Time	0.0
Africa/Bangui	Western African Time	1.0
Africa/Banjul	Greenwich Mean Time	0.0
Africa/Bissau	Greenwich Mean Time	0.0
Africa/Blantyre	Central African Time	2.0
Africa/Brazzaville	Western African Time	1.0
Africa/Bujumbura	Central African Time	2.0
Africa/Cairo	Eastern European Time	2.0
Africa/Casablanca	Western European Time	0.0
Africa/Ceuta	Central European Time	1.0
Africa/Conakry	Greenwich Mean Time	0.0
Africa/Dakar	Greenwich Mean Time	0.0
Africa/Dar_es_Salaam	Eastern African Time	3.0
Africa/Djibouti	Eastern African Time	3.0
Africa/Douala	Western African Time	1.0
Africa/El_Aaiun	Western European Time	0.0

Africa/Freetown	Greenwich Mean Time	0.0
Africa/Gaborone	Central African Time	2.0
Africa/Harare	Central African Time	2.0
Africa/Johannesburg	South Africa Standard Time	2.0
Africa/Juba	Eastern African Time	3.0
Africa/Kampala	Eastern African Time	3.0
Africa/Khartoum	Eastern African Time	3.0
Africa/Kigali	Central African Time	2.0
Africa/Kinshasa	Western African Time	1.0
Africa/Lagos	Western African Time	1.0
Africa/Libreville	Western African Time	1.0
Africa/Lome	Greenwich Mean Time	0.0
Africa/Luanda	Western African Time	1.0
Africa/Lubumbashi	Central African Time	2.0
Africa/Lusaka	Central African Time	2.0
Africa/Malabo	Western African Time	1.0
Africa/Maputo	Central African Time	2.0
Africa/Maseru	South Africa Standard Time	2.0
Africa/Mbabane	South Africa Standard Time	2.0
Africa/Mogadishu	Eastern African Time	3.0
Africa/Monrovia	Greenwich Mean Time	0.0
Africa/Nairobi	Eastern African Time	3.0
Africa/Ndjamena	Western African Time	1.0
Africa/Niamey	Western African Time	1.0
Africa/Nouakchott	Greenwich Mean Time	0.0

Africa/Ouagadougou	Greenwich Mean Time	0.0
Africa/Porto-Novo	Western African Time	1.0
Africa/Sao_Tome	Greenwich Mean Time	0.0
Africa/Timbuktu	Greenwich Mean Time	0.0
Africa/Tripoli	Eastern European Time	2.0
Africa/Tunis	Central European Time	1.0
Africa/Windhoek	Western African Time	1.0
America/Adak	Hawaii Standard Time	-10.0
America/Anchorage	Alaska Standard Time	-9.0
America/Anguilla	Atlantic Standard Time	-4.0
America/Antigua	Atlantic Standard Time	-4.0
America/Araguaina	Brasilia Time	-3.0
America/Argentina/Buenos_Aires	Argentine Time	-3.0
America/Argentina/Catamarca	Argentine Time	-3.0
America/Argentina /ComodRivadavia	Argentine Time	-3.0
America/Argentina/Cordoba	Argentine Time	-3.0
America/Argentina/Jujuy	Argentine Time	-3.0
America/Argentina/La_Rioja	Argentine Time	-3.0
America/Argentina/Mendoza	Argentine Time	-3.0
America/Argentina/Rio_Gallegos	Argentine Time	-3.0
America/Argentina/Salta	Argentine Time	-3.0
America/Argentina/San_Juan	Argentine Time	-3.0
America/Argentina/San_Luis	Argentine Time	-3.0
America/Argentina/Tucuman	Argentine Time	-3.0
America/Argentina/Ushuaia	Argentine Time	-3.0

America/Aruba	Atlantic Standard Time	-4.0
America/Asuncion	Paraguay Time	-4.0
America/Atikokan	Eastern Standard Time	-5.0
America/Atka	Hawaii Standard Time	-10.0
America/Bahia	Brasilia Time	-3.0
America/Bahia_Banderas	Central Standard Time	-6.0
America/Barbados	Atlantic Standard Time	-4.0
America/Belem	Brasilia Time	-3.0
America/Belize	Central Standard Time	-6.0
America/Blanc-Sablon	Atlantic Standard Time	-4.0
America/Boa_Vista	Amazon Time	-4.0
America/Bogota	Colombia Time	-5.0
America/Boise	Mountain Standard Time	-7.0
America/Buenos_Aires	Argentine Time	-3.0
America/Cambridge_Bay	Mountain Standard Time	-7.0
America/Campo_Grande	Amazon Time	-4.0
America/Cancun	Eastern Standard Time	-5.0
America/Caracas	Venezuela Time	-4.0
America/Catamarca	Argentine Time	-3.0
America/Cayenne	French Guiana Time	-3.0
America/Cayman	Eastern Standard Time	-5.0
America/Chicago	Central Standard Time	-6.0
America/Chihuahua	Mountain Standard Time	-7.0
America/Coral_Harbour	Eastern Standard Time	-5.0
America/Cordoba	Argentine Time	-3.0

America/Costa_Rica	Central Standard Time	-6.0
America/Creston	Mountain Standard Time	-7.0
America/Cuiaba	Amazon Time	-4.0
America/Curacao	Atlantic Standard Time	-4.0
America/Danmarkshavn	Greenwich Mean Time	0.0
America/Dawson	Pacific Standard Time	-8.0
America/Dawson_Creek	Mountain Standard Time	-7.0
America/Denver	Mountain Standard Time	-7.0
America/Detroit	Eastern Standard Time	-5.0
America/Dominica	Atlantic Standard Time	-4.0
America/Edmonton	Mountain Standard Time	-7.0
America/Eirunepe	Acre Time	-5.0
America/El_Salvador	Central Standard Time	-6.0
America/Ensenada	Pacific Standard Time	-8.0
America/Fort_Nelson	Mountain Standard Time	-7.0
America/Fort_Wayne	Eastern Standard Time	-5.0
America/Fortaleza	Brasilia Time	-3.0
America/Glace_Bay	Atlantic Standard Time	-4.0
America/Godthab	Western Greenland Time	-3.0
America/Goose_Bay	Atlantic Standard Time	-4.0
America/Grand_Turk	Atlantic Standard Time	-4.0
America/Grenada	Atlantic Standard Time	-4.0
America/Guadeloupe	Atlantic Standard Time	-4.0
America/Guatemala	Central Standard Time	-6.0
America/Guayaquil	Ecuador Time	-5.0

America/Guyana	Guyana Time	-4.0
America/Halifax	Atlantic Standard Time	-4.0
America/Havana	Cuba Standard Time	-5.0
America/Hermosillo	Mountain Standard Time	-7.0
America/Indiana/Indianapolis	Eastern Standard Time	-5.0
America/Indiana/Knox	Central Standard Time	-6.0
America/Indiana/Marengo	Eastern Standard Time	-5.0
America/Indiana/Petersburg	Eastern Standard Time	-5.0
America/Indiana/Tell_City	Central Standard Time	-6.0
America/Indiana/Vevay	Eastern Standard Time	-5.0
America/Indiana/Vincennes	Eastern Standard Time	-5.0
America/Indiana/Winamac	Eastern Standard Time	-5.0
America/Indianapolis	Eastern Standard Time	-5.0
America/Inuvik	Mountain Standard Time	-7.0
America/Iqaluit	Eastern Standard Time	-5.0
America/Jamaica	Eastern Standard Time	-5.0
America/Jujuy	Argentine Time	-3.0
America/Juneau	Alaska Standard Time	-9.0
America/Kentucky/Louisville	Eastern Standard Time	-5.0
America/Kentucky/Monticello	Eastern Standard Time	-5.0
America/Knox_IN	Central Standard Time	-6.0
America/Kralendijk	Atlantic Standard Time	-4.0
America/La_Paz	Bolivia Time	-4.0
America/Lima	Peru Time	-5.0
America/Los_Angeles	Pacific Standard Time	-8.0

America/Louisville	Eastern Standard Time	-5.0
America/Lower_Princes	Atlantic Standard Time	-4.0
America/Maceio	Brasilia Time	-3.0
America/Managua	Central Standard Time	-6.0
America/Manaus	Amazon Time	-4.0
America/Marigot	Atlantic Standard Time	-4.0
America/Martinique	Atlantic Standard Time	-4.0
America/Matamoros	Central Standard Time	-6.0
America/Mazatlan	Mountain Standard Time	-7.0
America/Mendoza	Argentine Time	-3.0
America/Menominee	Central Standard Time	-6.0
America/Merida	Central Standard Time	-6.0
America/Metlakatla	Alaska Standard Time	-9.0
America/Mexico_City	Central Standard Time	-6.0
America/Miquelon	Pierre & Miquelon Standard Time	-3.0
America/Moncton	Atlantic Standard Time	-4.0
America/Monterrey	Central Standard Time	-6.0
America/Montevideo	Uruguay Time	-3.0
America/Montreal	Eastern Standard Time	-5.0
America/Montserrat	Atlantic Standard Time	-4.0
America/Nassau	Eastern Standard Time	-5.0
America/New_York	Eastern Standard Time	-5.0
America/Nipigon	Eastern Standard Time	-5.0
America/Nome	Alaska Standard Time	-9.0
America/Noronha	Fernando de Noronha Time	-2.0

America/North_Dakota/Beulah	Central Standard Time	-6.0
America/North_Dakota/Center	Central Standard Time	-6.0
America/North_Dakota/New_Salem	Central Standard Time	-6.0
America/Ojinaga	Mountain Standard Time	-7.0
America/Panama	Eastern Standard Time	-5.0
America/Pangnirtung	Eastern Standard Time	-5.0
America/Paramaribo	Suriname Time	-3.0
America/Phoenix	Mountain Standard Time	-7.0
America/Port-au-Prince	Eastern Standard Time	-5.0
America/Port_of_Spain	Atlantic Standard Time	-4.0
America/Porto_Acre	Acre Time	-5.0
America/Porto_Velho	Amazon Time	-4.0
America/Puerto_Rico	Atlantic Standard Time	-4.0
America/Punta_Arenas	GMT-03:00	-3.0
America/Rainy_River	Central Standard Time	-6.0
America/Rankin_Inlet	Central Standard Time	-6.0
America/Recife	Brasilia Time	-3.0
America/Regina	Central Standard Time	-6.0
America/Resolute	Central Standard Time	-6.0
America/Rio_Branco	Acre Time	-5.0
America/Rosario	Argentine Time	-3.0
America/Santa_Isabel	Pacific Standard Time	-8.0
America/Santarem	Brasilia Time	-3.0
America/Santiago	Chile Time	-4.0
America/Santo_Domingo	Atlantic Standard Time	-4.0

America/Sao_Paulo	Brasilia Time	-3.0
America/Scoresbysund	Eastern Greenland Time	-1.0
America/Shiprock	Mountain Standard Time	-7.0
America/Sitka	Alaska Standard Time	-9.0
America/St_Barthelemy	Atlantic Standard Time	-4.0
America/St_Johns	Newfoundland Standard Time	-3.5
America/St_Kitts	Atlantic Standard Time	-4.0
America/St_Lucia	Atlantic Standard Time	-4.0
America/St_Thomas	Atlantic Standard Time	-4.0
America/St_Vincent	Atlantic Standard Time	-4.0
America/Swift_Current	Central Standard Time	-6.0
America/Tegucigalpa	Central Standard Time	-6.0
America/Thule	Atlantic Standard Time	-4.0
America/Thunder_Bay	Eastern Standard Time	-5.0
America/Tijuana	Pacific Standard Time	-8.0
America/Toronto	Eastern Standard Time	-5.0
America/Tortola	Atlantic Standard Time	-4.0
America/Vancouver	Pacific Standard Time	-8.0
America/Virgin	Atlantic Standard Time	-4.0
America/Whitehorse	Pacific Standard Time	-8.0
America/Winnipeg	Central Standard Time	-6.0
America/Yakutat	Alaska Standard Time	-9.0
America/Yellowknife	Mountain Standard Time	-7.0
Antarctica/Casey	Australian Western Standard Time	11.0
Antarctica/Davis	Davis Time	7.0

Antarctica/DumontDURville	Dumont-d'Urville Time	10.0
Antarctica/Macquarie	Macquarie Island Standard Time	11.0
Antarctica/Mawson	Mawson Time	5.0
Antarctica/McMurdo	New Zealand Standard Time	12.0
Antarctica/Palmer	Chile Time	-3.0
Antarctica/Rothera	Rothera Time	-3.0
Antarctica/South_Pole	New Zealand Standard Time	12.0
Antarctica/Syowa	Syowa Time	3.0
Antarctica/Troll	Coordinated Universal Time	0.0
Antarctica/Vostok	Vostok Time	6.0
Arctic/Longyearbyen	Central European Time	1.0
Asia/Aden	Arabia Standard Time	3.0
Asia/Almaty	Alma-Ata Time	6.0
Asia/Amman	Eastern European Time	2.0
Asia/Anadyr	Anadyr Time	12.0
Asia/Aqtau	Aqtau Time	5.0
Asia/Aqtobe	Aqtobe Time	5.0
Asia/Ashgabat	Turkmenistan Time	5.0
Asia/Ashkhabad	Turkmenistan Time	5.0
Asia/Atyrau	GMT+05:00	5.0
Asia/Baghdad	Arabia Standard Time	3.0
Asia/Bahrain	Arabia Standard Time	3.0
Asia/Baku	Azerbaijan Time	4.0
Asia/Bangkok	Indochina Time	7.0
Asia/Barnaul	GMT+07:00	7.0

Asia/Beirut	Eastern European Time	2.0
Asia/Bishkek	Kirgizstan Time	6.0
Asia/Brunei	Brunei Time	8.0
Asia/Calcutta	India Standard Time	5.5
Asia/Chita	Yakutsk Time	9.0
Asia/Choibalsan	Choibalsan Time	8.0
Asia/Chongqing	China Standard Time	8.0
Asia/Chungking	China Standard Time	8.0
Asia/Colombo	India Standard Time	5.5
Asia/Dacca	Bangladesh Time	6.0
Asia/Damascus	Eastern European Time	2.0
Asia/Dhaka	Bangladesh Time	6.0
Asia/Dili	Timor-Leste Time	9.0
Asia/Dubai	Gulf Standard Time	4.0
Asia/Dushanbe	Tajikistan Time	5.0
Asia/Famagusta	GMT+03:00	3.0
Asia/Gaza	Eastern European Time	2.0
Asia/Harbin	China Standard Time	8.0
Asia/Hebron	Eastern European Time	2.0
Asia/Ho_Chi_Minh	Indochina Time	7.0
Asia/Hong_Kong	Hong Kong Time	8.0
Asia/Hovd	Hovd Time	7.0
Asia/Irkutsk	Irkutsk Time	8.0
Asia/Istanbul	Eastern European Time	3.0
Asia/Jakarta	West Indonesia Time	7.0

Asia/Jayapura	East Indonesia Time	9.0
Asia/Jerusalem	Israel Standard Time	2.0
Asia/Kabul	Afghanistan Time	4.5
Asia/Kamchatka	Petropavlovsk-Kamchatski Time	12.0
Asia/Karachi	Pakistan Time	5.0
Asia/Kashgar	Xinjiang Standard Time	6.0
Asia/Kathmandu	Nepal Time	5.75
Asia/Katmandu	Nepal Time	5.75
Asia/Khandyga	Yakutsk Time	9.0
Asia/Kolkata	India Standard Time	5.5
Asia/Krasnoyarsk	Krasnoyarsk Time	7.0
Asia/Kuala_Lumpur	Malaysia Time	8.0
Asia/Kuching	Malaysia Time	8.0
Asia/Kuwait	Arabia Standard Time	3.0
Asia/Macao	China Standard Time	8.0
Asia/Macau	China Standard Time	8.0
Asia/Magadan	Magadan Time	11.0
Asia/Makassar	Central Indonesia Time	8.0
Asia/Manila	Philippines Time	8.0
Asia/Muscat	Gulf Standard Time	4.0
Asia/Nicosia	Eastern European Time	2.0
Asia/Novokuznetsk	Krasnoyarsk Time	7.0
Asia/Novosibirsk	Novosibirsk Time	7.0
Asia/Omsk	Omsk Time	6.0
Asia/Oral	Oral Time	5.0

Asia/Phnom_Penh	Indochina Time	7.0
Asia/Pontianak	West Indonesia Time	7.0
Asia/Pyongyang	Korea Standard Time	8.5
Asia/Qatar	Arabia Standard Time	3.0
Asia/Qyzylorda	Qyzylorda Time	6.0
Asia/Rangoon	Myanmar Time	6.5
Asia/Riyadh	Arabia Standard Time	3.0
Asia/Saigon	Indochina Time	7.0
Asia/Sakhalin	Sakhalin Time	11.0
Asia/Samarkand	Uzbekistan Time	5.0
Asia/Seoul	Korea Standard Time	9.0
Asia/Shanghai	China Standard Time	8.0
Asia/Singapore	Singapore Time	8.0
Asia/Srednekolymsk	Srednekolymsk Time	11.0
Asia/Taipei	China Standard Time	8.0
Asia/Tashkent	Uzbekistan Time	5.0
Asia/Tbilisi	Georgia Time	4.0
Asia/Tehran	Iran Standard Time	3.5
Asia/Tel_Aviv	Israel Standard Time	2.0
Asia/Thimbu	Bhutan Time	6.0
Asia/Thimphu	Bhutan Time	6.0
Asia/Tokyo	Japan Standard Time	9.0
Asia/Tomsk	GMT+07:00	7.0
Asia/Ujung_Pandang	Central Indonesia Time	8.0
Asia/Ulaanbaatar	Ulaanbaatar Time	8.0

Asia/Ulan_Bator	Ulaanbaatar Time	8.0
Asia/Urumqi	Xinjiang Standard Time	6.0
Asia/Ust-Nera	Ust-Nera Time	10.0
Asia/Vientiane	Indochina Time	7.0
Asia/Vladivostok	Vladivostok Time	10.0
Asia/Yakutsk	Yakutsk Time	9.0
Asia/Yangon	Myanmar Time	6.5
Asia/Yekaterinburg	Yekaterinburg Time	5.0
Asia/Yerevan	Armenia Time	4.0
Atlantic/Azores	Azores Time	-1.0
Atlantic/Bermuda	Atlantic Standard Time	-4.0
Atlantic/Canary	Western European Time	0.0
Atlantic/Cape_Verde	Cape Verde Time	-1.0
Atlantic/Faeroe	Western European Time	0.0
Atlantic/Faroe	Western European Time	0.0
Atlantic/Jan_Mayen	Central European Time	1.0
Atlantic/Madeira	Western European Time	0.0
Atlantic/Reykjavik	Greenwich Mean Time	0.0
Atlantic/South_Georgia	South Georgia Standard Time	-2.0
Atlantic/St_Helena	Greenwich Mean Time	0.0
Atlantic/Stanley	Falkland Is. Time	-3.0
Australia/ACT	Australian Eastern Standard Time (New South Wales)	10.0
Australia/Adelaide	Australian Central Standard Time (South Australia)	9.5
Australia/Brisbane	Australian Eastern Standard Time (Queensland)	10.0
Australia/Broken_Hill	Australian Central Standard Time (South Australia/New South Wales)	9.5

Australia/Canberra	Australian Eastern Standard Time (New South Wales)	10.0
Australia/Currie	Australian Eastern Standard Time (New South Wales)	10.0
Australia/Darwin	Australian Central Standard Time (Northern Territory)	9.5
Australia/Eucla	Australian Central Western Standard Time	8.75
Australia/Hobart	Australian Eastern Standard Time (Tasmania)	10.0
Australia/LHI	Lord Howe Standard Time	10.5
Australia/Lindeman	Australian Eastern Standard Time (Queensland)	10.0
Australia/Lord_Howe	Lord Howe Standard Time	10.5
Australia/Melbourne	Australian Eastern Standard Time (Victoria)	10.0
Australia/NSW	Australian Eastern Standard Time (New South Wales)	10.0
Australia/North	Australian Central Standard Time (Northern Territory)	9.5
Australia/Perth	Australian Western Standard Time	8.0
Australia/Queensland	Australian Eastern Standard Time (Queensland)	10.0
Australia/South	Australian Central Standard Time (South Australia)	9.5
Australia/Sydney	Australian Eastern Standard Time (New South Wales)	10.0
Australia/Tasmania	Australian Eastern Standard Time (Tasmania)	10.0
Australia/Victoria	Australian Eastern Standard Time (Victoria)	10.0
Australia/West	Australian Western Standard Time	8.0
Australia/Yancowinna	Australian Central Standard Time (South Australia/New South Wales)	9.5
Brazil/Acre	Acre Time	-5.0
Brazil/DeNoronha	Fernando de Noronha Time	-2.0
Brazil/East	Brasilia Time	-3.0
Brazil/West	Amazon Time	-4.0
CET	Central European Time	1.0

CST6CDT	Central Standard Time	-6.0
Canada/Atlantic	Atlantic Standard Time	-4.0
Canada/Central	Central Standard Time	-6.0
Canada/East-Saskatchewan	Central Standard Time	-6.0
Canada/Eastern	Eastern Standard Time	-5.0
Canada/Mountain	Mountain Standard Time	-7.0
Canada/Newfoundland	Newfoundland Standard Time	-3.5
Canada/Pacific	Pacific Standard Time	-8.0
Canada/Saskatchewan	Central Standard Time	-6.0
Canada/Yukon	Pacific Standard Time	-8.0
Chile/Continental	Chile Time	-4.0
Chile/EasterIsland	Easter Is. Time	-6.0
Cuba	Cuba Standard Time	-5.0
EET	Eastern European Time	2.0
EST5EDT	Eastern Standard Time	-5.0
Egypt	Eastern European Time	2.0
Eire	Greenwich Mean Time	0.0
Etc/GMT	Greenwich Mean Time	0.0
Etc/GMT+0	Greenwich Mean Time	0.0
Etc/GMT+1	GMT-01:00	-1.0
Etc/GMT+10	GMT-10:00	-10.0
Etc/GMT+11	GMT-11:00	-11.0
Etc/GMT+12	GMT-12:00	-12.0
Etc/GMT+2	GMT-02:00	-2.0
Etc/GMT+3	GMT-03:00	-3.0

Etc/GMT+4	GMT-04:00	-4.0
Etc/GMT+5	GMT-05:00	-5.0
Etc/GMT+6	GMT-06:00	-6.0
Etc/GMT+7	GMT-07:00	-7.0
Etc/GMT+8	GMT-08:00	-8.0
Etc/GMT+9	GMT-09:00	-9.0
Etc/GMT-0	Greenwich Mean Time	0.0
Etc/GMT-1	GMT+01:00	1.0
Etc/GMT-10	GMT+10:00	10.0
Etc/GMT-11	GMT+11:00	11.0
Etc/GMT-12	GMT+12:00	12.0
Etc/GMT-13	GMT+13:00	13.0
Etc/GMT-14	GMT+14:00	14.0
Etc/GMT-2	GMT+02:00	2.0
Etc/GMT-3	GMT+03:00	3.0
Etc/GMT-4	GMT+04:00	4.0
Etc/GMT-5	GMT+05:00	5.0
Etc/GMT-6	GMT+06:00	6.0
Etc/GMT-7	GMT+07:00	7.0
Etc/GMT-8	GMT+08:00	8.0
Etc/GMT-9	GMT+09:00	9.0
Etc/GMT0	Greenwich Mean Time	0.0
Etc/Greenwich	Greenwich Mean Time	0.0
Etc/UCT	Coordinated Universal Time	0.0
Etc/UTC	Coordinated Universal Time	0.0

Etc/Universal	Coordinated Universal Time	0.0
Etc/Zulu	Coordinated Universal Time	0.0
Europe/Amsterdam	Central European Time	1.0
Europe/Andorra	Central European Time	1.0
Europe/Astrakhan	GMT+04:00	4.0
Europe/Athens	Eastern European Time	2.0
Europe/Belfast	Greenwich Mean Time	0.0
Europe/Belgrade	Central European Time	1.0
Europe/Berlin	Central European Time	1.0
Europe/Bratislava	Central European Time	1.0
Europe/Brussels	Central European Time	1.0
Europe/Bucharest	Eastern European Time	2.0
Europe/Budapest	Central European Time	1.0
Europe/Busingen	Central European Time	1.0
Europe/Chisinau	Eastern European Time	2.0
Europe/Copenhagen	Central European Time	1.0
Europe/Dublin	Greenwich Mean Time	0.0
Europe/Gibraltar	Central European Time	1.0
Europe/Guernsey	Greenwich Mean Time	0.0
Europe/Helsinki	Eastern European Time	2.0
Europe/Isle_of_Man	Greenwich Mean Time	0.0
Europe/Istanbul	Eastern European Time	3.0
Europe/Jersey	Greenwich Mean Time	0.0
Europe/Kaliningrad	Eastern European Time	2.0
Europe/Kiev	Eastern European Time	2.0

Europe/Kirov	GMT+03:00	3.0
Europe/Lisbon	Western European Time	0.0
Europe/Ljubljana	Central European Time	1.0
Europe/London	Greenwich Mean Time	0.0
Europe/Luxembourg	Central European Time	1.0
Europe/Madrid	Central European Time	1.0
Europe/Malta	Central European Time	1.0
Europe/Mariehamn	Eastern European Time	2.0
Europe/Minsk	Moscow Standard Time	3.0
Europe/Monaco	Central European Time	1.0
Europe/Moscow	Moscow Standard Time	3.0
Europe/Nicosia	Eastern European Time	2.0
Europe/Oslo	Central European Time	1.0
Europe/Paris	Central European Time	1.0
Europe/Podgorica	Central European Time	1.0
Europe/Prague	Central European Time	1.0
Europe/Riga	Eastern European Time	2.0
Europe/Rome	Central European Time	1.0
Europe/Samara	Samara Time	4.0
Europe/San_Marino	Central European Time	1.0
Europe/Sarajevo	Central European Time	1.0
Europe/Saratov	GMT+04:00	4.0
Europe/Simferopol	Moscow Standard Time	3.0
Europe/Skopje	Central European Time	1.0
Europe/Sofia	Eastern European Time	2.0

Europe/Stockholm	Central European Time	1.0
Europe/Tallinn	Eastern European Time	2.0
Europe/Tirane	Central European Time	1.0
Europe/Tiraspol	Eastern European Time	2.0
Europe/Ulyanovsk	GMT+04:00	4.0
Europe/Uzhgorod	Eastern European Time	2.0
Europe/Vaduz	Central European Time	1.0
Europe/Vatican	Central European Time	1.0
Europe/Vienna	Central European Time	1.0
Europe/Vilnius	Eastern European Time	2.0
Europe/Volgograd	Moscow Standard Time	3.0
Europe/Warsaw	Central European Time	1.0
Europe/Zagreb	Central European Time	1.0
Europe/Zaporozhye	Eastern European Time	2.0
Europe/Zurich	Central European Time	1.0
GB	Greenwich Mean Time	0.0
GB-Eire	Greenwich Mean Time	0.0
GMT	Greenwich Mean Time	0.0
GMT0	Greenwich Mean Time	0.0
Greenwich	Greenwich Mean Time	0.0
Hongkong	Hong Kong Time	8.0
Iceland	Greenwich Mean Time	0.0
Indian/Antananarivo	Eastern African Time	3.0
Indian/Chagos	Indian Ocean Territory Time	6.0
Indian/Christmas	Christmas Island Time	7.0

Indian/Cocos	Cocos Islands Time	6.5
Indian/Comoro	Eastern African Time	3.0
Indian/Kerguelen	French Southern & Antarctic Lands Time	5.0
Indian/Mahe	Seychelles Time	4.0
Indian/Maldives	Maldives Time	5.0
Indian/Mauritius	Mauritius Time	4.0
Indian/Mayotte	Eastern African Time	3.0
Indian/Reunion	Reunion Time	4.0
Iran	Iran Standard Time	3.5
Israel	Israel Standard Time	2.0
Jamaica	Eastern Standard Time	-5.0
Japan	Japan Standard Time	9.0
Kwajalein	Marshall Islands Time	12.0
Libya	Eastern European Time	2.0
MET	Middle Europe Time	1.0
MST7MDT	Mountain Standard Time	-7.0
Mexico/BajaNorte	Pacific Standard Time	-8.0
Mexico/BajaSur	Mountain Standard Time	-7.0
Mexico/General	Central Standard Time	-6.0
NZ	New Zealand Standard Time	12.0
NZ-CHAT	Chatham Standard Time	12.75
Navajo	Mountain Standard Time	-7.0
PRC	China Standard Time	8.0
PST8PDT	Pacific Standard Time	-8.0
Pacific/Apia	West Samoa Standard Time	13.0

Pacific/Auckland	New Zealand Standard Time	12.0
Pacific/Bougainville	Bougainville Standard Time	11.0
Pacific/Chatham	Chatham Standard Time	12.75
Pacific/Chuuk	Chuuk Time	10.0
Pacific/Easter	Easter Is. Time	-6.0
Pacific/Efate	Vanuatu Time	11.0
Pacific/Enderbury	Phoenix Is. Time	13.0
Pacific/Fakaofu	Tokelau Time	13.0
Pacific/Fiji	Fiji Time	12.0
Pacific/Funafuti	Tuvalu Time	12.0
Pacific/Galapagos	Galapagos Time	-6.0
Pacific/Gambier	Gambier Time	-9.0
Pacific/Guadalcanal	Solomon Is. Time	11.0
Pacific/Guam	Chamorro Standard Time	10.0
Pacific/Honolulu	Hawaii Standard Time	-10.0
Pacific/Johnston	Hawaii Standard Time	-10.0
Pacific/Kiritimati	Line Is. Time	14.0
Pacific/Kosrae	Kosrae Time	11.0
Pacific/Kwajalein	Marshall Islands Time	12.0
Pacific/Majuro	Marshall Islands Time	12.0
Pacific/Marquesas	Marquesas Time	-9.5
Pacific/Midway	Samoa Standard Time	-11.0
Pacific/Nauru	Nauru Time	12.0
Pacific/Niue	Niue Time	-11.0
Pacific/Norfolk	Norfolk Time	11.0

Pacific/Noumea	New Caledonia Time	11.0
Pacific/Pago_Pago	Samoa Standard Time	-11.0
Pacific/Palau	Palau Time	9.0
Pacific/Pitcairn	Pitcairn Standard Time	-8.0
Pacific/Pohnpei	Pohnpei Time	11.0
Pacific/Ponape	Pohnpei Time	11.0
Pacific/Port_Moresby	Papua New Guinea Time	10.0
Pacific/Rarotonga	Cook Is. Time	-10.0
Pacific/Saipan	Chamorro Standard Time	10.0
Pacific/Samoa	Samoa Standard Time	-11.0
Pacific/Tahiti	Tahiti Time	-10.0
Pacific/Tarawa	Gilbert Is. Time	12.0
Pacific/Tongatapu	Tonga Time	13.0
Pacific/Truk	Chuuk Time	10.0
Pacific/Wake	Wake Time	12.0
Pacific/Wallis	Wallis & Futuna Time	12.0
Pacific/Yap	Chuuk Time	10.0
Poland	Central European Time	1.0
Portugal	Western European Time	0.0
ROK	Korea Standard Time	9.0
Singapore	Singapore Time	8.0
SystemV/AST4	Atlantic Standard Time	-4.0
SystemV/AST4ADT	Atlantic Standard Time	-4.0
SystemV/CST6	Central Standard Time	-6.0
SystemV/CST6CDT	Central Standard Time	-6.0

SystemV/EST5	Eastern Standard Time	-5.0
SystemV/EST5EDT	Eastern Standard Time	-5.0
SystemV/HST10	Hawaii Standard Time	-10.0
SystemV/MST7	Mountain Standard Time	-7.0
SystemV/MST7MDT	Mountain Standard Time	-7.0
SystemV/PST8	Pacific Standard Time	-8.0
SystemV/PST8PDT	Pacific Standard Time	-8.0
SystemV/YST9	Alaska Standard Time	-9.0
SystemV/YST9YDT	Alaska Standard Time	-9.0
Turkey	Eastern European Time	3.0
UCT	Coordinated Universal Time	0.0
US/Alaska	Alaska Standard Time	-9.0
US/Aleutian	Hawaii Standard Time	-10.0
US/Arizona	Mountain Standard Time	-7.0
US/Central	Central Standard Time	-6.0
US/East-Indiana	Eastern Standard Time	-5.0
US/Eastern	Eastern Standard Time	-5.0
US/Hawaii	Hawaii Standard Time	-10.0
US/Indiana-Starke	Central Standard Time	-6.0
US/Michigan	Eastern Standard Time	-5.0
US/Mountain	Mountain Standard Time	-7.0
US/Pacific	Pacific Standard Time	-8.0
US/Pacific-New	Pacific Standard Time	-8.0
US/Samoa	Samoa Standard Time	-11.0
UTC	Coordinated Universal Time	0.0

Universal	Coordinated Universal Time	0.0
W-SU	Moscow Standard Time	3.0
WET	Western European Time	0.0
Zulu	Coordinated Universal Time	0.0

JDBC tracing

In some cases, custom JDBC drivers work incorrectly in DBeaver - they show the wrong metadata like table columns, constraints or foreign keys.

It usually happens because the driver is not compliant with the JDBC API specification and DBeaver cannot correctly interpret the metadata provided by the driver.

To understand what is going on inside the driver, you can enable JDBC tracing:

1. Find `dbeaver.ini` file.

Tip: Detailed instructions on finding `dbeaver.ini` are available in [our article](#).

2. Add line `-Ddbeaver.jdbc.trace=true` in the end of `dbeaver.ini`.
3. Restart DBeaver.
4. Connect to your database and browse the metadata in the database navigator/object editors.
5. In DBeaver [Workspace](#) go to `.metadata` folder.
6. File `jdbc-api-trace.log` contains all JDBC API invocations and all queries with results.

Analyzing contents of `jdbc-api-trace.log` you can understand what is wrong with the metadata. Attach the piece of the trace file in the GitHub ticket if you think that something is wrong on DBeaver's side.

WARNING: disable JDBC tracing in your regular work. Enable it only for debugging. The trace generation decreases application performance and may produce huge log files.

Apache Hive

Table of contents

[Apache Hive](#)

[Connection setup](#)

[Schema and data browser](#)

[Limitations](#)

Apache Hive

Hive is a Hadoop-based storage system. Hive uses a special SQL dialect (HiveQL) to operate with data and metadata. Generally, it is quite similar to SQL.

There are multiple implementations of storage systems which utilize Hive on the server-side - including Apache Spark, Impala, etc. Most of them support the standard Hive JDBC driver which is used in DBeaver to communicate with the server.

DBeaver uses a so-called Hive JDBC Uber Jar driver (<https://github.com/timveil/hive-jdbc-uber-jar>) which includes all necessary dependencies. You do not need to download anything - DBeaver will download everything automatically (if you have internet access).

Connection setup

Create new connection

Select new connection type

Apache Hive JDBC

hive

Name	#
▼ Hadoop	6
Apache Hive	1
Spark Hive	1

Project

General

< Back

Next >

Finish

Cancel

Test Connection ...

Create new connection

Generic JDBC Connection Settings

Hadoop / Apache Hive connection settings

General

Driver properties

JDBC URL:

jdbc:hive2://hive.theserver.com:10000/foodmart

Host:

hive.theserver.com

Port:

10000

Database/Schema:

foodmart

User name:

Password:

☐ Save password locally

[Network settings \(SSH, SSL, Proxy, ...\)](#)

[Connection details \(name, type, ...\)](#)

Driver Name: Hadoop / Apache Hive

Edit Driver Settings

?

< Back

Next >

Finish

Cancel

Test Connection ...

Success

i

Server: Apache Hive 1.2.1000.2.6.5.0-292

Driver: org.apache.hive.jdbc.HiveDriver -1.-1

Connected (2349 ms)

OK

Schema and data browser

FancyName - foodmart.customer

File Edit Navigate Search SQL Editor Run Database Window Help

Auto Hive - default default 200

Database Navig Projects

Enter a part of table name here

Hive - default

default

foodmart

Tables

account

agg_c_10_sales_fact_1997

agg_c_14_sales_fact_1997

agg_c_special_sales_fact_1997

agg_g_ms_pcat_sales_fact_1997

agg_l_03_sales_fact_1997

agg_l_04_sales_fact_1997

agg_l_05_sales_fact_1997

agg_lc_06_sales_fact_1997

agg_lc_100_sales_fact_1997

agg_ll_01_sales_fact_1997

agg_pl_01_sales_fact_1997

category

currency

customer

days

department

employee

employee_closure

expense_fact

inventory_fact_1997

inventory_fact_1998

position

product

product_class

promotion

region

reserve_employee

salary

sales_fact_1997

Properties Data ER Diagram

customer Enter a SQL expression to filter results (use Ctrl+Space)

	customer_id	account_num	ABC lname	ABC fname	ABC mi	ABC address1	ABC address2	ABC address3
1	1	87,462,024,688	Nowmer	Sheri	A.	2433 Bailey Road		
2	2	87,470,586,299	Whelpy	Derrick	I.	2219 Dewing Avenue		
3	3	87,475,757,600	Derry	Jeanne		7640 First Ave.		
4	4	87,500,482,201	Spence	Michael	J.	337 Tosca Way		
5	5	87,514,054,179	Gutierrez	Maya		8668 Via Neruda		
6	6	87,517,782,449	Damstra	Robert	F.	1619 Stillman Court		
7	7	87,521,172,800	Kanagaki	Rebecca		2860 D Mt. Hood Circle		
8	8	87,539,744,377	Brunner	Kim	H.	6064 Brodia Court		
9	9	87,544,797,658	Blumberg	Brenda	C.	7560 Trees Drive		
10	10	87,568,712,234	Stanz	Darren	M.	1019 Kenwal Rd.		
11	11	87,572,821,378	Murraiin	Jonathan	V.	5423 Camby Rd.		
12	12	87,579,237,222	Creek	Jewel	C.	1792 Belmont Rd.		
13	13	87,587,122,917	Medina	Peggy	A.	3796 Keller Ridge		
14	14	87,592,626,810	Rutledge	Bryan	K.	3074 Ardith Drive		
15	15	87,597,749,829	Cavestany	Walter	G.	7987 Seawind Dr.		
16	16	87,603,285,908	Planck	Peggy	M.	4864 San Carlos		
17	17	87,625,473,141	Marshall	Brenda	S.	2687 Ridge Road		
18	18	87,637,655,735	Wolter	Daniel	P.	2473 Orchard Way		
19	19	87,650,814,652	Collins	Dianne		551 Rainier Dr		
20	20	87,653,979,700	Baker	Beverly		591 Merriewood Drive		
21	21	87,663,244,009	Castillo	Pedro		1579 Plaza Rosa		
22	22	87,675,641,200	Borges	Laurie		1873 Lyon Circle		
23	23	87,678,398,489	Wyro	Shauna	K.	3114 Notre Dame Ave.		
24	24	87,681,713,700	Wyllie	Jacqueline		6318 Marclair Dr.		
25	25	87,686,740,159	Conlev	Lin	N.	7814 Milburn Dr.		

Save Cancel Script

200 row(s) fetched - 332ms (+583ms) [Hive - default]

MSK en 96M of 518M Load row count

foodmart.account foodmart.customer <Hive - default> Script-108

Properties Data ER Diagram

Hive - default foodmart Tables account

Name	Value
Table Name	account
Table Type	TABLE
Catalog	
Schema	foodmart
Table Description	

Columns Indexes DDL

```

CREATE EXTERNAL TABLE `foodmart.account` (
  `account_id` int,
  `account_parent` int,
  `account_description` varchar(30),
  `account_type` varchar(30),
  `account_rollup` varchar(30),
  `custom_members` varchar(255))
ROW FORMAT SERDE
  'org.apache.hadoop.hive.q1.io.orc.OrcSerde'
STORED AS INPUTFORMAT
  'org.apache.hadoop.hive.q1.io.orc.OrcInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.q1.io.orc.OrcOutputFormat'
LOCATION
  'hdfs://sandbox-hdp.hortonworks.com:8020/apps/hive/warehouse/foodmart.db/account'
TBLPROPERTIES (
  'COLUMN_STATS_ACCURATE'='{\"COLUMN_STATS\":{\"account_id\":\"true\",\"account_parent\":\"true\",
  \"account_description\":\"true\",\"account_type\":\"true\",\"account_rollup\":\"true\",
  \"custom_members\":\"true\"}}',
  'numFiles'='1',
  'orc.compress'='SNAPPY',
  'totalSize'='1444',
  'transient_lastDdlTime'='1529335182')

```

Limitations

Hive does not support referential integrity so you will not see primary keys or foreign keys. ER diagrams also do not make much sense.

Cassandra

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Connecting to Cassandra cluster](#)

[Browsing Cassandra tables](#)

[Executing CQL](#)

[ERD](#)

Overview

DBeaver EE supports Cassandra schema browser, data viewer and CQL queries execution. It also supports various administrative tools.


Connecting to Cassandra cluster

You can connect directly to a server or use SSH tunneling or SOCKS proxy. DBeaver uses the DataStax Java driver to operate with a server. It supports Cassandra servers 2.x, 3.x or higher.

Create new connection

Apache Cassandra Connection Settings

Cassandra CQL connection settings



GeneralCassandra TCP settings

Connection Setting

Host(s):localhost

Keyspace:systemPort:9042

User:

Password:

☐ Use SSL [SSL settings](#)

☒ Save password locally


Misc

Default consistency level: ONE

[Network settings \(SSH, SSL, Proxy, ...\)](#)
[Connection details \(name, type, ...\)](#)

Driver Name: Cassandra CQL

Edit Driver Settings



< Back

Next >

Finish

Cancel

Test Connection ...

Create new connection

Network

Configure networks handlers and tunnels

SSH Tunnel

SOCKS Proxy

SSL

☒ Use SSH Tunnel

Settings

Host/IP:

theserver.com

Port:

22

User Name:

theuser

Authentication Method:

Password

Password

.....

Save Password:

☒

Advanced

Implementation:

JSch

Local port:

0

Keep-Alive interval (ms):

0

Tunnel connect timeout (ms):

10000

Test tunnel configuration

?

< Back

Next >

Finish

Cancel

Test Connection ...

Create new connection

Network
Configure networks handlers and tunnels

SSH Tunnel

SOCKS Proxy

SSL

☒ Use SSL

You can choose SSL context configuration type:

Trust Store

Client Trust Store:

Trust Store password:

Trust store file must be in Java KeyStore format (jks)

Client Authentication

Client Certificate:

Client Key:

Use it if you have client certificate and private key

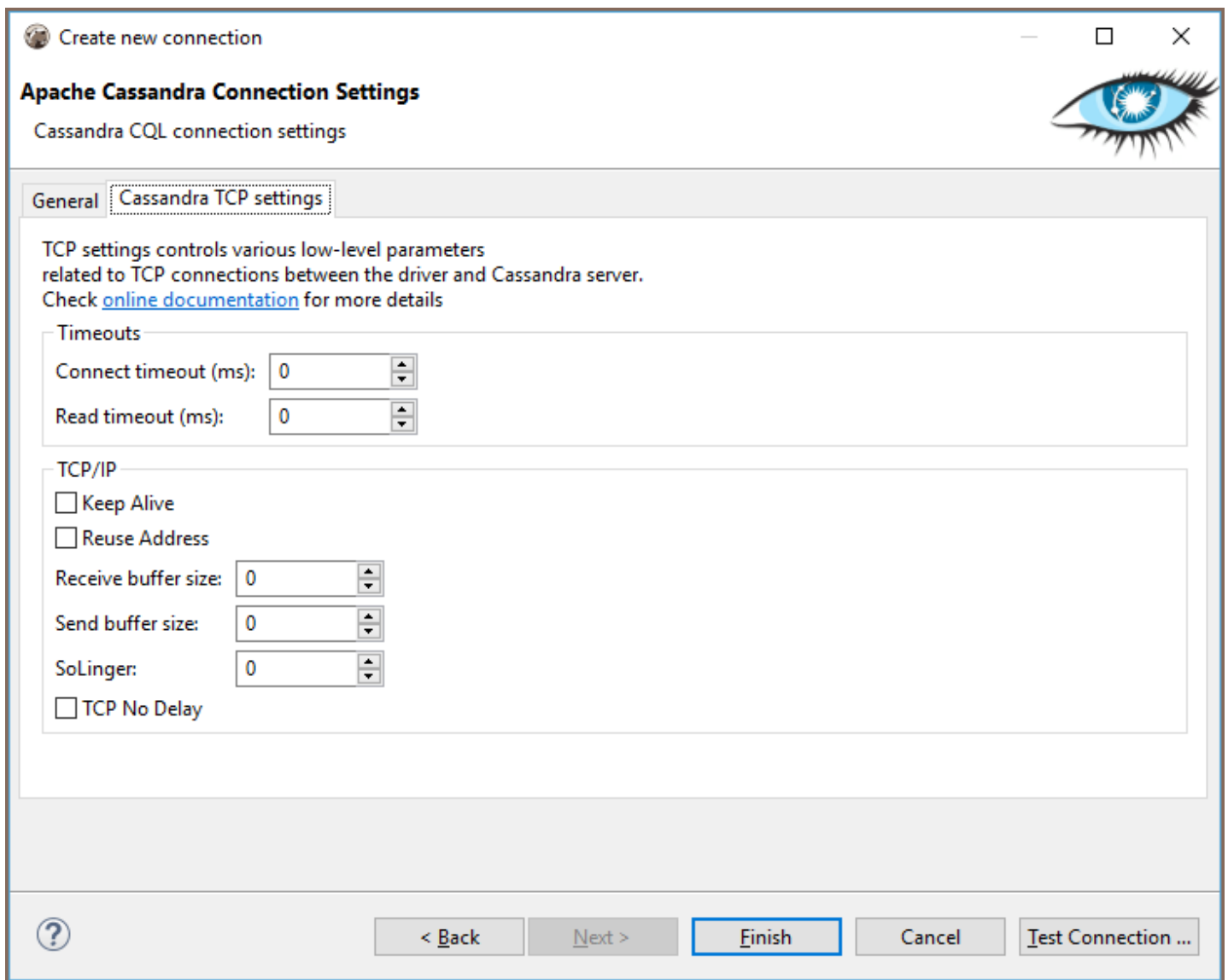
< Back

Next >

Finish

Cancel

Test Connection ...



Create new connection

Apache Cassandra Connection Settings

Cassandra CQL connection settings

General **Cassandra TCP settings**

TCP settings controls various low-level parameters related to TCP connections between the driver and Cassandra server. Check [online documentation](#) for more details

Timeouts

Connect timeout (ms): 0

Read timeout (ms): 0

TCP/IP

☐ Keep Alive

☐ Reuse Address

Receive buffer size: 0

Send buffer size: 0

SoLinger: 0

☐ TCP No Delay

? < Back Next > **Finish** Cancel Test Connection ...

Browsing Cassandra tables

You can browse, view, edit and filter Cassandra tables the same way as with regular (relational) tables. However, being a distributed key-value database, Cassandra does not support any kind of referential integrity. There are no foreign keys, references, etc.

You should note that Cassandra has a very advanced (comparing to relational databases) data type system. Each column may be a collection, map, or set of values (with very big number of values). In some cases this makes browsing data in the "Grid" mode inconvenient.

DBeeer Enterprise 5.2.1 - <Cassandra CQL - system> Script-19

File Edit Navigate Search SQL Editor Database Window Help

Auto Commit Rollback Auto

Cassandra CQL - system killrvideo 200 Quick Access

Database Navigator

Enter a part of table name here

- Keyspaces
 - killrvideo
 - Tables
 - comments_by_user
 - comments_by_video
 - encoding_job_notifications
 - latest_videos
 - tags_by_letter
 - uploaded_videos
 - uploaded_videos_by_jobid
 - user_credentials
 - user_videos
 - users
 - video_event
 - video_rating
 - video_ratings_by_user
 - videos
 - videos_by_tag
 - Materialized views
 - User types
 - Aggregates
 - Functions
 - system
 - system_auth
 - system_distributed
 - system_schema
 - system_traces
 - test
 - Tables
 - collect_things
 - driverperf_v1
 - job
 - list_tbl
 - messages
 - staged
 - Materialized views
 - User types

job

*<Cassandra CQL - system> Script-19

```

1 CREATE TABLE songs (
2   id uuid PRIMARY KEY,
3   title text,
4   album text,
5   artist text,
6   data blob
7 );
8
9 CREATE TABLE playlists (
10  id uuid,
11  song_order int,
12  song_id uuid,
13  title text,
14  album text,
15  artist text,
16  PRIMARY KEY (id, song_order) );
17 INSERT INTO playlists (id, song_order, song_id, title, artist, album)
18 VALUES ('00000000-0000-0000-0000-000000000000', 1, '00000000-0000-0000-0000-000000000000', 'The Gambia', 'The Gambia', 'The Gambia');

```

Result

SELECT * FROM test.job LIMIT 200 ALLOW FILTERING

	asc lot	asc gtn	asc
1	1683	Zambia	20.
2	1424	Vietnam	15.
3	1615	Uruguay	37.
4	631	Estonia	25.
5	403	Cabo Verde	0.6
6	492	Serbia	24.
7	517	The Gambia	0.5
8	1843	East Timor	0.9
9	789	The Gambia	0.1
10	947	Papua New Guinea	0.2
11	255	North Korea	0.2
12	326	Senegal	99.
13	374	Iran	8.1
14	151	Vietnam	2.2

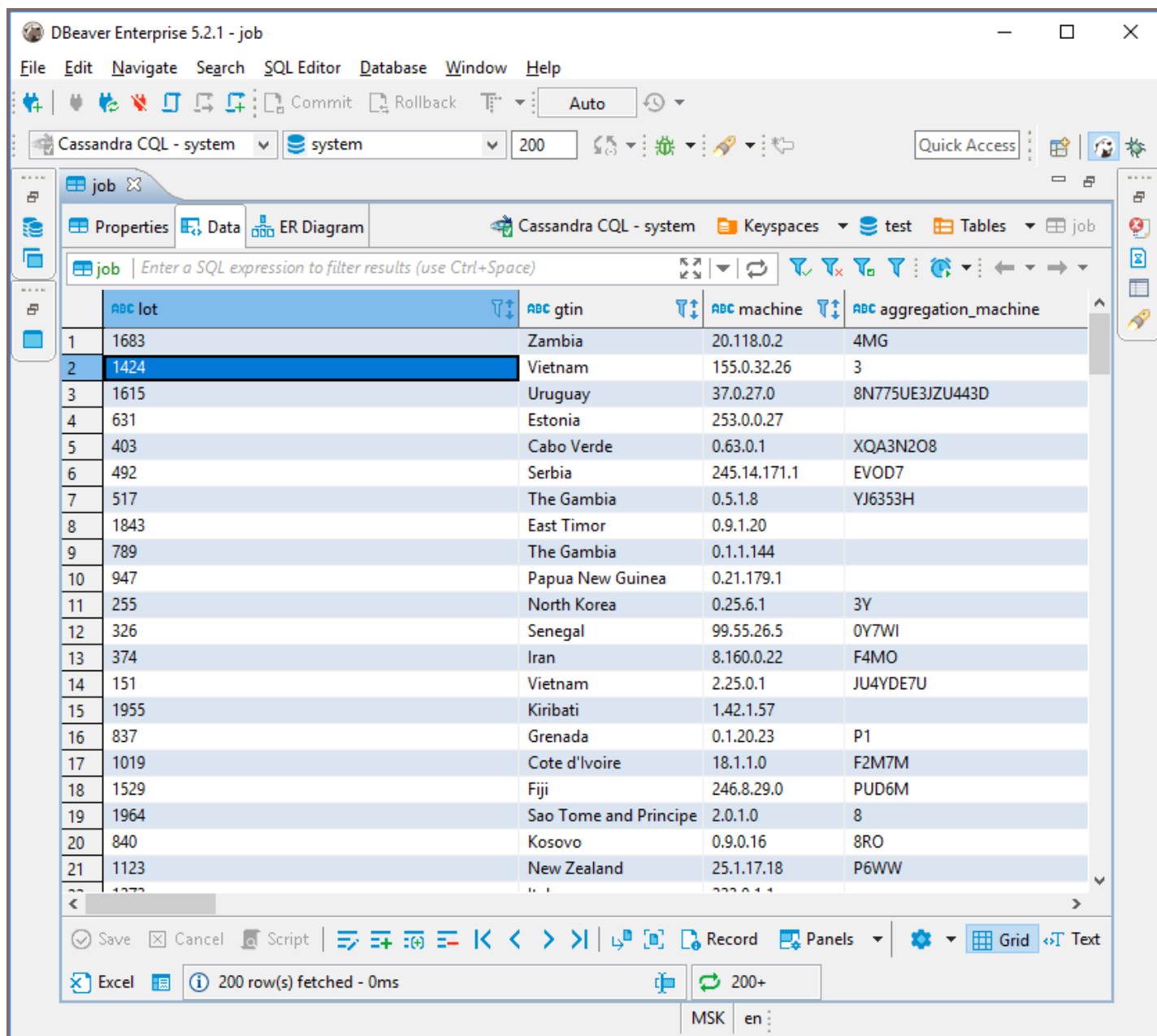
Value Grouping Query Trace

Event	Time	Source	Thread
Parsing SELECT * FROM test.job LIMIT 200 ALLOW FILTERING	137	/127.0.0.1	Native-Transport-Requests-1
Preparing statement	265	/127.0.0.1	Native-Transport-Requests-1
Computing ranges to query	453	/127.0.0.1	Native-Transport-Requests-1
Submitting range requests on 257 ranges with a concurrency ...	795	/127.0.0.1	Native-Transport-Requests-1
Submitted 1 concurrent range requests	1027	/127.0.0.1	Native-Transport-Requests-1
Executing seq scan across 1 sstables for (min(-9223372036854...	1844	/127.0.0.1	ReadStage-3
Read 200 live rows and 0 tombstone cells	3003	/127.0.0.1	ReadStage-3

Save Cancel Script

200 row(s) fetched - 124ms (+1ms)

MSK en Writable Smart Insert 11:18



Executing CQL

CQL [Cassandra Query Language](#) is a very simple SQL language dialect.

It supports simple SELECT queries, DDL statements (like CREATE TABLE) and some other query types.

You can use the standard DBeaver SQL editor to execute CQL queries. DBeaver supports Cassandra query execution, results scrolling, data export/import, mock data generation and other features. Data viewer (of individual tables or custom CQL query results) query tracing is supported.

The screenshot shows the DBeaver Enterprise 5.2.1 interface. The top menu bar includes File, Edit, Navigate, Search, SQL Editor, Database, Window, and Help. The toolbar contains various icons for file operations, SQL execution, and database management. The main window is titled "<Cassandra CQL - system> Script-19". The SQL Editor shows the following query:

```
1 SELECT * FROM test.job
2 LIMIT 200
3 ALLOW FILTERING
4
```

The Result tab is active, displaying a table with 15 rows. The columns are labeled 'ABC lot' and 'ABC gtn'. The data is as follows:

	ABC lot	ABC gtn
1	1683	Zambia
2	1424	Vietnam
3	1615	Uruguay
4	631	Estonia
5	403	Cabo Verde
6	492	Serbia
7	517	The Gambia
8	1843	East Timor
9	789	The Gambia
10	947	Papua New Guinea
11	255	North Korea
12	326	Senegal
13	374	Iran
14	151	Vietnam
15	1955	Kiribati

The Query Trace tab is also visible, showing the execution steps and their durations:

Event	Time	Source	Thread
Parsing SELECT * FROM test.job LIMIT 200 ALLOW FILTERING	137	/127.0.0.1	Native-Transport-Requests-1
Preparing statement	265	/127.0.0.1	Native-Transport-Requests-1
Computing ranges to query	453	/127.0.0.1	Native-Transport-Requests-1
Submitting range requests on 257 ranges with a concurrency ...	795	/127.0.0.1	Native-Transport-Requests-1
Submitted 1 concurrent range requests	1027	/127.0.0.1	Native-Transport-Requests-1
Executing seq scan across 1 sstables for (min(-9223372036854...	1844	/127.0.0.1	ReadStage-3
Read 200 live rows and 0 tombstone cells	3003	/127.0.0.1	ReadStage-3

The bottom status bar indicates "200 row(s) fetched - 124ms (+1ms)".

ERD

Physical ERD (Entity Relation Diagram) does not make much sense for Cassandra as there are no foreign keys.

However, you can make your own [custom ERD](#) and connect an actual Cassandra table with each other using logical associations.

ClickHouse

Table of contents

[Overview](#)

[Setting up](#)

[ClickHouse connection settings](#)

[Connection details](#)

[ClickHouse driver properties](#)

[ODBC and JDBC driver configuration](#)

[Secure connection configurations](#)

[Powering ClickHouse with DBeaver](#)

[ClickHouse database objects](#)

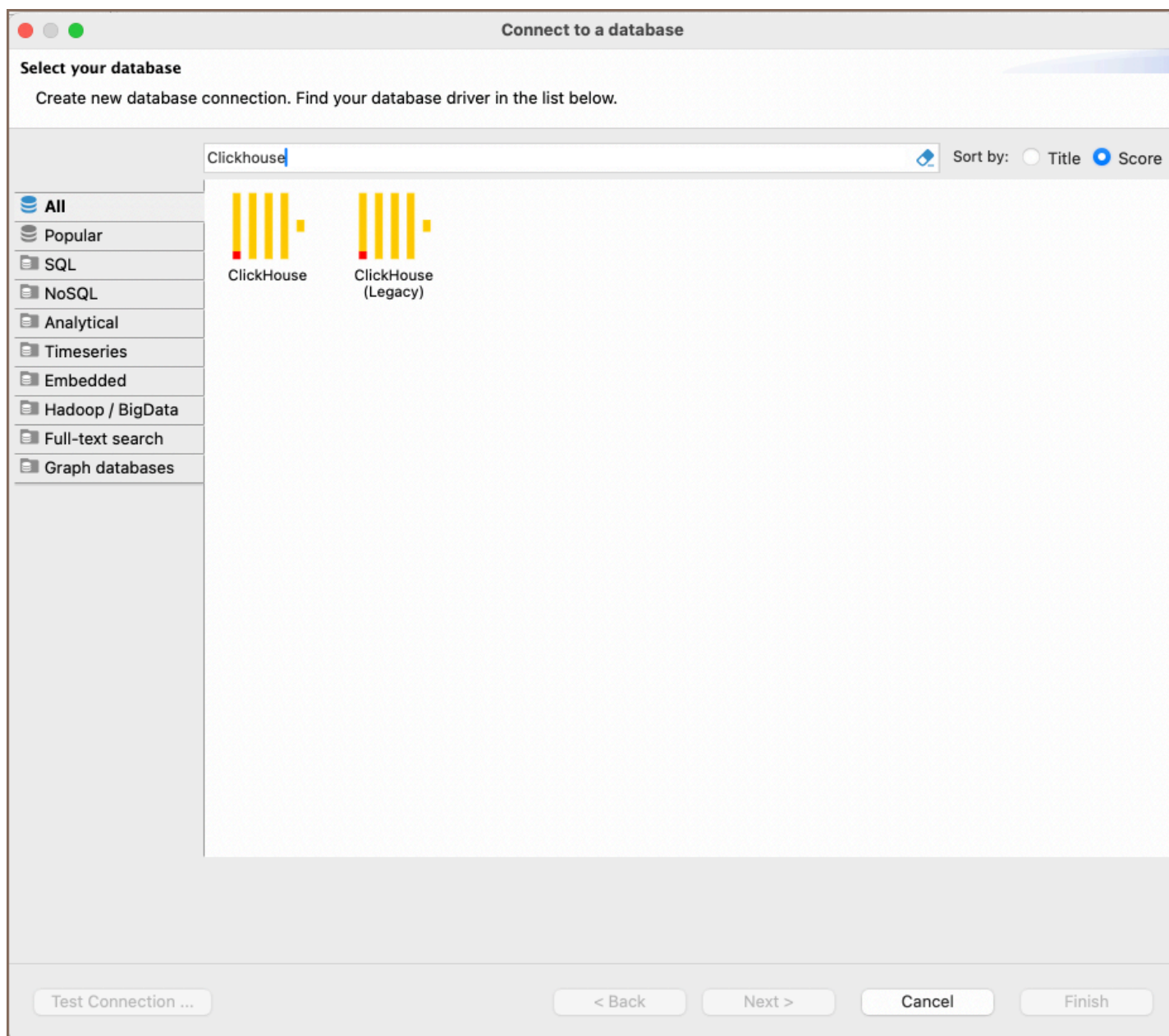
[ClickHouse features in DBeaver](#)

Overview

This guide provides instructions on how to set up and use ClickHouse with DBeaver.

Before you start, you must create a connection in DBeaver and select ClickHouse. If you have not done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the ClickHouse server using a specific driver. It supports all versions of ClickHouse, but the correct driver must be selected: use `ClickHouse (Legacy)` for driver version 21.3 and earlier, and `ClickHouse` for driver version 21.3 and later. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Setting up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, SSL, and the setup of ODBC/JDBC drivers for ClickHouse.

ClickHouse connection settings

In this subsection, we will outline the settings for establishing a direct connection to a ClickHouse database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your ClickHouse database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Generic JDBC Connection Settings

ClickHouse connection settings

Main

Driver properties

SSH

SSL

+ Network configurations...

General

Connect by: ☒ Host ☐ URL

Driver type: JDBC

JDBC URL: jdbc:clickhouse://localhost:8123

Host: localhost

Port: 8123

Database/Schema:

Authentication

Authentication: Database Native

Username:

Password:

☒ Save password

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: ClickHouse

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your ClickHouse database.
JDBC URL	If you are connecting via URL, enter the URL of your ClickHouse database here. This field is disabled if you're connecting via the host.
Host	If you are connecting via Host, enter the host address of your ClickHouse database here.
Database /Schema	Enter the name of the ClickHouse database you want to connect to.

DBeaver User Guide 24.1.ea. Page 256 of 1174.

Port	Enter the port number for your ClickHouse database. The default ClickHouse port is 8123 .
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Native Database Authentication - DBeaver Profile Authentication ★ <p>You can also read about security in DBeaver PRO.</p>
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with ClickHouse database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

ClickHouse driver properties

The settings for ClickHouse **Driver properties** enable you to adjust the performance of ClickHouse ClickHouse JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your ClickHouse database.

For a complete walkthrough on setting up ClickHouse JDBC and ODBC drivers properties, you can refer to the official [ClickHouse JDBC documentation](#) and [ClickHouse ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize ClickHouse database connections.

You can customize the ClickHouse driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your ClickHouse database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC driver configuration

DBeaver provides extensive capabilities for managing ClickHouse database connections via ODBC/JDBC drivers. This functionality enables you to connect to your ClickHouse database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

This process is similar across most databases that support ODBC/JDBC drivers. For a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver, you can refer to our [ODBC Driver Configuration](#) article.

Secure connection configurations

DBeaver supports secure connections to your ClickHouse database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#) ★
-

Powering ClickHouse with DBeaver

DBeaver provides a host of features designed for ClickHouse databases. This includes the ability to view and manage databases, along with numerous unique capabilities aimed at optimizing database operations.

ClickHouse database objects

DBeaver lets you view and manipulate a wide range of ClickHouse database objects. DBeaver has extensive support for various ClickHouse metadata types, allowing you to interact with a wide variety of database objects, such as:

- Schemas
 - Tables
 - Columns
- Views
- Procedures

ClickHouse features in DBeaver

DBeaver is not limited to typical SQL tasks. It also includes numerous unique features specifically for ClickHouse. Beyond regular SQL operations, DBeaver provides a range of ClickHouse-specific capabilities, such as:

Category	Feature
Data Types	ClickHouse data types such as Nested and Array
Database Management	ClickHouse Procedures

Additional features compatible with ClickHouse, but not exclusive to it:

Category	Feature

Data Transfer	Data Import
	Data Export
Schema Management	Schema Compare ★
Mock Data Generation	Mock data generation guide ★
Data Visualization	Query Execution plan ★

Note: As a column-oriented database management system, ClickHouse does not include support for referential integrity, such as foreign keys. This particular limitation means that Entity-Relationship (ER) diagrams, which are typically valuable in visualizing and enforcing database relationships, become less applicable and effective in the context of ClickHouse databases.

Couchbase

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Couchbase specialty](#)

[Setting Up](#)

[Couchbase connection configuration](#)

[Couchbase version 4 and below](#)

[Couchbase version 5 and above](#)

[Connection details](#)

[Couchbase driver properties](#)

[Secure Connection Configurations](#)

[Powering Couchbase with DBeaver](#)

[Couchbase database objects](#)

[Couchbase Features in DBeaver](#)

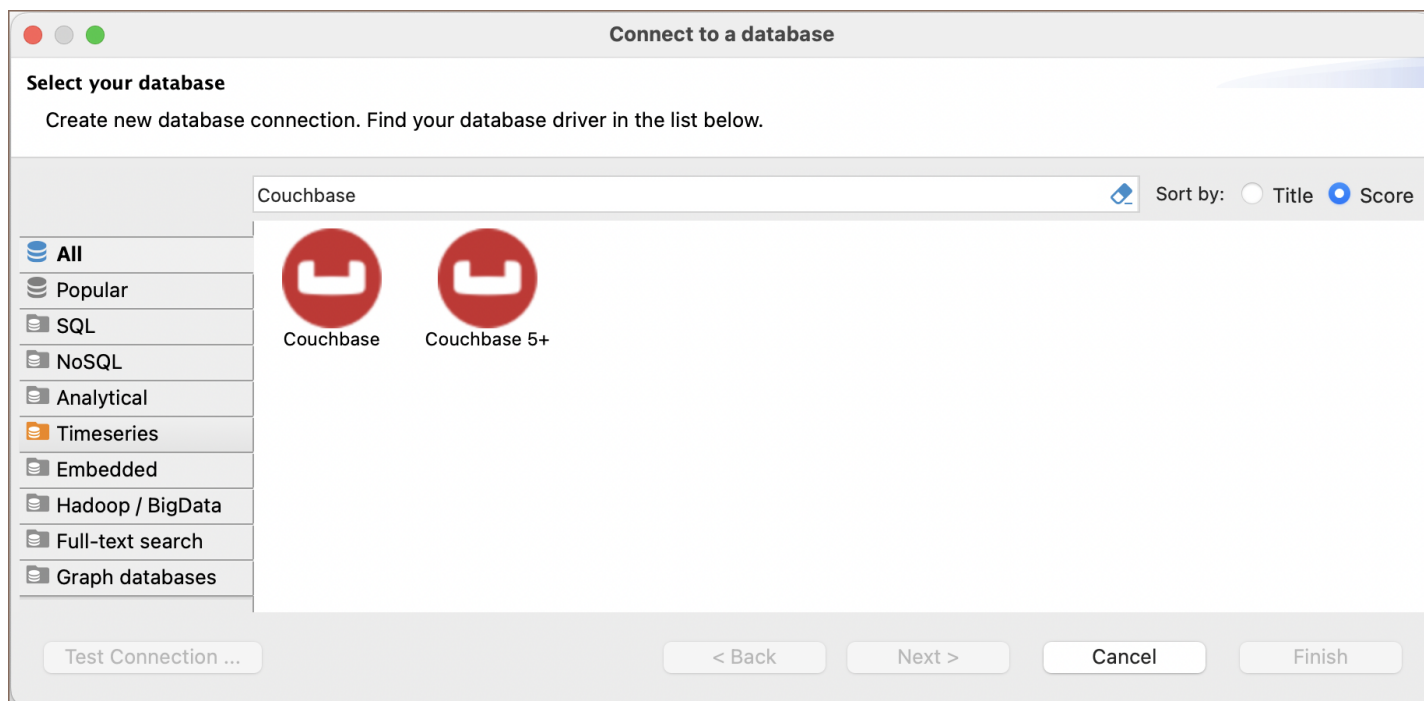
[Executing Couchbase queries](#)

Overview

This guide provides instructions on how to set up and use Couchbase with DBeaver.

Before you start, you must create a connection in DBeaver and select **Couchbase**. If you have not done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the Couchbase server using a specific driver. It supports all versions of Couchbase, but the correct driver must be selected: use **Couchbase** for versions 4.x and earlier, and **Couchbase 5+** for version 5 and later.



Couchbase specialty

Couchbase is a NoSQL database that diverges from traditional relational databases by storing data in JSON documents instead of tables. It employs N1QL, an SQL-like language specifically designed for querying JSON. In Couchbase, the concept typically known as databases in traditional systems is represented as Buckets. Unlike relational databases, Couchbase does not support some common features, such as triggers, stored procedures, views, complex joins, and the constructs of primary and foreign keys. For a more comprehensive understanding of Couchbase, please refer to the [official documentation](#).

The section [Powering Couchbase with DBeaver](#) details the features supported by DBeaver that are compatible with Couchbase.

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, SSL, proxies, and Kubernetes.

Couchbase connection configuration

Couchbase version 4 and below


In this subsection, we will outline the settings for establishing a direct connection to a Couchbase version 4 and below database using DBeaver.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Couchbase Connection Settings

Couchbase connection settings



Main

Driver properties

SSH

+ Network configurations...

Server

Node:localhost

Port:8091

Bucket:travel-sample

Authentication

Authentication:Database Native

Username:test

Password: Save password

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:Couchbase

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Node	Specify the node in the Couchbase cluster to connect to.
Bucket	Enter the name of the specific bucket within the Couchbase database to access.
Port	Enter the port number for your Couchbase database. The default Couchbase port is 8091 .
Authentication	<div>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</div> <div><div>- Native Database Authentication</div><div>- DBeaver Profile Authentication</div></div> <div>You can also read about security in DBeaver PRO.</div>
Connection Details	Provide additional connection details if necessary.

Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	Specify additional driver settings , such as Timeouts and SSL properties.

Couchbase version 5 and above

The connection settings for these versions are similar to those for version 4 and below, with the notable exception that the **Bucket** setting is not used.

The screenshot shows the 'Connect to a database' dialog box in DBeaver. The title bar says 'Connect to a database'. Inside, the 'Couchbase Connection Settings' section is active, with a subtitle 'Couchbase 5+ connection settings' and the Couchbase logo. There are three tabs: 'Main', 'Driver properties', and 'SSH'. The 'Main' tab is selected. Below the tabs, there's a 'Server' section with 'Node' set to 'localhost' and 'Port' set to '8091'. A 'Use SSL' checkbox is unchecked. The 'Authentication' section shows 'Authentication' set to 'Database Native' (with a dropdown arrow), 'Username' and 'Password' fields, and a 'Save password' checkbox which is checked. A hint icon and text say 'You can use variables in connection parameters.' To the right, there's a button 'Connection details (name, type, ...)'. Below this, the 'Driver name' is set to 'Couchbase 5+', with 'Driver Settings' and 'Driver license' buttons. At the bottom, there's a 'Test Connection ...' button and navigation buttons: '< Back', 'Next >', 'Cancel', and a blue 'Finish' button.

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with Couchbase database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- Security Settings Guide
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Couchbase driver properties

Couchbase is a NoSQL database system integrated into DBeaver through its API. While configuring connections to Couchbase in DBeaver, there are specific custom properties available, such as Timeouts and SSL settings.

Connect to a database

Couchbase Connection Settings
Couchbase 5+ connection settings

Couchbase

Main | **Driver properties** | SSH | [+ Network configurations...](#)

Name	Value
▼ Timeouts	
Analytics Timeout	75,000
Connection Timeout	5,000
KV Timeout	2,500
Management Timeout	75,000
Query Timeout	75,000
Search Timeout	75,000
View Timeout	75,000
▼ SSL	
Keystore file	
Keystore password	
▼ User Properties	
couchbase.sslEnabled	false

[Advanced driver properties](#) [Driver documentation](#)

Test Connection ... < Back Next > Cancel Finish

Property	Description
Analytics Timeout	The maximum time to wait for an analytics query before timing out.
Connection Timeout	The time to wait for a connection to be established before timing out.
KV Timeout	Default timeout for key-value operations.

Management Timeout	Time to wait for management operations to complete.
Query Timeout	The maximum time a query can run before it times out.
Search Timeout	Time to wait for a search query to complete before timing out.
View Timeout	Time to wait for a view query to complete before timing out.
Keystore file	Path to the keystore file used for SSL connections.
Keystore password	Password for the keystore file.
couchbase.sslEnabled	A Boolean property to enable or disable SSL. Set to <code>true</code> to enable SSL.

Tip: All timeout settings are specified in milliseconds.

Secure Connection Configurations

DBeaver supports secure connections to your Couchbase database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Note: For SSL configuration, the required properties are set within the DBeaver [Driver settings](#).

Powering Couchbase with DBeaver

DBeaver provides a host of features designed for Couchbase databases. Couchbase uses a document-oriented approach, which means each piece of data, or "document" can have a unique structure.

In DBeaver, you have the flexibility to view and edit the data within Couchbase buckets. This can be done similarly to how you would interact with tables in a traditional database, either through a grid view or as plain text. You can also work directly with JSON documents, making it easier to handle the different data formats that Couchbase supports.

Couchbase database objects

DBeaver lets you view and manipulate a few Couchbase database objects. DBeaver has extensive support for various Couchbase metadata types, allowing you to interact with:

- Buckets
 - Indexes
- Cluster
 - Users

Couchbase Features in DBeaver

DBeaver also includes additional features compatible with Couchbase, but not exclusive to it:

Category	Feature
Data Transfer	Data Export
Data Visualization	ERD Guide
Data Management	Working with JSON

Executing Couchbase queries

Couchbase uses N1QL (Non-First Normal Form Query Language), a query language similar to SQL, it lets you search through JSON documents with complex queries. For further details on N1QL, refer to the [official documentation](#).

Database driver IBM Db2

Table of contents

[Overview](#)

[Setting Up](#)

[Db2 connection settings](#)

[Connection details](#)

[Trace Settings](#)

[Db2 driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering Db2 with DBeaver](#)

[Db2 database objects](#)

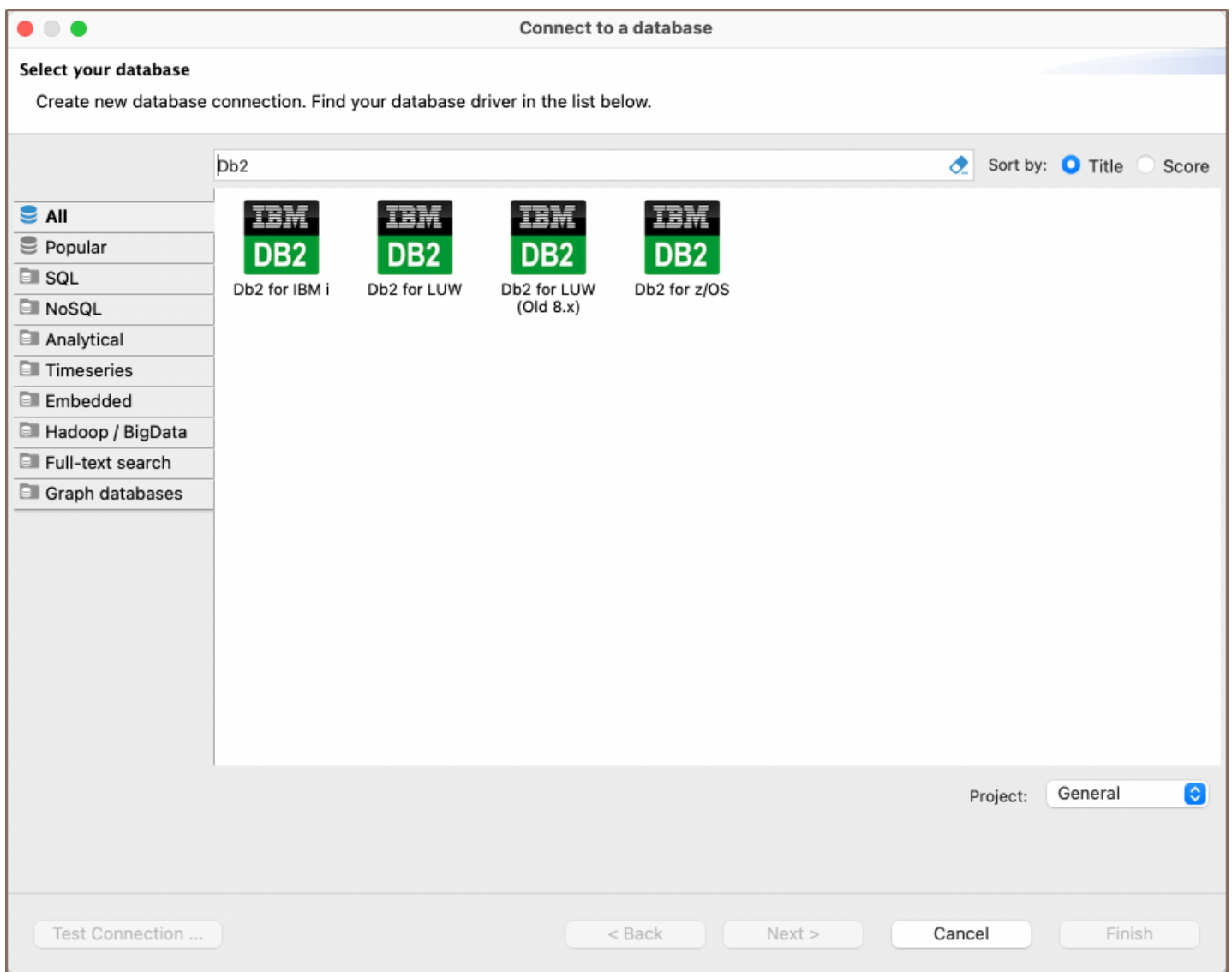
[Db2 Features in DBeaver](#)

Overview

This guide provides instructions on how to set up and use IBM DB2 with DBeaver.

Before you start, you must create a connection in DBeaver and select IBM DB2. If you have not done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the IBM DB2 server using a specific driver, it supports all versions of IBM DB2, but the correct driver must be selected: use **Db2 for LUW (Old 8.x)** for versions below 8.x and **Db2 for LUW** for newer versions. DBeaver also supports various IBM DB2 databases such as **Db2 for IBM i**, and **Db2 for z/OS**. You must select the appropriate driver in the **Connect to a database** window for these databases.



Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, SSL, and the setup of ODBC/JDBC drivers for Db2.

Db2 connection settings

In this subsection, we will outline the settings for establishing a direct connection to a Db2 database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Db2 database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your Db2 database.
URL	If you are connecting via URL, enter the URL of your Db2 database here. This field is hidden if you are connecting via the host.
Host	If you are connecting via host, enter the host address of your Db2 database here.
Database	Enter the name of the Db2 database you want to connect to.
Port	Enter the port number for your Db2 database.
Authentication	Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:

	<ul style="list-style-type: none"> - Native Database Authentication - DBeaiver Profile Authentication ★
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaver allows for further customization of your Db2 connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Trace Settings

In DBeaver, the trace settings feature allows you to monitor and record database operations in a detailed manner. To enable this feature, check the **Enable trace** box. You will need to specify a file where the traced information will be stored and formatted into a readable format.

The trace settings page allows you to configure various levels and settings to tailor the tracing process according to your preferences.

Connect to a database

DB2 Connection Settings

Db2 for LUW connection settings

IBM

DB2

Main
Trace settings
Driver properties
SSH

+ Network configurations...

☒ Enable trace

Folder: /Documents/test

File name: trace

Append: ☐

Levels

☐ Connection calls
☐ Statement calls

☐ Result set calls
☐ Driver configuration

☐ Connects
☐ DRDA flows

☐ Result set metadata
☐ Parameter metadata

☐ Diagnostics
☐ SQL J

☐ XA calls

Test Connection ...

< Back

Next >

Cancel

Finish

Setting	Description
Folder/File name	Specify the directory and file name where the trace data will be stored. This file will contain detailed information about the operations performed in the database.
Append	This option allows you to add information to an existing trace file, instead of creating a new file each time the trace is enabled.
Connection Calls	Records calls related to database connections.
Result Set Calls	Records calls pertaining to the result sets.
Connects	Records information about database connects.
Result Set Metadata	Records metadata information from the result sets.

Diagnostics	Records diagnostic information about database operations.
XA Calls	Records information on XA distributed transactions.
Statement Calls	Records calls related to SQL statements.
Driver Configuration	Records configuration details of the database driver.
DRDA Flows	Records information on DRDA data flows.
Parameter Metadata	Records metadata about SQL parameters.
SQL J	Records information on SQL J operations.

For more detailed information on these settings, please refer to the [IBM documentation](#).

Db2 driver properties

The settings for Db2 **Driver properties** enable you to adjust the performance of the Db2 JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your Db2 database.

For a complete walkthrough on setting up Db2 JDBC and ODBC drivers properties, you can refer to the official [Db2 JDBC documentation](#) and [Db2 ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Db2 database connections.

You can customize the Db2 driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your Db2 database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing Db2 database connections via ODBC/JDBC drivers. This functionality enables you to connect to your Db2 database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

You can find a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver in our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your Db2 database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **SSL**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering Db2 with DBeaver

DBeaver provides a host of features designed for Db2 databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

Db2 database objects

DBeaver lets you view and manipulate a wide range of Db2 database objects. DBeaver has extensive support for various Db2 metadata types, allowing you to interact with a wide variety of database objects, such as:

- Schemas
 - Tables
 - Columns
 - Unique Constraints
 - Foreign Keys

- Indexes
- References
- Check Constraints
- Partitions
- Periods
- Triggers
- Columns Masks
- Views
- MQTs
- Indexes
- Sequences
- Aliases
- Nicknames
- Triggers
- XML Schemas
- Application Objects
 - Functions
 - Modules
 - Packages
 - Procedures
 - User Defined Types
- Federation

- Wrappers
- Remote Servers
- User Mappings
- Global metadata
 - Types
 - Variables
 - XML Strings
- Storage
 - Bufferpools
 - Tablespaces
 - Storage Groups
- Administer
 - Session Manager
- Security
 - Roles
 - Groups
 - Users

Db2 Features in DBeaver

DBeaver is not limited to typical SQL tasks. It also includes numerous unique features specifically for DB2. Beyond regular SQL operations, DBeaver provides a range of Db2-specific capabilities, such as:

Category	Feature
Schemas	Unique Constraints

	Periods
	Column Masks
Views	MQTs
Federation	Wrappers
	Remote Servers
Global Metadata	XML Strings
Storage	Bufferpools
	Tablespaces
Security	Roles
	Groups

Additional features compatible with Db2, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Lock Management	Lock Manager
Schema Management	Schema Compare
Data Visualization	GIS Guide
	ERD Guide

Greenplum

Table of contents

[Overview](#)

[Greenplum specialty](#)

[Setting Up](#)

[Greenplum connection settings](#)

[Connection details](#)

[Greenplum driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering Greenplum with DBeaver](#)

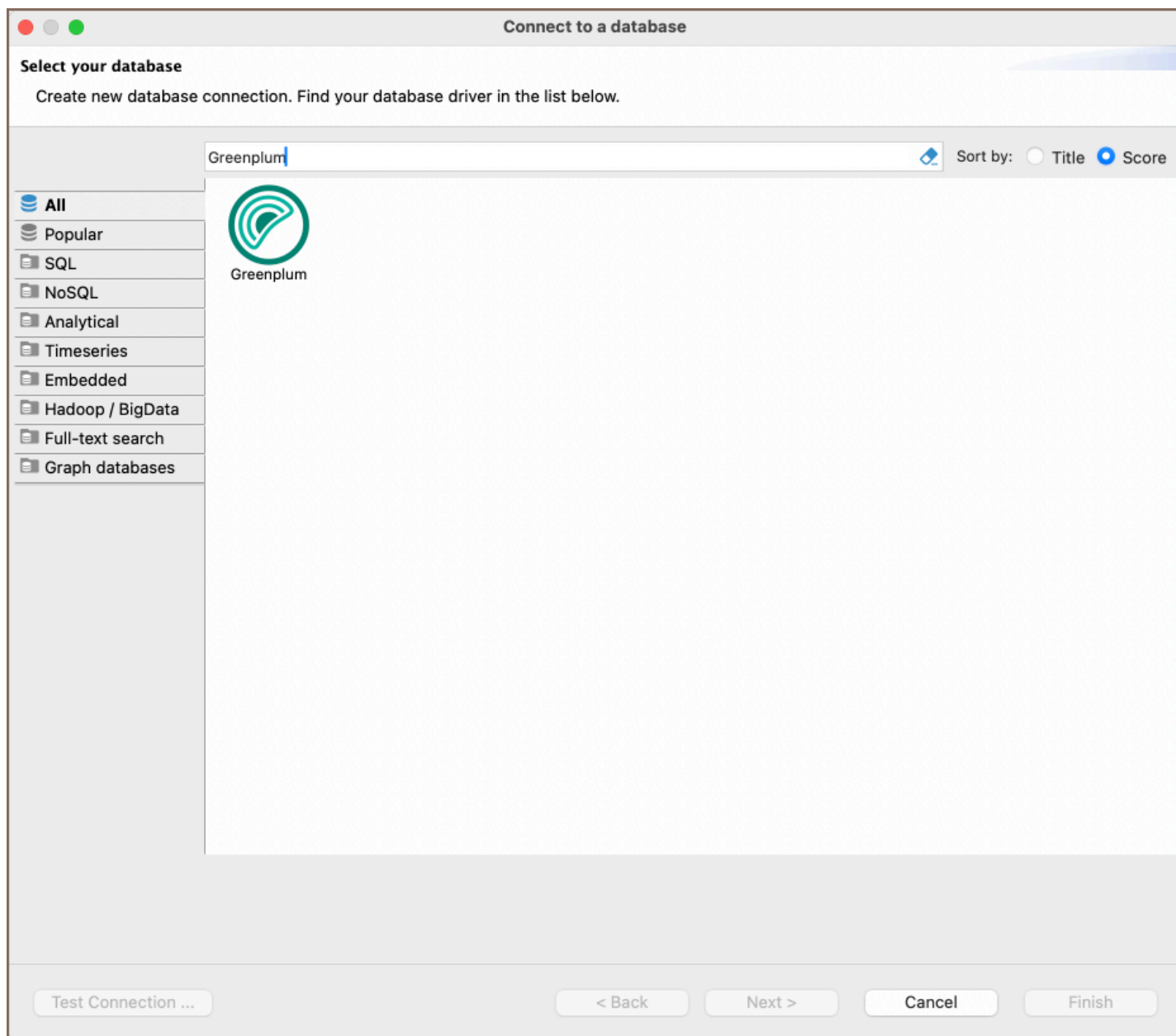
[Greenplum database objects](#)

[Greenplum Features in DBeaver](#)

Overview

This guide provides instructions on how to set up and use Greenplum with DBeaver.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **Greenplum** option to connect to the Greenplum database. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Greenplum specialty

Greenplum is a highly scalable, multi-node relational database system, primarily used for data warehousing and analytics. A key characteristic of Greenplum is its columnar data storage capabilities, making it particularly effective for analytics and large-scale data processing. Built on PostgreSQL, it supports rich metadata structures, including tables, views, indexes, sequences, and various data types. Its setup in DBeaver mostly replicates the PostgreSQL driver configuration. Greenplum includes support for SQL triggers, stored procedures, and user-defined functions, enhancing its utility for complex data operations. It also provides robust data partitioning options and supports diverse storage models like row and column store, optimizing performance for a variety of workloads. For a more comprehensive understanding of Greenplum, please refer to the [official documentation](#).

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, Kubernetes, SSL, and the setup of ODBC/JDBC drivers for Greenplum.

Greenplum connection settings


In this subsection, we will outline the settings for establishing a direct connection to a Greenplum database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Greenplum database.

1) The first page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Connection Settings

Greenplum connection settings



Main

Greenplum

Driver properties

SSH

SSL

+ Network configurations...

Server

Connect by: ☒ Host ☐ URL

Driver type:

JDBC

URL:

jdbc:postgresql://localhost:5432/postgres

Host:

localhost

Port:

5432

Database:

postgres

☐ Show all databases

Authentication

Authentication:


Database Native

Username:

test

Password:

.....


☒ Save password 

Advanced

Session role:

Local Client:

/Library/PostgreSQL/16

 You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:

Greenplum

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

DBeaver User Guide 24.1.ea. Page 280 of 1174.


Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your Greenplum database.
URL	If you are connecting via URL, enter the URL of your Greenplum database here. This field is hidden if you are connecting via the host.
Host	If you are connecting via host, enter the host address of your Greenplum database here.
Database	Enter the name of the Greenplum database you want to connect to.
Show all databases	Displays all databases in the database navigator . Note: This option works only when the Host connection type is selected. It does not work when connecting via URL .
Port	Enter the port number for your Greenplum database. The default Greenplum port is 5432 .
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Native Database Authentication - DBeaver Profile Authentication ★ - PostgreSQL Kerberos Authentication ★ - PostgreSQL PgPass Authentication - PostgreSQL SSPI Authentication ★ <p>You can also read about security in DBeaver PRO.</p>
Local Client	Specify the local client, see our article about Local client .
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

2) The second page of the connection settings offers additional options that allow you to customize your further connection to the Greenplum database.

Connect to a database

Connection Settings

Greenplum connection settings



GREENPLUM
DATABASE

Main

Greenplum

Driver properties

SSH

SSL

+ Network configurations...

Settings

☐ Show template databases

☐ Show databases not available for connection

☐ Show database statistics

☐ Read all data types

☐ Read table keys with columns

SQL

Show \$\$ quote as:

Code block

Show \$tagName\$ quote as:

Code block

Performance

☐ Use prepared statements

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Show template databases	Displays template databases. This option is available when Show all databases is checked.
Show databases unavailable for connection	Show databases not available for connection in database list. This option is available when Show all databases is checked.
Show databases statistics	Displays statistics for databases.
Read all data types	Reads all data types from the server (including arrays).
Read table keys with columns	Enable reading table constraints during the column reading stage to ensure accurate recognition of keys. Note that this setting may decrease metadata loading performance.

Show <code>\$\$</code> quote as	Displays <code>\$\$</code> quotes as code blocks or strings.
Show <code>\$tagName\$</code> quote as	Displays <code>\$tagName\$</code> quotes as code blocks or strings.
Use prepared statements	Uses prepared statements for SQL execution.

Connection details

The **Connection Details** section in DBeaver allows for further customization of your Greenplum connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Greenplum driver properties

The settings for Greenplum **Driver properties** enable you to adjust the performance of the Greenplum JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your Greenplum database.

For a complete walkthrough on setting up Greenplum JDBC and ODBC drivers properties, you can refer to the official [Greenplum JDBC documentation](#) and [Greenplum ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Greenplum database connections.

You can customize the Greenplum driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your Greenplum database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing Greenplum database connections via ODBC/JDBC drivers. This functionality enables you to connect to your Greenplum database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

You can find a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver in our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your Greenplum database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering Greenplum with DBeaver

DBeaver provides a host of features designed for Greenplum databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

Greenplum database objects

DBeaver lets you view and manipulate a wide range of Greenplum database objects. DBeaver has extensive support for various Greenplum metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Schemas
 - Data types
 - Tables
 - Columns
 - Constraints
 - Indexes
 - Dependencies
 - References
 - Partitions
 - Triggers
 - Rules
 - Policies
 - External Tables
 - Views
 - Functions
 - Sequences
 - Data types
 - Aggregate functions

- Event triggers
- Storage
 - Tablespaces
- Roles
- Administer
 - Jobs
 - Session Manager
 - Lock Manager

Greenplum Features in DBeaver

DBeaver is not limited to typical SQL tasks. It also includes numerous additional features compatible with Greenplum, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Lock Management	Lock Manager
Backup and Restore	How to Backup/Restore data
Schema Management	Schema Compare
Data Visualization	GIS Guide
	ERD Guide
	Query Execution plan

InfluxDB

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Connecting to Influx Server](#)

[Browsing InfluxDB schema](#)

[Executing InfluxQL](#)

[Executing Flux](#)

[Query example](#)

Overview

DBeaver EE supports InfluxDB schema browser, data viewer and InfluxQL queries execution.

DBeaver uses InfluxDB Java driver 2.12 to operate with the server over HTTP/HTTPS (standard InfluxDB protocol).

It supports InfluxDB servers of any version (in the moment of writing).


Connecting to Influx Server

You can connect directly to a server or use SSH tunneling or SOCKS proxy.

Create new connection

InfluxDB Connection Settings

InfluxDB connection settings



InfluxDB support: beta version

Server

Host:localhost

Port:8086

Database:NOAA_water_database

Security

User Name:user

Password:

☒ Save password locally

☐ Use SSL

Misc

Default consistency level:ONE

[Network settings \(SSH, SSL, Proxy, ...\)](#)

[Connection details \(name, type, ...\)](#)

Driver Name: InfluxDB

Edit Driver Settings

?

< Back

Next >

Finish

Cancel

Test Connection ...

The screenshot shows the 'Create new connection' dialog box in DBeaver. The 'Network' tab is active, and the 'SSH Tunnel' sub-tab is selected. The 'Use SSH Tunnel' checkbox is checked. The 'Settings' section includes fields for 'Host/IP' (influx.theserver.com), 'Port' (22), 'User Name' (theuser), 'Authentication Method' (Password), 'Password' (masked with dots), and 'Save Password' (checked). The 'Advanced' section includes 'Implementation' (JSch), 'Local port' (0), 'Keep-Alive interval (ms)' (0), and 'Tunnel connect timeout (ms)' (10000). A 'Test tunnel configuration' button is located below the advanced settings. At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Finish' (highlighted with a blue border), 'Cancel', and 'Test Connection ...'.

Browsing InfluxDB schema

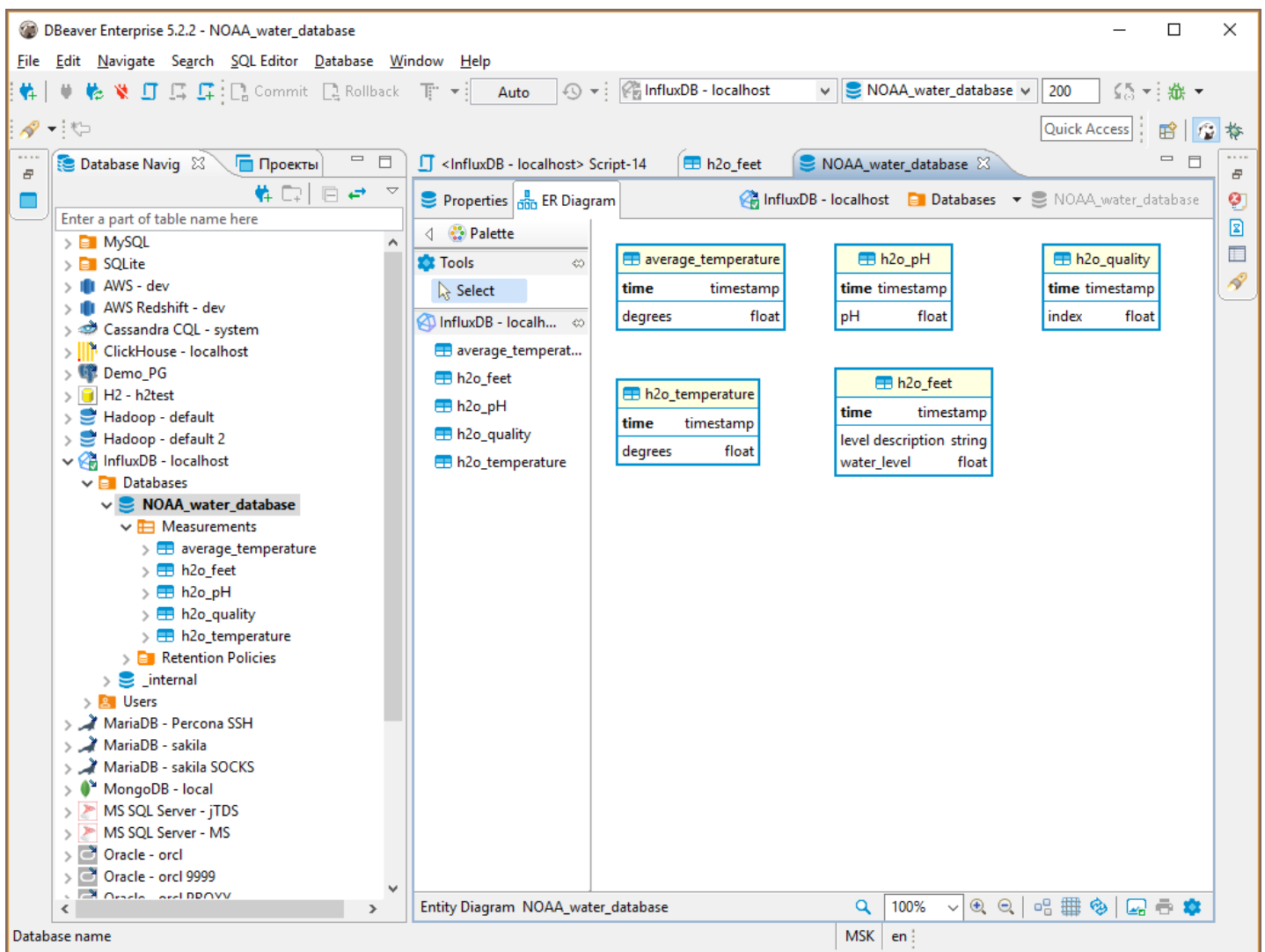
InfluxDB is [TimeSeries database](#), it does not support tables, foreign keys and other relational entities.

DBeaver does not support data insert/update in InfluxDB. Database is basically a in read-only state for DBeaver.

You can browse schema and view/analyse data.

While data itself is loaded by various sensors/data collectors in real time.

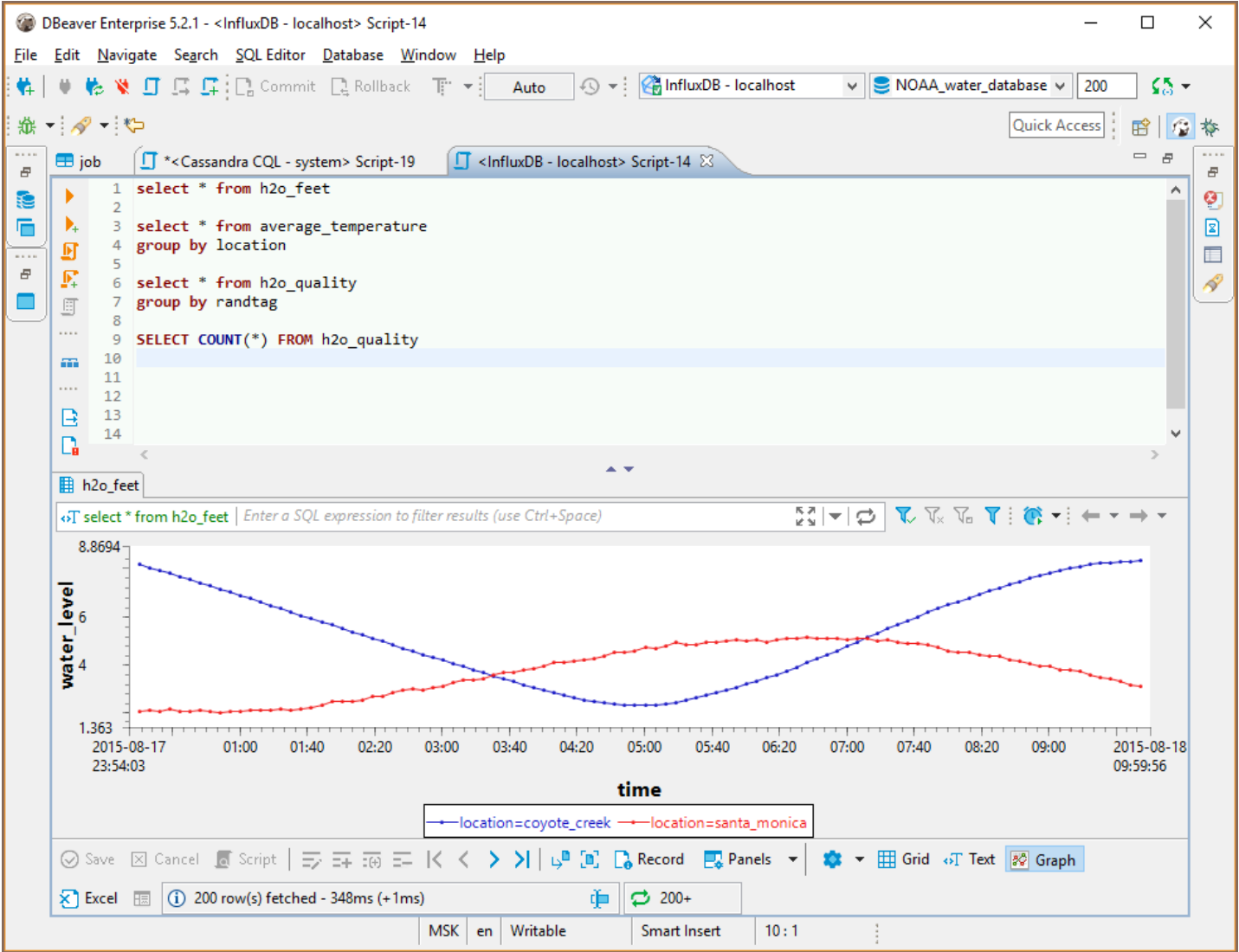
Instead of tables InfluxDB has [measurements](#). Instead of columns it has [fields](#) and [tags](#).

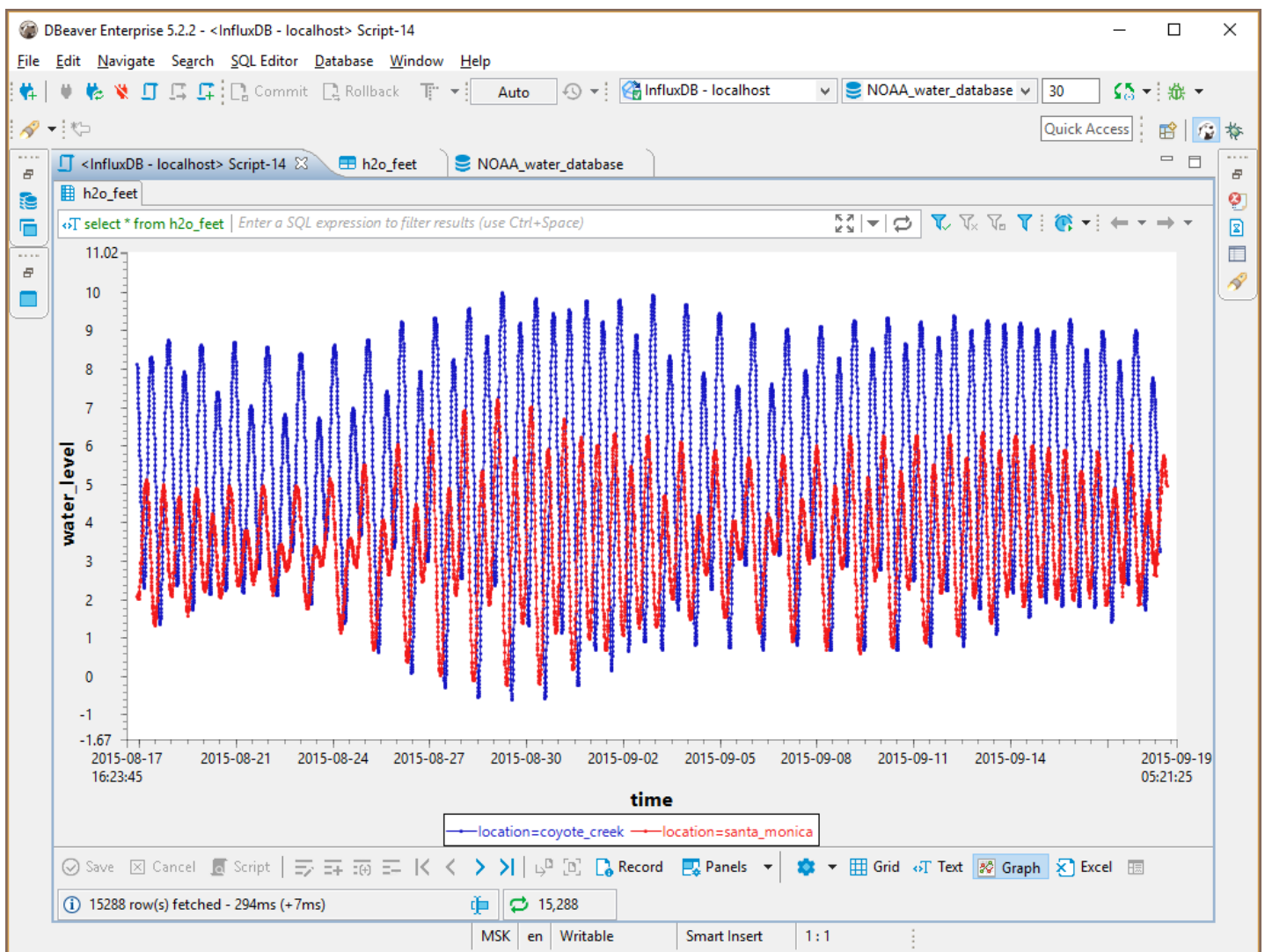


Executing InfluxQL

versions lower than 2 [InfluxQL](#) is a query language similar to SQL.

DBeaver fully supports all InfluxQL statements. Query results are presented as grid or as graphs:





Executing Flux

versions higher than 2 [Flux](#) is a query language that is used in the new versions of Influx. It is not similar to InfluxQL and doesn't use SQL syntax.

Query example

```
from(bucket: "example-bucket")
  |> range(start: -1d)
  |> filter(fn: (r) => r._measurement == "example-measurement")
  |> mean()
  |> yield(name: "_results")
```

Microsoft SQL Server

Table of contents

[Overview](#)

[Setting Up](#)

[SQL Server connection settings](#)

[Connection details](#)

[SQL Server driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering SQL Server with DBeaver](#)

[SQL Server database objects](#)

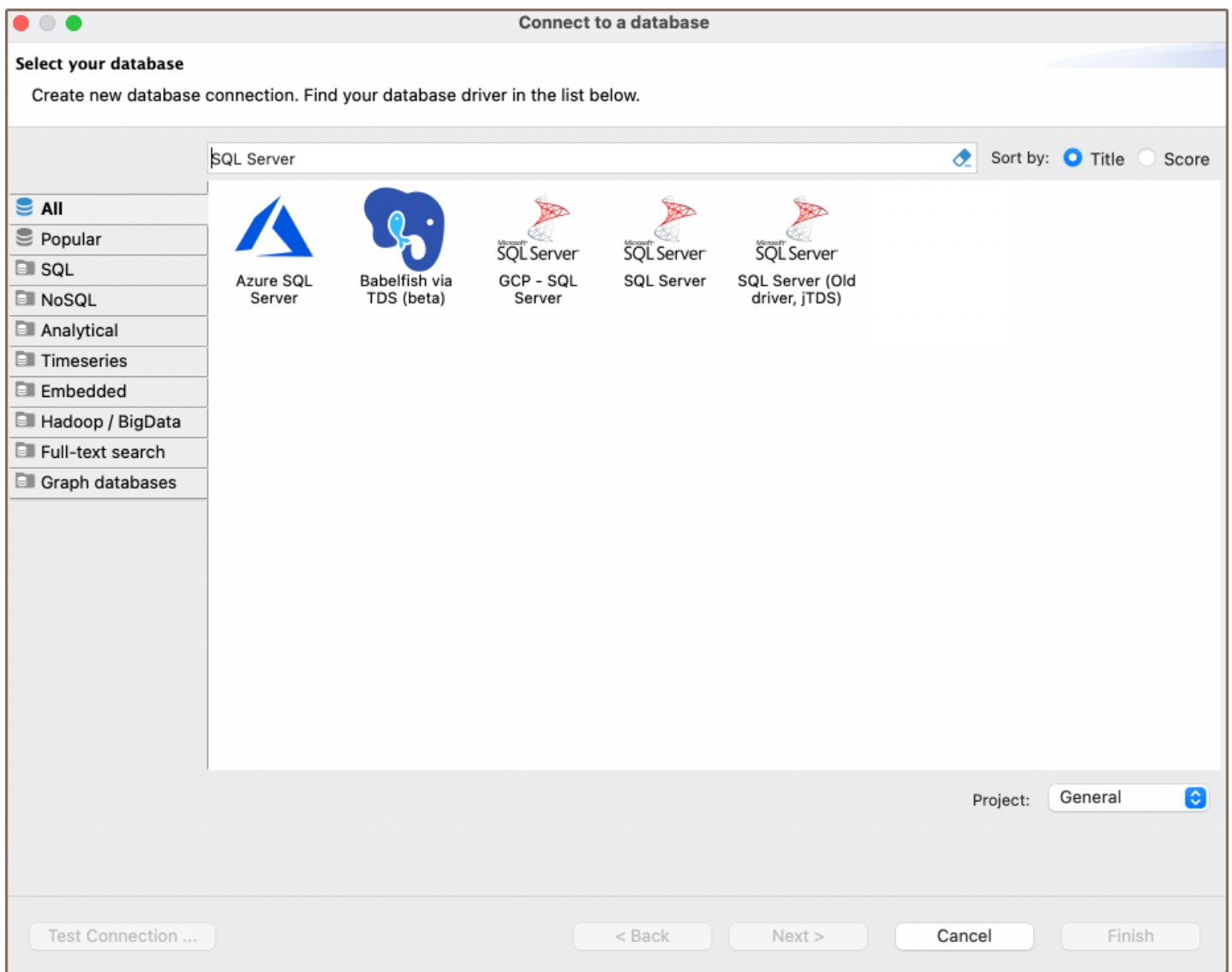
[SQL Server Features in DBeaver](#)

Overview

This guide provides instructions on how to set up and use SQL Server with DBeaver.

Before you start, you must create a connection in DBeaver and select SQL Server. If you have not done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the SQL Server using a specific driver, it supports all versions of SQL Server, but the correct driver must be selected: use **SQL Server (Old driver, jTDS)** for versions below 2012 and **SQL Server** for newer versions. DBeaver also supports SQL Server extensions such as **Azure SQL Server**, **Babelfish via TDS** and **Google Cloud Platform SQL Server**. You must select the appropriate driver in the **Connect to a database** window for these extensions.



Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection, the configuration of secure connections, and the setup of ODBC/JDBC drivers for SQL Server.

SQL Server connection settings

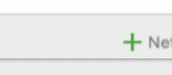
In this subsection, we will outline the settings for establishing a direct connection to a SQL Server database using DBeaver.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

SQL Server Connection Settings

MS SQL Server / SQL Server connection settings



MainDriver propertiesSSHSSL

+ Network configurations...

Server

Connect by: ☒ Host ☐ URL

Driver type: JDBC

URL: jdbc:sqlserver://;serverName=localhost;databaseName=master

Host: localhostPort: 1433

Database/Schema: master

Authentication

Authentication: SQL Server Authentication

Username: dbeaver

Password:

••••••••

☒ Save password

Settings

☐ Show All Schemas

☐ Trust Server Certificate

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: MS SQL Server / SQL Server

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your SQL Server database.
URL	If you are connecting via URL, enter the URL of your SQL Server database here. This field is hidden if you are connecting via the host.
Host	If you are connecting via host, enter the host address of your SQL Server database here.
Database /Schema	Enter the name of the SQL Server database you want to connect to.
Port	Enter the port number for your SQL Server database. The default SQL Server port is 1433 .
	Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:

Authentication	<ul style="list-style-type: none"> - SQL Server Authentication (coming soon) - Active Directory - Integrated (coming soon) - Active Directory - MFA (coming soon) - Active Directory - MSI (coming soon) - Active Directory - Password (coming soon) - Azure AD for SQL Server ★ - Custom (coming soon) - DBeaver profile ★ - Kerberos ★ - NTLM (coming soon) - Window Authentication (coming soon) <p>You can also read about security in DBeaver PRO.</p>
Show All Schemas	Shows all database schemas. If disabled shows only non-empty schemas.
Trust Server Certificate	If <code>true</code> , the SQL Server SSL certificate is automatically trusted when the communication layer is encrypted using SSL. If <code>false</code> , the Microsoft JDBC Driver for SQL Server validates the server SSL certificate. If the server certificate validation fails, the driver raises an error and terminates the connection. To configure certificate settings in DBeaver, refer to our detailed article on Truststore .
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with SQL Server database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)

- [Database Navigator](#)
- [Security Settings Guide](Managing-security-restrictions-for-database-connection.md)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

SQL Server driver properties

The settings for SQL Server **Driver properties** enable you to adjust the performance of the SQL Server JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your SQL Server database.

For a complete walkthrough on setting up SQL Server JDBC and ODBC drivers properties, you can refer to the official [SQL Server JDBC documentation](#) and [SQL Server ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize SQL Server database connections.

You can customize the SQL Server driver in DBeaver via the **Edit Driver** window, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This window offers a range of settings that can influence your SQL Server database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing SQL Server database connections via ODBC/JDBC drivers. This functionality enables you to connect to your SQL Server database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

You can find a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver in our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your SQL Server database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering SQL Server with DBeaver

DBeaver provides a host of features designed for SQL Server databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

SQL Server database objects

DBeaver lets you view and manipulate a wide range of SQL Server database objects. DBeaver has extensive support for various SQL Server metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Schemas
 - Tables
 - External Tables
 - Views
 - Columns
 - Triggers
 - External Properties

- Indexes
- Procedures
- Sequences
- Synonyms
- Triggers
- Data Types
- Database triggers
- Security
 - Logins
- Administer
 - Sessions
- Agent
 - Jobs

SQL Server Features in DBeaver

DBeaver includes many features that go beyond basic SQL operations like querying, updating, inserting, and deleting data. It also includes numerous unique features specifically for SQL Server. Beyond regular SQL operations, DBeaver offers a range of SQL Server-specific capabilities, such as:

Category	Feature
Data Structure	SQL Server Tables & Views
	SQL Server Procedures
	SQL Server Sequences
	SQL Server Synonyms
	SQL Server External Tables

Job Management	Agent Jobs
-------------------	------------

Additional features compatible with SQL Server, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Schema Management	Schema Compare
Data Visualization	GIS Guide
	ERD Guide

MongoDB

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Setting Up](#)

[MongoDB connection settings](#)

[Connection details](#)

[MongoDB driver properties](#)

[Secure Connection Configurations](#)

[Powering MongoDB with DBeaver](#)

[MongoDB database objects](#)

[MongoDB Features in DBeaver](#)

[Browsing MongoDB collections](#)

[Database operations](#)

[Executing JavaScript](#)

[Executing SQL](#)

[SELECT queries](#)

[Conditions](#)

[Nested fields](#)

[Working with object IDs](#)

[Working with JOINS](#)

[Aggregate functions](#)

[INSERT statement](#)

[UPDATE statement](#)

[DELETE statement](#)

[CREATE TABLE statement](#)

[DROP TABLE statement](#)

[Working with dates](#)

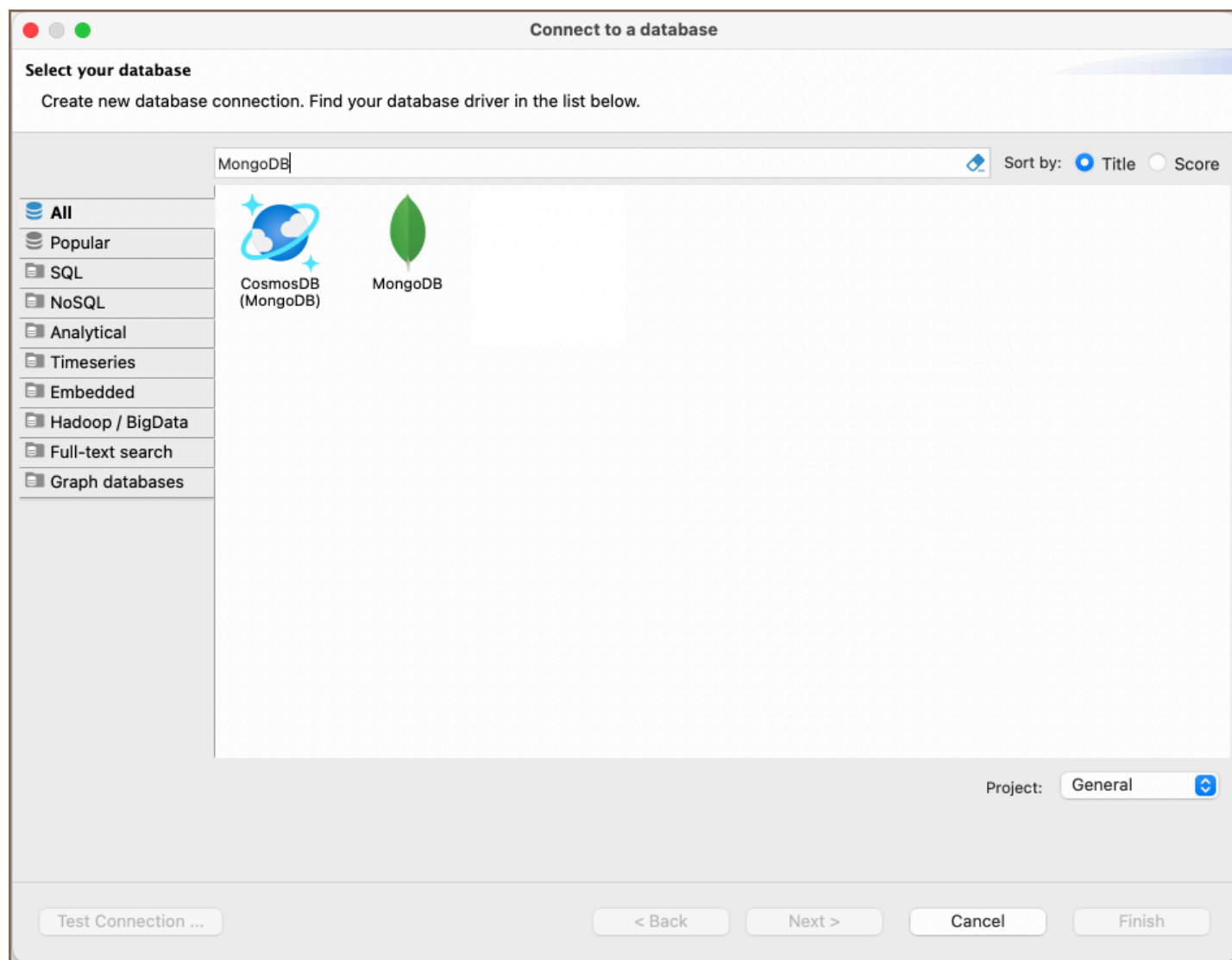
Overview

This guide provides instructions on how to set up and use MongoDB with DBeaver.

One of the standout features of DBeaver's MongoDB support is its flexibility in presentation. You can view MongoDB collections as standard relational tables, JSON documents, or even in chart presentations.

DBeaver interacts with MongoDB servers using a specific driver, supporting versions from 2.x to the current version. DBeaver also supports MongoDB extension such as CosmosDB, you can find more information about this driver in our [article](#).

Before you start, you must create a connection in DBeaver and select MongoDB. If you have not done this, please refer to our [Database Connection](#) article.



Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies and SSL.

MongoDB connection settings

In this subsection, we will outline the settings for establishing a direct connection to a MongoDB database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your MongoDB database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Connection Settings

MongoDB connection settings

Main

Driver properties

SSH

SSL

+ Network configurations...

Address

Type: ☒ Manual ☐ URL

Host:localhost

Port:27017

Database:local

Replica Set:

Authentication

Authentication:MongoDB authentication

Mechanism:NONE

Source:admin

Username:

Password:

☒ Save password

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: MongoDB

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description

Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
URL	If you are connecting via URL, enter the URL of your MongoDB database here. This field is disabled if you're connecting via the host.
Host	If you are connecting via host, enter the host address of your MongoDB database here.
Database	Enter the name of the MongoDB database you want to connect to.
Replica Set	Specify the name of the replica set if your MongoDB instance is a part of a replica set configuration.
Port	Enter the port number for your MongoDB database. The default MongoDB port is 27017 .
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - MongoDB Authentication - DBeaiver Profile Authentication ★ <p>You can also read about security in DBeaiver PRO.</p>
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaiver allows you to customize your experience while working with MongoDB database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)

- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

MongoDB driver properties

The settings for MongoDB **Driver properties** enable you to adjust the performance of the MongoDB JDBC driver. These adjustments can influence the efficiency, compatibility, and features of your MongoDB database.

For a complete walkthrough on setting up MongoDB JDBC driver properties, you can refer to the official [MongoDB JDBC documentation](#). This guide detail driver's properties and how they can be used to optimize MongoDB database connections.

You can customize the MongoDB driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your MongoDB database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your MongoDB database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering MongoDB with DBeaver

DBeaver provides a host of features designed for MongoDB databases. This includes the ability to view and manage collections, along with numerous unique capabilities aimed at optimizing database operations.

MongoDB database objects

DBeaver lets you view and manipulate a wide range of MongoDB database objects. DBeaver has extensive support for various MongoDB metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Collections
 - Java Script
 - Users
- Administration
 - Active Operations

MongoDB Features in DBeaver

DBeaver is not confined to handling typical SQL tasks. It also embraces the NoSQL database spectrum, offering numerous unique features specifically designed for MongoDB. Beyond standard SQL operations, DBeaver facilitates a plethora of MongoDB-specific capabilities, such as:

Category	Feature
Data Types	BSON Data Types (e.g., Object ID, ISODate)
Security	User-Based Access Control
Database Management	MongoDB Stored Procedures (JavaScript Functions)

Additional features compatible with MongoDB, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Schema Management	Schema Compare

Browsing MongoDB collections

You can view or edit MongoDB collection content as standard relational tables (in grid/plain text presentations /chart) or as [JSON documents](#). You can switch between these presentations using the [toolbar](#) of the Data Editor.

Properties Data ER Diagram test Databases sample_airbnb Collections listingsAndReviews

listingsAndReviews Enter a SQL expression to filter results (use Ctrl+Space)

[1] 10006546

```
{
  _id: "10006546",
  listing_url: "https://www.airbnb.com/rooms/10006546",
  name: "Ribeidraa Charming Duplexx",
  summary: "Fantastic duplex apartment with three bedrooms, located in the historic area of Porto, Ribeira (Cube) - UNESCO",
  space: "Privileged views of the Douro River and Ribeira square, our apartment offers the perfect conditions to discover",
  description: "Fantastic duplex apartment with three bedrooms, located in the historic area of Porto, Ribeira (Cube) - UN",
  neighborhood_overview: "In the neighborhood of the river, you can find several restaurants as varied flavors, but withou",
  notes: "Lose yourself in the narrow streets and staircases zone, have lunch in pubs and typical restaurants, and find th",
  transit: "Transport: • Metro station and S. Bento railway 5min; • Bus stop a 50 meters; • Lift Guindais (Funicular) 50 m",
  access: "We are always available to help guests. The house is fully available to guests. We are always ready to assist g",
  interaction: "Cot - 10 € / night Dog - € 7,5 / night",
  house_rules: "Make the house your home...",
  property_type: "House",
  room_type: "Entire home/apt",
  bed_type: "Real Bed",
  minimum_nights: "2",
  maximum_nights: "30",
  cancellation_policy: "moderate",
  last_scraped: ISODate('2019-02-16T05:00:00.000Z'),
  calendar_last_scraped: ISODate('2019-02-16T05:00:00.000Z'),
  first_review: ISODate('2016-01-03T05:00:00.000Z'),
  last_review: ISODate('2019-01-20T05:00:00.000Z'),
  accommodates: 8,
  bedrooms: 3,
  beds: 5,
  number_of_reviews: 51,
  bathrooms: 1,
  amenities: [ ... ],
  price: 80,
  security_deposit: 200,
  cleaning_fee: 35,
  extra_people: 15,
  guests_included: 6,
  images: {
    thumbnail_url: "",
    medium_url: "",
    picture_url: "https://a0.muscache.com/im/pictures/e83e702f-ef49-40fb-8fa0-6512d7e26e9b.jpg?aki_policy=large",
    xl_picture_url: ""
  },
  host: {
    host_id: "51399391",
    host_url: "https://www.airbnb.com/users/show/51399391",
  }
}
```

Refresh Save Cancel Export data 200 200+

Database operations

Executing JavaScript

Execute JavaScript statements in the [SQL editor](#) as usual. DBeaver supports all JavaScript queries for MongoDB versions 2 and 3, as well as a subset of the `mongo` shell queries.

Here is an example that creates a user in the current database:

```
db.createUser({
  user: 'testuser',
  pwd: 'test',
  roles: []
})
```

This example returns all documents in the collection 'test_col':

```
db.test_col.find()
```

Note: Scripts will be executed in the current database. You can not set an explicit database name in your query. The current database can be changed on the SQL Editor toolbar or on the [Database Navigator](#).

Executing SQL

You can use standard SQL statements (`SELECT` , `INSERT` , `UPDATE` , `DELETE`) to manipulate data in MongoDB.

SELECT queries

SELECT queries can include `WHERE` , `ORDER BY` , `GROUP BY` , `JOIN` and `HAVING` clauses.

```
SELECT * FROM test_col
WHERE propName.subProp='value'

UPDATE FROM test_col
SET propsName.val1=123
WHERE propName.subProp='value'
```

Note: The MongoDB dialect does not support SQL sub-queries.

Conditions

SELECT queries with `WHERE` clauses support `AND` , `OR` , `<` , `<=` , `>` , `>=` , `=` and `!=` operators:

```
SELECT * FROM Employees
WHERE (Country = 'CA' OR Country = 'RU') AND Age > 20;
```

TIP: Be aware that `AND` has a higher precedence than `OR` and will be evaluated first; enclose it with parentheses to maintain the correct order.

Nested fields

You can differentiate nested JSON fields using a dot. Enclose fields containing special characters (like spaces or dashes) with double quotes, as demonstrated below:

```
SELECT title FROM movies WHERE info."imdb-details".rating > 6
```

Working with object IDs

To find a document by ID, use the `ObjectId` function:

```
SELECT * FROM documents
WHERE _id = ObjectId('5f9c458018e3c69d0adc0fbd')
ORDER BY value DESC
```

Working with JOINS

The SQL dialect for MongoDB supports `LEFT JOIN` and `INNER JOIN` currently:

```
SELECT
  ar.Name as Artist,
  al.Title as Album,
  SUM(tr.Milliseconds) as Duration
FROM Track tr
INNER JOIN Album al ON tr.AlbumId = al.AlbumId
INNER JOIN Artist ar ON al.ArtistId = ar.ArtistId
GROUP BY Artist, Album
ORDER BY Duration DESC
```

Remember to specify aliases for both the source and target tables in a defined order, as shown:

```
SELECT *
FROM <source> <source-alias>
INNER JOIN <target> <target-alias> ON <source-alias>.column = <target-alias>.column
```

Executing the script below does not yield a merged document but produces separate documents for `Track` and

`Album`:

```
SELECT *
FROM Track tr
INNER JOIN Album al ON tr.AlbumId = al.AlbumId
```

Aggregate functions

In version 22.x only `COUNT` function is supported.

INSERT statement

You cannot use conditions in `INSERT` statements, only the basic form is supported:

```
INSERT INTO <collection-name> (field1, field2) VALUES (val1, val2);
```

UPDATE statement

While you can use various expressions in the `WHERE` clause, sub-selects or joins are not permissible.

```
UPDATE <collection-name> SET field2=val3 WHERE field1=val1;
```

DELETE statement

You may use any expression in the `WHERE` clause, but sub-selects or joins are not allowed.

```
DELETE FROM <collection-name> WHERE field1=val1;
```

CREATE TABLE statement

In the `CREATE TABLE` statement, only the collection name can be specified, column lists are not allowed.

```
CREATE TABLE <collection-name>;
```

DROP TABLE statement

```
DROP TABLE <collection-name>;
```

Working with dates

When working with dates, ensure to specify them in ISO format. This is applicable in both JavaScript and SQL dialects:

```
db.dates.insert([
  { value: new Date('2016-05-18T16:00:00Z') },
  { value: new Date('2017-05-18T16:00:00Z') },
  { value: new Date('2018-05-18T16:00:00Z') },
  { value: new Date('2019-05-18T16:00:00Z') },
  { value: new Date('2020-05-18T16:00:00Z') }
])
```

To query data in JavaScript, follow this example:

```
db.dates.find({
  value: { $gte: new Date('2018-05-18T16:00:00Z') }
})
```

When querying data in the SQL dialect, you can use either the ISO format or UNIX timestamp (in milliseconds):

```
SELECT value FROM dates
WHERE value > ISODate('2018-05-18T16:00:00.000Z')
ORDER BY value DESC

SELECT value FROM dates
WHERE value > ISODate(1526659200000)
ORDER BY value DESC
```

Authentication MongoDB

Table of contents

[Overview](#)

[Settings](#)

Overview

DBeaver is fully compatible with the MongoDB authentication method, offering you a secure way to access your [MongoDB](#) databases.

Connection "test" configuration

Connection settings
MongoDB connection settings

Connection settings
Initialization
Shell Commands
Client identification
Transactions
General
Metadata
Errors and timeouts
Tableau
> Data Editor
> SQL Editor

Main Driver properties SSH SSL + Network configurations...

Address

Type: ☒ Manual ☐ URL

Host: localhost Port: 27017

Database: test

Replica Set:

Authentication

Authentication: MongoDB authentication

Mechanism: SCRAM-SHA-1

Source: admin

Username: dbeaver

Password: ☒ Save password

[You can use variables in connection parameters.](#)

Driver name: MongoDB Driver Settings Driver license

Test Connection ... Cancel OK

Settings

Here are the steps to configure MongoDB authentication:

1. From the **Authentication** dropdown menu, select the **MongoDB authentication** method.
2. From the **Mechanism** dropdown menu, select an option. The following table outlines the available authentication mechanisms:

--	--

Mechanism	Description
SCRAM-SHA-1	A password-based authentication mechanism using SHA-1 hashing algorithm. Suitable for securing data transmissions.
SCRAM-SHA-256	An upgraded version of <code>SCRAM-SHA-1</code> , using the stronger <code>SHA-256</code> hashing algorithm. Offers enhanced security.
MONGODB-CR	A protocol specific to MongoDB, utilizing a challenge-response mechanism for authentication.
PLAIN	A simple, unencrypted password-based authentication mechanism. Not recommended for production environments due to security vulnerabilities.
GSSAPI	Based on the Kerberos protocol, this mechanism offers secure password authentication.
MONGODB-X509	A certificate-based authentication mechanism that validates the client's <code>X.509</code> certificates.
NONE	No authentication is required. This method should only be used in secure, isolated environments.

You can read more about these mechanisms on the [MongoDB documentation](#) page.

3. Fill in the **Source** field if necessary.

4. Enter the username and password in the **Username** and **Password** fields respectively.

Tip: If you check the **Save credentials locally** option, you will not be asked to input your credentials every time you connect.

5. Once you have input your credentials, click the **Test Connection** button to verify your settings. If everything is configured correctly, DBeaver should now be able to connect to your database using the MongoDB authentication method.

MySQL

Table of contents

[Overview](#)

[Setting Up](#)

[MySQL connection settings](#)

[Connection details](#)

[MySQL driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering MySQL with DBeaver](#)

[MySQL database objects](#)

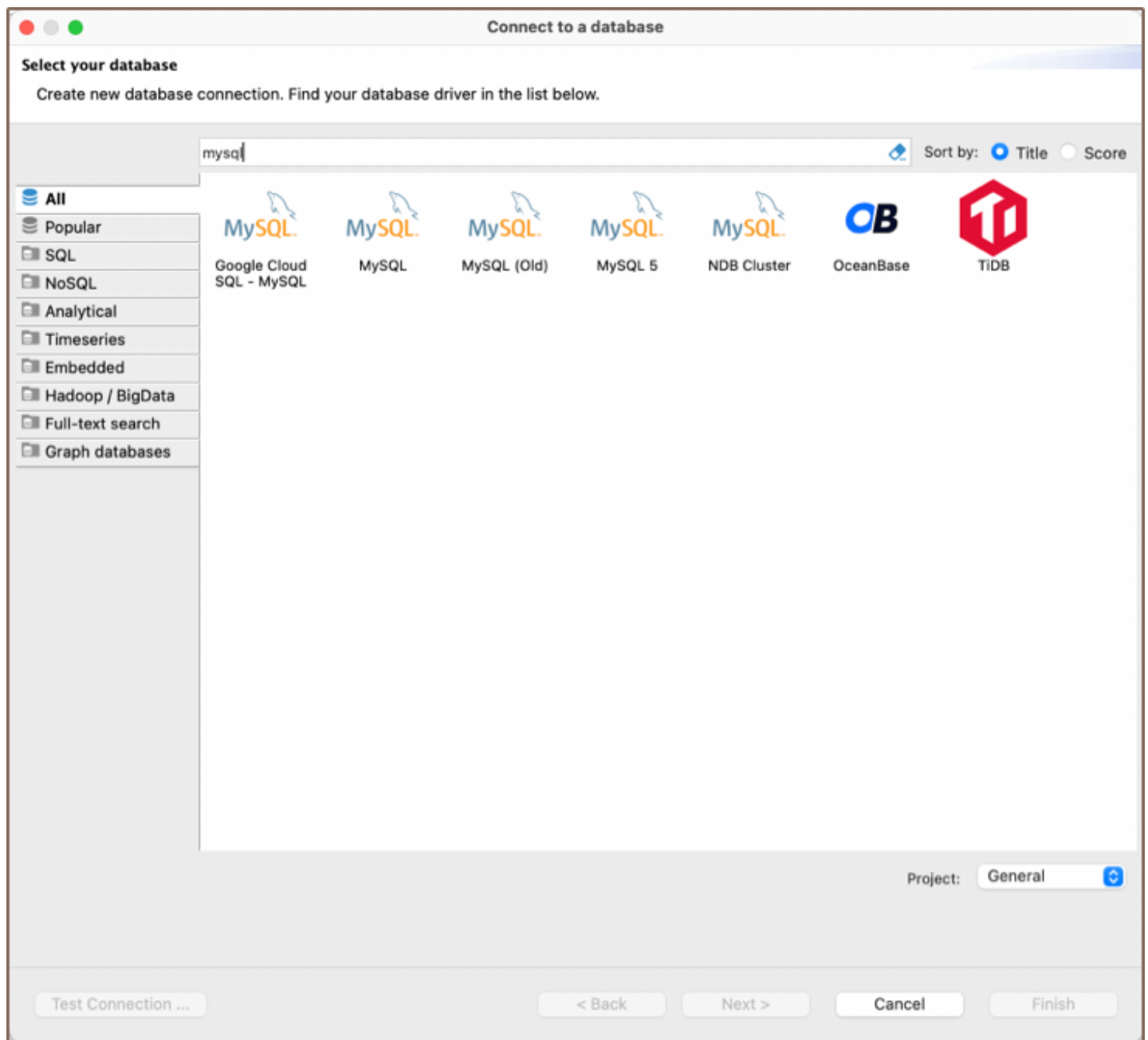
[MySQL Features in DBeaver](#)

Overview

This guide provides instructions on how to set up and use MySQL with DBeaver.

Before you start, you must create a connection in DBeaver and select MySQL. If you have not done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the MySQL server using a specific driver. It supports all versions of MySQL, but the correct driver must be selected: use MySQL (old) for versions 4.x and earlier, MySQL 5 for 5.x versions, and MySQL for version 8 and later. DBeaver also supports MySQL extensions such as NDB Cluster, OceanBase, and TiDB. For Google Cloud SQL, there is a dedicated Google Cloud SQL (MySQL) driver that supports MySQL version 8 and later. You must select the appropriate driver in the **Connect to a database** window for these extensions.



Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, SSL, and the setup of ODBC/JDBC drivers for MySQL.

MySQL connection settings

In this subsection, we will outline the settings for establishing a direct connection to a MySQL database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your MySQL database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

MySQL connection settings

MySQL

Main

Driver properties

SSH

SSL

+ Network configurations...

Server

Connect by: ☒ Host ☐ URL

Driver type: JDBC

URL: jdbc:mysql://localhost:3306/

Server Host: localhost

Port: 3306

Database:

Authentication

Authentication: Database Native

Username: root

Password:

☒ Save password

Advanced

Server Time Zone: Auto-detect

Local Client: MySQL 8.0.33

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: MySQL

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your MySQL database.
URL	If you are connecting via URL, enter the URL of your MySQL database here. This field is disabled if you're connecting via the host.
Host	If you are connecting via host, enter the host address of your MySQL database here.
Database	Enter the name of the MySQL database you want to connect to.
Port	Enter the port number for your MySQL database. The default MySQL port is 3306 .
	Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:

DBeaver User Guide 24.1.ea. Page 318 of 1174.

Authentication	<ul style="list-style-type: none"> - Native Database Authentication - AWS RDS IAM Authentication ★ - Azure AD Authentication for MySQL ★ - DBeaver Profile Authentication ★ - MySQL Kerberos Authentication ★ - Two-factor Authentication ★ <p>You can also read about security in DBeaver PRO.</p>
Local Client	Specify the local client, see our article about Local client.
Server Time Zone	This setting is responsible for determining the time zone used by the MySQL server.
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with MySQL database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)

MySQL driver properties

The settings for MySQL **Driver properties** enable you to adjust the performance of the MySQL JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your MySQL database.

For a complete walkthrough on setting up MySQL JDBC and ODBC drivers properties, you can refer to the official [MySQL JDBC documentation](#) and [MySQL ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize MySQL database connections.

You can customize the MySQL driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your MySQL database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing MySQL database connections via ODBC/JDBC drivers. This functionality enables you to connect to your MySQL database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

This process is similar across most databases that support ODBC/JDBC drivers. For a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver, you can refer to our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your AlloyDB database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)

- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering MySQL with DBeaver

DBeaver provides a host of features designed for MySQL databases. This includes the ability to view and manage databases, along with numerous unique capabilities aimed at optimizing database operations.

MySQL database objects

DBeaver lets you view and manipulate a wide range of MySQL database objects. DBeaver has extensive support for various MySQL metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Tables
 - Columns
 - Constraints
 - Indexes
 - Foreign Keys
 - References
 - Triggers
 - Views
 - Indexes
 - Procedures

- Triggers
- Events
- Users
- Administer
- Session Manager

MySQL Features in DBeaver

DBeaver is not limited to typical SQL tasks. It also includes numerous unique features specifically for MySQL. Beyond regular SQL operations, DBeaver provides a range of MySQL-specific capabilities, such as:

Category	Feature
Data Types	MySQL Enumerations and Sets
Security	MySQL User Management
Storage Engines	InnoDB, MyISAM, MRG_MyISAM, Archive, Blackhole, CSV, Memory, Performance_schema
Database Management	MySQL Events
	MySQL Stored Procedures

Additional features compatible with MySQL, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Backup and Restore	How to Backup/Restore data
Schema	

Management	Schema Compare
Data Visualization	GIS Guide
	ERD Guide

Mysql two-factor authentication

Table of contents

[Overview](#)

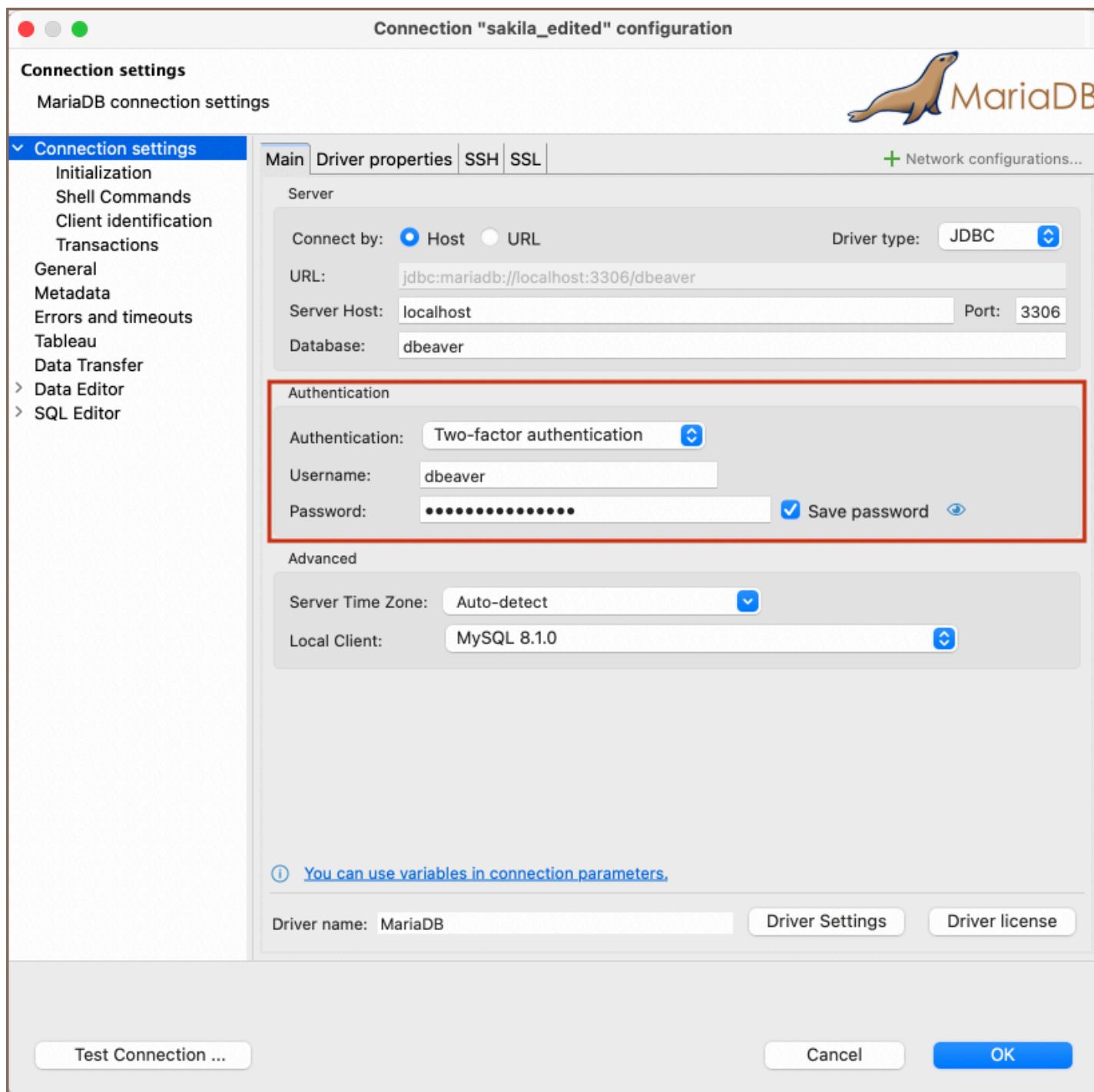
[Settings](#)

[How to Use](#)

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Overview

In addition to Database Native authentication, DBeaver also provides an option for Two-Factor Authentication (2FA). This feature requires a secondary password, offering enhanced security for database connections.



Settings

To set up Two-Factor Authentication with a secondary password, please refer to the official MySQL documentation: [Multi-Factor Authentication](#).

How to Use

When Two-Factor Authentication is enabled, you need to enter the secondary password along with your primary credentials:

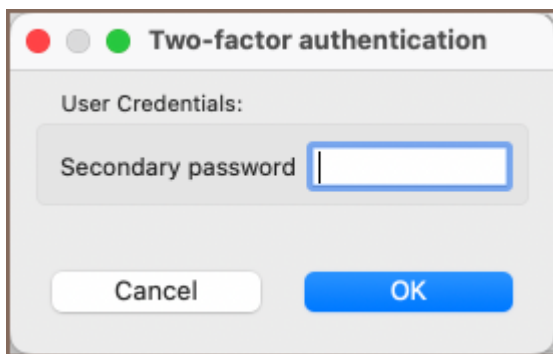
1) In the **Authentication** dropdown menu, choose the **Two-factor authentication** method.

Tip: If you check the **Save credentials locally** option, you will not be asked to input your credentials every time you connect.

2) Input your **Username** and **Password** associated with your database.

3) Click the **Test Connection** button or attempt to connect to the database.

4) A window for the **Secondary Password** will appear. Enter your secondary password.



5) If all details are correct, DBeaver will establish a connection using both your primary and secondary passwords.

Netezza

Table of contents

[Overview](#)

[Netezza specialty](#)

[Setting Up](#)

[Netezza connection settings](#)

[Connection details](#)

[Netezza driver properties](#)

[ODBC and JDBC driver configuration](#)

[Secure Connection Configurations](#)

[Powering Netezza with DBeaver](#)

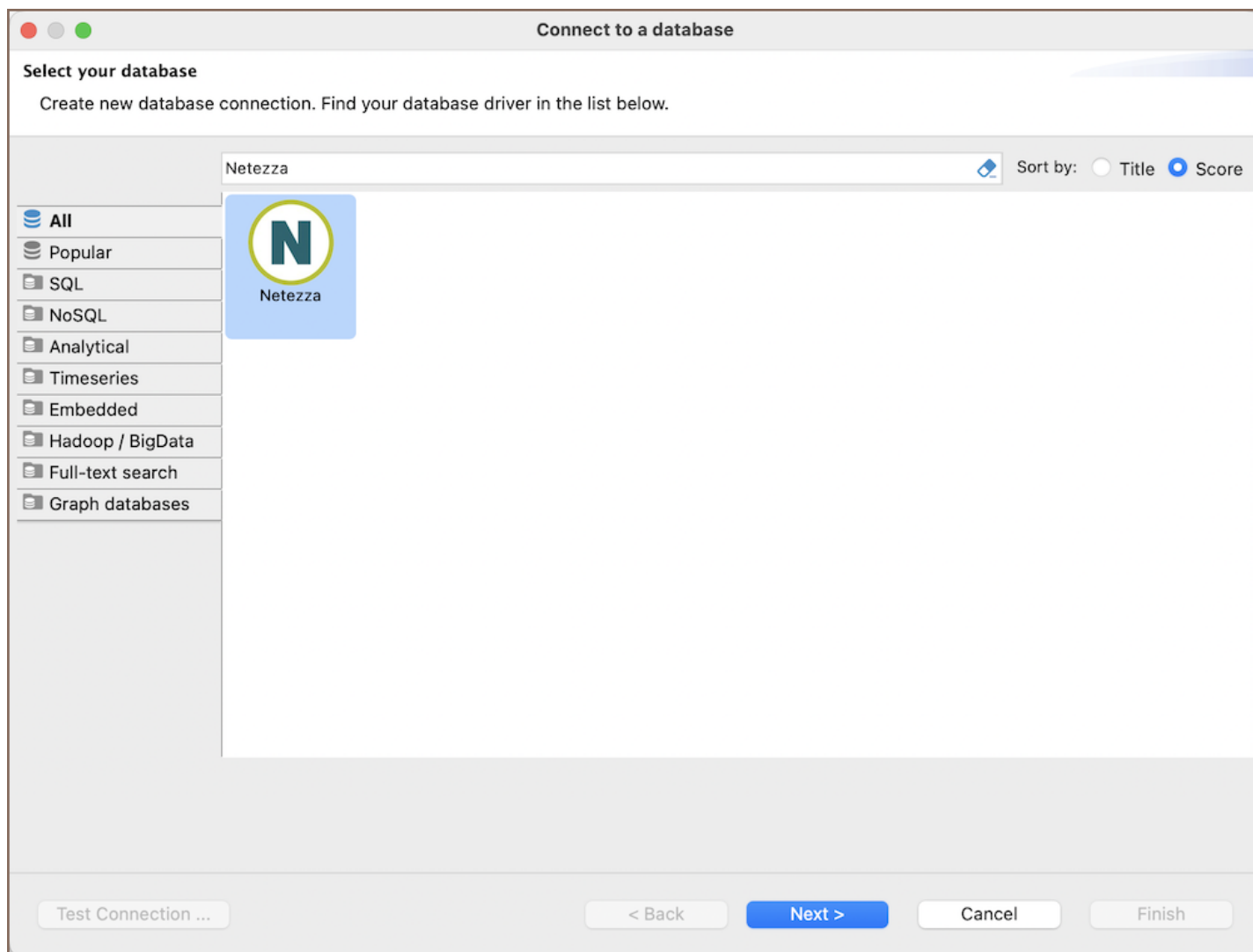
[Netezza database objects](#)

[Netezza features in DBeaver](#)

Overview

This guide provides instructions on how to set up and use Netezza with DBeaver.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **Netezza** option to connect to the Netezza database. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Netezza specialty

Netezza is a high-performance data warehousing appliance that excels in delivering fast data analytics and warehousing solutions through its massively parallel processing (MPP) architecture. This system is tailored for complex analytics on large datasets, often scaling to petabytes of data. Netezza simplifies analytics by allowing operations directly on compressed data, significantly reducing the storage requirements and enhancing query performance.

The integrated architecture of Netezza, which combines database, server, and storage into a single, streamlined system, facilitates the easy management and scaling of data warehousing capabilities.

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, Kubernetes, SSL, and the setup of ODBC/JDBC drivers for Netezza.

Netezza connection settings

In this subsection, we will outline the settings for establishing a direct connection to a Netezza database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Netezza database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Netezza Connection Settings

Netezza connection settings

Main

Driver properties

SSH

+ Network configurations...

General

Connect by:

Host

URL

Driver type:

JDBC

JDBC URL:

jdbc:netezza://localhost:5480/DBeaver

Host:

localhost

Port:

5480

Database/Schema:

DBeaver

Authentication

Authentication:

Database Native

Username:

test_name

Password:

.....

Save password

Advanced Settings

Show all databases

Use Owners for table structure view

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:

Netezza

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your Netezza database.

DBeaver User Guide 24.1.ea. Page 329 of 1174.

URL	If you are connecting via URL, enter the URL of your Netezza database here. This field is hidden if you are connecting via the host.
Host	If you are connecting via host, enter the host address of your Netezza database here.
Database/Schema	Enter the name of the Netezza database/schema you want to connect to.
Port	Enter the port number for your Netezza database. The default Netezza port is <code>5480</code> .
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Native Database Authentication - DBeaver Profile Authentication ★ <p>You can also read about security in DBeaver PRO.</p>
Show all databases ★	<p>Displays all databases in the Database Navigator.</p> <p>Note: This option works only when the Host connection type is selected. It does not work when connecting via URL.</p>
Use Owners for table structure view ★	Show fully qualified table names in Database Navigator with owners names instead schemas.
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaver allows for further customization of your Netezza connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)

- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Netezza driver properties

The settings for Netezza **Driver properties** enable you to adjust the performance of the Netezza JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your Netezza database.

For a complete walkthrough on setting up Netezza JDBC and ODBC drivers properties, you can refer to the official [Netezza JDBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Netezza database connections.

You can customize the Netezza driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your Netezza database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC driver configuration

DBeaver provides extensive capabilities for managing Netezza database connections via ODBC/JDBC drivers. This functionality enables you to connect to your Netezza database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

You can find a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver in our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your Netezza database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering Netezza with DBeaver

DBeaver provides a host of features designed for Netezza databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

Netezza database objects

DBeaver lets you view and manipulate a wide range of Netezza database objects. DBeaver has extensive support for various Netezza metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Schemas
 - Tables
 - Unique Keys
 - Primary Keys
 - Foreign Keys
 - Views
 - Procedures
 - Sequences ★

● Synonyms ★

● Users ★

● Data types

Netezza features in DBeaver

DBeaver is not limited to typical SQL tasks. It also includes numerous additional features compatible with Netezza, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management ★	Session Manager
Schema Management ★	Schema Compare
Data Visualization	ERD Guide
	Query Execution plan ★
Data Generation	Mock Data Generation ★

Oracle

Table of contents

[Overview](#)

[Setting Up](#)

[Oracle connection settings](#)

[Connection details](#)

[Oracle driver properties](#)

[Secure Connection Configurations](#)

[Powering Oracle with DBeaver](#)

[Oracle database objects](#)

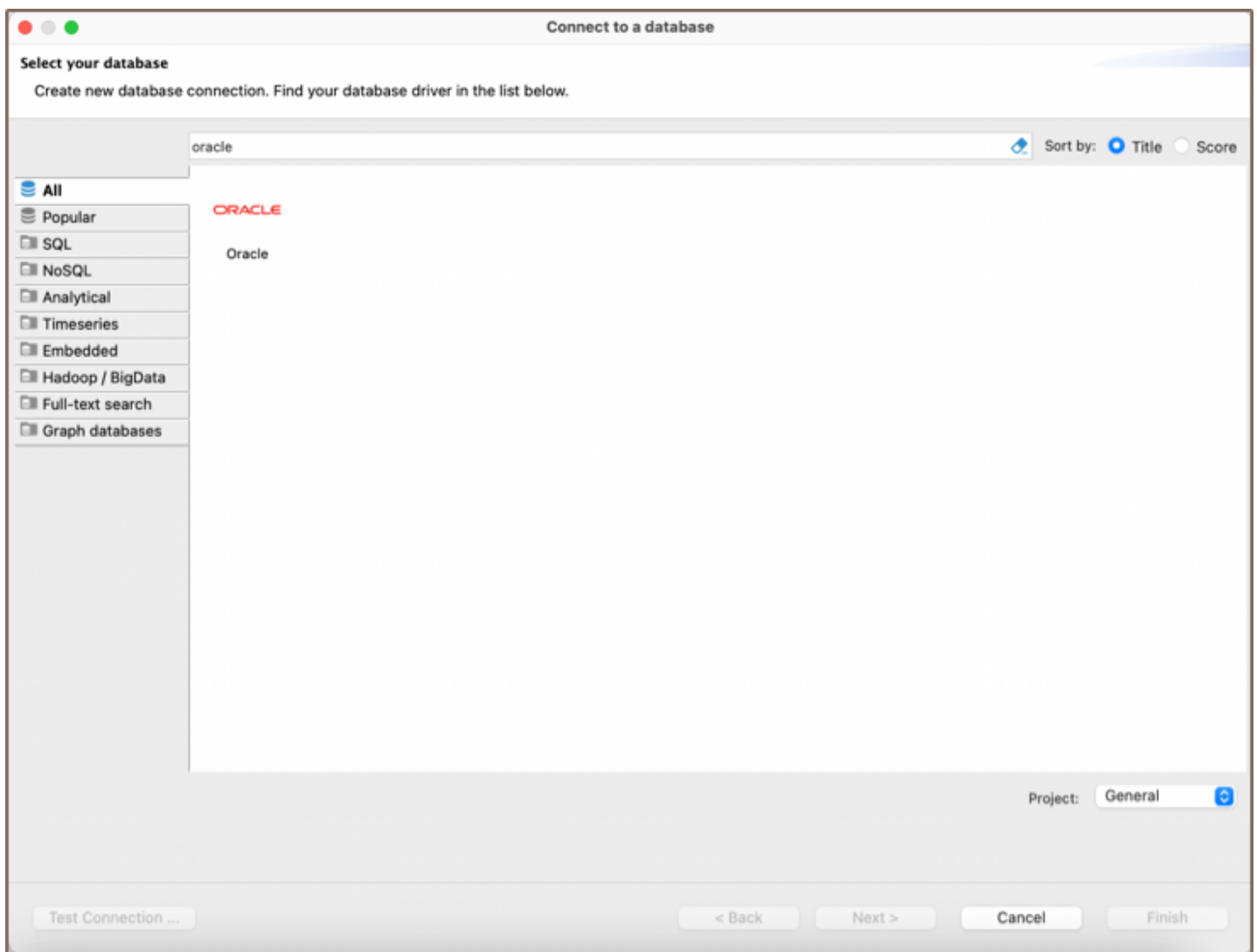
[Oracle Features in DBeaver](#)

Overview

This guide provides instructions for setting up and using Oracle with DBeaver.

Before you start, you must create a connection in DBeaver and select Oracle. If you haven't done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the Oracle server using a specific driver. It also supports Oracle cloud database extensions such as Oracle Cloud JSON and Oracle NetSuite, both of which can be connected through [Cloud Explorer](#).



Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, and the driver setup for Oracle.

Oracle connection settings

In this subsection, we'll outline the settings for establishing a direct connection to a Oracle database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Oracle database.

1) The first page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Oracle Connection Settings

Oracle connection settings

ORACLE

MainOracle propertiesDriver propertiesSSH

+ Network configurations...

Connection Type:

BasicTNSCustom

Host:localhost

Port:1521

Database:ORCL

Service Name

Authentication

Authentication:Oracle Database Native

Username:system

Role:Normal

Password:

Save password

Client:<not present>

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: Oracle

Driver Settings

Driver license

Test Connection ...

< Back

Next >

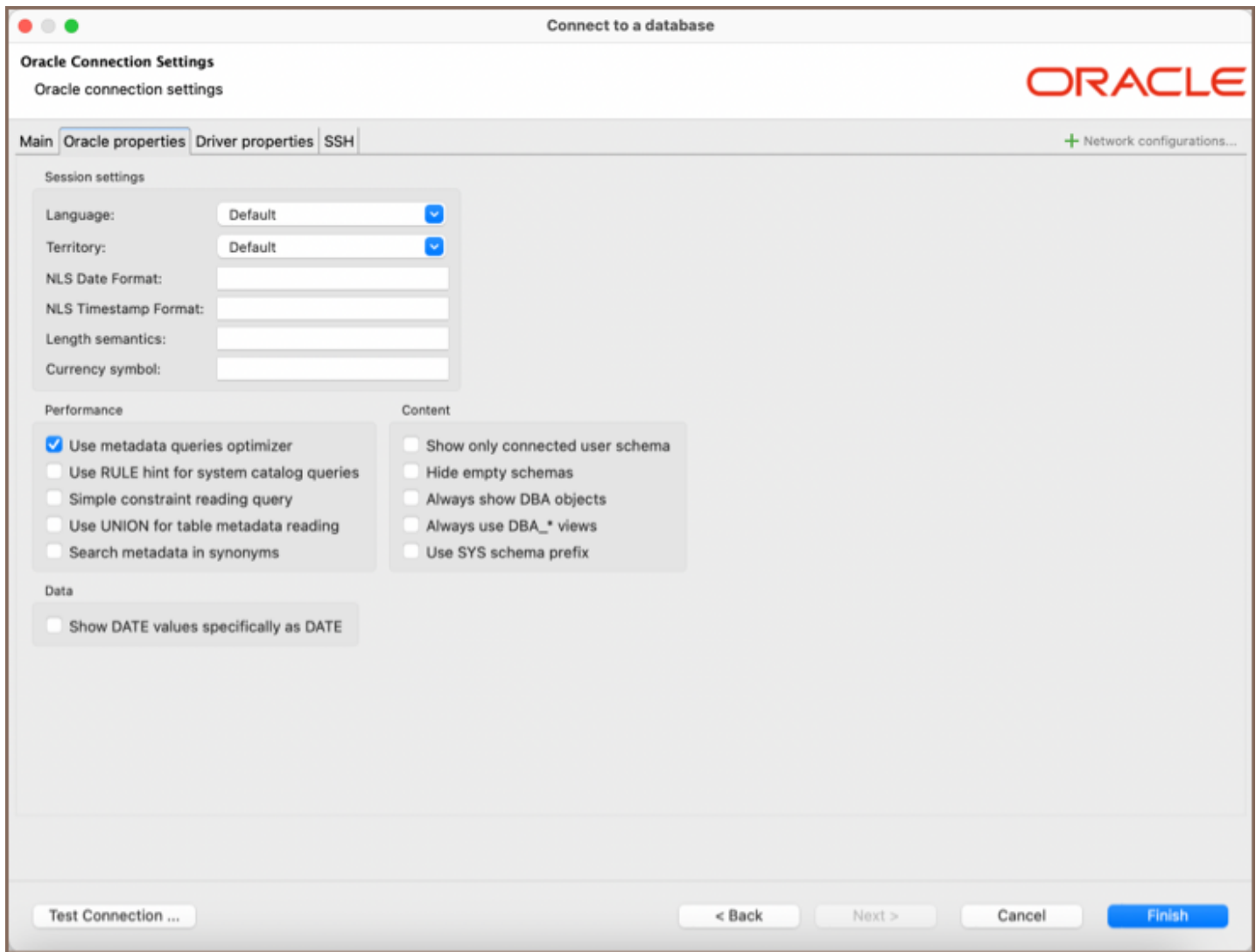
Cancel

Finish

Field	Description
Host	If you're connecting via host, enter the host address of your Oracle database here.
Database	Enter the name of the Oracle database you want to connect to.
Port	Enter the port number for your Oracle database. The default Oracle port is 1521 .
Connection identifiers	Choose whether you want to connect using a Service name or a SID identifier.
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none">- Oracle Database Native- OS Authentication ★- DBeaver Profile Authentication ★- Oracle Kerberos Authentication ★- Oracle Wallet ★

	You can also read about security in DBeaver PRO .
Role	Choose whether you want to connect using a <code>Normal</code> , <code>SYSDABA</code> or a <code>SYSOPER</code> role. For more details, you can refer to the official Administering User Accounts and Security article.
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

2) The second page of the connection settings offers additional options that allow you to customize your further connection to the Oracle database.



Field	Description
Language	Specify the Session Language.

Territory	Specify the Session Territory.
NLS Date Format	Specify NLS (National Language Support) Date Format. The default value of the <code>NLS_DATE_FORMAT</code> parameter is determined by the <code>TERRITORY</code> parameter.
NLS Timestamp Format	Specify NLS Timestamp format. The default value of the <code>NLS_TIMESTAMP_FORMAT</code> parameter is determined by the <code>TERRITORY</code> parameter.
Length semantics	specify the length semantics for <code>VARCHAR2</code> and <code>CHAR</code> table columns, user-defined object attributes, and PL/SQL variables in database objects created in the session. The length semantics can be specified as either <code>BYTE</code> or <code>CHAR</code> .
Currency symbol	Specify the Currency symbol.You can find out which currency symbol your current session uses by querying the <code>V\$NLS_PARAMETERS</code> view.
Show only connected user schema	Show only a scheme of a connected user in the Database Navigator .
Hide empty schemas	Check existence of objects within schema and do not show empty schemas in tree. Enabled by default but it may cause performance problems on databases with very big number of objects.
Always show <code>DBA</code> objects	Always shows DBA-related metadata objects in tree even if user do not has <code>DBA</code> role.
<i>*Always use <code>'DBA_</code> <code>views**</code> Use <code>DBA</code> * <code>views instead of ALL*</code> views wherever it is possible</i>	
Use SYS schema prefix	Use SYS schema prefix in all metadata queries. Otherwise use view names without explicit schema.
Simple constraint reading query	Use simple metadata queries. May work slower but it is more stable for all Oracle versions.
Use UNION for table metadata reading	Use legacy table metadata query. With <code>UNION</code> instead <code>JOIN</code> . It helps in some cases speed up reading of table data.
Search metadata in synonyms	Search for metadata in synonyms among other places. May significantly slow down metadata search as well as autocompletion.
Use RULE hint for system catalog queries	Adds <code>RULE</code> hint for some system catalog queries (like columns and constraints reading).It significantly increases performance on some Oracle databases (and decreases on others).

Show DATE values specifically as DATE	Show DATE data type values specifically as DATE , not as a TIMESTAMP .This setting will not work with disabled date/time formatting.
Use metadata queries optimizer	Use metadata queries optimizer. May significantly improve metadata reading performance on some systems.

Connection details

The **Connection Details** section in DBeaver allows to customize your experience while working with Oracle database. This includes options for adjusting the **Navigator View**, setting up **Security** measures, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Oracle driver properties

The settings for Oracle **Driver properties** enable you to adjust the performance of the Oracle driver. These adjustments can influence the efficiency, compatibility, and features of your Oracle database.

For a complete walkthrough on setting up Oracle JDBC driver properties, you can refer to the official [Oracle JDBC documentation](#). These guide detail each driver's properties and how they can be used to optimize Oracle database connections.

You can customize the Oracle driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your

Oracle database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your Oracle database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering Oracle with DBeaver

DBeaver provides a host of features designed for Oracle databases. This includes the ability to view and manage databases, along with numerous unique capabilities aimed at optimizing database operations.

Oracle database objects

DBeaver lets you view and manipulate a wide range of Oracle database objects. DBeaver has extensive support for various Oracle metadata types, allowing you to interact with a wide variety of database objects, such as:

- Schemas
 - Tables
 - Columns
 - Constraints

- Foreign Keys
- References
- Triggers
- Indexes
- Partitions
- Dependencies
- Views
- Materialized Views
- Indexes
- Sequences
- Queues
- Types
- Packages
- Procedures
- Functions
- Synonyms
- Schema Triggers
- Table Triggers
- Database Links
- Java
- Jobs
- Scheduler

- Jobs
- Programs
- Recycle bin
- Global metadata
 - Types
 - Public Synonyms
 - Public Database Links
 - User Recycle bin
- Storage
 - Tablespaces
 - Files
 - Objects
- Security
 - Users
 - Roles
 - Profiles
- Administer
 - Session Manager
 - Locks Manager

Oracle Features in DBeaver

DBeaver isn't limited to typical SQL tasks. It also includes numerous unique features specifically for Oracle. Beyond regular SQL operations, DBeaver provides a range of Oracle-specific capabilities, such as:

--	--

Category	Feature
Data Types	Oracle Nested Tables
PL/SQL Support	PL/SQL Procedures
	PL/SQL Functions
Security	Oracle Permissions
	Oracle Profiles
	Oracle Roles
Data Organization	Oracle Partitions
Database Management	Oracle Dependencies
Performance Tuning	Oracle Performance Reports
	Oracle Execution Plans

Additional features compatible with Oracle, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Backup and Restore	How to Backup/Restore data
Schema Management	Schema Compare
Data Visualization	GIS Guide
	ERD Guide

Oracle authentication models

Table of contents

[Configuration types](#)

[Basic connections](#)

[TNS](#)

[Custom URL](#)

[Authentication](#)

[Database](#)

[OS authentication](#)

[Oracle Wallet](#)

[Kerberos](#)

[Oracle Cloud connections](#)

[Plain URL connection](#)

[Oracle Wallet connection](#)

There are several ways to configure a database connection and several ways to perform an authentication.

Connect to a database

Oracle Connection Settings
Oracle connection settings

Main | Oracle properties | Driver properties | SSH | [+ Network configurations...](#)

Connection Type: Basic | TNS | Custom

Host: localhost Port: 1521
Database: ORCL Service Name

Authentication

Authentication: Oracle Database Native
Username: system Role: Normal
Password: Save password

Client: <not present>

[You can use variables in connection parameters.](#) Connection details (name, type, ...)

Driver name: Oracle Driver Settings Driver license

Test Connection ... < Back Next > Cancel Finish

Configuration types

Basic connections

Host/port based configuration

Parameter	Description	Example
Host	Server host name	192.168.1.25
Post number	Server listener port	1521 (default)
Database	Service or SID name	ORCL
Service/SID	It depends on the server configuration. SID must be selected for some servers and the Service Name for	SID

	others	
--	--------	--

TNS

TNS configuration is the simplest, but it requires you to have the `tnsnames.ora` file somewhere on your disk. `tnsnames.ora` contains information about all accessible Oracle server connections. DBeaver can determine the default location of this file, but sometimes you will need to manually specify it.

Parameter	Description	Example
Network Alias	Name of configuration from <code>tnsnames.ora</code>	ORCL1
TNS names path	Path to <code>tnsnames.ora</code> file. By default, it is got from the <code>TNS_ADMIN</code> environment variable or from Windows' registry	C: \oracle\network\admin

Custom URL

For a more sophisticated configuration, you can specify the full JDBC URL manually (see [Data Sources and URLs](#)).

Sample URL (Oracle Cloud):

```
jdbc:oracle:thin:@(description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1522)(host=ac
```

Authentication

Database

Parameter	Description	Example
User name	Database user name	SYS
Password	Database user password	
Role	Role for connection. Roles SYSDBA and SYSOPER are needed for some administrative	Normal

	operations	
Save password	Saves the user/password information in the local DBeaver configuration	SID

OS authentication

The Oracle driver gets user information from the current OS user.

You do not need to explicitly specify any credentials.

Oracle Wallet

A more secure way to connect is to use the Oracle Wallet. Wallet is a directory with security keys and some other optional connection information. Wallets are usually distributed as ZIP archives. You need to extract the ZIP archive to a folder on a disk and specify this folder in the `Wallet location` field.

Wallet may contain information about a database user. This, however, is optional. You will sometimes need to specify the user too.

Wallet may also contain a TNS configuration. If it does, you can use the TNS connection configuration easily by setting the `TNS path` to the same value as the `Wallet location`.

Parameter	Description	Example
User name, Password, Role	See Database authentication	
Wallet location	Oracle wallet directory	C:\oracle\network\wallet\example
Wallet password	Optional. Some wallets are password-protected	

Kerberos

Kerberos is the most complicated authentication in Oracle.

Parameter	Description	Example
-----------	-------------	---------

Username	Database user name	c##testuser
Kerberos user	Kerberos / Active directory user name	testuser@THE-REALM
Realm	Kerberos realm	THE-REALM
KDC server	KDC server address	krb5.your-domain.com
Password	Kerberos user password	

Oracle Cloud connections

DBeaver supports Oracle Cloud Autonomous databases connectivity.

There are two ways to authenticate:

Plain URL connection

- To use a plain URL connection you must enable the `Access control list` for the Oracle autonomous database.
- Then add your IP address to the IP list.
- Use the Custom connection configuration ([URL](#)). You can copy the URL from the Oracle Cloud database page (link "DB Connection").

Oracle Wallet connection

It is the default authentication type for the Oracle Cloud.

- Download Wallet from the Oracle Cloud website
- Expand the wallet archive to a folder
- Set [TNS](#) configuration type
- Set the `TNS path` to the wallet location directory

- Choose the proper `Network Alias` from the drop-down menu
- Set `Authentication` to the Oracle Wallet
- Set the database username and password (you can get them from the Oracle Cloud database information page)
- Set the `Wallet location` to the wallet location directory

PostgreSQL

Table of contents

[Overview](#)

[Setting Up](#)

[PostgreSQL connection settings](#)

[Connection details](#)

[PostgreSQL driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering PostgreSQL with DBeaver](#)

[PostgreSQL database objects](#)

[PostgreSQL Features in DBeaver](#)

Overview

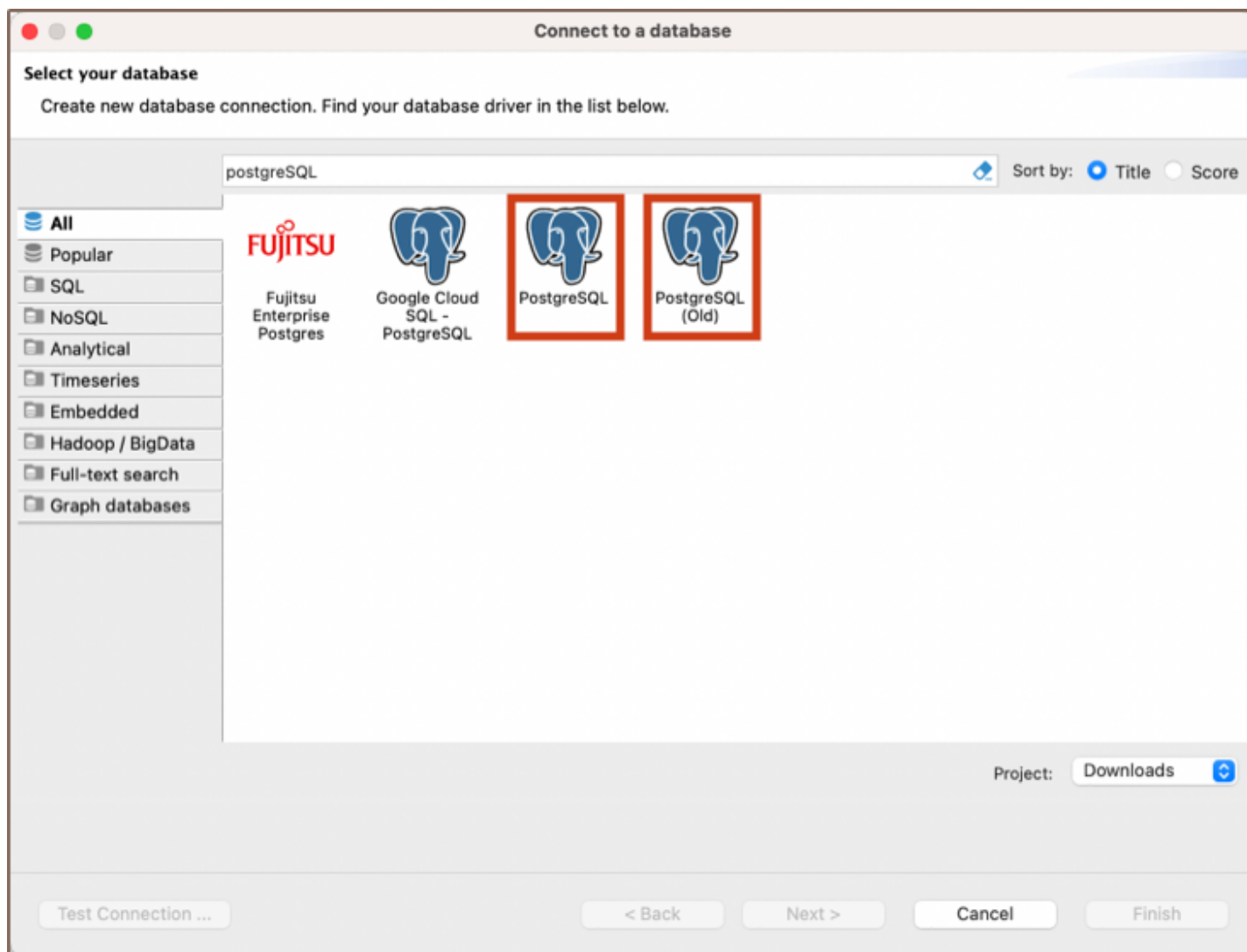
This guide provides instructions on how to set up and use PostgreSQL with DBeaver.

DBeaver offers extensive support for PostgreSQL, but it might not cover all possible functionalities. If DBeaver is missing a feature you need, please submit a request through our [GitHub page](#). If you are a developer, you have the option to enhance DBeaver's capabilities by contributing your own code. For guidelines on how to do this, check out our [Contribution Guide](#).

Before you start, you must create a connection in DBeaver and select PostgreSQL. If you have not done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the PostgreSQL server using a specific driver, it supports all versions of PostgreSQL, but the correct driver must be selected: use **PostgreSQL (old)** for versions below 8.x and **PostgreSQL** for newer versions.

DBeaver also supports PostgreSQL extensions such as **EDB**, **Google Cloud SQL for PostgreSQL** and **Fujitsu Enterprise Postgres**. You must select the appropriate driver in the **Connect to a database** window for these extensions.



Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, SSL, and the setup of ODBC/JDBC drivers for PostgreSQL.

PostgreSQL connection settings

In this subsection, we will outline the settings for establishing a direct connection to a PostgreSQL database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your PostgreSQL database.

1) The first page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Connection Settings

PostgreSQL connection settings

PostgreSQL

Main

PostgreSQL

Driver properties

SSH

SSL

+ Network configurations...

Server

Connect by:

☒ Host
☐ URL

Driver type:

JDBC

URL:

jdbc:postgresql://localhost:5432/postgres

Host:

localhost

Port:

5432

Database:

postgres

☐ Show all databases

Authentication

Authentication:

Database Native

Username:

postgres

Password:

•••••

☒ Save password

Advanced

Session role:

Local Client:

/Library/PostgreSQL/16

[You can use variables in connection parameters.](#)

Connection details (name, type, ...)

Driver name:

PostgreSQL

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your PostgreSQL database.
URL	If you are connecting via URL, enter the URL of your PostgreSQL database here. This field is hidden if you are connecting via the host.
Host	If you are connecting via host, enter the host address of your PostgreSQL database here.
Database	Enter the name of the PostgreSQL database you want to connect to.
Show all	Displays all databases in the database navigator . Note: This option works only when the Host connection type is selected. It does not work when connecting

DBeaver User Guide 24.1.ea. Page 352 of 1174.


databases	via URL .
Port	Enter the port number for your PostgreSQL database. The default PostgreSQL port is 5432 .
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Native Database Authentication - AWS RDS IAM Authentication ★ - Azure AD Authentication for PostgreSQL ★ - DBeaver Profile Authentication ★ - PostgreSQL Kerberos Authentication ★ - PostgreSQL PgPass Authentication - PostgreSQL SSPI Authentication (Windows SSO) ★ <p>You can also read about security in DBeaver PRO.</p>
Local Client	Specify the local client, see our article about Local client.
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

2) The second page of the connection settings offers additional options that allow you to customize your further connection to the PostgreSQL database.

Connect to a database

Connection Settings

PostgreSQL connection settings

 PostgreSQL

Main
PostgreSQL
Driver properties
SSH
SSL

+ Network configurations...

Settings

☐ Show template databases
☐ Show databases not available for connection
☐ Show database statistics
☐ Read all data types
☐ Read table keys with columns

SQL

Show \$\$ quote as: Code block

Show \$tagName\$ quote as: Code block

Performance

☐ Use prepared statements

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Show template databases	Displays template databases. This option is available when Show all databases is checked.
Show databases unavailable for connection	Show databases not available for connection in database list. This option is available when Show all databases is checked.
Show databases statistics	Displays statistics for databases.
Read all data types	Reads all data types from the server (including arrays).
Read table keys with	Enable reading table constraints during the column reading stage to ensure accurate recognition of

columns	keys. Note that this setting may decrease metadata loading performance.
Show <code>\$\$</code> quote as	Displays <code>\$\$</code> quotes as code blocks or strings.
Show <code>\$tagName\$</code> quote as	Displays <code>\$tagName\$</code> quotes as code blocks or strings.
Use prepared statements	Uses prepared statements for SQL execution.

Connection details

The **Connection Details** section in DBeaver allows for further customization of your PostgreSQL connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

PostgreSQL driver properties

The settings for PostgreSQL **Driver properties** enable you to adjust the performance of the PostgreSQL JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your PostgreSQL database.

For a complete walkthrough on setting up PostgreSQL JDBC and ODBC drivers properties, you can refer to the official [PostgreSQL JDBC documentation](#) and [PostgreSQL ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize PostgreSQL database connections.

You can customize the PostgreSQL driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your PostgreSQL database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing PostgreSQL database connections via ODBC/JDBC drivers. This functionality enables you to connect to your PostgreSQL database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

You can find a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver in our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your PostgreSQL database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering PostgreSQL with DBeaver

DBeaver provides a host of features designed for PostgreSQL databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

PostgreSQL database objects

DBeaver lets you view and manipulate a wide range of PostgreSQL database objects. DBeaver has extensive support for various PostgreSQL metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Schemas
 - Data types
 - Tables
 - Columns
 - Constraints
 - Indexes
 - Foreign Keys
 - Dependencies
 - References
 - Partitions
 - Triggers
 - Rules
 - Policies
 - Foreign Tables
 - Views
 - Materialized Views
 - Functions
 - Sequences
 - Data types

- Aggregate functions
- Event Triggers
- Extensions
- Storage
 - Tablespaces
- Roles
- Administer
 - Jobs
 - Session Manager
 - Lock Manager

For a practical guide on creating new tables and working with various database objects in PostgreSQL, check out our [tutorial](#).

PostgreSQL Features in DBeaver

DBeaver is not limited to typical SQL tasks. It also includes numerous unique features specifically for PostgreSQL. Beyond regular SQL operations, DBeaver provides a range of PostgreSQL-specific capabilities, such as:

Category	Feature
Data Types	PostgreSQL Arrays
	PostgreSQL Structures
Extensions	PostgreSQL Extensions
Security	PostgreSQL Permissions
	PostgreSQL Policies
	PostgreSQL Roles
Data Organization	PostgreSQL Partitions

Database Management	PostgreSQL Dependencies
	PostgreSQL Tools
External Data	PostgreSQL Foreign Tables

Additional features compatible with PostgreSQL, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Lock Management	Lock Manager
Backup and Restore	How to Backup/Restore data
Schema Management	Schema Compare
Data Visualization	GIS Guide
	ERD Guide

Arrays

Table of contents

[Creating Array column](#)

[Creating Array Data](#)

[Modifying and Querying Array Data](#)

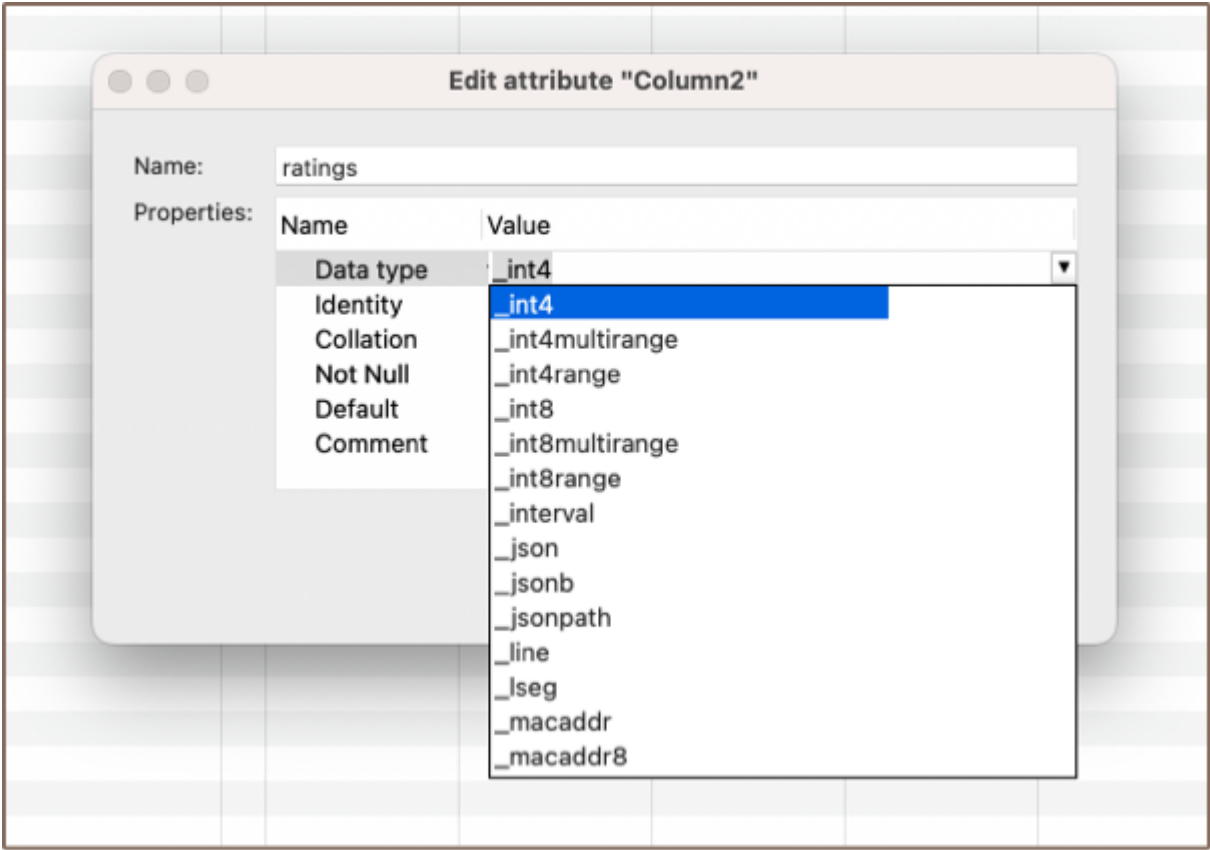
DBeaver offers support for array data types in managing [PostgreSQL](#) databases. This feature enables the storage of multiple values in a single variable, allowing for more efficient and organized data management. A complete list of types is available in the **Edit Attribute** window in the data type selection dropdown when [creating a column](#).

The representation of these array data types in DBeaver requires a specific notation. An underscore `_` is placed before the data type, distinguishing array data types from scalar ones.

Creating Array column

You have the flexibility to create an array column either via the [Properties Editor](#) or the [SQL Editor](#) in DBeaver.

In the **Properties Editor**, navigate to the **Edit Attribute** window. Choose your desired array data type, For an array of integers, such as `_int4` , from the data type selection dropdown.



Alternatively, you can execute a script in the **SQL Editor** to add an array column to an existing table. For example, the following command will add a new integer array column called 'ratings' to your table:

```
ALTER TABLE film ADD ratings _int4 NULL;
```

Creating Array Data

Once the table is created, you can use the SQL script in the **SQL Editor** to insert data into the array column. For example, this script utilizes the array constructor with nested arrays:

```
INSERT INTO film (film_title, ratings)
VALUES ('Inception', array[array[8, 9], array[7, 9], array[8, 10]]);
```

This script, "Inception" has received three sets of ratings. Each set could represent ratings from a different platform or group of users, for example. Each set contains two ratings, in this case.

Modifying and Querying Array Data

Once you've established an array column, DBeaver allows you to modify and query the array data just like you would perform any other operations in PostgreSQL. You can use standard SQL commands like `UPDATE`, `INSERT`, and `SELECT` to manipulate and retrieve data from your array columns. The array data is treated the same way as any other data type in PostgreSQL, granting you all the flexibility of SQL operations.

Extensions

Table of contents

[Install Extensions](#)

[Database Navigator](#)

[SQL Editor](#)

[Using Extensions](#)

[Deleting Extensions](#)

DBeaver offers support for [PostgreSQL](#) extensions. These tools allow you to customize and extend your database's functionality beyond its default capabilities. From adding new data types and functions to enhancing performance and database operations, PostgreSQL extensions offer a range of possibilities.

The ability to install, use, and delete PostgreSQL extensions within DBeaver provides a seamless and integrated experience, making it simpler for you to leverage the full potential of your PostgreSQL database system.

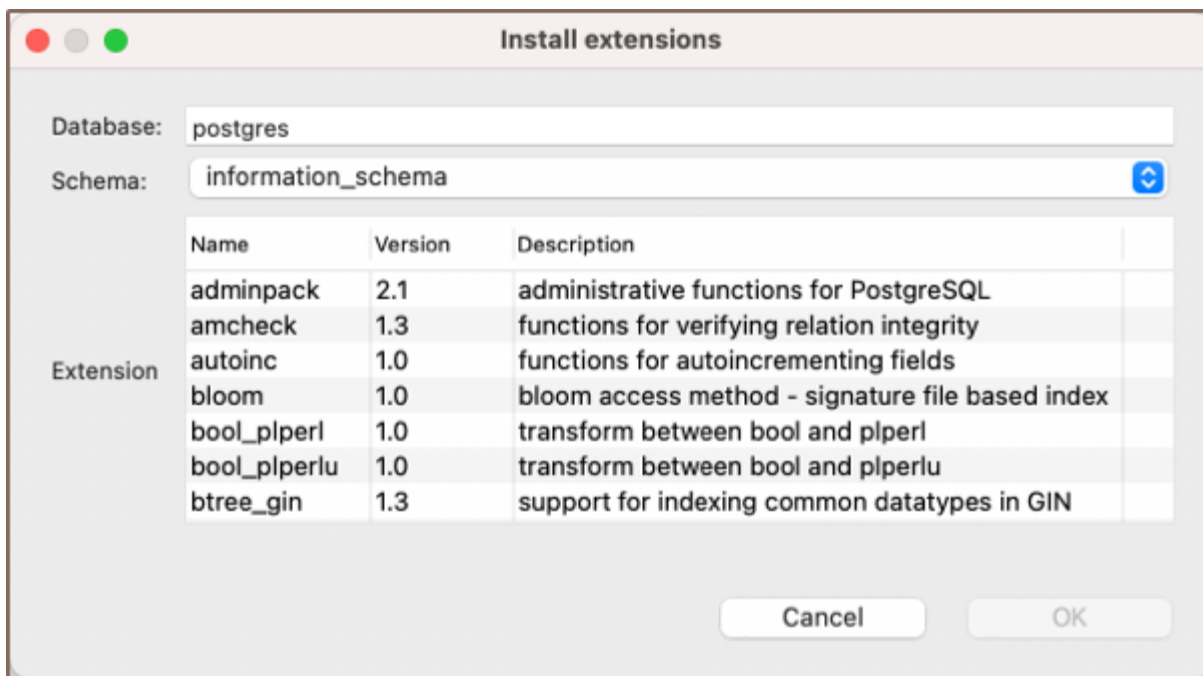
Install Extensions

With DBeaver, you have two ways to install PostgreSQL Extensions - through the [Database Navigator](#) and using the [SQL Editor](#).

Note: Be sure to have the appropriate permissions to perform the installation. The ability to add extensions typically requires database administrative privileges. Without these, you may encounter errors during the installation process.

Database Navigator

Navigate to the **Database Navigator** -> **Databases** -> **Extensions**. Right-click on the **Extensions** and select **Create New Extension**. In the opened dialog box, you can select the **Schema**, required **Extension** from the list and confirm the installation.



SQL Editor

Alternatively, you can install an extension directly using a SQL command. Open the SQL Editor, and type the appropriate SQL command to create the extension. For example, to install the "uuid-oss" extension, you would run:

```
CREATE EXTENSION IF NOT EXISTS "uuid-oss";
```

Then, [execute](#) the command. DBeaver will run this command on your PostgreSQL server to install the extension.

Using Extensions

After installing the extension, its functionality becomes available. New functions, data types, operators, or other features may be available in your SQL queries depending on the extension.

For example, if you have the `uuid-oss` extension installed, you can use the `uuid_generate_v4()` function to generate universally unique identifiers (UUID). It can be used directly in SQL queries or as a default value when creating a new table.

Deleting Extensions

Deleting an extension in DBeaver is as simple as installing one. Navigate to the **Extensions** in the **Database Navigator**, right-click on the extension you wish to remove, and select **Delete**. Confirm the deletion in the dialog box that appears.

Important: Dropping an extension will remove all functionalities and data types associated with it. Ensure you have a clear understanding of the impact on your database before performing this operation.

PostgreSQL Pgpss

Table of contents

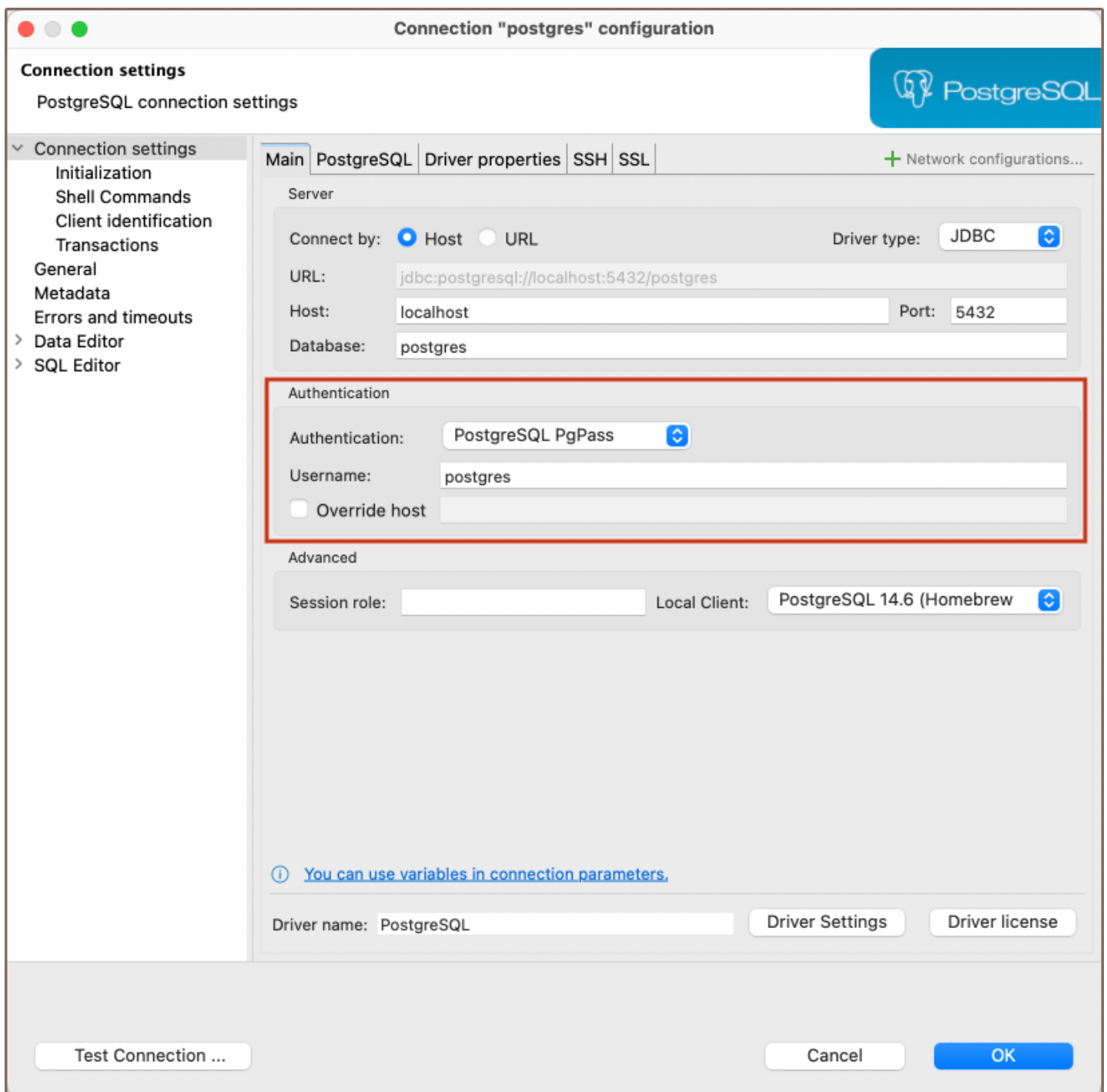
[Overview](#)

[Settings](#)

[Override host](#)

Overview

DBeaver is fully compatible with the PgPass authentication method, offering you a secure way to access your [PostgreSQL](#) databases.



Settings

PostgreSQL PgPass method allows you to authenticate using credentials stored in a `.pgpass` file. Here are the steps to configure this:

- 1) From the **Authentication** dropdown menu, select the **PostgreSQL Pgpass** method.
- 2) Enter the username that is associated with the database in the **Username** field.

3) DBeaver will use the `.pgpass` file for authentication. If you haven't created this file yet, follow the official [PostgreSQL documentation](#) to create it, format it correctly, and understand where it should be located on your system.

Note: Ensure that the permissions for your `.pgpass` file are set correctly. The file should only be readable and writable by the file owner. Improper permissions can lead to security issues.

4) Once your `.pgpass` file is set up as per the PostgreSQL documentation, you can test the connection from within the **Test Connection** button. If everything is configured correctly, DBeaver should now be able to connect to your PostgreSQL database using the PgPass authentication method.

Override host

DBeaver gives you the option to override the default host. If you wish to do this, select the checkbox next to **Override host** and then input the new host information in the field that appears.

Tip: This is useful in situations where you need to connect to a different server without changing the `.pgpass` file, or if the `.pgpass` file is shared among several users or applications with different host requirements.

Authentication PostgreSQL SSPI

Table of contents

[Overview](#)

[Settings](#)

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Overview

DBeaver comes with SSPI (Security Support Provider Interface) authentication support, allowing secure access to [PostgreSQL](#) databases. This authentication type enables Windows Single Sign-On (SSO), allowing users to authenticate using their existing Windows account credentials.

Connection "postgres" configuration

PostgreSQL connection settings

Connection settings

- Initialization
- Shell Commands
- Client identification
- Transactions
- General
- Metadata
- Errors and timeouts
- > Data Editor
- > SQL Editor

Main PostgreSQL Driver properties SSH SSL + Network configurations...

Server

Connect by: ☒ Host ☐ URL Driver type: JDBC

URL: jdbc:postgresql://localhost:5432/postgres

Host: localhost Port: 5432

Database: postgres

Authentication

Authentication: PostgreSQL SSPI

Username: postgres

Password: [masked] ☒ Save password

Advanced

Session role: Local Client: PostgreSQL 14.6 (Homebrew)

[You can use variables in connection parameters.](#)

Driver name: PostgreSQL Driver Settings Driver license

Test Connection ... Cancel OK

Settings

To configure SSPI authentication in DBeaver, follow these steps:

1) From the **Authentication** dropdown menu, select the **PostgreSQL SSPI** method. 2) For the authentication process, your system will utilize the relevant credentials automatically. On Windows systems, your current login credentials are used, and on non-Windows platforms that support GSSAPI, the integrated authentication system will apply.

Note: Your system (whether it's Windows or a non-Windows platform with GSSAPI support), and PostgreSQL server must be correctly configured for SSPI authentication. If you encounter any issues during this process, please follow the official [PostgreSQL documentation](#). 3) Click **Test Connection** to confirm your settings. If the test is successful, your setup is complete.

Redis

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Redis specialty](#)

[Setting Up](#)

[Redis connection settings](#)

[Connection details](#)

[Secure Connection Configurations](#)

[Redis Features in DBeaver](#)

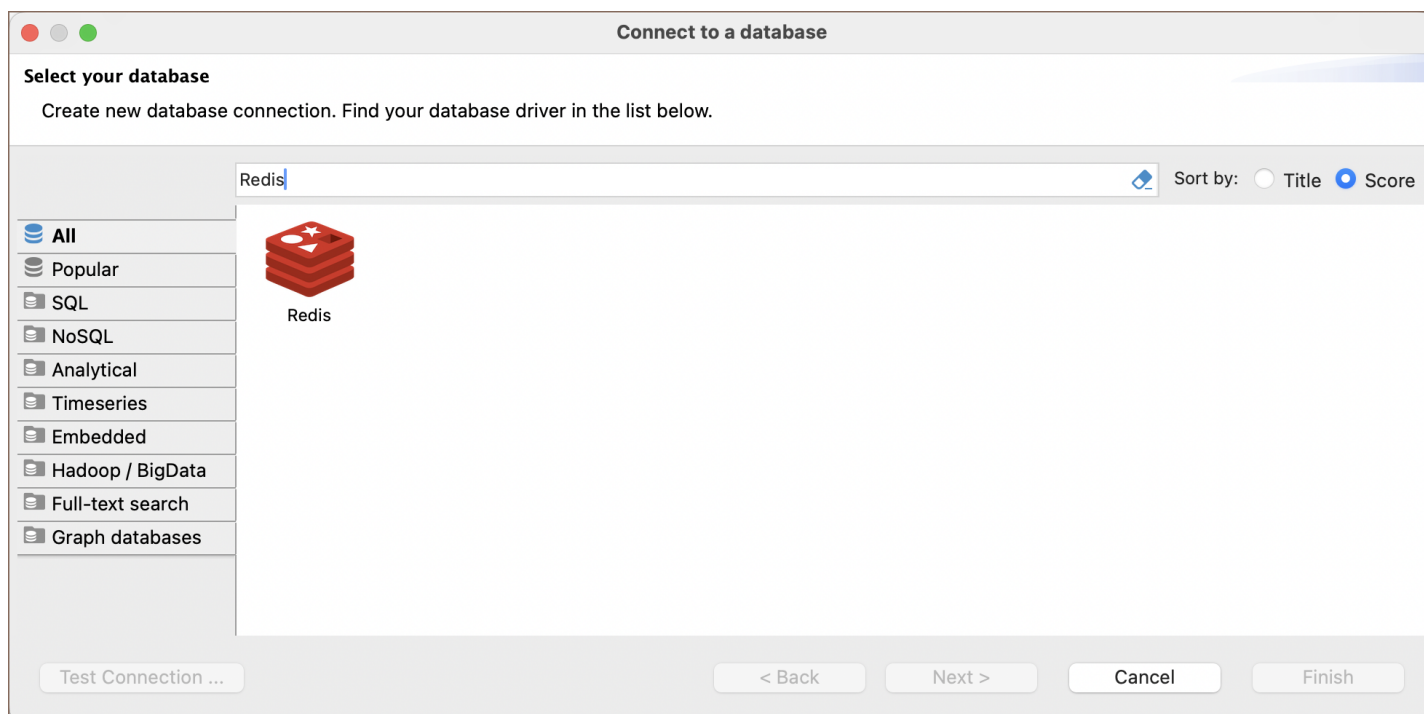
[Browsing keys](#)

[Executing commands](#)

Overview

This guide provides instructions on how to set up and use Redis with DBeaver.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **Redis** option to connect to the Redis. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Redis specialty

Redis is an open-source, in-memory data structure store, used as a database, cache, and message broker. It supports various data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperloglogs, geospatial indexes, and streams. Redis has built-in replication, Lua scripting, LRU eviction, transactions, and different levels of on-disk persistence. Additionally, Redis offers high availability via Redis Sentinel and automatic partitioning with Redis Cluster.

Redis is particularly well-suited for real-time applications like leaderboards, session stores, real-time analytics, and geospatial services due to its fast performance and support for complex data types and operations. For detailed information on Redis and its features, please refer to the [official documentation](#).

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, Kubernetes and SSL.

Redis connection settings


In this subsection, we will outline the settings for establishing a direct connection to a Redis database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Redis database.

1) The first page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Redis Connection Settings

Redis connection settings



Main

Keys settings

SSH

SSL

+ Network configurations...

Server

Type: ☒ Manual ☐ URL

Host: localhost Port: 6379


Database: 0

Deployment: Standalone

Authentication

Authentication: Database Native

Username:


Password: ☒ Save password 

Connection settings

Connection settings

Connect timeout: 10000 Socket timeout: 100000

☒ Send client name

 You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: Redis

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Manual/URL)	Choose whether you want to connect using a host or a URL.
URL	If you are connecting via URL, enter the URL of your Redis database here. This field is disabled if you're connecting via the host.
Host	If you are connecting via host, enter the host address of your Redis database here.
Database	Enter the database number you want to connect to.
Port	Enter the port number for your Redis database. The default Redis port is 6379 .
Deployment	Select the type of Redis deployment for your connection. Options include:


	<ul style="list-style-type: none"> - Standalone: A single-node Redis deployment.
	<ul style="list-style-type: none"> - Redis Cluster: A multi-node Redis deployment for higher performance and fault tolerance.
	<ul style="list-style-type: none"> - Redis Sentinel: A system designed to help manage Redis instances, providing high availability and monitoring services.
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Native Database Authentication - DBeaver Profile Authentication - Redis Sentinel (coming soon) <p>You can also read about security in DBeaver PRO.</p>
Connection settings	Customize connection settings:
	<p>Connect timeout: Specify the duration, in milliseconds, the application waits to establish a connection before it times out. <code>10000</code> milliseconds is the default value.</p>
	<p>Socket timeout: Set the time, in milliseconds, the application waits for a server response to a query. The default setting here is <code>100000</code> milliseconds.</p>
	<p>Send client name: Check this option to transmit the client's name to the server when connecting. Some Redis servers do not support this feature, which may cause the connection attempt to fail.</p>
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings, configure them here.

2) The second page of the connection settings offers additional options that allow you to customize the keys settings.

Connect to a database

Redis Connection Settings

Redis connection settings



Main
Keys settings
SSH
SSL

+ Network configurations...

Keys

Key group divider: :

Max keys read: 10000

Max cache read: 100000

Key filters:

+

-

Test Connection ...

< Back

Next >

Cancel

Finish

Setting	Description	Default Value
Key group divider	Determines the delimiter used to group keys in the keyspace.	:
Max keys read	Limits the number of keys to read.	10000
Max cache read	Specifies the number of keys to read with pattern (cache).	100000
Key filters	Allows you to define patterns to include or exclude certain keys.	*

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with Redis database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)

- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Secure Connection Configurations

DBeaver supports secure connections to your Redis database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Redis Features in DBeaver

DBeaver is not limited to typical SQL tasks. It also includes numerous additional features compatible with Redis, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Task Management	Task Management

Browsing keys

Redis keys are often viewed and edited as a simple list. However, when dealing with a large number of keys, which can run into millions or billions, a list view is not always practical or possible.

DBeaver offers a hierarchical view of Redis keys. Although Redis itself does not inherently support hierarchical data structures, it is common practice to organize keys into groups using specific delimiters like commas, dashes, or colons. DBeaver leverages these naming patterns to display keys in a hierarchical format. You can configure the group separator in the [connection properties](#).

The key browser feature is useful, but for databases with a large key count, locating a specific key through the navigator might be challenging. In such cases, using the SQL editor with Redis commands is the most effective method to manage keys.

Executing commands

Redis operates differently from traditional SQL databases. It uses built-in commands and LUA scripting for operations.

To execute Redis commands, use the syntax similar to the Redis command line shell:

```
SET mykey "Hello"
```

This command will set the value of `mykey` to `Hello`. To run this command in DBeaver, you would type it into the SQL editor and execute it using either `CTRL+Enter` or `ALT+X`.

To learn more about Redis commands, refer to the [official Redis commands documentation](#).

For LUA script execution, enclose the script in curly braces `{}` and execute it as a single statement. If your LUA script includes empty lines or special characters, highlight the script text before executing it. For instance:

```
{
  -- Retrieve the value of a Redis key
  local key = 'key1038' -- Specify the key you want to retrieve
  local value = redis.call('GET', key)
  return value
}
```

For LUA scripting in Redis, see [LUA scripts in Redis](#).

Salesforce

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Setting Up](#)

[Salesforce connection settings](#)

[Connection details](#)

[Salesforce driver properties](#)

[Secure Connection Configurations](#)

[Powering Salesforce with DBeaver](#)

[Salesforce database objects](#)

[Salesforce Object Query Language](#)

[SOQL vs SQL](#)

[Examples](#)

[Simple Select](#)

[Select with Filtering](#)

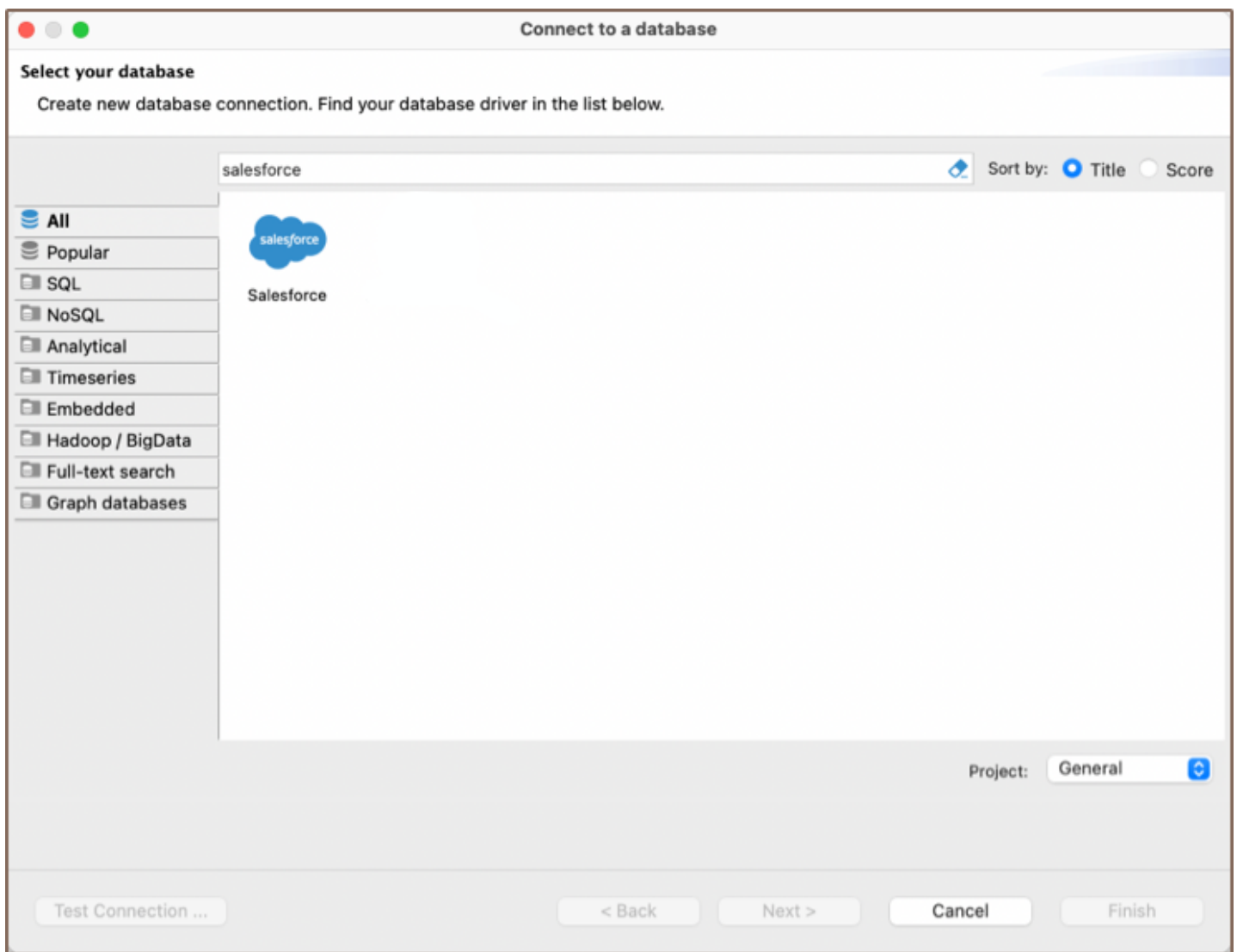
[Select with Grouping](#)

Overview

This guide provides instructions on how to set up and use Salesforce with DBeaver.

Before you start, you must create a connection in DBeaver and select Salesforce. If you have not done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the Salesforce server using a specific driver. To use it, choose the right Salesforce driver in the Connect to a database window.



Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH and proxies.

Salesforce connection settings


In this subsection, we will outline the settings for establishing a direct connection to a Salesforce database using DBeaver.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Generic JDBC Connection Settings

Salesforce connection settings



Main

Driver properties

SSH

+

Network configurations...

General

Connect by:

☒ Host
 ☐ URL

Driver type:

JDBC

JDBC URL:

jdbc:dbeaver:salesforce:login.salesforce.com

Host:

login.salesforce.com

Port:

443

Authentication

Authentication:

Salesforce authentication

Username:

Password:

☒ Save password

Security token:

When you access Salesforce from an untrusted IP address you need a security token to log in

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:

Salesforce

Driver Settings

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your Salesforce database.
URL	If you are connecting via URL, enter the URL of your Salesforce database here. This field is disabled if you're connecting via the host.
Host	If you are connecting via host, enter the host address of your Salesforce database here.
Port	Enter the port number for your Salesforce database. The default Salesforce port is 443 .
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - DBeaver Profile Authentication - Salesforce authentication

	You can also read about security in DBeaver PRO .
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with Salesforce database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Salesforce driver properties

Salesforce is not a traditional Relational Database Management System (RDBMS), and as such, it does not come with a standard JDBC driver. DBeaver uses REST API for connecting to the Salesforce server.

There are no particular properties for configuration at this time. The driver is designed with the current functionality of the Salesforce platform in mind, focusing solely on providing the ability to read data.

Secure Connection Configurations

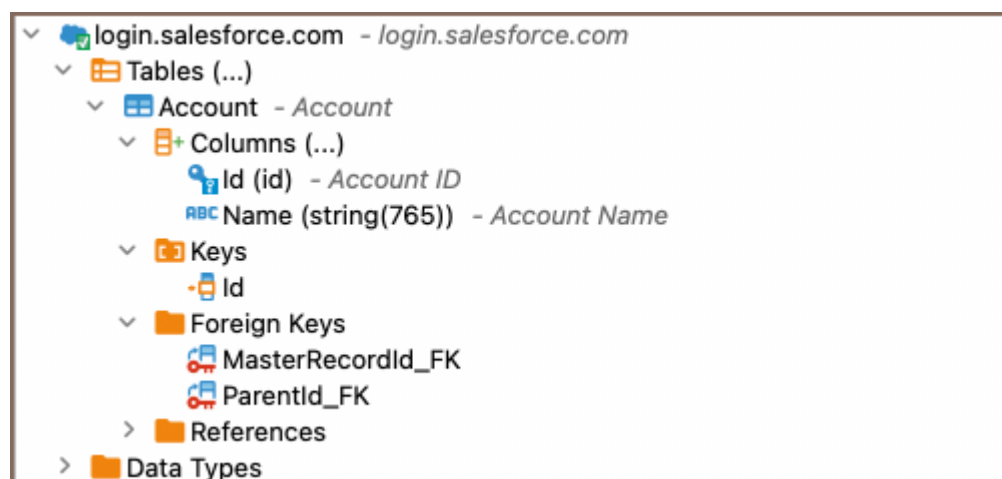
DBeaver supports secure connections to your Salesforce database. Guidance on configuring such connections, specifically **SSH Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering Salesforce with DBeaver

At the current version of DBeaver, the Salesforce database driver allows read-only access to its data, enabling you to execute **SELECT** queries to view data. This includes the ability to use [filters](#), [groupings](#), [data export](#), [ER diagrams](#) and other functionalities of DBeaver for viewing and analyzing the data.

Salesforce database objects



DBeaver lets you view Salesforce database objects and has support for a limited number of Salesforce metadata types, allowing you to interact with a select variety of database objects, such as:

- Tables
 - Columns
 - Keys
 - Foreign Keys
 - References
- Data Types

Salesforce Object Query Language

SOQL is a query language used to search Salesforce data. Within DBeaver, you can utilize SOQL to execute read-only `SELECT` queries on Salesforce data.

SOQL vs SQL

SOQL is specifically tailored for querying Salesforce data and has some distinct differences from traditional SQL:

- Read-Only Queries: SOQL is used primarily for querying data, not for data manipulation. In Salesforce, data insertion and manipulation are typically handled through the platform's user interface, APIs, or specialized data loading tools, rather than through SOQL.
- Limited `JOIN` Operations: Unlike SQL, SOQL does not support all types of `JOIN` operations.
- No Arbitrary Expressions: SOQL does not allow arbitrary expressions or calculations in the `SELECT` clause, unlike SQL.

Examples

Here are some brief examples, you can find more details in the official [SOQL tutorial](#):

Simple Select

Retrieve the `Name` and `Id` of all accounts:

```
SELECT Id, Name
FROM Account
```

Select with Filtering

Retrieve accounts with a specific status:

```
SELECT Id, Name
FROM Account
WHERE Status = 'Active'
```

Select with Grouping

Group accounts by `status` and count them:

```
SELECT Status, COUNT(Id)
FROM Account
GROUP BY Status
```

Authentication Salesforce

Table of contents

[Overview](#)

[Settings](#)

Overview

DBeaver is fully compatible with the Salesforce authentication method, offering you a secure way to access your [Salesforce](#) databases.

Connection "login.salesforce.com" configuration

Connection settings

Salesforce connection settings

Connection settings

- Initialization
- Shell Commands
- Client identification
- Transactions
- General
- Metadata
- Errors and timeouts
- Tableau
- > Data Editor
- > SQL Editor

Main Driver properties SSH

+ Network configurations...

General

Connect by: ☒ Host ☐ URL Driver type: JDBC

JDBC URL: jdbc:dbeaver:salesforce:login.salesforce.com

Host: login.salesforce.com Port:

Authentication

Authentication: Salesforce authentication

Username:

Password: ☒ Save password

Security token:

[You can use variables in connection parameters.](#)

Driver name: Salesforce Driver Settings

Test Connection ... Cancel OK

Settings

Note: Having a security token is necessary for securely connecting to Salesforce from DBeaver. This token can be generated in your Salesforce account settings.

Here are the steps to configure Salesforce authentication:

- 1) From the **Authentication** dropdown menu, select the **Salesforce authentication** method.

2) Enter the username, password and security token that are associated with your Salesforce database in the **Username**, **Password** and **Security token** fields respectively.

Tip: If you check the **Save credentials locally** option, you will not be asked to input your credentials every time you connect.

3) Once you have input your credentials, click the **Test Connection** button to verify your settings. If everything is configured correctly, DBeaver should now be able to connect to your database using the Salesforce authentication method.

SQLite

Table of contents

[Overview](#)

[Setting Up](#)

[SQLite connection configuration](#)

[General SQLite connection settings](#)

[SQLite Crypt connection settings](#)

[SQLite extensions](#)

[Remote database connection](#)

[Synchronization](#)

[Connection details](#)

[SQLite driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering SQLite with DBeaver](#)

[SQLite database objects](#)

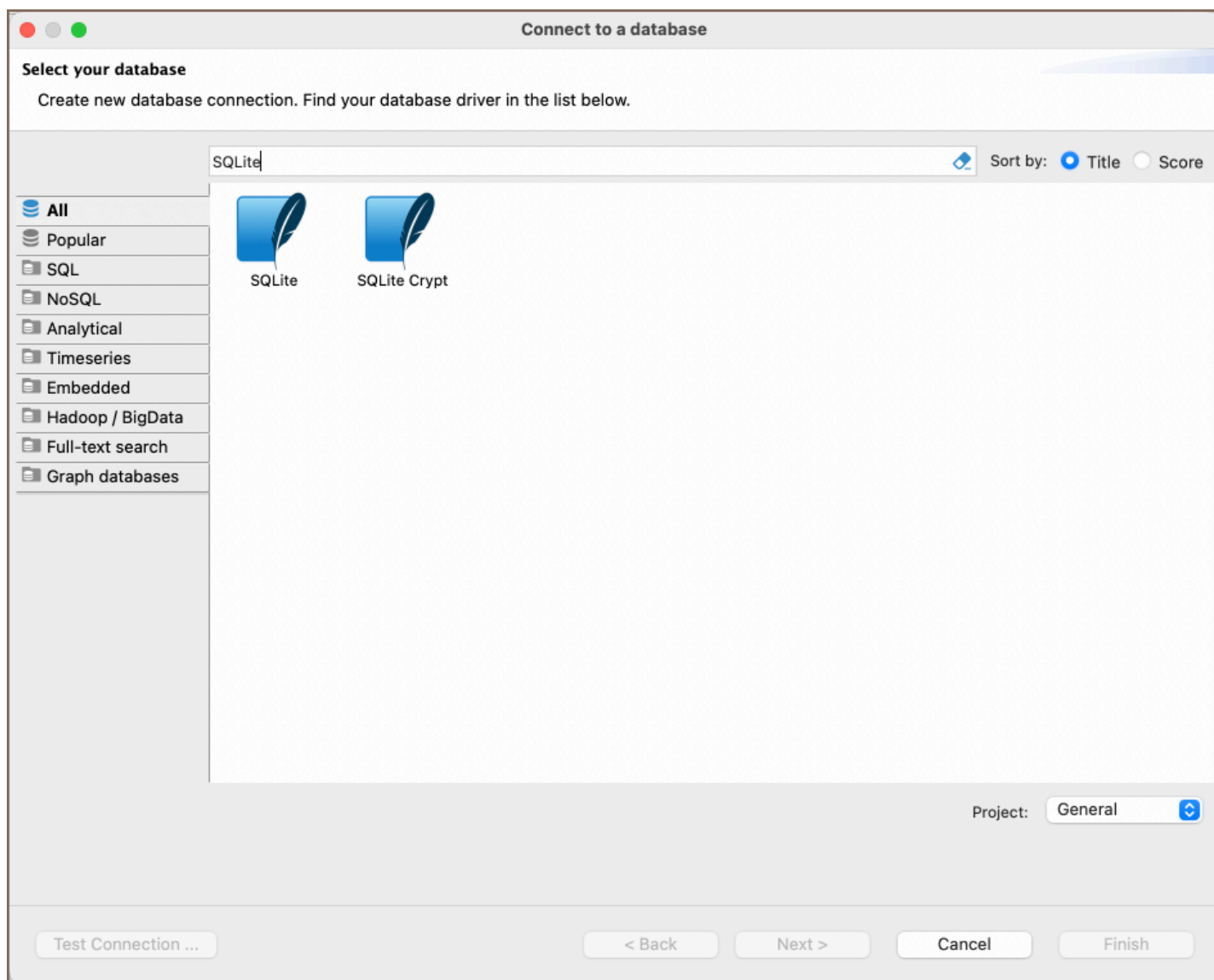
Overview

This guide provides instructions on how to set up and use SQLite with DBeaver.

Before you start, you must create a connection in DBeaver and select SQLite. If you have not done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the SQLite server using a specific driver, supporting all versions of SQLite. DBeaver also supports SQLite extensions such as SQLite Crypt (Cipher).

You can also create a sample database in DBeaver. See our [Creating a Sample Database article](#) for more information.



Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, Kubernetes, and the setup of ODBC/JDBC drivers for SQLite.

SQLite connection configuration

General SQLite connection settings

In this subsection, we will outline the settings for establishing a direct connection to a SQLite database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your SQLite database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

SQLite Connection Settings

SQLite connection settings

Main

Extensions

Driver properties

SSH

+ Network configurations...

General

Connect by: ☒ Host ☐ URL

Driver type: JDBC

JDBC URL: jdbc:sqlite:{file}

Path:

Open ...

Advanced

☐ Database is remote

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: SQLite

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host /URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your SQLite database.
URL	If you are connecting via URL, enter the URL of your SQLite database here. This field is disabled if you're connecting via the host.
Path	Enter the path to the database file.
Database is remote	If the SQLite database resides on a remote server, check this box. Reed more.
Connection Details	Provide additional connection details if necessary.

DBeaver User Guide 24.1.ea. Page 391 of 1174.

Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

SQLite Crypt connection settings

This subsection outlines the settings to establish a connection to encrypted SQLite databases.

Connect to a database

SQLite Crypt Connection Settings
SQLite Crypt connection settings

Main | Extensions | Driver properties

Settings

Path:

Cipher: AES 128

Password: ☒ Save password

Cipher parameters

Parameter	Value
kdf_iter	
fast_kdf_iter	
hmac_use	
hmac_pgno	
hmac_salt_mask	
legacy	0
legacy_page_size	0
kdf_algorithm	
hmac_algorithm	
plaintext_header_size	

[You can use variables in connection parameters.](#) Connection details (name, type, ...)

Driver name: SQLite Crypt Driver Settings Driver license

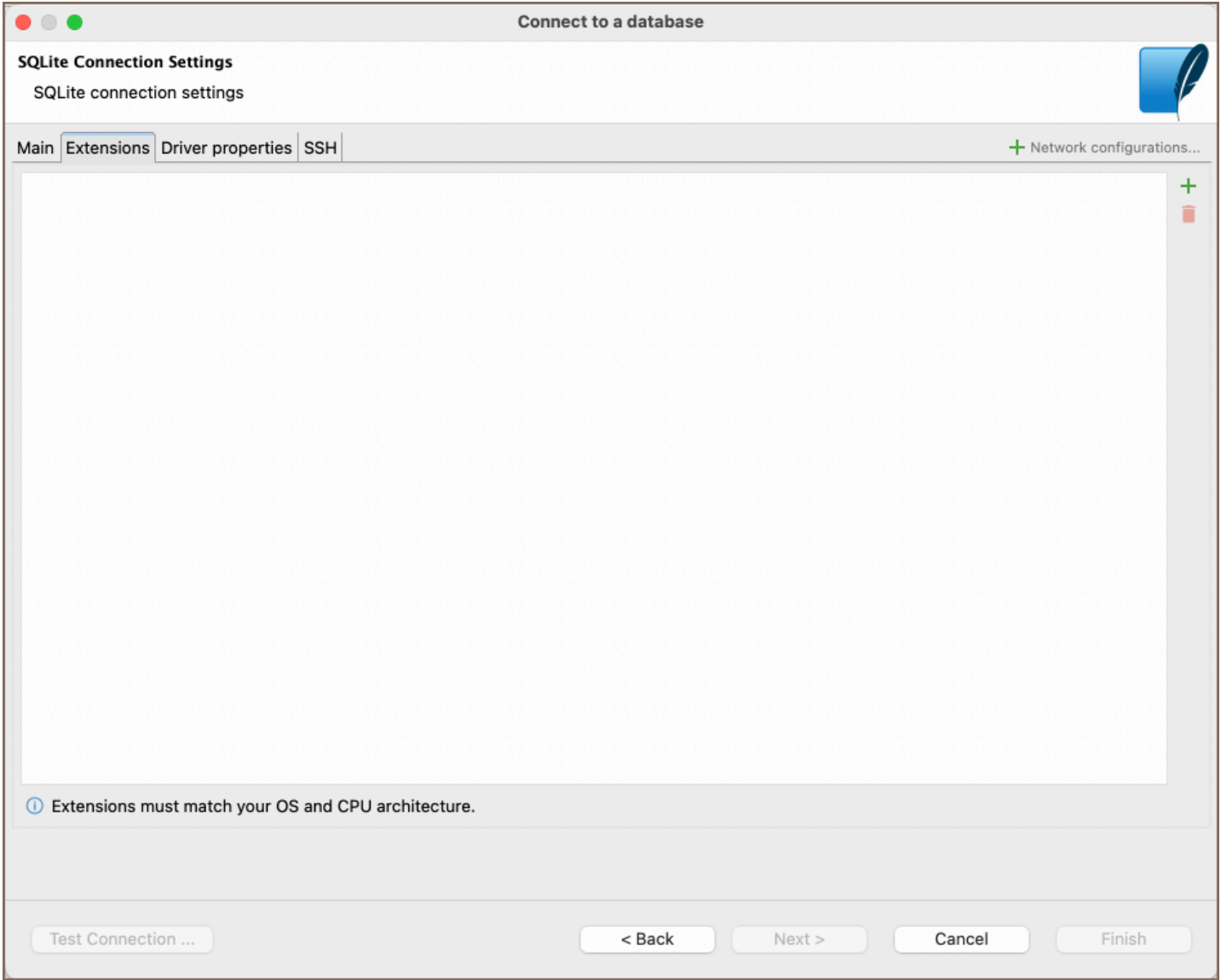
Test Connection ... < Back Next > Cancel Finish

Field	Description
Path	Enter the path to the database file.
Cipher	Choose the encryption algorithm for your database. Available options include AES 128 , AES 256 , ChaCha20 , SQLCipher , SQLCipher v1 , SQLCipher v2 , SQLCipher v3 , SQLCipher v4 , RC4 , and Custom option. For more details, refer to the SQLite Ciphers documentations .
Password	This field will be auto-filled based on your selected driver type.

Cipher parameters	These parameters are read-only and change according to the chosen Cipher. You can customize them only if you select Custom in the Cipher field.
--------------------------	--

SQLite extensions

DBeaver supports the addition of SQLite extensions to enhance database functionality. Extensions can introduce new custom functions, collating sequences, or virtual tables, which are useful for advanced data manipulation or utilizing encryption algorithms not natively supported.



To import an SQLite extension, do the following:

1. Locate the **Extensions** tab in your SQLite connection settings.
2. Click on the **Add** button.
3. Enter the path to the extension file you wish to import.

Note: Make sure the extension you are importing matches your operating system and CPU architecture to ensure compatibility.


Remote database connection

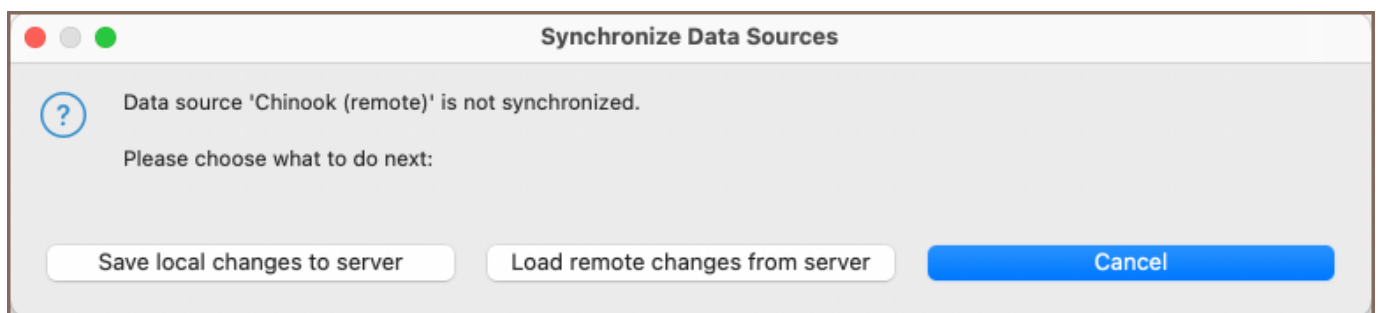
When connecting to a remote SQLite database in DBeaver:

1. Specify a local path in the connection settings. This path should mirror the path on the remote server.
2. Set up an [SSH](#) tunnel.

Synchronization

To maintain consistency between your local SQLite database and the remote server, follow these steps:

1. Make local changes to your database.
2. Save changes.
3. Click the **Synchronize** button  in the main [toolbar](#).
4. A dialog will prompt you with two options:



- **Save local changes to server:** This will push your local changes to the server.
- **Load remote changes from server:** This will pull the server's data to your local environment. After selecting this, refresh the result set to view updated data.

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with SQLite database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

SQLite driver properties

The settings for SQLite **Driver properties** enable you to adjust the performance of the SQLite JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your SQLite database.

For a complete walkthrough on setting up SQLite JDBC and ODBC drivers properties, you can refer to the official [SQLite JDBC documentation](#) and [SQLite ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize SQLite database connections.

You can customize the SQLite driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your SQLite database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing SQLite database connections via ODBC/JDBC drivers. This functionality enables you to connect to your SQLite database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

This process is similar across most databases that support ODBC/JDBC drivers. For a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver, you can refer to our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your SQLite database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering SQLite with DBeaver

DBeaver provides a host of features designed for SQLite databases. This includes the ability to view and manage tables, along with numerous unique capabilities aimed at optimizing database operations.

SQLite database objects

DBeaver lets you view and manipulate a few SQLite database objects. DBeaver has extensive support for various SQLite metadata types, allowing you to interact with:

- Tables
- Views

Teradata

Table of contents

[Overview](#)

[Setting Up](#)

[Teradata connection settings](#)

[Connection details](#)

[Teradata driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering Teradata with DBeaver](#)

[Teradata database objects](#)

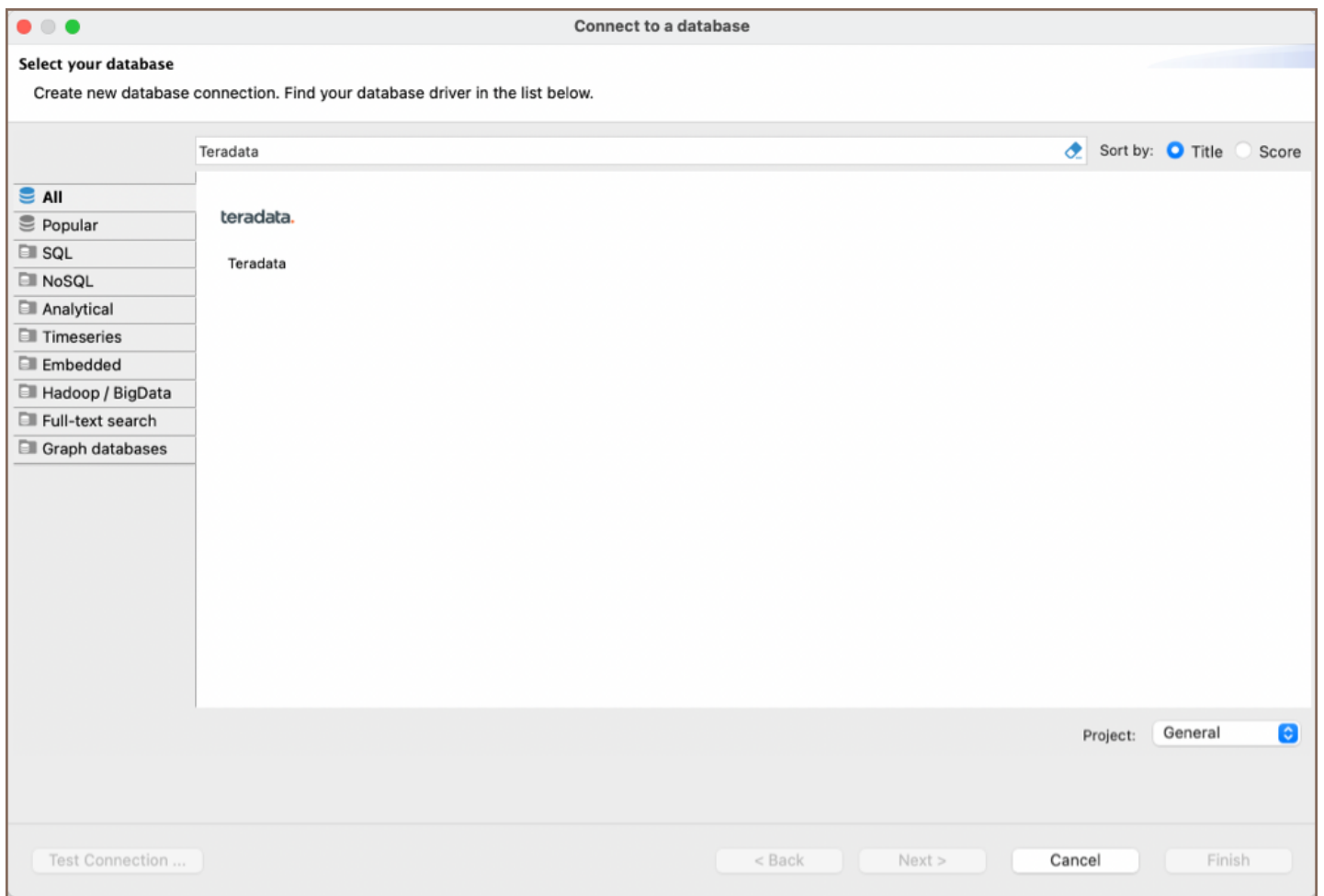
[Teradata Features in DBeaver](#)

Overview

This guide provides instructions on how to set up and use Teradata with DBeaver. DBeaver enhances your experience with specialized support for Teradata features, including unique data types like Vargraphic and Long Vargraphic, as well as advanced security options through Roles and Profiles.

Before you start, you must create a connection in DBeaver and select Teradata. If you have not done this, please refer to our [Database Connection](#) article.

DBeaver interacts with the Teradata server using a specific driver, supporting all versions of Teradata.



Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, Kubernetes, and the setup of ODBC/JDBC drivers for Teradata.

Teradata connection settings

In this subsection, we will outline the settings for establishing a direct connection to a Teradata database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Teradata database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Teradata Connection Settings

Teradata connection settings

teradata

MainDriver propertiesSSH

+ Network configurations...

General

Connect by: ☒ Host ☐ URL

Driver type: JDBC

JDBC URL: jdbc:teradata://localhost/DATABASE={database},DBS_PORT=1025

Host: localhostPort: 1025

Database/Schema:

Authentication

Authentication: Database Native

Username:

Password: ☒ Save password

Advanced Settings

☐ Use TLS protocol

☐ Show databases and users hierarchically

☐ Show only databases in the navigator tree

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: Teradata

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your Teradata database.
URL	If you are connecting via URL, enter the URL of your Teradata database here. This field is disabled if you're connecting via the host.
Host	If you are connecting via host, enter the host address of your Teradata database here.
Database/Schema	Enter the name of the Teradata database/schema you want to connect to.
Port	Enter the port number for your Teradata database. The default Teradata port is 1025 .
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Native Database Authentication - DBEaver Profile Authentication ★ - Kerberos ★ - LDAP Mechanism ★

	You can also read about security in DBeaver PRO .
Use TLS protocol	If you enable this checkbox, the connection will use TLS (Transport Layer Security) to encrypt the data that is transmitted between DBeaver and the Teradata database.
Show databases and users hierarchically	Enable this checkbox to view databases and associated user permissions in a hierarchical structure.
Show only databases in the navigator tree	Displays databases in the navigator tree and exclude other types of objects like schemas, tables, and users.
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with MySQL database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Teradata driver properties

The settings for Teradata **Driver properties** enable you to adjust the performance of the Teradata JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your Teradata database.

For a complete walkthrough on setting up Teradata JDBC and ODBC drivers properties, you can refer to the official [Teradata JDBC documentation](#) and [Teradata ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Teradata database connections.

You can customize the Teradata driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your Teradata database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing Teradata database connections via ODBC/JDBC drivers. This functionality enables you to connect to your Teradata database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

This process is similar across most databases that support ODBC/JDBC drivers. For a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver, you can refer to our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your Teradata database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering Teradata with DBeaver

DBeaver provides a host of features designed for Teradata databases. This includes the ability to view and manage databases, along with numerous unique capabilities aimed at optimizing database operations.

Teradata database objects

DBeaver lets you view and manipulate a wide range of Teradata database objects. DBeaver has extensive support for various Teradata metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Tables
 - Columns
 - Keys
 - Foreign keys
 - Indexes
 - References
 - Triggers
 - Views
 - Indexes
 - Procedures
 - Macros
 - Data Types
- Administration
 - Zones

- Maps
- Users
- Roles
- Profiles

Teradata Features in DBeaver

DBeaver is not limited to typical SQL tasks. It also includes numerous unique features specifically for Teradata. Beyond regular SQL operations, DBeaver provides a range of Teradata-specific capabilities, such as:

Category	Feature
Data Types	Teradata Vargraphic and Long Vargraphic
Security	Teradata Roles and Profiles
Database Management	Teradata Procedures
	Teradata Macros

Additional features compatible with Teradata, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Schema Management	Schema Compare
Data Visualization	ERD Guide

Athena

Table of contents

[Overview](#)

[Athena specialty](#)

[Setting up](#)

[Athena connection settings](#)

[Connection details](#)

[Secure connection configurations](#)

[Powering Athena with DBeaver](#)

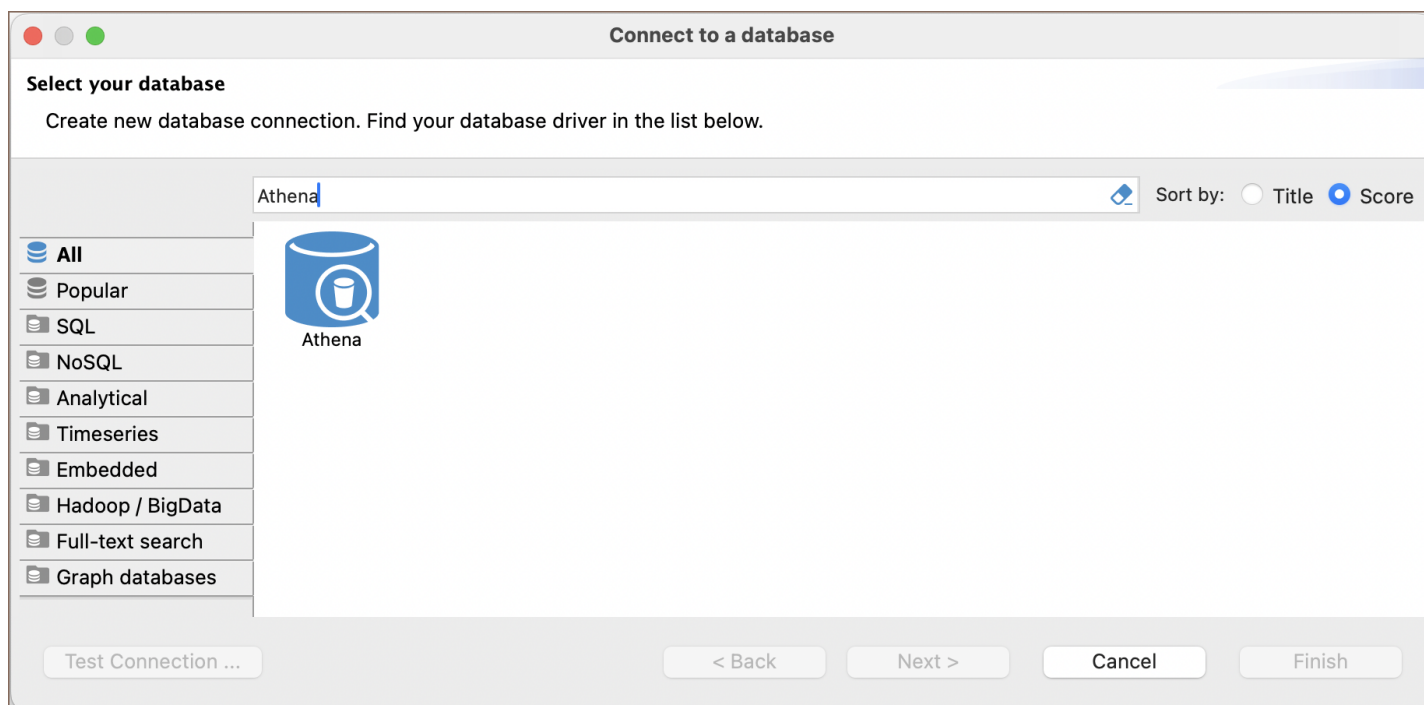
[Athena database objects](#)

[Athena additional features in DBeaver](#)

Overview

This documentation outlines the process of configuring and using AWS Athena with DBeaver.

To manage databases using AWS Athena in DBeaver, establishing a connection is essential. This is done by selecting the **Athena** option in the database connection settings of DBeaver. For those new to this process, our [Creating a Connection](#) article provides detailed guidance.



Tip: Users with [Ultimate](#) and [Team](#) editions of DBeaver can connect to their AWS Athena databases using the [Cloud Explorer](#).

Athena specialty

AWS Athena, as an analytical database service, has limitations compared to traditional relational databases. It does not support real-time data changes, such as adding or updating data. Also, Athena cannot modify data using **DELETE** or **UPDATE** commands. It lacks constraint enforcement like primary keys and foreign keys. For more detailed information, refer to the [AWS Athena database documentation](#).

Important: In DBeaver, AWS Athena is read-only.

For a detailed overview of the supported features and capabilities of AWS Athena within DBeaver, please see the section on [Powering Athena with DBeaver](#).

Setting up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, Proxies, and Kubernetes.

Athena connection settings

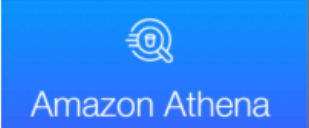
In this subsection, we will outline the settings for establishing a direct connection to a Athena database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Athena database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

AWS Athena Connection Settings


AWS / Athena connection settings



Main

Driver properties

SSH

 Network configurations...

Connection

Region:

S3 location:

Browse ...

☐ Show catalogs

Authentication

Authentication:

Credentials:

Details


Access key: Secret key:

☒ Save credentials

Role Name:

☐ 3rd party account

☐ Use AWS Secrets Manager

 You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Region	Specify the AWS region where your Athena database is located, such as <code>us-west-1</code> . This should match the region of the S3 bucket used for storing query results.
	Enter the output location for query results in the format

DBeaver User Guide 24.1.ea. Page 406 of 1174.

S3 location	<p><code>s3://aws-athena-query-results-[your-account-id]-[region]/</code> . Replace <code>[your-account-id]</code> with your actual AWS account ID and <code>[region]</code> with the region code, for example,</p> <p><code>s3://aws-athena-query-results-80980980-us-east-1/</code> . Ensure you have the necessary permissions set for this S3 location.</p>
Browse	Click to open a dialog that allows you to navigate and select an S3 bucket through DBeaver's interface.
Show catalogs	Check this option if you want to view the data catalog in the DBeaver connection. This will show the databases and tables defined in Athena's data catalog.
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Database native - AWS IAM - DBeaver Profile ★ <p>You can also read about security in DBeaver PRO.</p>
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings, configure them here. For detailed guide on driver properties please refer to the official AWS Athena documentation .

Connection details

The **Connection Details** section in DBeaver allows for further customization of your Athena connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- Security Settings Guide

- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Secure connection configurations

DBeaver supports secure connections to your Athena database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering Athena with DBeaver

DBeaver provides a host of features designed for Athena databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

Athena database objects

DBeaver lets you view a range of Athena database objects, such as:

- Schemas
 - Tables
 - Columns

- Views
 - Columns
- Procedures

Athena additional features in DBeaver

DBeaver provides additional features compatible with Athena, but not exclusive to it:

Category	Feature
Data Transfer	Data Export
Schema Management	Schema Compare ★

DocumentDB

Table of contents

[Connections](#)

[Queries](#)

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only. AWS DocumentDB is based on the [MongoDB](#) engine.

It has several minor differences in the query processing and network configuration.

However, most features which work for MongoDB will work for DocumentDB as well. Please refer to the [MongoDB](#) article.

Connections

AWS restricts direct access to DocumentDB clusters from outside of the cloud (region). So you can connect to it directly (using a cluster host name) only when DBeaver is deployed on the EC2 instance.

In other cases you will need to use the SSH tunnel through a proxy machine to access DocumentDB instance.

Please read the AWS Documentation about proxy configurations: <https://docs.aws.amazon.com/documentdb/latest/developerguide/connect-from-outside-a-vpc.html>

In DBeaver you can use the SSH tab on the connection settings page. Just enter proxy host, user name and specify a private key file (it is provided by AWS as a keypair).

Queries

DBeaver processes DocDB SQL queries exactly like in [MongoDB](#). It supports SELECT, UPDATE, INSERT and DELETE queries.

SELECT queries support WHERE, ORDER BY, GROUP BY and HAVING clauses.

DocumentDB restricts the `eval` function so all JavaScript queries will be parsed on the client's side and then evaluated at a DocDB cluster one by one. Most JS functions work exactly like in Mongo Shell.

DynamoDB

Table of contents

[Supported features](#)

[DynamoDB connection](#)

[Database navigation](#)

[Viewing table data](#)

[Viewing data in JSON document format](#)

[Executing queries](#)

[Exporting and importing data](#)

Note: This driver is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Supported features

- Table data view
- Table data edit in document (json) mode
- Data filters
- SQL queries execution
- JSON queries execution
- Data export and import

DynamoDB connection

Connection "DynamoDB - AWS" configuration

Connection settings
DynamoDB connection settings

Connection settings
General
Metadata
Errors and timeouts
Data viewer
SQL Editor

Main SSH Proxy

Server
Type: ☒ AWS ☐ Standalone
AWS Region: eu-central-1
Endpoint: http://localhost:8000

Authentication
Authentication: AWS IAM
User name: N/A Region:
☐ Use Default credentials
Access key: AKIAJQTLGDGH4MASDADS AWS access key Secret key:
☐ 3rd party account
Account ID: Role Name:

You can use variables in connection parameters.

Driver name: DynamoDB Edit Driver Settings

Test Connection ... OK Cancel

DBeaver supports AWS Cloud and Standalone versions of DynamoDB.

For standalone server you need to enter endpoint (http or https URL).

For cloud server you must enter the AWS region. DynamoDB exists in all available regions in your AWS account but the tables are different.

AWS Access Key and Secret Key are used for authentication.

For 3rd-party account access you must specify the 3rd party account ID (12-digits number) and the 3rd party role name. This role will be used for permission management. Your account must be added to the whitelist in the 3rd party account.

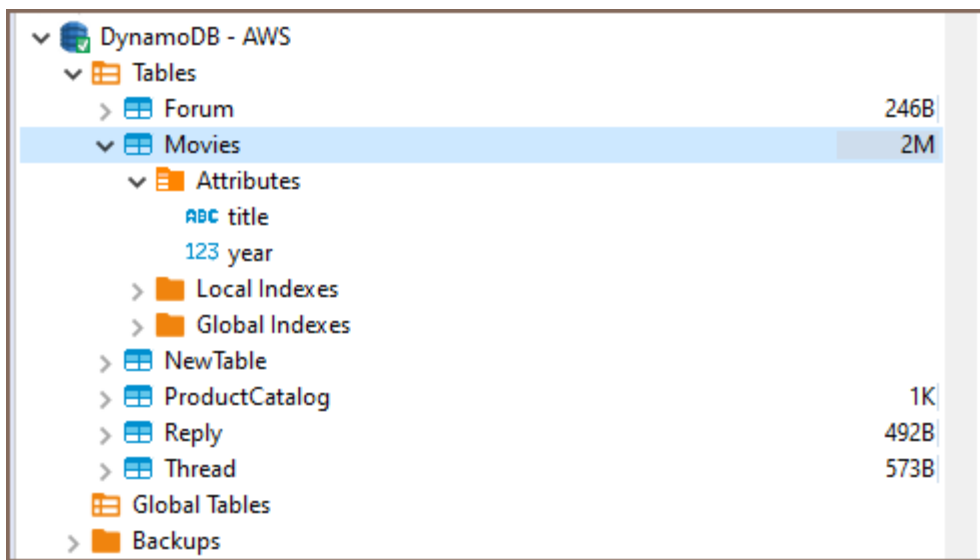
Press "Test Connection" to validate your connection settings.

Database navigation

DynamoDB has a simple metadata structure. Basically, you can only access Table and Global tables.

Table has primary attributes (a kind of primary key) and indexes.

DynamoDB is a document-oriented database. Each table may have its own set of attributes and sub-attributes.



Viewing table data

You can open table editor and see the table data.

You may need to switch to the "Data" tab. DBeaver converts DynamoDB documents into a table format by default,

but you can switch to another data representation.

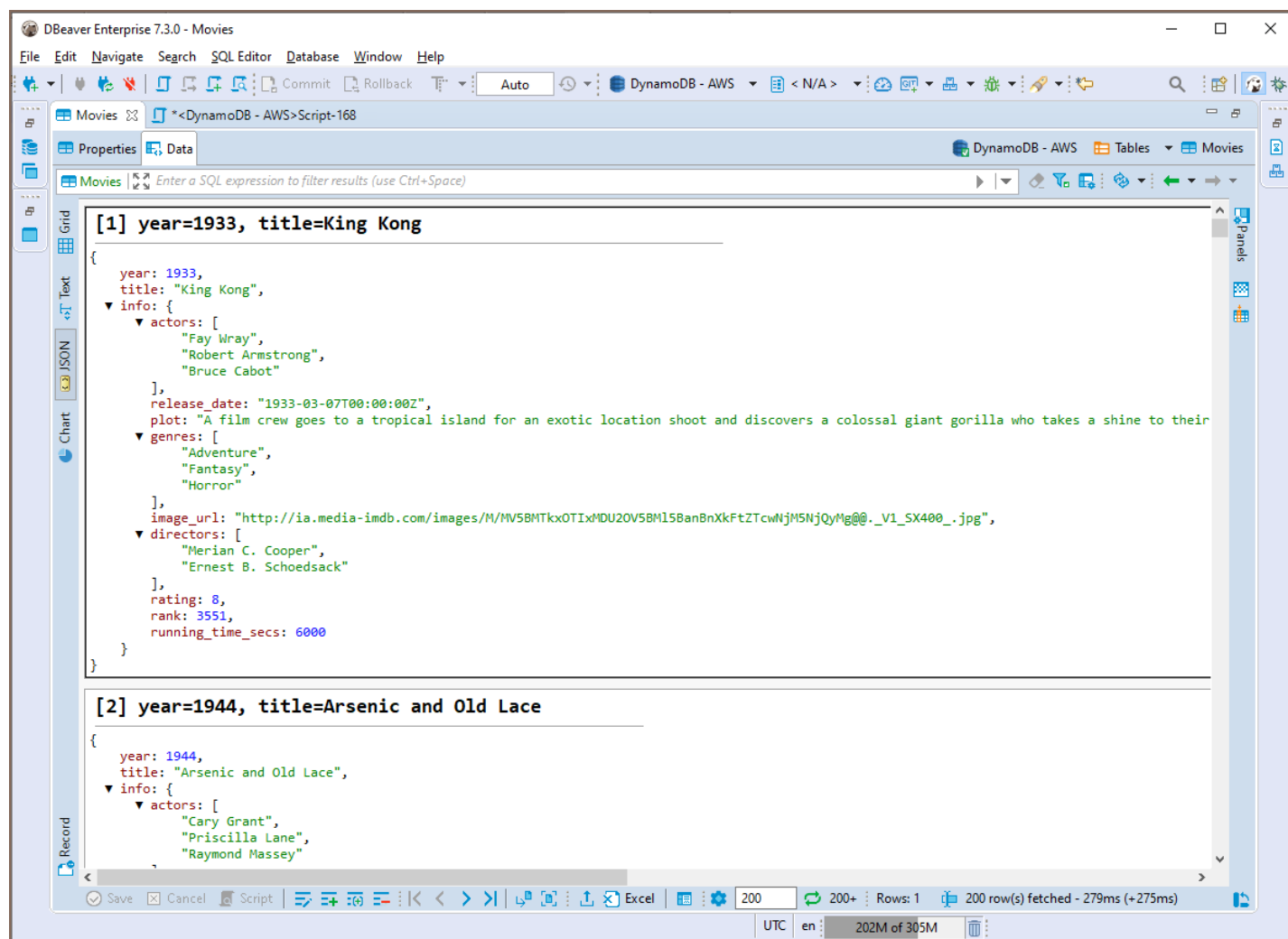
You can use data filters in order to find documents.

The screenshot shows the DBeaver Enterprise 7.3.0 interface. The main window is titled "DBeaver Enterprise 7.3.0 - Movies". The menu bar includes File, Edit, Navigate, Search, SQL Editor, Database, Window, and Help. The toolbar contains various icons for database operations. The left sidebar has icons for Grid, Text, JSON, Chart, and Record. The main area displays a table of movie data with columns: year, title, info, release_date, plot, and image_url. The table is filtered by "year > 1950". The status bar at the bottom shows "Rows: 1" and "200 row(s) fetched - 679ms (+627ms)".

	year	title	info	release_date	plot	image_url
1	1958	Cat on a Hot Tin Roof		1958-09-18T00:00:00Z	Brick, an alcoholic ex-football player, drinks his day	http://ia.media-imdb.com/images/M/MV5BMjA4...
2	1958	Monster on the Campus		1958-12-17T00:00:00Z	The blood of a primitive fish exposed to gamma r	http://ia.media-imdb.com/images/M/MV5BMTQ...
3	1958	No Time for Sergeants		1958-05-29T00:00:00Z	Will Stockdale is a country bumpkin drafted into th	http://ia.media-imdb.com/images/M/MV5BMTI4...
4	1958	Teacher's Pet		1958-04-01T00:00:00Z	[NULL]	http://ia.media-imdb.com/images/M/MV5BMTI1...
5	1958	Touch of Evil		1958-04-23T00:00:00Z	A stark, perverse story of murder, kidnapping, and	http://ia.media-imdb.com/images/M/MV5BNjMj...
6	1958	Vertigo		1958-05-09T00:00:00Z	A retired San Francisco detective suffering from ac	http://ia.media-imdb.com/images/M/MV5BNzYk...
7	2013	+		2013-03-10T00:00:00Z	Three college friends hit the biggest party of the y	http://ia.media-imdb.com/images/M/MV5BMTQ...
8	2013	100 Degrees Below Zero		2013-03-29T00:00:00Z	After freak climate and weather events destroy the	http://ia.media-imdb.com/images/M/MV5BMTk...
9	2013	12 Years a Slave		2013-08-30T00:00:00Z	In the antebellum United States, Solomon Northup	http://ia.media-imdb.com/images/M/MV5BMjEx...
10	2013	2 Guns		2013-07-30T00:00:00Z	A DEA agent and a naval intelligence officer find th	http://ia.media-imdb.com/images/M/MV5BNTQ...
11	2013	20 Feet from Stardom		2013-01-17T00:00:00Z	Backup singers live in a world that lies just beyond	http://ia.media-imdb.com/images/M/MV5BMTQ...
12	2013	200 Cartas		2013-06-14T00:00:00Z	[NULL]	http://ia.media-imdb.com/images/M/MV5BMTQ...
13	2013	21 & Over		2013-02-28T00:00:00Z	The night before his big medical school interview,	http://ia.media-imdb.com/images/M/MV5BMjI0...
14	2013	3 Geezers!		2013-05-22T00:00:00Z	[NULL]	http://ia.media-imdb.com/images/M/MV5BOTg...
15	2013	3096 Tage		2013-02-28T00:00:00Z	A young Austrian girl is kidnapped and held in cap	[NULL]
16	2013	42		2013-04-12T00:00:00Z	The life story of Jackie Robinson and his history-m	http://ia.media-imdb.com/images/M/MV5BMTQ...
17	2013	47 Ronin		2013-12-06T00:00:00Z	A band of samurai set out to avenge the death an	http://ia.media-imdb.com/images/M/MV5BMTA...
18	2013	7500		2013-10-01T00:00:00Z	Passengers aboard a flight across the Pacific Ocea	http://ia.media-imdb.com/images/M/MV5BMjE...
19	2013	A Belfast Story		2013-09-20T00:00:00Z	[NULL]	http://ia.media-imdb.com/images/M/MV5BMTY...
20	2013	A Case of You		2013-04-21T00:00:00Z	A young writer tries to impress a girl he meets onl	http://ia.media-imdb.com/images/M/MV5BMjI2...
21	2013	A Field in England		2013-07-04T00:00:00Z	[NULL]	http://ia.media-imdb.com/images/M/MV5BMzI4...
22	2013	A Good Day to Die Hard		2013-02-07T00:00:00Z	John McClane travels to Russia to help out his see	http://ia.media-imdb.com/images/M/MV5BMTQ...
23	2013	A Madea Christmas		2013-12-13T00:00:00Z	Madea dispenses her unique form of holiday spirit	http://ia.media-imdb.com/images/M/MV5BMTY...
24	2013	A Most Wanted Man		2013-11-21T00:00:00Z	A Chechen Muslim illegally immigrates to Hamburg	http://ia.media-imdb.com/images/M/MV5BNjE...
25	2013	A Resurrection		2013-03-22T00:00:00Z	[NULL]	http://ia.media-imdb.com/images/M/MV5BODY...
26	2013	A Single Shot		2013-02-09T00:00:00Z	The tragic death of a beautiful young girl starts a t	http://ia.media-imdb.com/images/M/MV5BMjM...
27	2013	A C.O.D.		2013-01-23T00:00:00Z	A grown man caught in the crossfire of his parents	http://ia.media-imdb.com/images/M/MV5BMTQ...

Viewing data in JSON document format

You view, search and edit JSON documents. Double-click on a document to activate the editor.

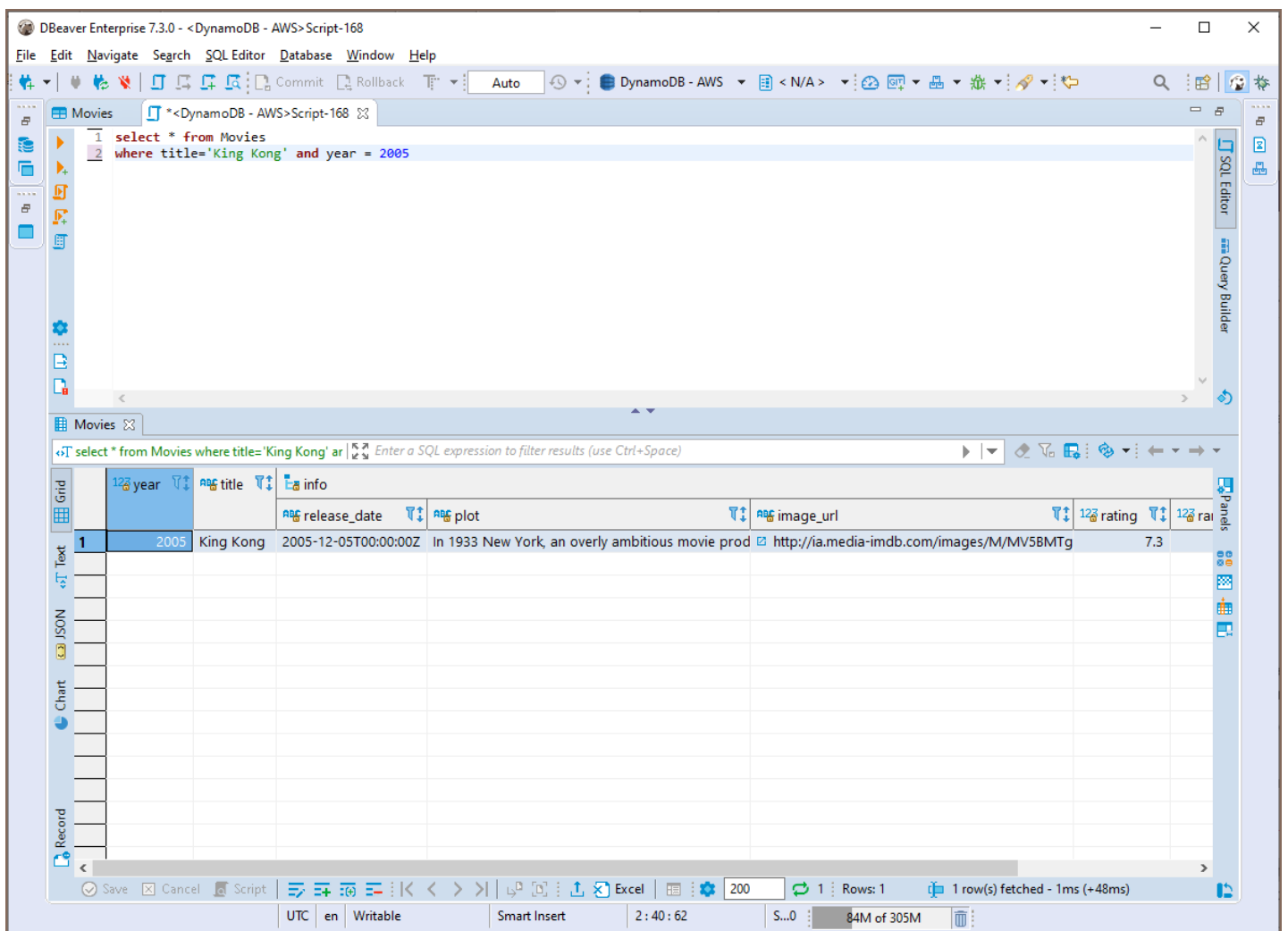


Executing queries

DBeaver supports simple SQL dialect for DynamoDB.

You can use the WHERE clause in the same fashion as in regular SQL in order to find or filter documents.

You can also use JSON requests syntax to query documents. See [Amazon DynamoDB query reference](#).



Exporting and importing data

You can export data from a DynamoDB table in different file formats (CSV, XLSX, XML, JSON, etc,) or export data directly to another table.

Keyspaces

Table of contents

[Overview](#)

[Connecting to Keyspaces](#)

[Browsing Keyspaces tables](#)

[Executing CQL](#)

[ERD](#)

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Overview

[AWS Keyspaces](#) is a key-value database based on [Apache Cassandra](#).

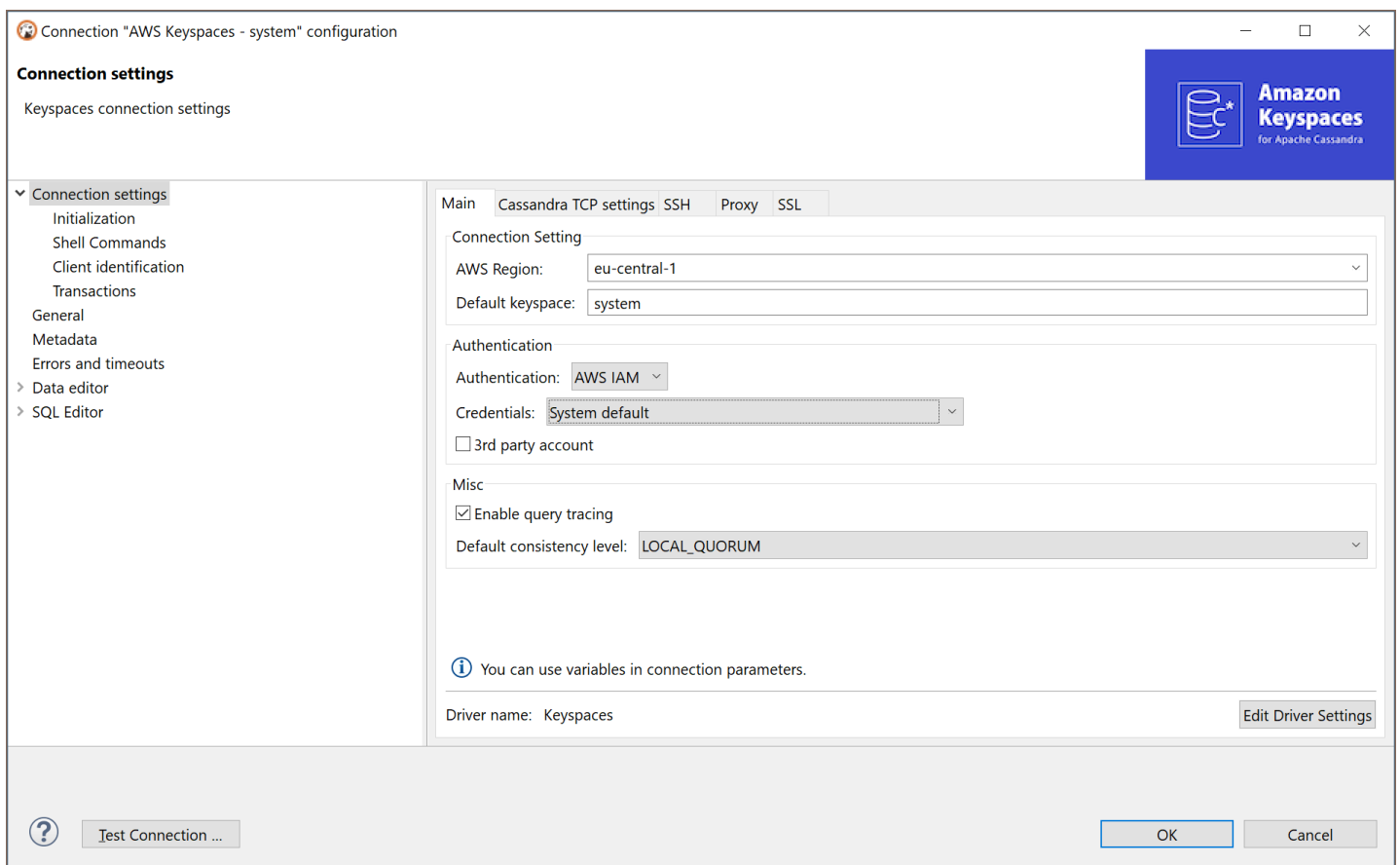
DBeaver EE supports the Keyspaces schema browser, data viewer and CQL queries execution. It also supports various administrative tools.

Connecting to Keyspaces

AWS Keyspaces uses AWS IAM authentication.

You need to specify your IAM credentials and AWS region. There is no specific endpoint - there is only one cluster per account per AWS region.

There is also no need to configure SSH or SSL - DBeaver uses default AWS settings to access the Keyspace cluster.



Browsing Keyspaces tables

You can browse, view, edit and filter Cassandra tables the same way as with regular (relational) tables. However, being a distributed key-value database, Keyspaces does not support any kind of referential integrity. There are no foreign keys, references, etc.

You should note that Cassandra has a very advanced (comparing to relational databases) data type system. Each column may be a collection, map, or set of values (with a very big number of values). In some cases this makes browsing data in the "Grid" mode inconvenient.

DBeeer Enterprise 5.2.1 - <Cassandra CQL - system> Script-19

File Edit Navigate Search SQL Editor Database Window Help

Database Navigator

- Enter a part of table name here
- Keyspaces
 - killrvideo
 - Tables
 - comments_by_user
 - comments_by_video
 - encoding_job_notifications
 - latest_videos
 - tags_by_letter
 - uploaded_videos
 - uploaded_videos_by_jobid
 - user_credentials
 - user_videos
 - users
 - video_event
 - video_rating
 - video_ratings_by_user
 - videos
 - videos_by_tag
 - Materialized views
 - User types
 - Aggregates
 - Functions
 - system
 - system_auth
 - system_distributed
 - system_schema
 - system_traces
 - test
 - Tables
 - collect_things
 - driverperf_v1
 - job
 - list_tbl
 - messages
 - staged
 - Materialized views
 - User types

job

```

1 CREATE TABLE songs (
2   id uuid PRIMARY KEY,
3   title text,
4   album text,
5   artist text,
6   data blob
7 );
8
9 CREATE TABLE playlists (
10  id uuid,
11  song_order int,
12  song_id uuid,
13  title text,
14  album text,
15  artist text,
16  PRIMARY KEY (id, song_order) );
17 INSERT INTO playlists (id, song_order, song_id, title, artist, album)
18 VALUES ('00000000-0000-0000-0000-000000000000', 1, '00000000-0000-0000-0000-000000000000', 'The Gambia', 'The Gambia', 'The Gambia');

```

Result

SELECT * FROM test.job LIMIT 200 ALLOW FILTERING

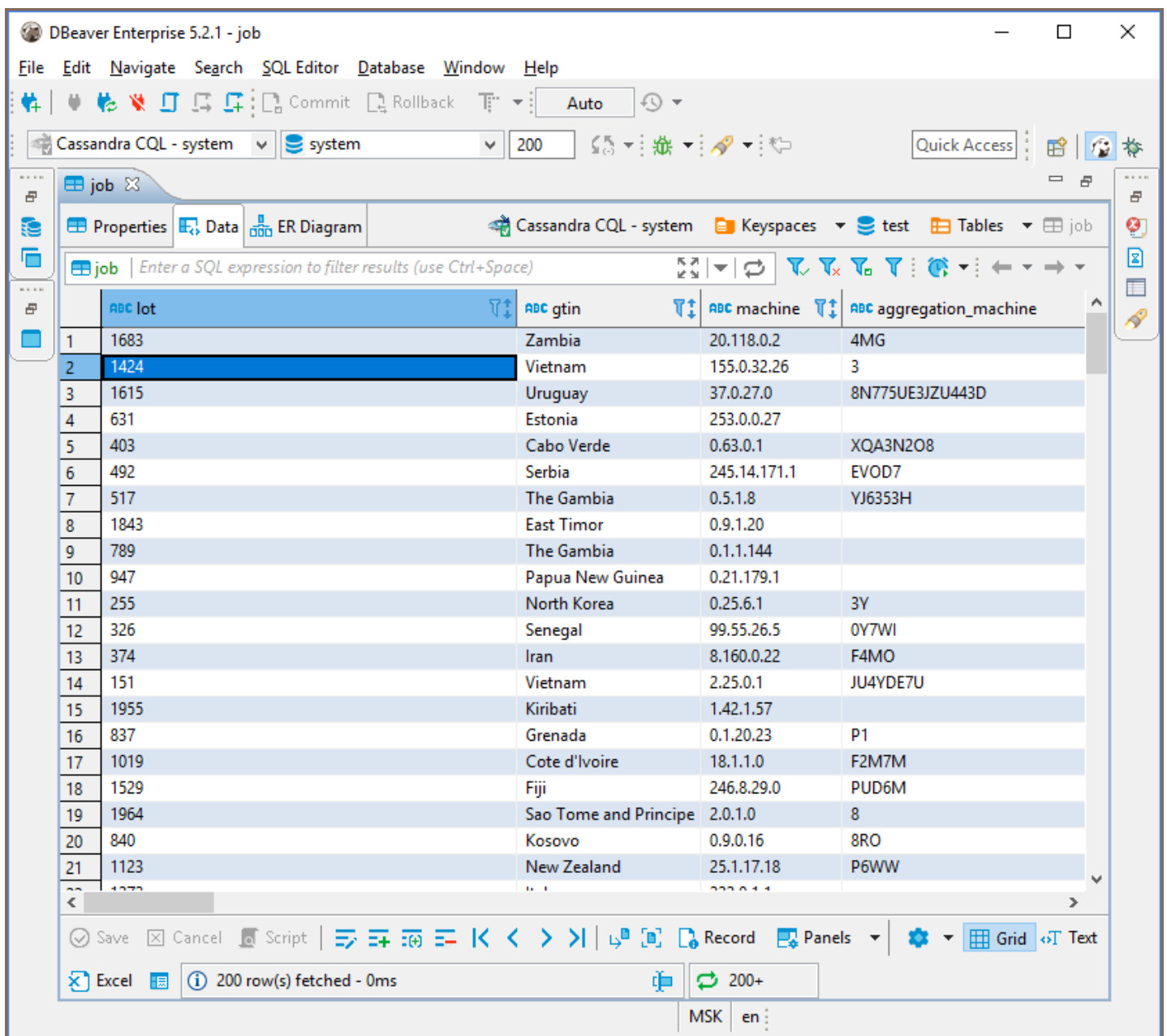
	asc lot	asc gtin	asc
1	1683	Zambia	20.
2	1424	Vietnam	15.
3	1615	Uruguay	37.
4	631	Estonia	25.
5	403	Cabo Verde	0.6
6	492	Serbia	24.
7	517	The Gambia	0.5
8	1843	East Timor	0.9
9	789	The Gambia	0.1
10	947	Papua New Guinea	0.2
11	255	North Korea	0.2
12	326	Senegal	99.
13	374	Iran	8.1
14	151	Vietnam	2.2

Query Trace

Event	Time	Source	Thread
Parsing SELECT * FROM test.job LIMIT 200 ALLOW FILTERING	137	/127.0.0.1	Native-Transport-Requests-1
Preparing statement	265	/127.0.0.1	Native-Transport-Requests-1
Computing ranges to query	453	/127.0.0.1	Native-Transport-Requests-1
Submitting range requests on 257 ranges with a concurrency ...	795	/127.0.0.1	Native-Transport-Requests-1
Submitted 1 concurrent range requests	1027	/127.0.0.1	Native-Transport-Requests-1
Executing seq scan across 1 sstables for (min(-9223372036854...	1844	/127.0.0.1	ReadStage-3
Read 200 live rows and 0 tombstone cells	3003	/127.0.0.1	ReadStage-3

200 row(s) fetched - 124ms (+1ms)

MSK en Writable Smart Insert 11:18



Executing CQL

CQL [Cassandra Query Language](#) is a very simple kind of SQL language dialect.

It supports simple SELECT queries, DDL statements (like CREATE TABLE) and some other query types.

You can use the standard DBeaver SQL editor to execute CQL queries. DBeaver supports Cassandra query execution, results scrolling, data export/import, mock data generation and other features.

ERD

Physical ERD (Entity Relation Diagram) does not make much sense for Keyspaces as there are no foreign keys. However, you can make you own [custom ERD](#) and connect Keyspaces tables with each other using logical associations.

Redshift

Table of contents

[Overview](#)

[Redshift specialty](#)

[Setting up](#)

[Redshift connection settings](#)

[Connection details](#)

[Redshift driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering Redshift with DBeaver](#)

[Redshift database objects](#)

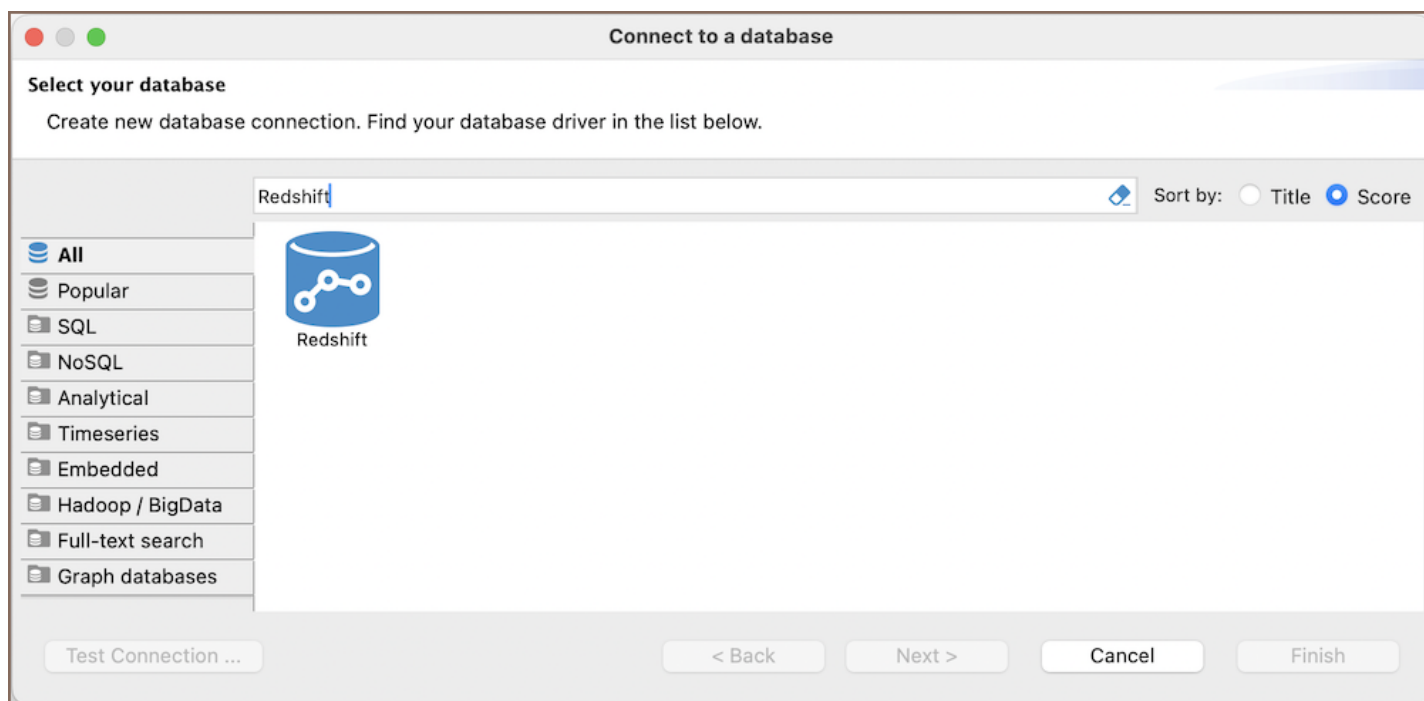
[Redshift additional features in DBeaver](#)

Overview

This section provides guidance on configuring and using AWS Redshift with DBeaver.

AWS Redshift is a fully managed data warehouse service known for its high performance and scalability in handling large datasets. An important aspect of Redshift is that it is based on open standard PostgreSQL, which means it shares numerous settings and capabilities similar to PostgreSQL databases. For detailed information about AWS Redshift's features, visit the [AWS Redshift documentation](#).

To manage databases using AWS Redshift in DBeaver, select the **Redshift** option in the database connection settings of DBeaver. Our [Creating a Connection](#) article offers comprehensive instructions for those new to this process.



Tip: Users with [Ultimate](#) and [Team](#) editions of DBeaver can connect to their AWS Redshift databases using the [Cloud Explorer](#).

Redshift specialty

AWS Redshift is an optimized database service designed for large-scale data warehousing and analytics, supporting a wide array of SQL operations. Its foundation in PostgreSQL, through a modified version tailored for its query processing, allows Redshift to support many of the standard SQL features of PostgreSQL. In addition to its PostgreSQL roots, a key feature of AWS Redshift is its columnar data storage. This approach to data storage is particularly efficient for analytics, as it allows for faster retrieval of specific columns, optimizing the performance for query-heavy workloads typically associated with data warehousing. However, it is important to note that AWS Redshift does not fully support referential integrity constraints, such as foreign keys. For more detailed information, refer to the [AWS Redshift database documentation](#).

For a detailed overview of the supported features and capabilities of AWS Redshift within DBeaver, please see the section on [Powering Redshift with DBeaver](#).

Setting up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, SSL, Proxies, Kubernetes, and the setup of ODBC/JDBC drivers for Redshift.

Redshift connection settings


In this subsection, we will outline the settings for establishing a direct connection to a Redshift database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Redshift database.

1) The first page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

New Redshift connection

AWS / Redshift connection settings



Main

Redshift

Driver properties

SSH

SSL

+ Network configurations...

Server

Connect by: ☒ Host ☐ URL

Driver type: JDBC

URL: jdbc:redshift://localhost:5439/dev

Host/Instance: localhost

Database: dev

Port: 5439

☐ Show all databases

Authentication

Authentication: Database Native

Username: awsuser

Password:

☒ Save password

[You can use variables in connection parameters.](#)

Connection details (name, type, ...)

Driver name: AWS / Redshift

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host /URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your Redshift database.
	If you are connecting via URL, enter the URL of your Redshift database here. This field is hidden if you are


URL	connecting via the host.
Host/Instance	If you are connecting via host or instance, enter the address of your AWS Redshift database here.
Database	Enter the name of the Redshift database you want to connect to.
Show all databases	Displays all databases in the database navigator . Note: This option works only when the Host connection type is selected. It does not work when connecting via URL .
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

2) The second page of the connection settings offers additional options that allow you to customize your further connection to the Redshift database.

Connect to a database

New Redshift connection

AWS / Redshift connection settings



Main

Redshift

Driver properties

SSH

SSL

+ Network configurations...

Settings

☐ Show template databases
☐ Show databases not available for connection
☐ Show database statistics
☐ Read all data types
☐ Read table keys with columns

SQL

Show \$\$ quote as: Code block

Show \$tagName\$ quote as: Code block

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Show template databases	Displays template databases. This option is available when Show all databases is checked.
Show databases not available for connection	Show databases not available for connection in database list. This option is available when Show all databases is checked.
Show databases statistics	Displays statistics for databases.
Read all data types	Reads all data types from the server (including arrays).
Read table keys with columns	Enable reading table constraints during the column reading stage to ensure accurate recognition of keys. Note that this setting may decrease metadata loading performance.
Show \$\$ quote as	Displays \$\$ quotes as code blocks or strings.

Show `$tagName$`

quote as

Displays `$tagName$` quotes as code blocks or strings.

Connection details

The **Connection Details** section in DBeaver allows for further customization of your Redshift connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- Security Settings Guide
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Redshift driver properties

The settings for Redshift **Driver properties** enable you to adjust the performance of the Redshift JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your Redshift database.

For a complete walkthrough on setting up Redshift JDBC and ODBC drivers properties, you can refer to the official [AWS Redshift JDBC documentation](#) and [AWS Redshift ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Redshift database connections.

You can customize the Redshift driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your Redshift database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing Redshift database connections via ODBC/JDBC drivers. This functionality enables you to connect to your Redshift database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

You can find a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver in our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your Redshift database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering Redshift with DBeaver

DBeaver provides a host of features designed for Redshift databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

Redshift database objects

DBeaver lets you view and manipulate a wide range of Redshift database objects. DBeaver has extensive support for various Redshift metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Schemas
 - Tables
 - Columns
 - Constraints
 - Foreign Keys
 - Dependencies
 - References
 - Views
 - Functions
 - Data types
 - Aggregate functions
 - Users ★
 - Groups ★
 - Roles ★
- Administer
 - Jobs
 - Session Manager
- Datashares ★
 - Objects ★
 - Consumers ★

Redshift additional features in DBeaver

DBeaver provides additional features compatible with Redshift, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Data Visualization	ERD Guide
	Query Execution plan
Session Management	Session Manager
Schema Management	Schema Compare ★
Data Generation	Mock Data Generation ★

Timestream

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Timestream specialty](#)

[Setting up](#)

[Timestream connection settings](#)

[Connection details](#)

[Timestream driver properties](#)

[Secure connection configurations](#)

[Powering Timestream with DBeaver](#)

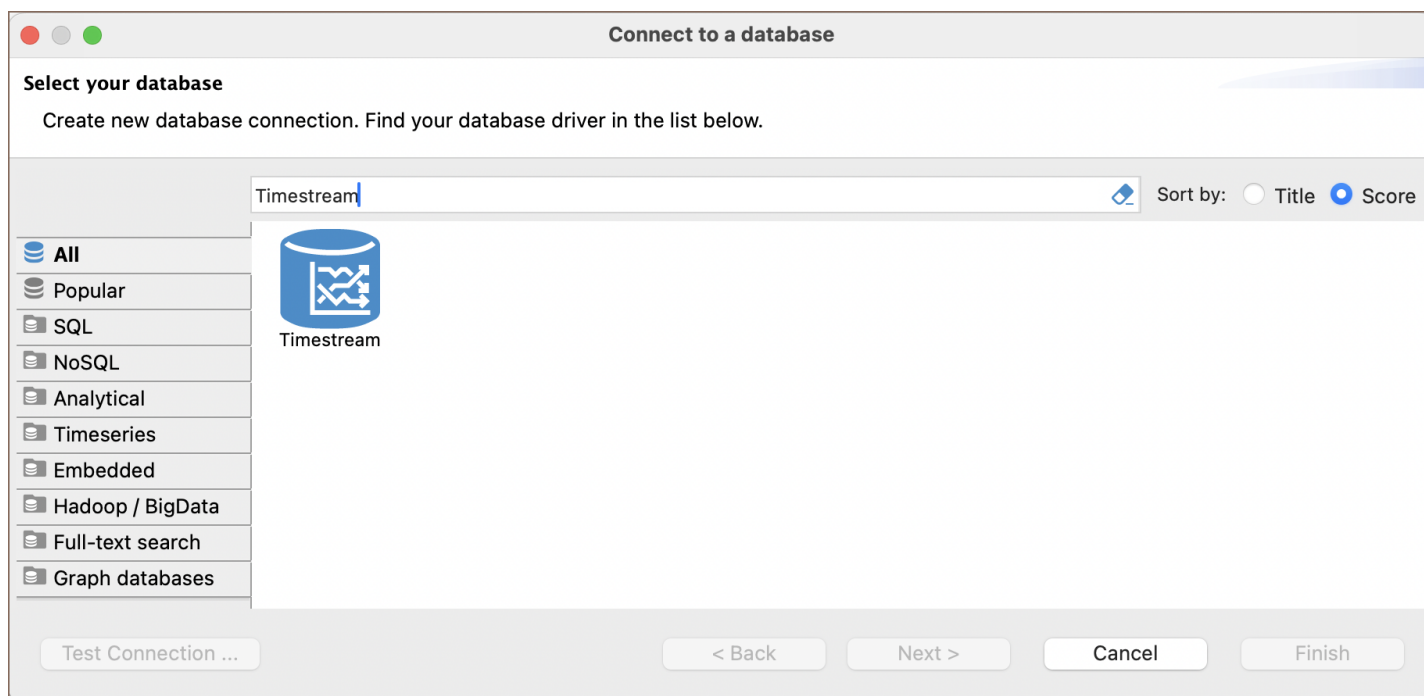
[Timestream database objects](#)

[Timestream features in DBeaver](#)

Overview

This documentation outlines the process of configuring and using AWS Timestream with DBeaver. For more information, visit the official [AWS Timestream website](#).

To manage databases using AWS Timestream in DBeaver, establishing a connection is essential. This is done by selecting the **Timestream** option in the database connection settings of DBeaver. For those new to this process, our [Creating a Connection](#) article provides detailed guidance.



Tip: Users with [Ultimate](#) and [Team](#) editions of DBeaver can connect to their AWS Timestream databases using the [Cloud Explorer](#).

Timestream specialty

AWS Timestream, while optimized for time-series data, lacks some features common in traditional relational databases. It does not have features like constraints, foreign keys, triggers, views, or complex joins, and sub-queries limit DBeaver's capabilities when working with Timestream. For more detailed information, refer to the AWS Timestream database documentation. Due to the absence of certain features in AWS Timestream, there is an inability to link tables, automate tasks based on database changes, create virtual tables from queries, or conduct an in-depth analysis of how data is related within Timestream. For more detailed information, refer to the [AWS Timestream database documentation](#).

Important: In DBeaver, AWS Timestream is read-only.

For a detailed overview of the supported features and capabilities of AWS Timestream within DBeaver, please see the section on [Powering Timestream with DBeaver](#).

Setting up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, Proxies, and Kubernetes.

Timestream connection settings


This subsection will outline the settings for establishing a direct connection to a Timestream database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Timestream database.

The connection settings page requires you to fill in specific fields to establish the initial connection.

Connect to a database

Timestream Connection Settings

Timestream connection settings

 Amazon Timestream

Main

Driver properties

SSH

+ Network configurations...

Settings

AWS Region:

Authentication

Authentication:

AWS Timestream IAM

Credentials:

Access/secret keys

[Details](#)

Access key: AWS access key Secret key: AWS secret key

☒ Save credentials

Role Name: Optional AWS role (ARN or simple name)

☐ 3rd party account

☐ Use AWS Secrets Manager

?

 You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: Timestream

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Region	Specify the AWS region where your Timestream database is located, such as <code>us-west-1</code> .
Authentication	<div>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</div> <div><div>- AWS Timestream IAM</div><div>- DBeaver Profile</div></div> <div>You can also read about security in DBeaver PRO.</div>

Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings, configure them here. For a detailed guide on driver properties, please refer to the official AWS Timestream documentation .

Connection details

The **Connection Details** section in DBeaver allows for further customization of your Timestream connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- Security Settings Guide
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Timestream driver properties

The settings for Timestream **Driver properties** enable you to adjust the performance of the Timestream JDBC driver.

For a complete walkthrough on setting up Timestream JDBC driver properties, you can refer to the official [Timestream JDBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Timestream database connections.

You can customize the Timestream driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your Timestream database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

Secure connection configurations

DBeaver supports secure connections to your Timestream database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering Timestream with DBeaver

DBeaver provides a host of features designed for Timestream databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

Timestream database objects

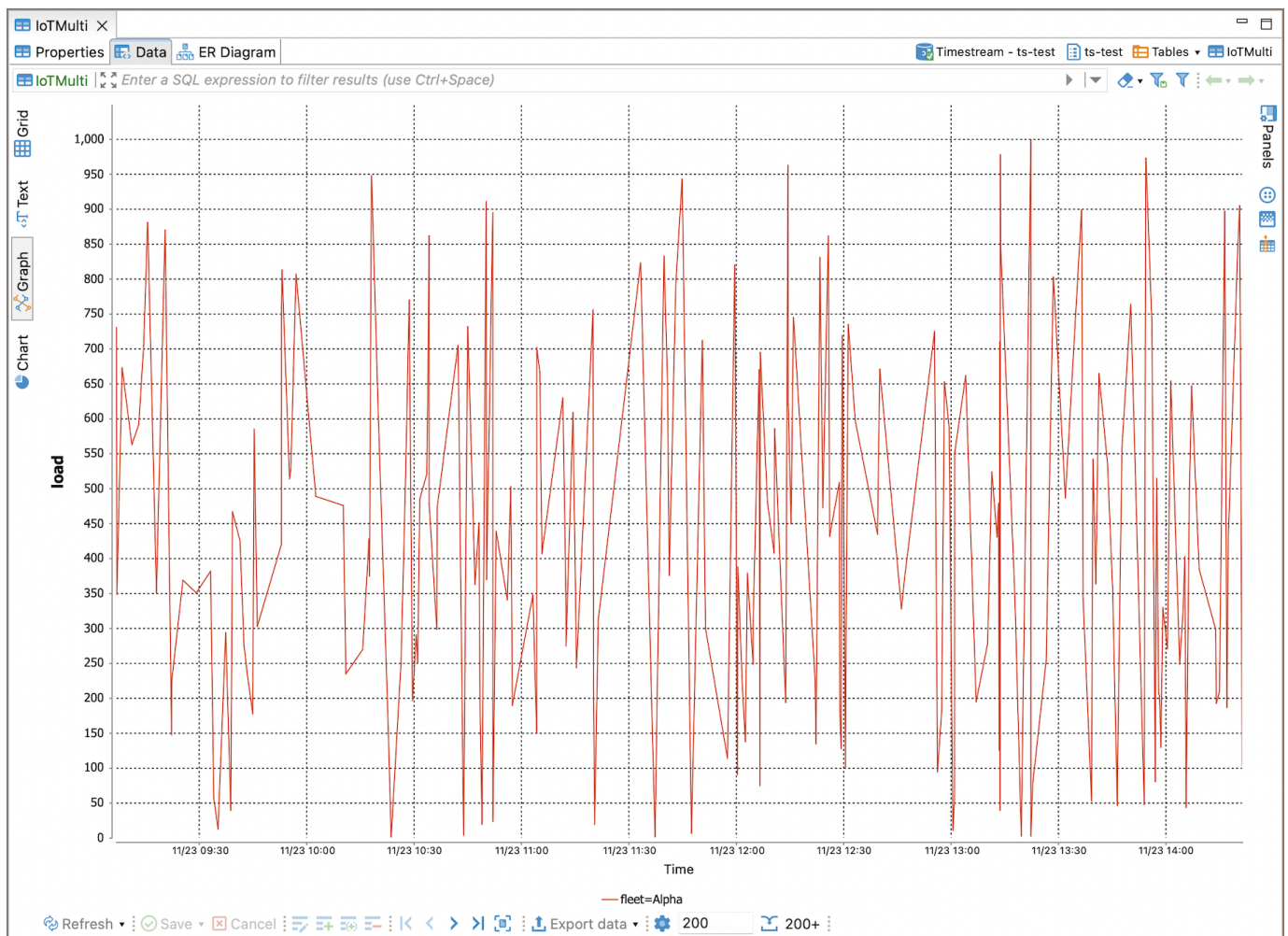
DBeaver lets you view a few of the Timestream database objects, such as:

- Schemas
 - Tables
 - Columns

Timestream features in DBeaver

DBeaver provides the following features compatible with Timestream, but not exclusive to it:

- **Data Export:** Enables the exportation of data. See [Data Export](#) for more details.
- **Graph Construction:** Allows the building of detailed and informative graphs.



CosmosDB

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Setting Up](#)

[Secure Connection Configurations](#)

[Differences between CosmosDB and MongoDB in DBeaver](#)

[Session Manager](#)

[Azure Connection](#)

Overview

DBeaver offers extensive support for CosmosDB, a globally distributed, multimodel database service provided by Microsoft Azure.

Before you start, you must create a connection in DBeaver and select CosmosDB. If you have not done this, please refer to our [Database Connection](#) article.

CosmosDB is similar to MongoDB in many ways, and DBeaver's support for CosmosDB includes collections browsing, data viewing, and SQL queries execution. For a detailed guide, refer to the [MongoDB article](#).

Setting Up

Setting up CosmosDB in DBeaver is also similar to setting up MongoDB.

Secure Connection Configurations

DBeaver supports secure connections to your CosmosDB database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

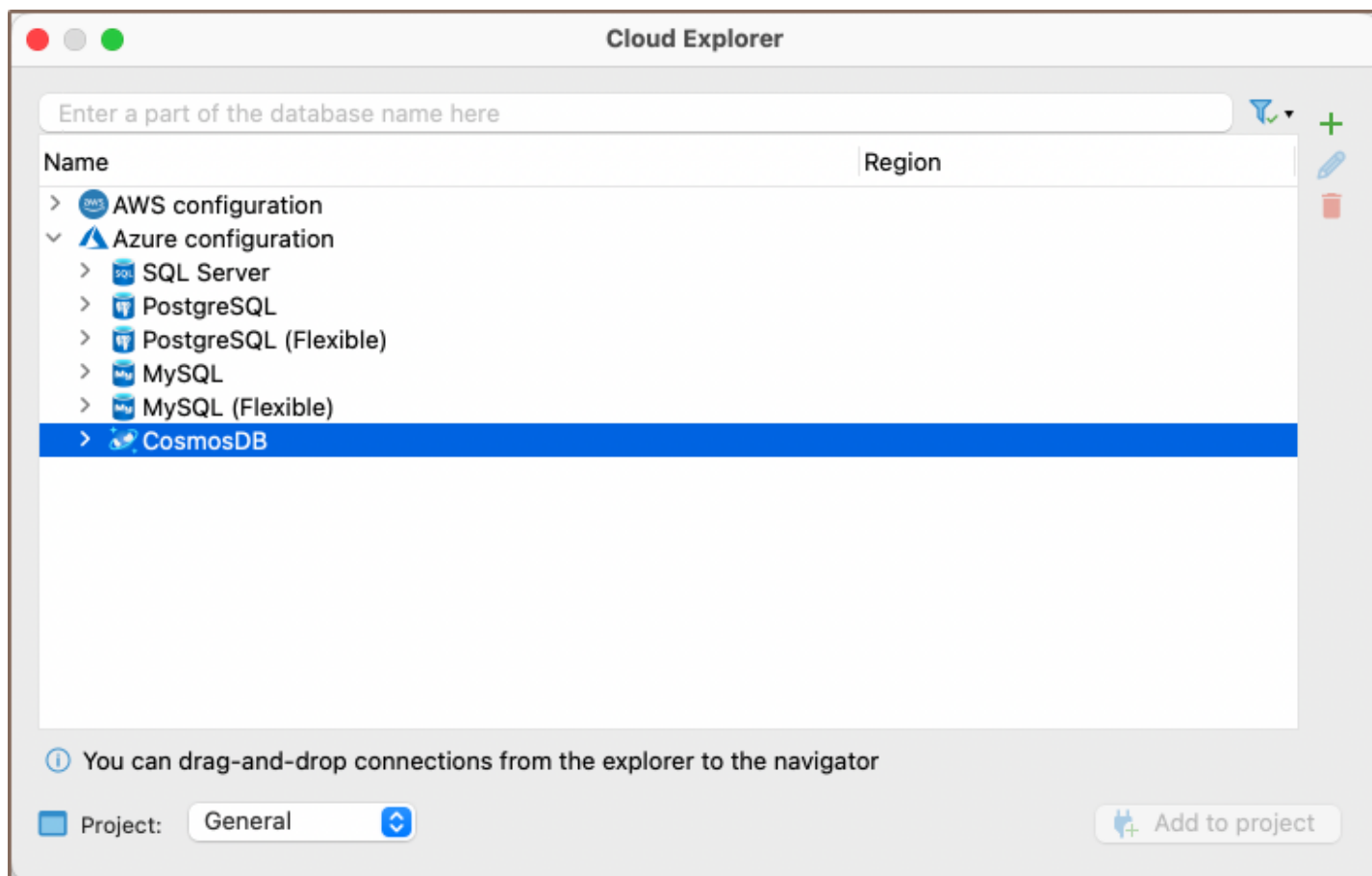
Differences between CosmosDB and MongoDB in DBeaver

Session Manager

Unlike MongoDB, CosmosDB does not have a [Session Manager](#) in DBeaver. This means that you would not be able to manage server sessions directly within DBeaver when using CosmosDB.

Azure Connection

Unique to CosmosDB, DBeaver provides the option to connect through the [Cloud Explorer](#) feature, specifically for Azure. This feature allows you to browse and manage your Azure resources directly from DBeaver. You can view your CosmosDB instances, along with other Azure resources, in a structured tree view.



Databricks

Table of contents

[Overview](#)

[Databricks specialty](#)

[Setting up](#)

[Databricks connection settings](#)

[Connection details](#)

[Databricks driver properties](#)

[Secure connection configurations](#)

[Powering Databricks with DBeaver](#)

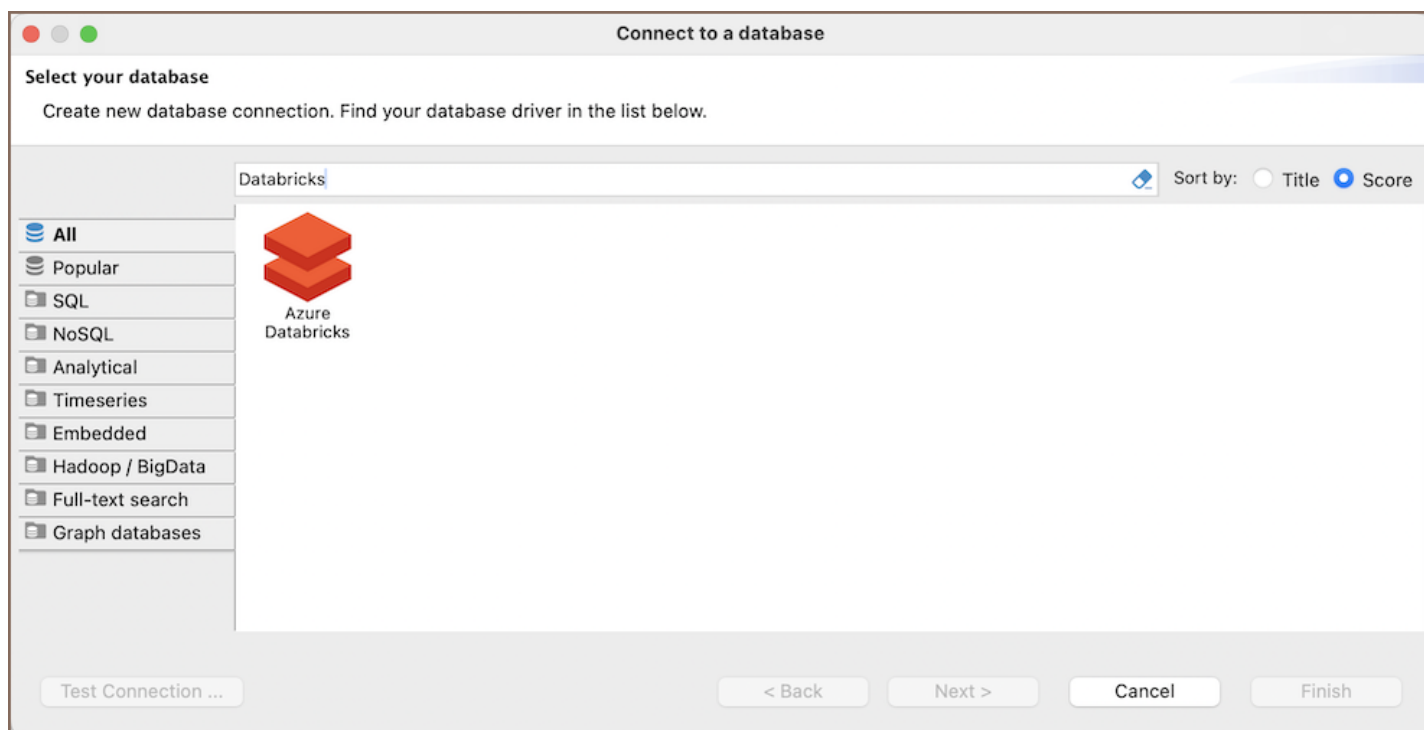
[Databricks database objects](#)

[Databricks Features in DBeaver](#)

Overview

This guide provides instructions on how to set up and use Databricks with DBeaver.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **Databricks** option to connect to the Databricks database. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Databricks specialty

Azure Databricks, based on Apache Spark, is a distributed computing system primarily used for machine learning and big data analytics. Its architecture is specially optimized for handling large-scale data processing and advanced analytics tasks. Azure Databricks integrates with MLlib in Apache Spark, offering scalable algorithms and utilities for machine learning tasks. It also facilitates collaborative work through its notebooks, allowing teams to share code, visualizations, and insights effectively. For more detailed information and a comprehensive understanding of Azure Databricks, please refer to the [official documentation](#).

In the section [Powering Databricks with DBeaver](#) you can find features supported by DBeaver that are compatible with Azure Databricks.

Setting up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH and proxies for Databricks.

Databricks connection settings

In this subsection, we will outline the settings for establishing a direct connection to a Databricks database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Databricks database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Generic JDBC Connection Settings

Azure Databricks connection settings

Main

Driver properties

SSH

+ Network configurations...

General

JDBC URL:

Authentication

Authentication:

Database Native

Username:

Password: ☒ Save password

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:

Azure Databricks

Driver Settings

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
JDBC URL	Enter the URL of your Databricks database here.
Authentication	<div>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</div> <div><div>- Native Database Authentication</div><div>- DBeaver Profile Authentication ★</div></div> <div>You can also read about security in DBeaver PRO.</div>

Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with a Databricks database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- Security Settings Guide
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Databricks driver properties

The settings for Databricks **Driver properties** enable you to adjust the performance of Databricks JDBC driver. These adjustments can influence the efficiency, compatibility, and features of your Databricks database.

For a complete walkthrough on setting up Databricks JDBC driver properties, you can refer to the official [Databricks JDBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Databricks database connections.

You can customize the Databricks driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your

Databricks database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

Secure connection configurations

DBeaver supports secure connections to your Databricks database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#) ★
- [AWS SSM](#) ★

Powering Databricks with DBeaver

DBeaver provides a host of features designed for Databricks databases. This includes the ability to view and manage databases, along with numerous unique capabilities aimed at optimizing database operations.

Databricks database objects

DBeaver lets you view and manipulate a wide range of Databricks database objects. DBeaver has extensive support for various Databricks metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Tables
 - Columns
 - Keys
 - Foreign keys
 - References

- Views
- Procedures

Databricks Features in DBeaver

DBeaver is not limited to typical SQL tasks. It also includes numerous additional features compatible with Databricks, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Data Visualization	ERD Guide
Data Generation	Mock Data Generation

AlloyDB for PostgreSQL

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[AlloyDB specialty](#)

[Setting Up](#)

[AlloyDB connection settings](#)

[Connection details](#)

[Driver properties](#)

[ODBC and JDBC driver configuration](#)

[Secure connection configurations](#)

[Powering AlloyDB with DBeaver](#)

[AlloyDB database objects](#)

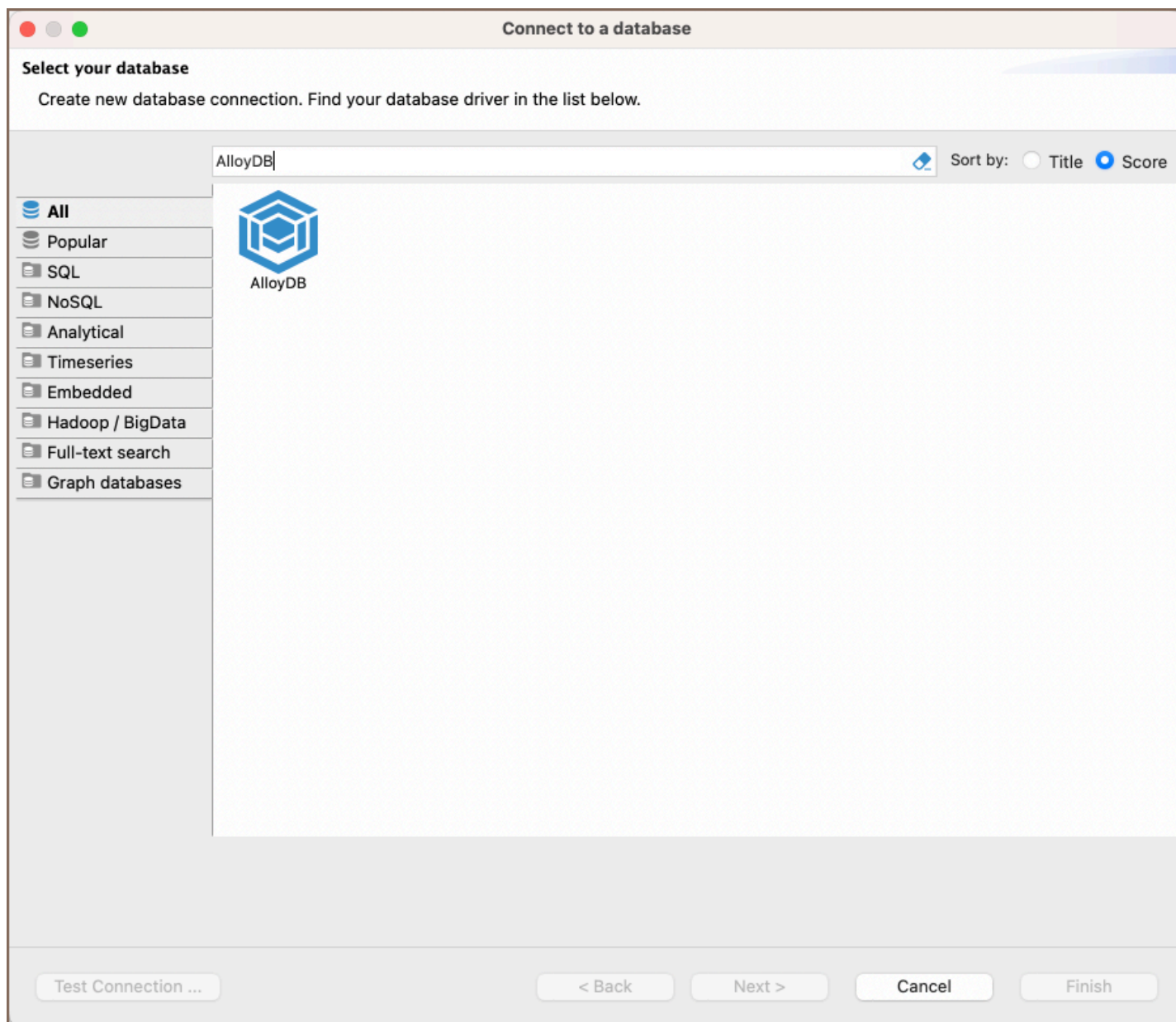
[AlloyDB additional features in DBeaver](#)

Overview

AlloyDB for PostgreSQL is a fully managed database service, tailored for demanding enterprise workloads. It blends the reliability and familiarity of PostgreSQL with Google's advanced technology. This service delivers enhanced performance, scalability, and high availability, making it an ideal choice for critical business applications.

For more detailed information, visit [Google AlloyDB](#) webpage.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **AlloyDB** option to connect to the AlloyDB database. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



AlloyDB specialty

AlloyDB for PostgreSQL is a fully managed database service, tailored for demanding enterprise workloads. It blends the reliability and familiarity of PostgreSQL with Google's advanced technology and supports rich metadata structures, including tables, views, indexes, sequences, and various data types. Its setup in DBeaver mostly replicates the PostgreSQL driver configuration. AlloyDB includes support for SQL triggers, stored procedures, and user-defined functions, enhancing its utility for complex data operations. For a more comprehensive understanding of AlloyDB, please refer to the [official documentation](#).

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, SSL, and the setup of ODBC/JDBC drivers for AlloyDB.

Note: The connection process in DBeaver utilizes the standard PostgreSQL driver, tailored to interact with the AlloyDB service, highlighting its cloud-based characteristics.

AlloyDB connection settings


In this subsection, we will outline the settings for establishing a direct connection to an AlloyDB database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your AlloyDB database.

1) The first page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Connection Settings

AlloyDB connection settings

 AlloyDB

Main AlloyDB Driver properties SSH SSL

+ Network configurations...

Server

Connect by:
☒ Host
☐ URL

Driver type:
JDBC

URL: jdbc:postgresql://localhost:5432/postgres

Host: localhost
Port: 5432

Database: postgres

Authentication

Authentication: Database Native

Username: postgres

Password:
☒ Save password

Advanced

Session role:

Local Client: PostgreSQL 16.0

ⓘ You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: AlloyDB

Driver Settings
Driver license

Test Connection ...

< Back
Next >
Cancel
Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your AlloyDB database.
URL	If you are connecting via URL, enter the URL of your AlloyDB database here. This field is hidden if you are connecting via the host.
Host	If you are connecting via host, enter the host address of your AlloyDB database here.
Database	Enter the name of the AlloyDB database you want to connect to.
Show all databases	Displays all databases in the database navigator . Note: This option works only when the Host connection type is selected. It does not work when connecting via URL .


Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Native Database Authentication - DBeaver Profile Authentication ★ - PostgreSQL Kerberos Authentication ★ - PostgreSQL PgPass Authentication - PostgreSQL SSPI Authentication (Windows SSO) ★ <p>You can also read about security in DBeaver PRO.</p>
Local Client	Specify the local client, see our article about local client configuration .
Session role	This option allows you to define a specific role that your database session assumes upon connection. Setting a session role can be useful for managing permissions and access control within your AlloyDB database.
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

2) The second page of the connection settings offers additional options that allow you to customize your further connection to the AlloyDB database.


Connect to a database

Connection Settings

AlloyDB connection settings

 AlloyDB

Main
AlloyDB
Driver properties
SSH
SSL

 Network configurations...

Settings

☐ Show all databases
☐ Show template databases
☐ Show databases not available for connection
☐ Show database statistics
☐ Read all data types
☐ Read table keys with columns

SQL

Show \$\$ quote as: Code block

Show \$tagName\$ quote as: Code block

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Show template databases	Displays template databases. This option is available when Show all databases is checked.
Show databases unavailable for connection	Show databases not available for connection in database list. This option is available when Show all databases is checked.
Show databases statistics	Displays statistics for databases.
Read all data types	Reads all data types from the server (including arrays).
Read table keys with columns	Enable reading table constraints during the column reading stage to ensure accurate recognition of keys. Note that this setting may decrease metadata loading performance.

Show <code>\$\$</code> quote as	Displays <code>\$\$</code> quotes as code blocks or strings.
Show <code>\$tagName\$</code> quote as	Displays <code>\$tagName\$</code> quotes as code blocks or strings.
Use prepared statements	Uses prepared statements for SQL execution.

Connection details

The **Connection Details** section in DBeaver allows for further customization of your AlloyDB connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Driver properties

While managing AlloyDB with DBeaver, it's important to understand that the underlying driver is the standard PostgreSQL JDBC and ODBC driver.

The settings for AlloyDB **Driver properties** enable you to adjust the performance of the JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your AlloyDB database.

For a complete walkthrough on setting up JDBC and ODBC drivers properties, you can refer to the official [JDBC documentation](#) and [ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize AlloyDB database connections.

You can customize the driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your AlloyDB database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC driver configuration

DBeaver provides extensive capabilities for managing AlloyDB database connections via ODBC/JDBC drivers. This functionality enables you to connect to your AlloyDB database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

You can find a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver in our [ODBC Driver Configuration](#) article.

Secure connection configurations

DBeaver supports secure connections to your AlloyDB database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering AlloyDB with DBeaver

DBeaver provides a host of features designed for AlloyDB databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

AlloyDB database objects

DBeaver lets you view and manipulate a wide range of AlloyDB database objects. DBeaver has extensive support for various AlloyDB metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Schemas
 - Data types
 - Tables
 - Columns
 - Constraints
 - Indexes
 - Foreign Keys
 - Dependencies
 - References
 - Partitions
 - Triggers
 - Rules
 - Policies
 - Foreign Tables
 - Views
 - Materialized Views
 - Functions
 - Sequences
 - Data types

- Aggregate functions
- Event Triggers
- Extensions
- Storage
 - Tablespaces
- Roles
- Administer
 - Jobs
 - Session Manager
 - Lock Manager

AlloyDB additional features in DBeaver

DBeaver provides additional features compatible with AlloyDB, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Lock Management	Lock Manager
Backup and Restore	How to Backup/Restore data
Schema Management	Schema Compare
Data Visualization	GIS Guide
	ERD Guide

BigQuery

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Feature highlights](#)

[Setting up](#)

[BigQuery connection settings](#)

[Connection details](#)

[Secure connection configurations](#)

[Powering BigQuery with DBeaver](#)

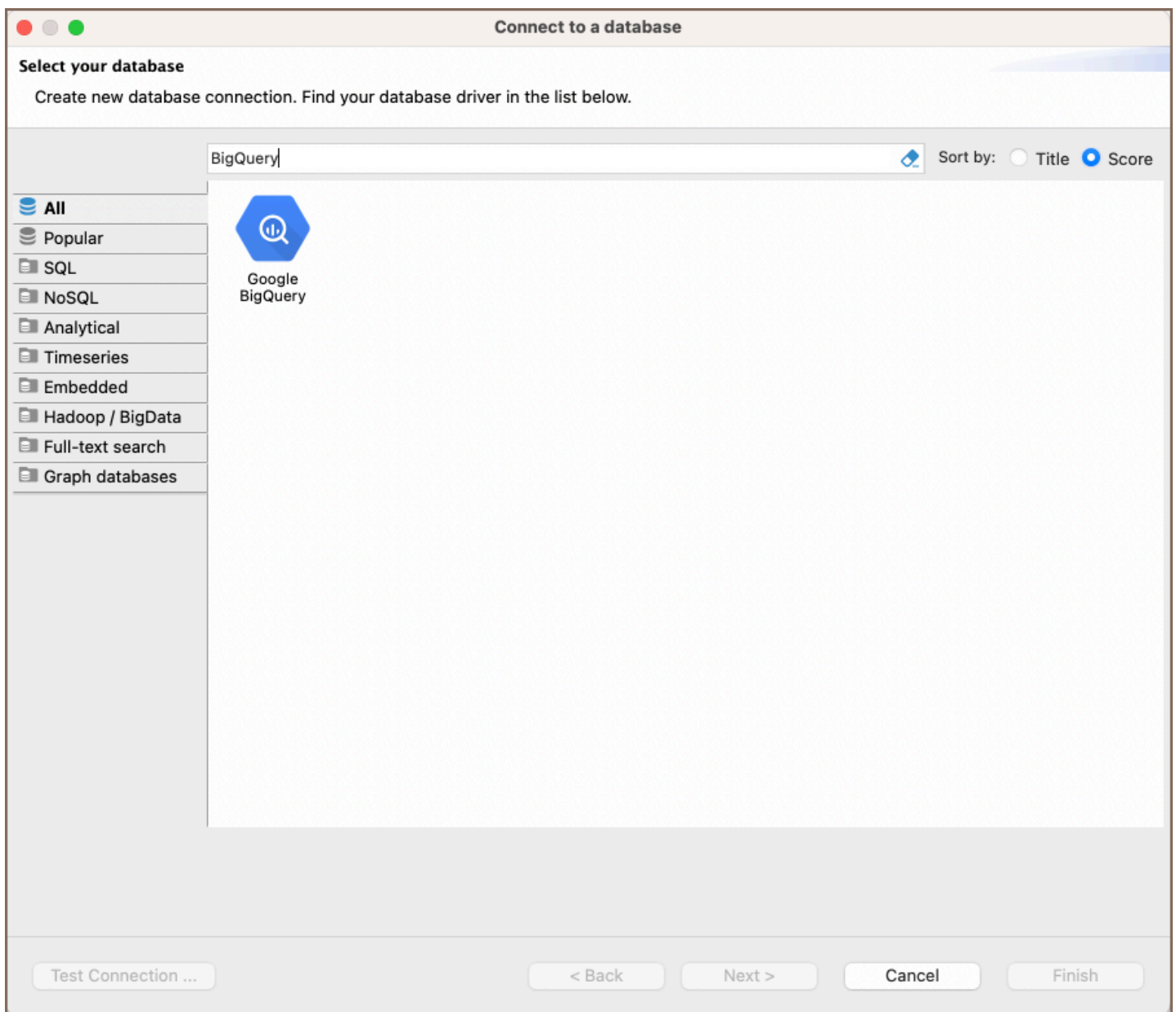
[BigQuery database objects](#)

[BigQuery additional features in DBeaver](#)

Overview

This documentation details the steps for configuring and utilizing BigQuery with DBeaver. The integration in DBeaver supports distinctive BigQuery functionalities, accommodating its serverless infrastructure, robust scalability, and compatibility with multi-cloud data analytics.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **BigQuery** option to connect to the BigQuery. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Feature highlights

DBeaver's database management capabilities include support for various data types tailored for complex data operations, such as:

- **BIGNUMERIC:** Handles large-scale or high-precision decimal numbers.
- **GEOGRAPHY:** Manages spatial information.
- **INTERVAL:** Tracks durations with precision.

Additionally, DBeaver provides tools for database schema customization and task automation:

- **Views:** Enables the creation of virtual tables based on SQL queries.

- **Procedures:** Allows for the definition and execution of stored procedures for routine database tasks.

Setting up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, Proxies, and Kubernetes.

BigQuery connection settings


In this subsection, we will outline the settings for establishing a direct connection to a BigQuery database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your BigQuery database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Google BigQuery Connection Settings

Google BigQuery connection settings



Main

Driver properties

SSH

+ Network configurations...

Connection

Project:

Additional project(s):

Server info

Host:

https://www.googleapis.com/bigquery/v2

Port:

443

Authentication

Authentication:

Google Cloud IAM (BigQuery)

Credentials:

Default credentials

Details

☐ Enable IAM

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: Google BigQuery

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Project	Enter the project ID for the Google Cloud Project where the BigQuery service is located.
Additional project(s)	Specify any additional project IDs if applicable.
Host	Set the host URL to the BigQuery API endpoint.
Port	Enter the port number for your BigQuery database. The default BigQuery port is 443 .
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Google Cloud IAM - DBeaiver Profile Authentication

	You can also read about security in DBeaver PRO .
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings, configure them here.

Connection details

The **Connection Details** section in DBeaver allows for further customization of your BigQuery connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Secure connection configurations

DBeaver supports secure connections to your BigQuery database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)

- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering BigQuery with DBeaver

DBeaver provides a host of features designed for BigQuery databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

BigQuery database objects

DBeaver lets you view and manipulate a range of BigQuery database objects, such as:

- Schemas
 - Tables
 - Columns
 - Keys
 - Foreign keys
 - Indexes
 - References
- Views
- Procedures

BigQuery additional features in DBeaver

DBeaver provides additional features compatible with BigQuery, but not exclusive to it:

Category	Feature
Data Transfer	Data Import

	Data Export
Schema Management	Schema Compare
	ERD Guide
Data Generation	Mock Data Generation
Query insights	Result details

Bigtable

Note: This driver is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Bigtable connection](#)

Supported features:


- Table data view
- Table data edit in document (json) mode
- Data filters
- SQL queries execution
- Data export and import

Bigtable connection

Connection "BigTable - test-bq" configuration

Connection settings

BigTable connection settings



Connection settings

Initialization

Shell Commands

Client identification

Transactions

General

Metadata

Errors and timeouts

> Data editor

> SQL Editor

MainSSHProxy

Authentication

Auth type:Service accountUse default credentials

Auth key file path:C:\keys\google-cloud\pragmatic-blend-123456-e2572b125d16.json

Settings

Project:pragmatic-blend-123456

Instance:

You can use variables in connection parameters.

Driver name: BigTable

Edit Driver Settings

?

Test Connection ...

OK

Cancel

Cloud SQL for MySQL

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Feature highlights](#)

[Setting Up](#)

[Google Cloud MySQL connection settings](#)

[Connection details](#)

[Driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering Google Cloud MySQL with DBeaver](#)

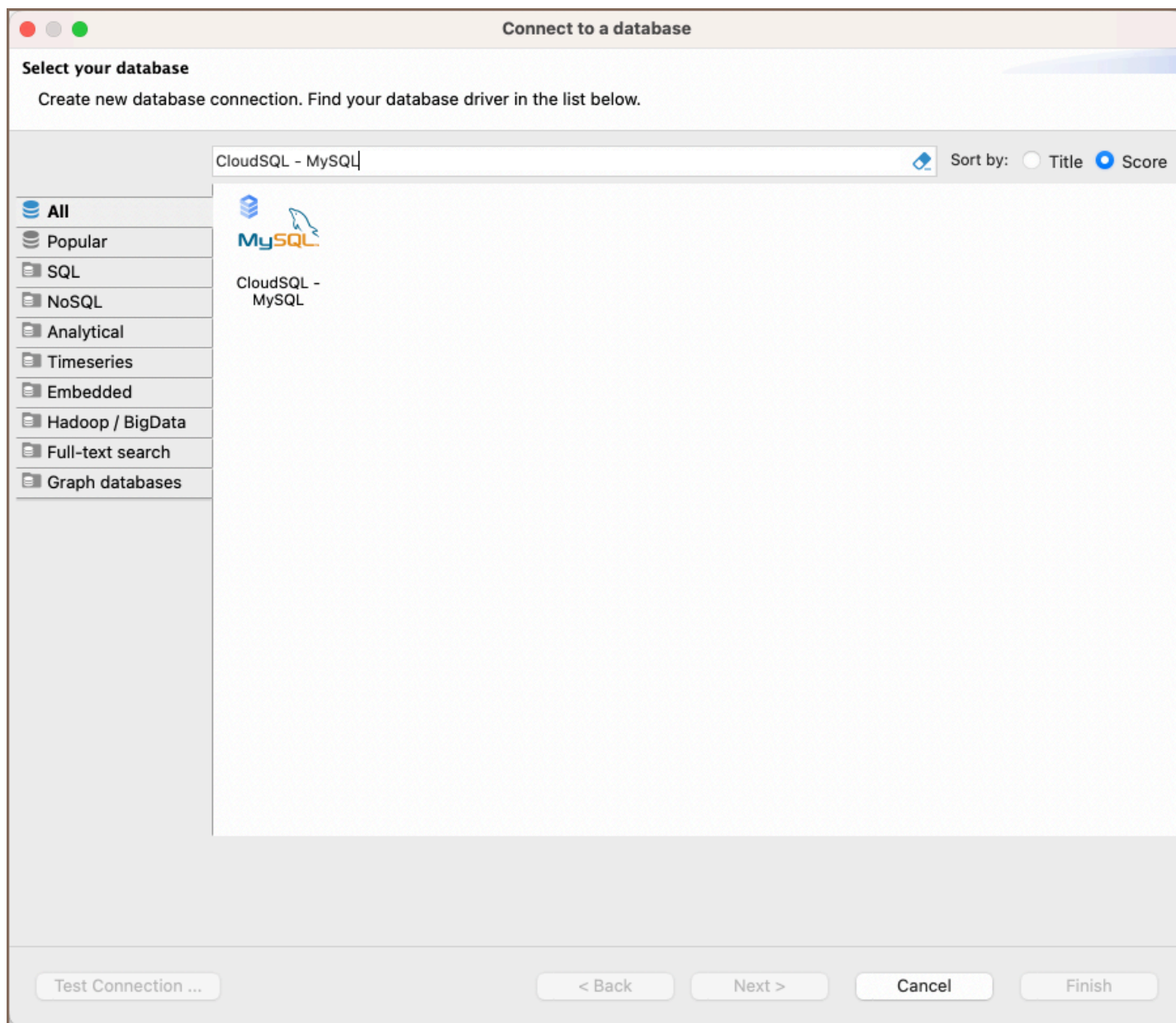
[Google Cloud MySQL database objects](#)

[MySQL additional features in DBeaver](#)

Overview

Google Cloud SQL for MySQL is a managed database service provided by [Google Cloud](#), offering the robust capabilities of MySQL with the added benefits of cloud hosting. This guide is designed to assist in the setup and management of Google Cloud SQL for MySQL databases using DBeaver.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the `CloudSQL - MySQL` option to connect to the Google Cloud SQL service. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Tip: Users with [Ultimate](#) and [Team](#) editions of DBeaver can connect to their Google Cloud SQL for MySQL databases using the [Cloud Explorer](#).

Feature highlights

DBeaver extends beyond standard SQL tasks, offering a suite of MySQL-specific features. These capabilities enhance the functionality and management of Google Cloud MySQL databases, catering to specific needs and advanced use cases.

- **Data Types:** DBeaver fully supports MySQL-specific data types such as Enumerations and Sets. This enables precise data structuring and validation directly within the interface.

- **Security Management:** An integral part of managing databases is user security. DBeaver includes comprehensive tools for MySQL User Management, allowing for detailed control over user permissions and roles.
- **Storage Engines:** Understanding and choosing the right storage engine is crucial for database performance. DBeaver supports various MySQL storage engines, including InnoDB, MyISAM, and specialized engines like MRG_MyISAM, Archive, Blackhole, CSV, Memory, and Performance_schema.
- **Database Management Tools:** DBeaver provides advanced database management features specific to MySQL. This includes the management of MySQL Events and Stored Procedures, allowing for automation and efficient management of routine database tasks.

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, SSL, and the setup of ODBC/JDBC drivers for Google Cloud MySQL.

Note: The connection process in DBeaver utilizes the standard MySQL driver, tailored to interact with the Google Cloud SQL for MySQL service, highlighting its cloud-based characteristics.

Google Cloud MySQL connection settings

In this subsection, we will outline the settings for establishing a direct connection to a Google Cloud MySQL database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Google Cloud MySQL database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Connection Settings

CloudSQL - MySQL connection settings

Main

Driver properties

SSH

SSL

+ Network configurations...

Server

Connect by:

☒ Host
☐ URL

Driver type:

JDBC

URL:

jdbc:mysql:///mysql?cloudSqlInstance=project:region:instance

Host/Instance:

project:region:instance

Database:

mysql

Authentication

Authentication:

Google Cloud IAM

Credentials:

Default credentials

Details

User:

root

Password:

☒ Save password
☐

☐ Enable IAM

Advanced

Server Time Zone:

Auto-detect

Local Client:

MySQL 8.1.0

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:

CloudSQL - MySQL

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your Google Cloud MySQL database.
URL	If you are connecting via URL, enter the URL of your Google Cloud MySQL database here. This field is disabled if you're connecting via the host.
Host/Instance	For a direct connection to a Google Cloud SQL instance, use the format <code>project:region:instance</code> . Here, <code>project</code> refers to your Google Cloud project ID, <code>region</code> denotes the geographical location of your instance, and <code>instance</code> is the unique ID of your Google Cloud SQL instance.
Database	Enter the name of the Google Cloud MySQL database you want to connect to.
	Choose the type of authentication you want to use for the connection. For detailed guides on authentication

Authentication	<p>types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Google Cloud IAM - DBever Profile Authentication <p>You can also read about security in DBever PRO.</p>
Local Client	Specify the local client, see our article about local client configuration .
Server Time Zone	This setting is responsible for determining the time zone used by the Google Cloud MySQL server.
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBever allows you to customize your experience while working with Google Cloud MySQL database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Driver properties

While managing MySQL databases hosted on Google Cloud with DBeaver, it's important to understand that the underlying driver is the standard MySQL JDBC and ODBC driver.

The settings for Google Cloud MySQL **Driver properties** enable you to adjust the performance of the MySQL JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your Google Cloud MySQL database.

For a complete walkthrough on setting up MySQL JDBC and ODBC drivers properties, you can refer to the official [MySQL JDBC documentation](#) and [MySQL ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Google Cloud MySQL database connections.

You can customize the MySQL driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your Google Cloud MySQL database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing Google Cloud MySQL database connections via ODBC/JDBC drivers. This functionality enables you to connect to your Google Cloud MySQL database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

This process is similar across most databases that support ODBC/JDBC drivers. For a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver, you can refer to our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your Google Cloud MySQL database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)

- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering Google Cloud MySQL with DBeaver

DBeaver offers an array of features specifically tailored for MySQL databases, including those hosted on Google Cloud. This includes the ability to view and manage databases, along with numerous unique capabilities aimed at optimizing database operations.

Google Cloud MySQL database objects

DBeaver lets you view and manipulate a wide range of database objects. DBeaver has extensive support for various MySQL metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Tables
 - Columns
 - Constraints
 - Indexes
 - Foreign Keys
 - References
 - Triggers
 - Views
 - Indexes
 - Procedures

- Triggers
- Events
- Users
- Administer
- Session Manager

MySQL additional features in DBeaver

DBeaver provides additional features compatible with Google Cloud MySQL, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Backup and Restore	How to Backup/Restore data
Schema Management	Schema Compare
Data Visualization	GIS Guide
	ERD Guide
	Query Execution plan

Cloud SQL for PostgreSQL

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Feature Highlights](#)

[Setting Up](#)

[PostgreSQL connection settings](#)

[Connection details](#)

[Driver properties](#)

[ODBC and JDBC driver configuration](#)

[Secure connection configurations](#)

[Powering Google Cloud PostgreSQL with DBeaver](#)

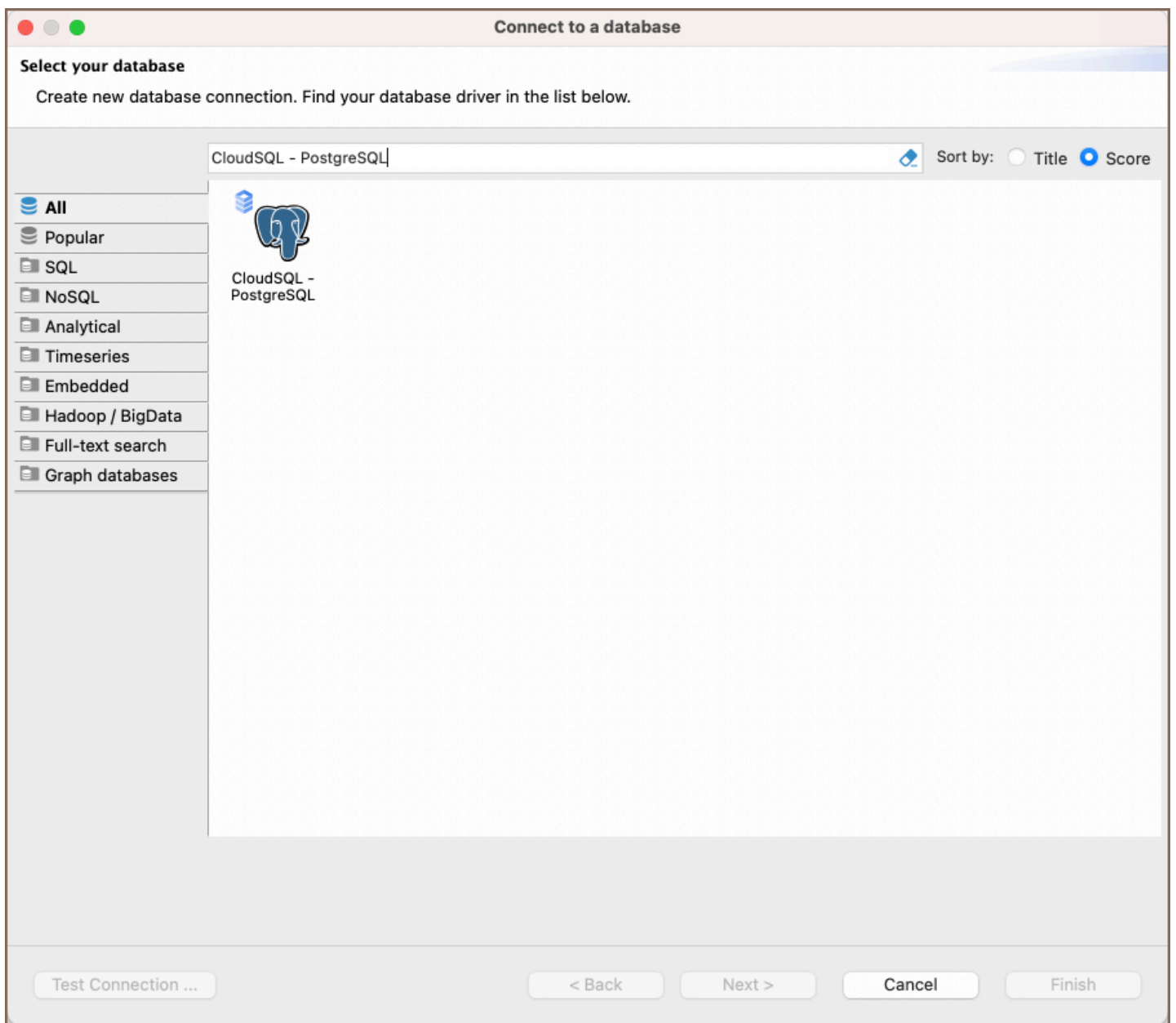
[Google Cloud PostgreSQL database objects](#)

[Google Cloud PostgreSQL additional features in DBeaver](#)

Overview

Google Cloud SQL for PostgreSQL is a managed database service provided by [Google Cloud](#), offering the robust capabilities of Google Cloud PostgreSQL with the added benefits of cloud hosting. This guide is designed to assist in the setup and management of Google Cloud SQL for PostgreSQL databases using DBeaver.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **CloudSQL - PostgreSQL** option to connect to the Google Cloud SQL service. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Tip: Users with [Ultimate](#) and [Team](#) editions of DBeaver can connect to their Google Cloud SQL for PostgreSQL databases using the [Cloud Explorer](#).

Feature Highlights

DBeaver extends beyond standard SQL tasks, offering a suite of PostgreSQL-specific features. These capabilities enhance the functionality and management of Google Cloud PostgreSQL databases, catering to specific needs and advanced use cases.

- **Data Types:** DBeaver supports complex data types unique to this system. This includes Arrays and various structures, enabling intricate data organization and manipulation.

- **Extensions and Security:** DBeaver includes robust support for Extensions, broadening the database's functionality. It also provides comprehensive security features, encompassing Permissions, Policies, and Roles management, ensuring thorough control over database access and operations.
- **Data Organization:** For effective data handling, DBeaver supports Partitions. This feature assists in efficient data segmentation and organization, enhancing database performance and management.
- **External Data Integration:** The platform also facilitates the integration of Foreign Tables. This feature enables seamless access and manipulation of external data sources, providing a more flexible and expansive data management environment.

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, SSL, and the setup of ODBC/JDBC drivers for Google Cloud PostgreSQL.

Note: The connection process in DBeaver utilizes the standard PostgreSQL driver, tailored to interact with the Google Cloud SQL for PostgreSQL service, highlighting its cloud-based characteristics.

PostgreSQL connection settings


In this subsection, we will outline the settings for establishing a direct connection to a Google Cloud PostgreSQL database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Google Cloud PostgreSQL database.

- 1) The first page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Connection Settings

CloudSQL - PostgreSQL connection settings



Main

CloudSQL - PostgreSQL

Driver properties

SSH

SSL

+ Network configurations...

Server

Connect by:

☒ Host
☐ URL

Driver type:

JDBC

URL:

jdbc:postgresql:///postgres?cloudSqlInstance=project:region:instance

Host/Instance:

project:region:instance

Database:

postgres

☐ Show all databases

Authentication

Authentication:

Google Cloud IAM

Credentials:

Default credentials

Details

User:

postgres

Password:

•••••

☒ Save password

☐ Enable IAM

Advanced

Session role:

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:

CloudSQL - PostgreSQL

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your Google Cloud PostgreSQL database.
URL	If you are connecting via URL, enter the URL of your Google Cloud PostgreSQL database here. This field is hidden if you are connecting via the host.
Host/Instance	For a direct connection to a Google Cloud SQL instance, use the format <code>project:region:instance</code> . Here, <code>project</code> refers to your Google Cloud project ID, <code>region</code> denotes the geographical location of your instance, and <code>instance</code> is the unique ID of your Google Cloud SQL instance.
Database	Enter the name of the Google Cloud PostgreSQL database you want to connect to.

DBeaver User Guide 24.1.ea. Page 477 of 1174.


Show all databases	<p>Displays all databases in the database navigator.</p> <p>Note: This option works only when the Host connection type is selected. It does not work when connecting via URL.</p>
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Google Cloud IAM - DBeaiver Profile Authentication <p>You can also read about security in DBeaiver PRO.</p>
Local Client	<p>Specify the local client, see our article about local client configuration.</p>
Connection Details	<p>Provide additional connection details if necessary.</p>
Driver Name	<p>This field will be auto-filled based on your selected driver type.</p>
Driver Settings	<p>If there are any specific driver settings, configure them here.</p>

2) The second page of the connection settings offers additional options that allow you to customize your further connection to the Google Cloud PostgreSQL database.

Connect to a database

Connection Settings

CloudSQL - PostgreSQL connection settings



Main

CloudSQL - PostgreSQL

Driver properties

SSH

SSL

+ Network configurations...

Settings

☐ Show template databases
 ☐ Show databases not available for connection
 ☐ Show database statistics
 ☐ Read all data types
 ☐ Read table keys with columns

SQL

Show \$\$ quote as: Code block

Show \$tagName\$ quote as: Code block

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Show template databases	Displays template databases. This option is available when Show all databases is checked.
Show databases unavailable for connection	Displays databases unavailable. This option is available when Show all databases is checked.
Show databases statistics	Displays statistics for databases.
Read all data types	Reads all data types from the server.
Read table keys with columns	Reads table keys with columns.
Show \$\$ quote as	Displays \$\$ quotes as code blocks or strings.

Show <code>\$tagName\$</code> quote as	Displays <code>\$tagName\$</code> quotes as code blocks or strings.
Use prepared statements	Uses prepared statements for SQL execution.

Connection details

The **Connection Details** section in DBeaver allows for further customization of your Google Cloud PostgreSQL connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Driver properties

While managing PostgreSQL databases hosted on Google Cloud with DBeaver, it's important to understand that the underlying driver is the standard PostgreSQL JDBC and ODBC driver.

The settings for Google Cloud PostgreSQL **Driver properties** enable you to adjust the performance of the PostgreSQL JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your Google Cloud PostgreSQL database.

For a complete walkthrough on setting up PostgreSQL JDBC and ODBC drivers properties, you can refer to the official [PostgreSQL JDBC documentation](#) and [PostgreSQL ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Google Cloud PostgreSQL database connections.

You can customize the PostgreSQL driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your Google Cloud PostgreSQL database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC driver configuration

DBeaver provides extensive capabilities for managing Google Cloud PostgreSQL database connections via ODBC /JDBC drivers. This functionality enables you to connect to your Google Cloud PostgreSQL database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

You can find a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver in our [ODBC Driver Configuration](#) article.

Secure connection configurations

DBeaver supports secure connections to your Google Cloud PostgreSQL database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering Google Cloud PostgreSQL with DBeaver

DBeaver provides a host of features designed for Google Cloud PostgreSQL databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

Google Cloud PostgreSQL database objects

DBeaver lets you view and manipulate a wide range of Google Cloud PostgreSQL database objects. DBeaver has extensive support for various PostgreSQL metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Schemas
 - Data types
 - Tables
 - Columns
 - Constraints
 - Indexes
 - Foreign Keys
 - Dependencies
 - References
 - Partitions
 - Triggers
 - Rules
 - Policies
 - Foreign Tables
 - Views
 - Materialized Views
 - Functions

- Sequences
- Data types
- Aggregate functions
- Event Triggers
- Extensions
- Storage
 - Tablespaces
- Roles
- Administer
 - Jobs
 - Session Manager
 - Lock Manager

Google Cloud PostgreSQL additional features in DBeaver

DBeaver provides additional features compatible with Google Cloud PostgreSQL, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Lock Management	Lock Manager
Backup and Restore	How to Backup/Restore data
Schema Management	Schema Compare

Data Visualization	GIS Guide
	ERD Guide
	Query Execution plan

Cloud SQL for SQL Server

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Feature Highlights](#)

[Setting Up](#)

[Google Cloud SQL Server connection settings](#)

[Connection details](#)

[Driver properties](#)

[ODBC and JDBC Driver Configuration](#)

[Secure Connection Configurations](#)

[Powering Google Cloud SQL Server with DBeaver](#)

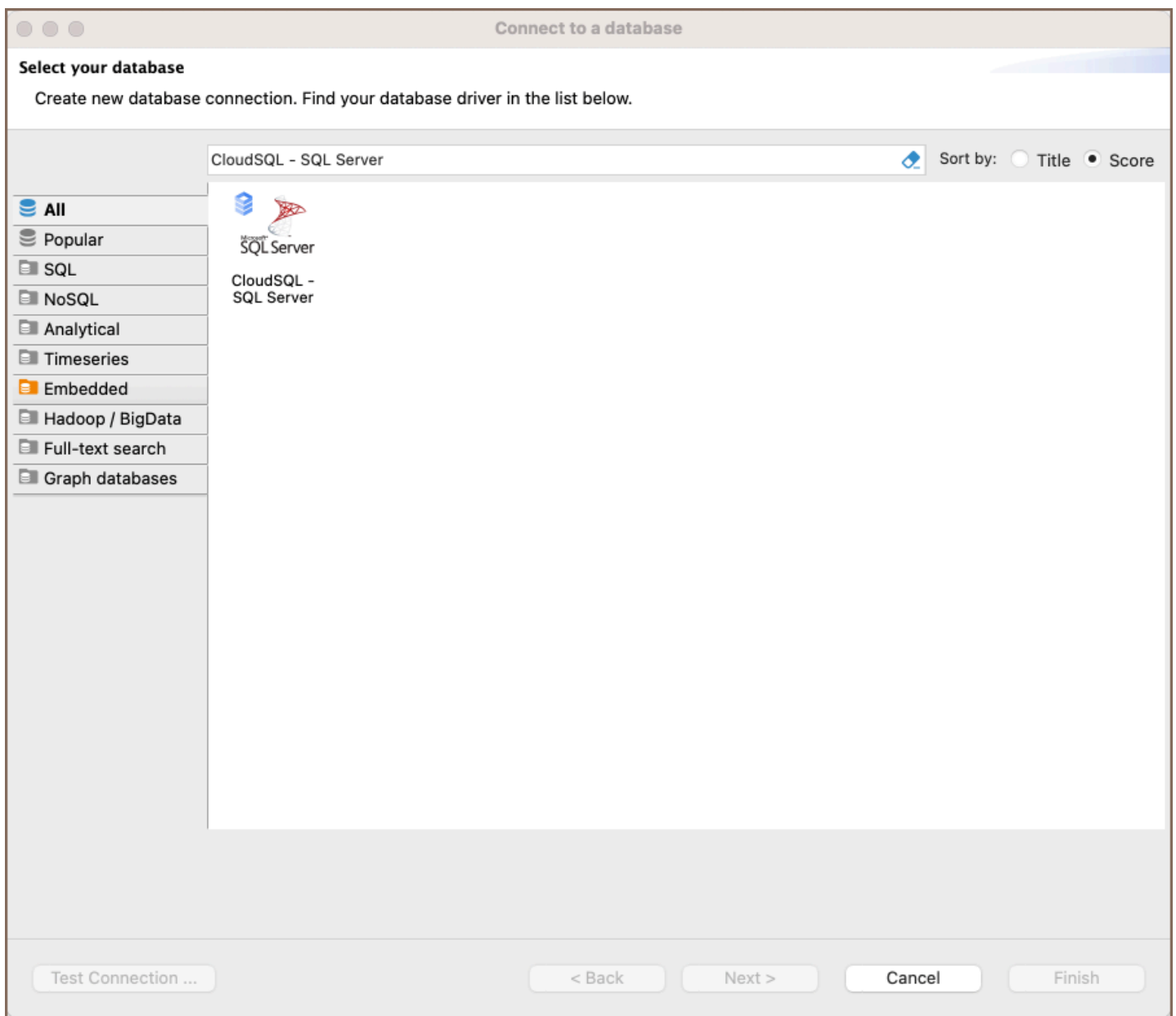
[Google Cloud SQL Server database objects](#)

[Google Cloud SQL Server additional features in DBeaver](#)

Overview

Google Cloud SQL for SQL Server is a managed database service provided by [Google Cloud](#), offering the robust capabilities of Google Cloud SQL Server with the added benefits of cloud hosting. This guide is designed to assist in the setup and management of Google Cloud SQL for SQL Server databases using DBeaver.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **CloudSQL - SQL Server** option to connect to the Google Cloud SQL service. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Tip: Users with [Ultimate](#) and [Team](#) editions of DBeaver can connect to their Google Cloud SQL for SQL Server databases using the [Cloud Explorer](#).

Feature Highlights

DBeaver extends beyond standard SQL tasks, offering a suite of SQL Server-specific features. These capabilities enhance the functionality and management of Google Cloud SQL Server databases, catering to specific needs and advanced use cases.

- **Data Structure:** DBeaver provides extensive support for SQL Server's data structures, ensuring comprehensive management and interaction capabilities. This includes:
 - **SQL Server Tables & Views:** Facilitating advanced data representation and querying capabilities.

- **SQL Server Procedures:** Supporting the creation, modification, and management of stored procedures for efficient data handling.
- **SQL Server Sequences:** Enabling the creation and management of sequences for generating unique numbers.
- **SQL Server Synonyms:** Offering the ability to use synonyms for database objects, simplifying SQL queries and enhancing data accessibility.
- **SQL Server External Tables:** Supporting the integration of external data sources, allowing for extended data management beyond traditional database boundaries.
- **Job Management:** DBeaver extends its functionalities to job management in SQL Server databases. This includes support for Agent Jobs, empowering users to schedule and automate various tasks, enhancing the efficiency of database operations and maintenance.

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, SSL, and the setup of ODBC/JDBC drivers for Google Cloud SQL Server.

Note: The connection process in DBeaver utilizes the standard SQL Server driver, tailored to interact with the Google Cloud SQL for SQL Server service, highlighting its cloud-based characteristics.

Google Cloud SQL Server connection settings


In this subsection, we will outline the settings for establishing a direct connection to a Google Cloud SQL Server database using DBeaver.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

SQL Server Connection Settings

CloudSQL - SQL Server connection settings



Main

Driver properties

SSH

SSL

+ Network configurations...

Server

Connect by:

☒ Host
 ☐ URL

Driver type:

JDBC

URL:

jdbc:sqlserver://localhost;databaseName=master;socketFactoryConstructorArg=project:region:instance

Host:

project:region:instance

Database/Schema:

master

Authentication

Authentication:

SQL Server Authentication

Username:

sqlserver

Password:

☒ Save password

Settings

☐ Show All Schemas

☐ Trust Server Certificate

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:

CloudSQL - SQL Server

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Connect by (Host/URL)	Choose whether you want to connect using a host or a URL.
Driver Type	Select the appropriate driver for your Google Cloud SQL Server database.
URL	If you are connecting via URL, enter the URL of your Google Cloud SQL Server database here. This field is hidden if you are connecting via the host.
Host/Instance	For a direct connection to a Google Cloud SQL instance, use the format <code>project:region:instance</code> . Here, <code>project</code> refers to your Google Cloud project ID, <code>region</code> denotes the geographical location of your instance, and <code>instance</code> is the unique ID of your Google Cloud SQL instance.
Database /Schema	Enter the name of the Google Cloud SQL Server database you want to connect to.

Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Google Cloud IAM - DBeaver Profile Authentication <p>You can also read about security in DBeaver PRO.</p>
Show All Schemas	Shows all database schemas. If disabled shows only non-empty schemas.
Trust Server Certificate	<p>If <code>true</code>, the Google Cloud SQL Server SSL certificate is automatically trusted when the communication layer is encrypted using SSL. If <code>false</code>, the Microsoft JDBC Driver for SQL Server validates the server SSL certificate. If the server certificate validation fails, the driver raises an error and terminates the connection. To configure certificate settings in DBeaver, refer to our detailed article on Truststore.</p>
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings , configure them here.

Connection details

The **Connection Details** section in DBeaver allows you to customize your experience while working with Google Cloud SQL Server database. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)

Driver properties

While managing Google Cloud SQL Server databases hosted on Google Cloud with DBeaver, it's important to understand that the underlying driver is the standard SQL Server JDBC and ODBC driver.

The settings for Google Cloud SQL Server **Driver properties** enable you to adjust the performance of the SQL Server JDBC and ODBC driver. These adjustments can influence the efficiency, compatibility, and features of your Google Cloud SQL Server database.

For a complete walkthrough on setting up SQL Server JDBC and ODBC drivers properties, you can refer to the official [SQL Server JDBC documentation](#) and [SQL Server ODBC documentation](#). These guides detail each driver's properties and how they can be used to optimize Google Cloud SQL Server database connections.

You can customize the SQL Server driver in DBeaver via the **Edit Driver** window, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This window offers a range of settings that can influence your Google Cloud SQL Server database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

ODBC and JDBC Driver Configuration

DBeaver provides extensive capabilities for managing Google Cloud SQL Server database connections via ODBC /JDBC drivers. This functionality enables you to connect to your Google Cloud SQL Server database using native ODBC drivers, offering an alternative when specific DBeaver drivers are not available.

You can find a comprehensive, step-by-step guide on how to install the driver manager, set up drivers, configure data sources, and establish connections in DBeaver in our [ODBC Driver Configuration](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your Google Cloud SQL Server database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes**, **AWS SSM** and **SSL** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [SSL Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering Google Cloud SQL Server with DBeaver

DBeaver provides a host of features designed for Google Cloud SQL Server databases. This includes the ability to view schemas, along with numerous unique capabilities aimed at optimizing database operations.

Google Cloud SQL Server database objects

DBeaver lets you view and manipulate a wide range of Google Cloud SQL Server database objects. DBeaver has extensive support for various SQL Server metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Schemas
 - Tables
 - External Tables
 - Views
 - Columns
 - Triggers
 - External Properties
 - Indexes
 - Procedures

- Sequences
- Synonyms
- Triggers
- Data Types
- Database triggers
- Security
 - Logins
- Administer
 - Sessions
- Agent
 - Jobs

Google Cloud SQL Server additional features in DBeaver

DBeaver provides additional features compatible with Google Cloud SQL Server, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Session Management	Session Manager
Schema Management	Schema Compare
Data Visualization	GIS Guide
	ERD Guide
	Query Execution plan

Firestore

Note: This driver is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Firestore specialty](#)

[Setting Up](#)

[Firestore connection settings](#)

[Connection details](#)

[Secure Connection Configurations](#)

[Powering Firestore with DBeaver](#)

[Working with Collections and Subcollections](#)

[Creating a Subcollection](#)

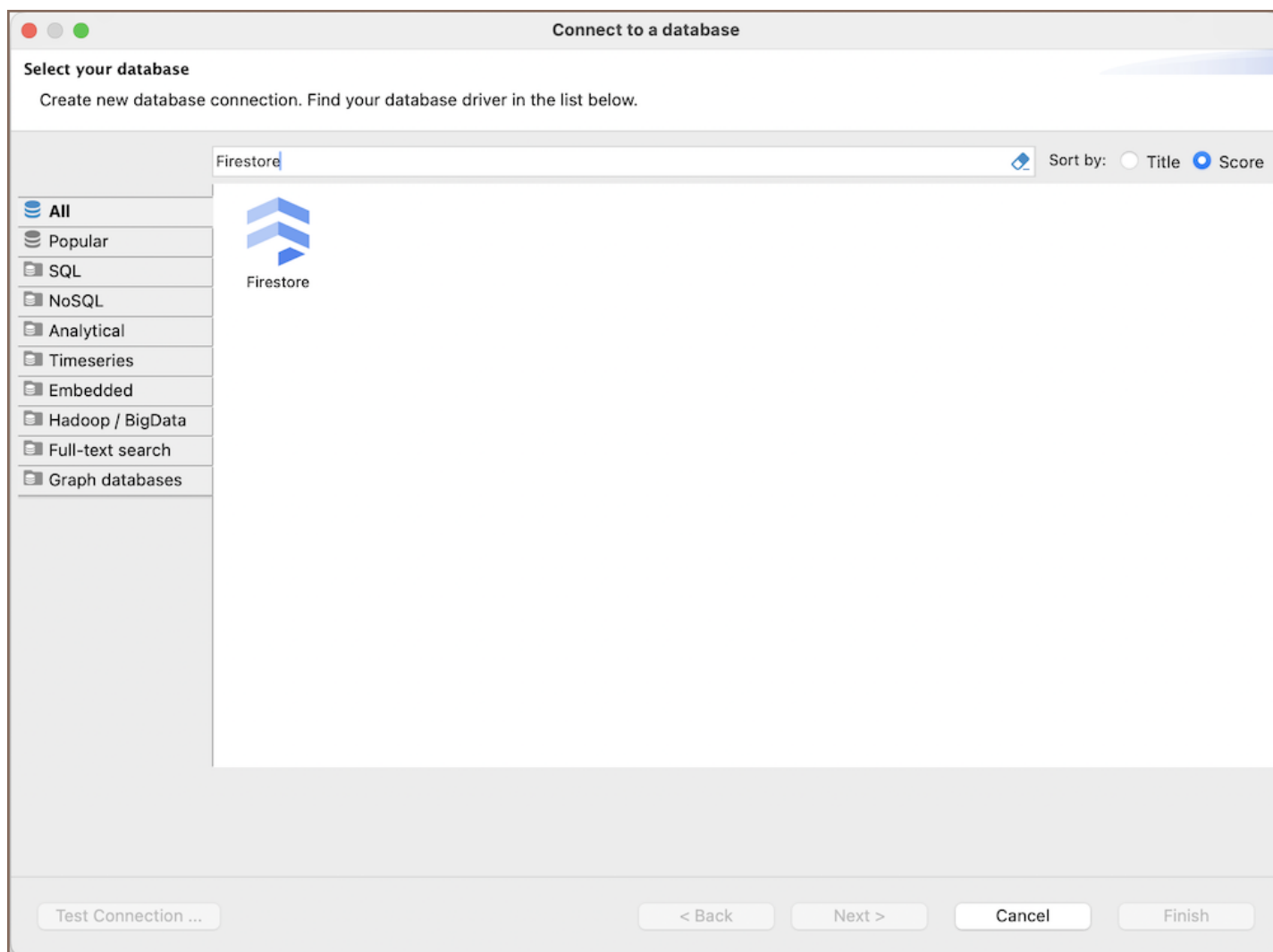
[Viewing Subcollections](#)

[Firestore Features in DBeaver](#)

Overview

This guide provides instructions on how to set up and use Firestore with DBeaver.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **Firestore** option to connect to the Firestore database. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Tip: Users with [Ultimate](#) and [Team](#) editions of DBeaver can connect to their Firestore databases using the [Cloud Explorer](#).

Firestore specialty

Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform. Firestore is a NoSQL database that stores data in documents, which are organized into collections. Each document contains a set of key-value pairs and is identified by a unique identifier. Firestore's real-time data synchronization capabilities allow developers to build collaborative applications by enabling real-time updates to data across all connected clients. It supports offline data access, enabling applications to function seamlessly without an internet connection. Security and data validation are enforced through Firestore Security Rules, allowing developers to control access to documents and collections. Firestore is designed to scale automatically, handling large datasets with ease. It supports powerful query capabilities, allowing efficient data retrieval and filtering. For more details, refer to the official [Firestore documentation](#).

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, and Kubernetes.

Firestore connection settings


In this subsection, we will outline the settings for establishing a direct connection to a Firestore database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Firestore database.

The connection settings page requires you to fill in specific fields to establish the initial connection.

Connect to a database

Google Firestore Connection Settings

Firestore connection settings



MainDriver propertiesSSH

+ Network configurations...

Settings

Project: dbeaver_test

Database: sample

Authentication

Authentication: Google Cloud IAM (Firestore)

Credentials: Key fileDetails

Configuration: /Users/dbeaver/dbeavertest.json

Enable IAM

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: Firestore

Driver Settings

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Project	Choose the name of the Project you want to connect to.

DBeaver User Guide 24.1.ea. Page 495 of 1174.

Database	Enter the name of the Firestore database you want to connect to.
Authentication	<p>Choose the type of authentication you want to use for the connection. For detailed guides on authentication types, please refer to the following articles:</p> <ul style="list-style-type: none"> - DBeaver Profile Authentication - Google Cloud IAM (coming soon) <p>You can also read about security in DBeaver PRO.</p>
Connection Details	Provide additional connection details if necessary.
Driver Name	This field will be auto-filled based on your selected driver type.
Driver Settings	If there are any specific driver settings, configure them here.

Connection details

The **Connection Details** section in DBeaver allows for further customization of your Firestore connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- [Security Settings Guide](#)
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Secure Connection Configurations

DBeaver supports secure connections to your Firestore database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering Firestore with DBeaver

DBeaver provides support for Firestore, focusing on facilitating the management of collections and subcollections.



Working with Collections and Subcollections

Firestore organizes data into collections, which are containers for documents. Additionally, collections can contain subcollections, enabling a hierarchical structure for data organization.

The process for managing Firestore collections mirrors the familiar experience of creating and handling objects in the [Database Navigator](#).


Creating a Subcollection

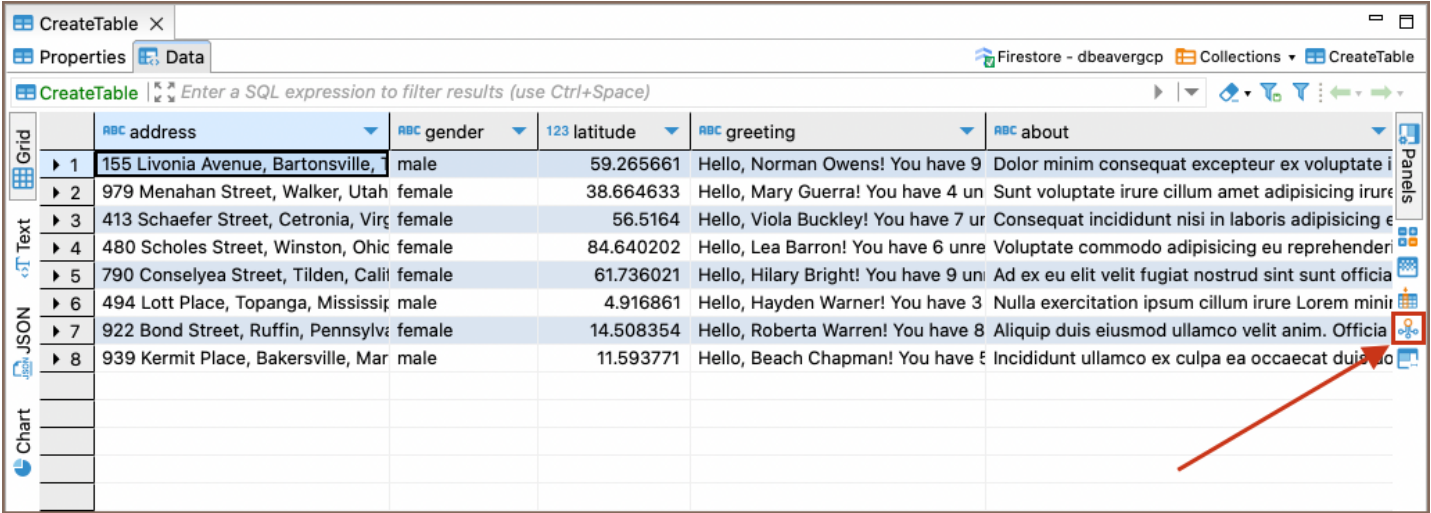
To work with Firestore subcollections, follow these instructions:

- 1) Open the collection in [Data Editor](#) and click the Subcollections button  located in the [right sidebar](#) to access the management interface for Firestore's subcollections.
- 2) In the Subcollections window, you will find options to create new subcollections, delete the existing ones, or refresh the list to view the latest data.
- 3) To add a Subcollection, click the **Create collection** button . You will be prompted to enter a name for the new collection.
- 4) After setting up the subcollection's name, it is ready to be filled with data, structured according to your application's requirements.

Viewing Subcollections

To view subcollections:

- 1) Navigate to the collection in Data Editor. 2) Select the object (collection) for which you wish to view the subcollections. 3) Click the Subcollections button .



Tip: To view data in **JSON** format, click the corresponding button in the [left sidebar](#).

Firestore Features in DBeaver

In addition to facilitating the management of collections and subcollections, DBeaver's integration with Firestore includes a range of features designed to enhance the user experience and efficiency when working with Firestore databases. Here are some of the key features:

Category	Feature
Data Transfer	Data Export

Spanner

Note: This driver is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Google Spanner specialty](#)

[Setting Up](#)

[Google Spanner connection settings](#)

[Connection details](#)

[Google Spanner driver properties](#)

[Secure Connection Configurations](#)

[Powering Google Spanner with DBeaver](#)

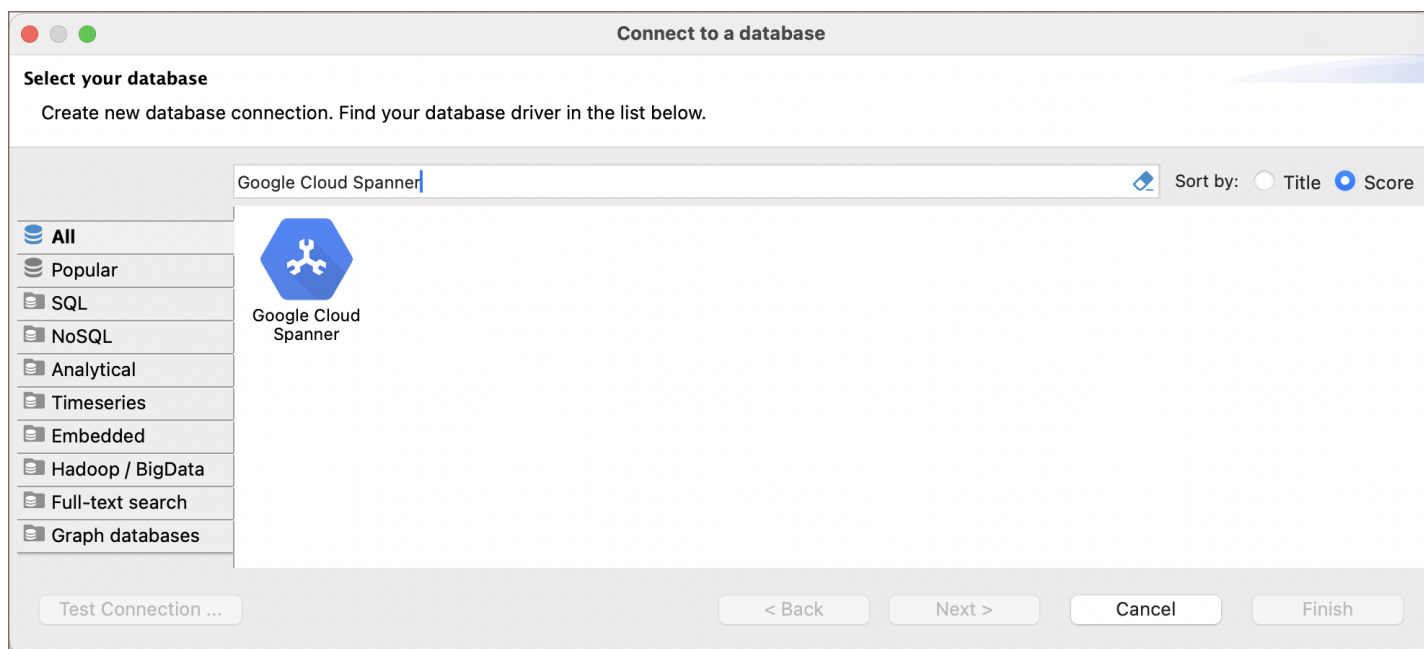
[Google Spanner database objects](#)

[Google Spanner additional features in DBeaver](#)

Overview

This guide provides step-by-step instructions on how to set up and manage Google Cloud Spanner databases using DBeaver.

Before you can start managing your database, it's essential to establish a connection in DBeaver. This involves selecting the **Google Cloud Spanner** option to connect to your database. If you have not yet created a connection in DBeaver, please refer to our [Creating a Connection](#) article for guidance.



Tip: Users with [Ultimate](#) and [Team](#) editions of DBeaver can connect to their Google Spanner databases using the [Cloud Explorer](#).

Google Spanner specialty

Google Spanner is a distributed database system that combines features from both relational and NoSQL databases. It stands out for its large-scale capabilities, designed to manage extensive data across numerous nodes while maintaining high availability and global consistency.

In Google Spanner, the metadata types resemble those in traditional relational databases, with tables, columns, schemas, and key constraints, including primary keys, foreign keys, and unique constraints. A significant aspect of Google Spanner is its synchronization mechanism for nodes. It uses atomic clocks and GPS receivers to power its TrueTime system, ensuring precise synchronization across all nodes. This advanced timekeeping infrastructure is crucial for maintaining external consistency in distributed transactions and offers a robust solution for strong consistency on a global scale. For a more comprehensive understanding of Google Spanner, please refer to the [official documentation](#).

DBeaver supports the features compatible with Google Spanner. For detailed information on these supported features, refer to the section [Powering Google Spanner with DBeaver](#).

Setting Up

This section provides an overview of DBeaver's settings for establishing a direct connection and the configuration of secure connections using SSH, proxies, and Kubernetes.

Google Spanner connection settings


In this subsection, we will outline the settings for establishing a direct connection to a Google Spanner database using DBeaver. Correctly configuring your connection ensures seamless interaction between DBeaver and your Google Spanner database.

The page of the connection settings requires you to fill in specific fields to establish the initial connection.

Connect to a database

Google Spanner Connection Settings

Google Cloud Spanner connection settings



Main

Driver properties

SSH

+ Network configurations...

Connection

Project:spanner

Instance:cloudspanner

Database:sample

Authentication

Authentication:Google Cloud IAM (Spanner)

Credentials:Default credentialsDetails

☐ Enable IAM

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name:Google Cloud Spanner

Driver Settings

Driver license

Test Connection ...

< Back

Next >

Cancel

Finish

Field	Description
Project	Enter the project name of your Google Spanner database here.
Instance	For a direct connection to a Google Spanner instance, use the format <code>project:region:instance</code> . Here, <code>project</code> refers to your Google Cloud project ID, <code>region</code> denotes the geographical location of your instance, and <code>instance</code> is the unique ID of your Google Cloud SQL instance.
Database	Enter the name of the Google Spanner database you want to connect to.
	Choose the type of authentication you want to use for the connection. For detailed guides on authentication

Authentication	<p>types, please refer to the following articles:</p> <ul style="list-style-type: none"> - Google Cloud IAM (Spanner) - DBeaiver Profile Authentication <p>You can also read about security in DBeaiver PRO.</p>
Connection Details	<p>Provide additional connection details if necessary.</p>
Driver Name	<p>This field will be auto-filled based on your selected driver type.</p>
Driver Settings	<p>If there are any specific driver settings, configure them here.</p>

Connection details

The **Connection Details** section in DBeaiver allows for further customization of your Google Spanner connection. This includes options for adjusting the **Navigator View**, setting up **Security measures**, applying **Filters**, configuring **Connection Initialization** settings, and setting up **Shell Commands**. Each of these settings can significantly impact your database operations and workflow. For detailed guides on these settings, please refer to the following articles:

- [Connection Details Configuration](#)
- [Database Navigator](#)
- Security Settings Guide
- [Filters Settings Guide](#)
- [Connection Initialization Settings Guide](#)
- [Shell Commands Guide](#)

Google Spanner driver properties

The settings for Google Spanner **Driver properties** enable you to adjust the performance of the Google Spanner JDBC driver. These adjustments can influence the efficiency, compatibility, and features of your Google Spanner database.

For a complete walkthrough on setting up Google Spanner JDBC driver properties, you can refer to the official [Google Spanner JDBC documentation](#).

You can customize the Google Spanner driver in DBeaver via the **Edit Driver** page, accessible by clicking on the **Driver Settings** button on the first page of the driver settings. This page offers a range of settings that can influence your Google Spanner database connections. For a comprehensive guide on these settings, please refer to our [Database drivers](#) article.

Secure Connection Configurations

DBeaver supports secure connections to your Google Spanner database. Guidance on configuring such connections, specifically **SSH**, **Proxy**, **Kubernetes** and **AWS SSM** connections, can be found in various referenced articles. For a comprehensive understanding, please refer to these articles:

- [SSH Configuration](#)
- [Proxy Configuration](#)
- [Kubernetes Configuration](#)
- [AWS SSM](#)

Powering Google Spanner with DBeaver

DBeaver provides a host of features designed for Google Spanner databases. This includes the ability to view and manage databases, along with numerous unique capabilities aimed at optimizing database operations.

Google Spanner database objects

DBeaver lets you view and manipulate a wide range of Google Spanner database objects. DBeaver has extensive support for various Google Spanner metadata types, allowing you to interact with a wide variety of database objects, such as:

- Databases
 - Tables
 - Columns
 - Keys
 - Foreign Keys
 - Indexes
 - References
 - Views
 - Procedures

Google Spanner additional features in DBeaver

DBeaver provides additional features compatible with Google Spanner, but not exclusive to it:

Category	Feature
Data Transfer	Data Import
	Data Export
Data Visualization	ERD Guide
Data Generation	Mock Data Generation

Snowflake

Table of contents

[Connecting to the Snowflake database.](#)

[Authentication](#)

[Database](#)

[Private key](#)

[SSO Browser](#)


[DBeaver Profile](#)

[Metadata](#)

You can read more about the Snowflake database in the Snowflake [documentation](#).

Connecting to the Snowflake database.

There are several ways to configure a database connection.

 Connect to a database

Snowflake Connection Settings

Snowflake connection settings

Main

Snowflake

Driver properties

SSH

Proxy

Connection

Host: Port:

Database:

Warehouse:


Schema:

Click on [Test Connection](#) to load warehouse/schema list from the server

Authentication

Authentication:

Username:

Password: ☒ Save password locally 

Role:

Authenticator:

You can use variables in connection parameters.

Connection details (name, type, ...)

Driver name: Snowflake

Driver Settings

Test Connection ...

< Back

Next >

Finish

Cancel

Connection parameters

Parameter	Description	Example
Host	Server host name	https://account.region.snowflakecomputing.com
Post number	Server listener port	443 (default)
Database*	Database name	SNOWFLAKE_SAMPLE_DATA
Warehouse*	Cluster of compute resources in Snowflake. Warehouses are required for queries, as well as all DML operations, including loading data into tables	SNOWFLAKE

Schema*	Schema name	SAMPLE_DATA
---------	-------------	-------------

*Click on the "Test Connection" button to load the warehouse/database/schema list from the server.

Authentication

Database

Parameter	Description	Example
Username	Database user name	ADMIN
Password	Database user password	*****
Save password	Saves the user/password information in the local DBeaver configuration	
Role	Role for connection	SYS
Authenticator	Authenticator parameter/option	https://.okta.com

Private key

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Parameter	Description	Example
Username	Database user name	ADMIN
Private key	Path to encrypted or unencrypted private key	C:/Users/admin/Downloads/rsa_key.p8
Key Password	Private key password	*****
Save password	Saves the user/password information in the local DBeaver configuration	
Role	Role for connection	SYS

SSO Browser

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Connection with the help of an external browser will be used.

Parameter	Description	Example
Username	Database user name	ADMIN
Role	Role for connection	SYS

DBeaver Profile

You can use preconfigured connection profile here.

Metadata

DBeaver provides an opportunity to view and manipulate such types of metadata:

- Databases
- Schemas
- Data types
- Tables
- Views
- Procedures and functions
- Sequences ★
- Stages ★
- Pipes ★

- Streams ★
- Tasks ★
- Materialized Views ★

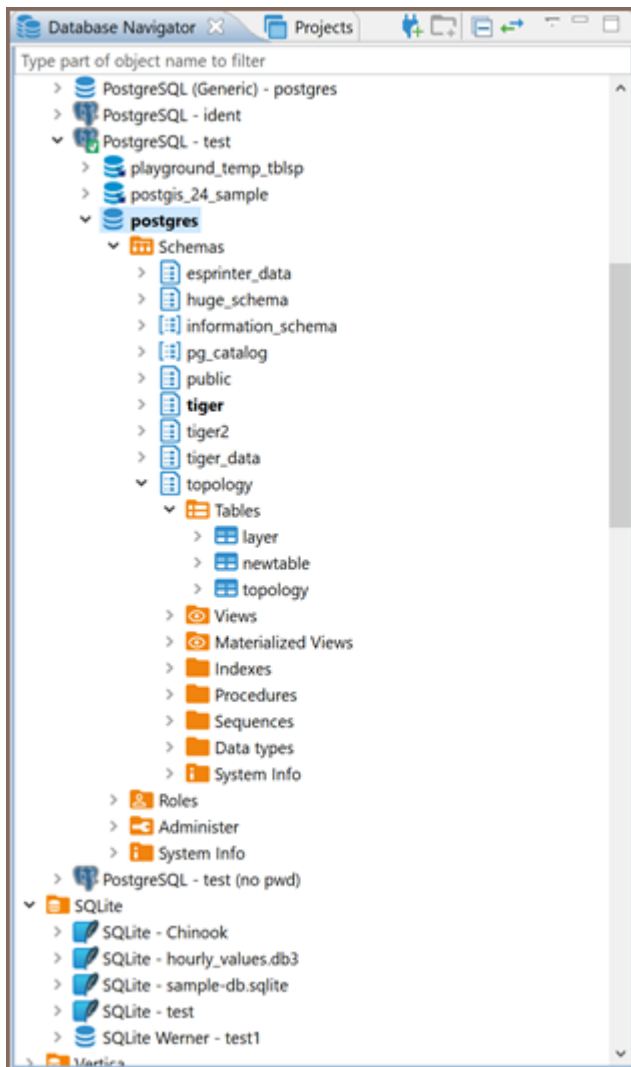
WEATHER - Weather forecast data		
Tables		
DAILY_14_TOTAL	12 days/daily forecast of 22,000 cities	37G
DAILY_16_TOTAL	12 days/daily forecast of 209,000 cities	277G
HOURLY_14_TOTAL	4 days/3 hour forecast of 22,000 cities	1.7T
HOURLY_16_TOTAL	4 days/3 hour forecast of 209,000 cities	913G
WEATHER_14_TOTAL	Current weather of 20,000 cities	49G
Views		
Procedures		
Sequences		
Stages		
Pipes		
Streams		
Tasks		

Table statistics are also available only in PRO DBeaver versions. As well as expanded metadata about tables and schemes.

Properties		ER Diagram	
Schema Name:	WEATHER	Catalog:	SNOWFLAKE SAMPLE DATA
Retention Time:	1	Schema Owner:	
<input type="checkbox"/> Is managed access <input type="checkbox"/> Is transient		Create time:	2021-11-09 21:04:43.727
Schema Description:	Weather forecast data	Last Altered:	2021-11-09 21:04:43.727


Database Navigator overview

Database Navigator is the main view to work with the structure and content of databases. To open Database Navigator, on the Windows menu, click **Database Navigator**. For information on how to change the view layout, please see the [Application Window Overview](#) article.








Database Navigator contains a tree of objects, a toolbar and View menu which contain generic items. Each object in the tree has its own context menu. The tree contains the following objects:

- Folders -
- Database connections - and other (icons differ depending on the database type)
- Database objects - various depending on the database type, such as Tables , Views , Columns **123 ABC**, Indexes , etc.

To open the view menu of Database Navigator, click the View Menu button () in the upper-right corner of the window. For more information on where to find the view toolbar and menu, please see the [Views](#) article.

The menu contains the following items:

Icon	Menu item	Description
	Driver Manager	Opens the Driver Manager window that allows you to create, edit and delete drivers for databases. See Database Drivers for information about managing database drivers.
	New Connection	Opens the Create new connection wizard. See Create Connection for information about creating connections.
(empty)	Active Project	Displays a submenu which allows you to choose a project. See Projects and Projects view for information about projects.
	New Folder	Opens a dialog box for creating a new folder
	Collapse All	Collapses the tree to the root level
	Link with editor	Synchronizes the active editor with the element in the database navigator

The toolbar is located in the title bar of the window. Its buttons duplicate the menu items, except for the **Active Project**.

To open the context menu for an object, right-click the object in the tree. The following table summarizes the context menu items for all types of objects that may appear in the tree. Note that the presence or absence of the context menu items for an object depends on the database and object types.

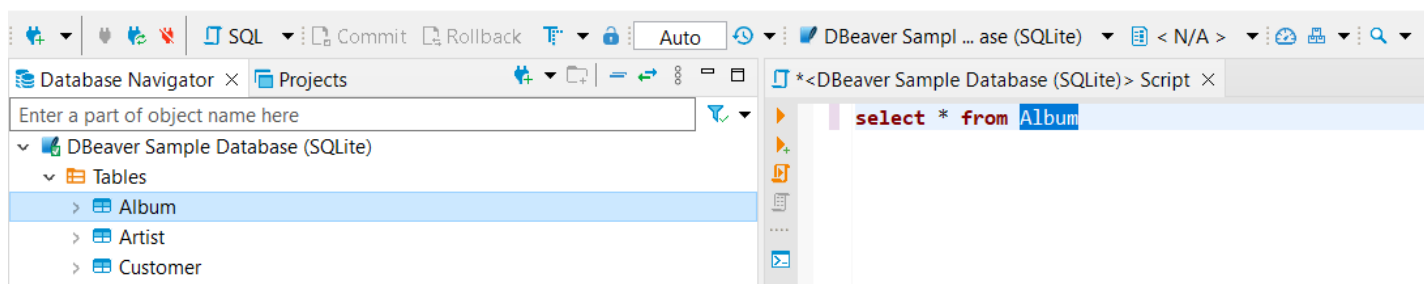
Menu item	Description
Open folder	Opens a folder in a separate view
Create new connections / Create New Connection	Opens the Create new connection wizard
New Folder	Opens a dialog box for creating a new folder
Copy	Copies an object to the clipboard
	Inserts the copied object into a selected folder - most convenient for copy-pasting

Paste	connections
Delete	Deletes an object WARNING! The Delete menu item removes the object not only from the tree but from the database itself or the file system, and this action is irreversible.
Rename	Opens the Rename [object] dialog box
Properties	Opens the Properties for [object] window which allows viewing and modifying the object's properties
Refresh	Depending on the object, refreshes the object itself, or its parent, or its subnodes – mostly used for refreshing tables and schemes
Connect	Attempts to connect to the database
Invalidate/Reconnect	Checks the status of connection, if it is broken, attempts to reconnect
Disconnect	Disconnects from the database
SQL Editor	Opens a new SQL editor for the connection
Recent SQL Editor	Opens the most recently opened SQL editor
Edit Connection	Opens the Connection Configuration window that allows configuring connection settings
Connection view	Adjusts the display of database objects, offering Simple , Advanced , or Custom views for tailored navigation.
View [objects]	Opens the object in a separate viewer
Edit [object]	Opens the object in a separate editor
Create new [object]	Opens an editor in which you can specify properties and save the new object
Filter	Opens a submenu of one or more filtering options (depending on the object): <ul style="list-style-type: none"> - Hide [object] - Show only [object] - Configure [objects] filter - Toggle filter - Clear filter See Filters for information.
Copy Advanced Info	Copies the full name of an object
Read Data in SQL Console	Opens an SQL console displaying the object's data

Compare	<ul style="list-style-type: none"> - Appears only if you select several objects of the same level - Opens the Compare objects wizard which guides you through the steps to generate a comparison report for the selected objects
Generate SQL	<p>Opens a submenu on which you can select the type of SQL query to generate:</p> <ul style="list-style-type: none"> - SELECT - INSERT - UPDATE - DELETE - MERGE - DDL <p>Clicking one of the items (for example INSERT) generates a relevant query in a separate window.</p>
Export Table Data	Opens the Data Transfer wizard that helps you select a format and export table data
Import Table Data	Opens a window with existing database connections in which you can select a table to import data from
Tools	Opens a submenu that provides tools for database backup and restore, vacuum, etc.

For information on how to filter database objects in Database Navigator, please see the [Filter Database Objects](#) article.

While working in the [SQL Editor](#), you can drag the desired object from the Database Navigator to the part of your query. Please pay attention to the selected object name. Running the query immediately with the selected object will cause an error because only the selected part of the script is being executed and fails as it's not a complete query. An extra step to deselect it by clicking somewhere in the SQL Editor is required to remove the selection. Nevertheless, we cannot deselect the object name automatically after dragging. This action is needed when you drag and drop an object into a complex query or long script, and it would be difficult to track what and where exactly you've just dropped, and you can get an unexpected error.



Simple and advanced View

Table of contents

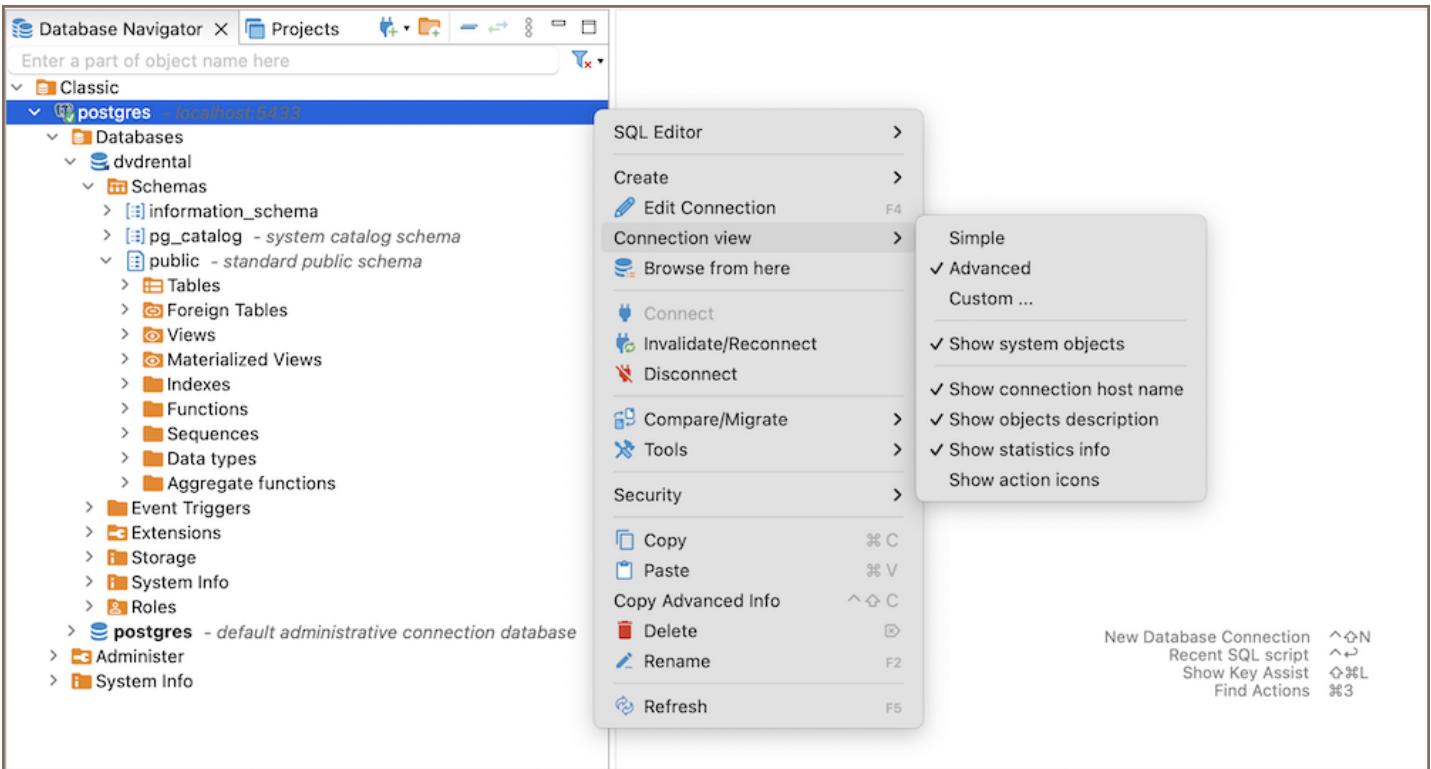
- [Overview](#)
- [Accessing Connection view modes](#)
- [View modes features](#)
- [Additional settings](#)

Overview

The Database Navigator in DBeaver displays various database objects by default, including system objects, schemas, tables, indexes, utilities, and more.

To customize or simplify the [Database Navigator's](#) contents, you may choose from three view modes:

- **Simple:** This mode offers a streamlined view, ideal for users who primarily interact with tables.
- **Advanced:** This mode provides a comprehensive view of all database objects that DBeaver can manage.
- **Custom:** This mode allows for a personalized configuration of the Database Navigator display.



Accessing Connection view modes

To access these settings:

1. Right-click on the desired database within the Database Navigator.
2. Select **Connection view**.
3. Choose between **Simple**, **Advanced**, or **Custom** to suit your preferences.

For the **Custom** view:

1. After selecting **Custom**, an additional window will appear.
2. In this window, you can configure the custom settings for the Database Navigator view.

Important: Changing the view modes settings affects the currently selected database. To customize the display of objects in each database individually, you must select a specific view mode for each one.

View modes features

Each view mode offers a unique set of features:

- **Simple View:** Tailored for ease of use, the **Simple view** offers a minimalist approach, displaying only schemas and tables. It is optimal for users who focus predominantly on these database objects.
- **Advanced View:** The **Advanced view** presents a comprehensive display of the database, showing all database objects, including system objects, all folders, and all supported objects. This mode is intended for users who require visibility across the entire database structure.
- **Custom View:** For those who prefer a personalized setup, the **Custom view** mode provides the ability to configure the Database Navigator.


Following is a detailed comparison of the features across the different view modes, along with descriptions for the configurable options available in the **Custom View**:

	Simple	Advanced	Custom View	
--	--------	----------	-------------	--

Feature	View	View	(Configurable)	Description
Show system objects	No	Yes	Yes/No	Includes or excludes system objects such as <code>pg_catalog</code> or <code>SYSTEM</code> .
Show utility objects	No	Yes	Yes/No	Includes or excludes utility objects like template databases for new database creation.
Show only schemas and tables	Yes	No	Yes/No	Displays only schemas and tables, hiding all other objects for a simplified view.
Show all tables in a single list	No	No	Yes/No	Consolidates all tables into one list, regardless of their schema.
Hide folders	Yes	No	Yes/No	Hides all intermediate logical folders, displaying objects directly.

Additional settings

In addition to selecting a view mode, there are other settings accessible via the **Connection view** context menu. These settings provide additional information about the database objects and enhance your experience:

Setting	Description
Show connection host name	Displays the host name or tunnel of the current database next to the connection name.
Show objects description	Shows object (table, column, etc.) description next to it.
Show statistics info	Provides statistical information on objects (e.g. table size), on the right side of the Database Navigator.
Show action icons	Shows action icons for database elements. For example, you can see a read-only icon  on the right side of the Database Navigator, indicating that a particular connection is in a read-only state.

These settings can be toggled on or off based on your preference, allowing for a customized Database Navigator. To access and modify these settings:

1. Right-click on the desired database within the Database Navigator.
2. Navigate to the relevant options.
3. Select or deselect the options to show or hide the desired information.

Note: These settings alter the display for the entire Database Navigator, affecting how all database objects are presented.

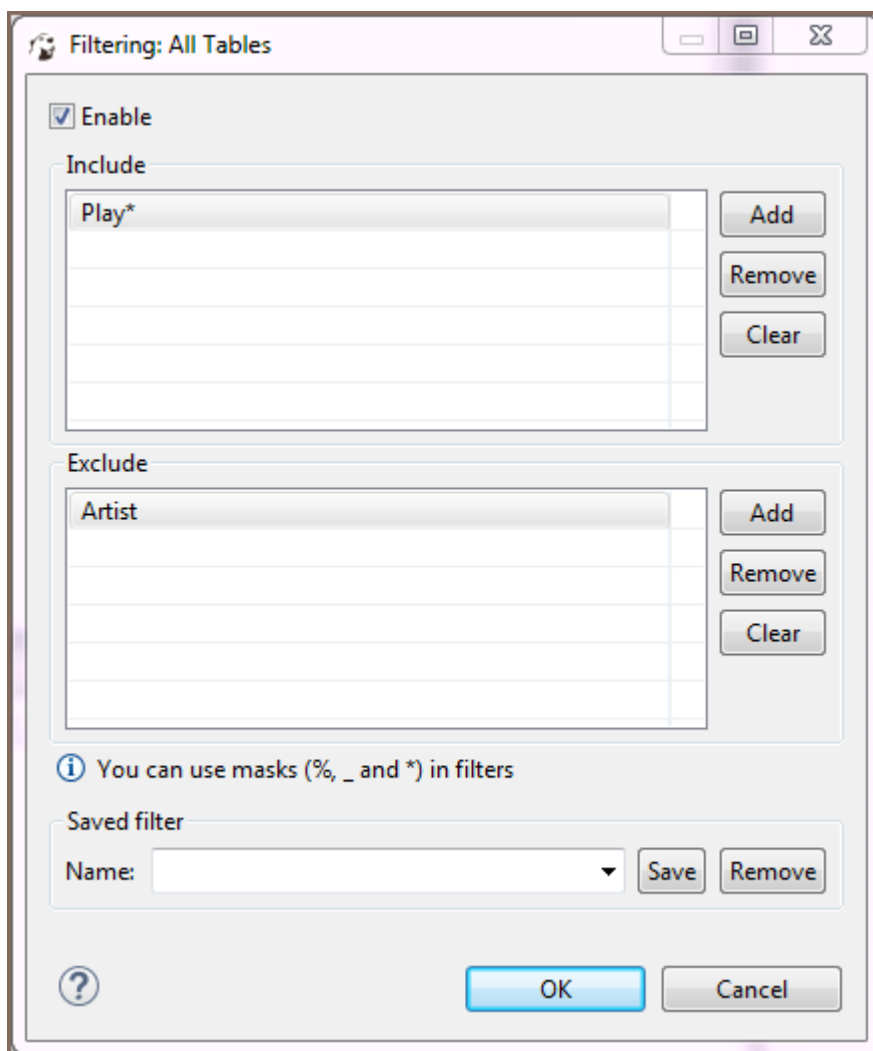
Configuring Filters

You can configure custom filters to filter database objects in the [Database Navigator](#) and [Database Object Editor](#).

To configure a custom filter:

1. In the **Database Navigator**, right-click the object and on the context menu click **Filter -> Configure [objects] filter**. In the **Database Object editor**, in the toolbar of the **Properties** tab, click the **Filter settings** button (🔍).


The Filtering window opens.



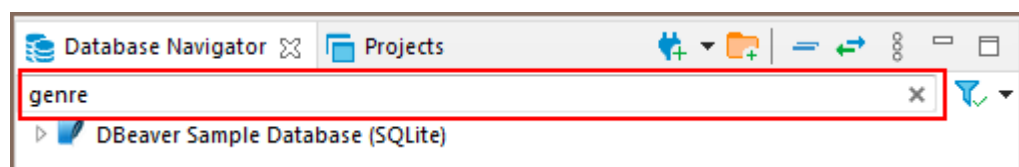
2. Select the **Enable** checkbox to activate the fields of the window.
3. If you want the filter to apply to all objects of a certain type, for example to all schemes, click **Show global filter**. Otherwise, the filter will apply only to the current object. NOTE: Once you apply the global filter, you cannot revert back to the local filter in the same window. To create a local filter, reopen the Filtering window, see Step 1.

4. For objects that you want to show, click **Add** next to the **Include** field and then, in the field itself, enter the name or combination of symbols to search. For objects that you want to hide, click **Add** next to the **Exclude** field and then, in the field itself, enter the name or combination of symbols to search. **NOTE:** In your search pattern, you can use wildcard characters. The **%** symbol is used to replace any number of characters, while **_** is used to replace a single character. In some database and search systems, the ***** character is also used as a wildcard to replace any number of characters. Always check the system's documentation to ensure you're using the correct symbols.
5. To remove one filtering combination, click the combination in the field and then click **Remove**. To remove all combinations from either of the fields, click **Clear** next to the field.
6. Once you set all filtering criteria, you can save a filter to use for other objects. To save the filter, in the Saved filter area, in the **Name** field, enter the filter's name and click **Save**.
7. You can also remove any of the saved filters. To remove a filter, in the **Name** drop-down list, click the filter name and then click **Remove**.
8. Click **OK** to apply the filtering criteria. Otherwise, click **Cancel**.

Filter Database objects

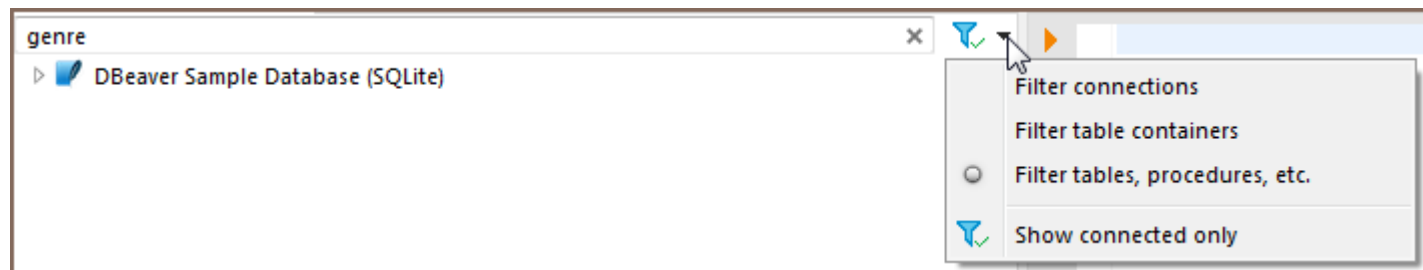
In the [Database Navigator](#) and [Database Object Editor](#) you can filter database objects to include or exclude some of them from the view. You can filter connections, schemas, tables, views, and procedures. A dots sign (...) next to the node's name indicates that a filter is applied to its sub-nodes:  Tables (...)

There are several ways in which you can filter objects. One of the ways is to filter objects by the names of tables and views using the filter field above the tree of objects:

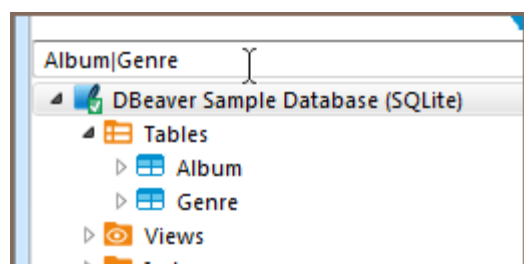


To filter objects by name, type the object name in the field. The tree dynamically updates to show connections /containers/tables/views with that name. To reset the filter, click the Clear icon (X) on the right end of the field.

You can select the types of filtering objects in the drop-down list on the right. And also use the filter only for active connections.



For multiple filtering use special symbols in the filter field - pipe ("|"), comma (","), or space between object names. You can also use an asterisk symbol to replace the part of the name.



Another way to filter objects is to use the **Filter** item on the context menu of a single object. To filter objects using the **Filter** menu, right-click the object, then click **Filter** on the context menu, and then click one of the items on the submenu:

Filter submenu item	Description
Hide '[object name]'	Hides the current object while displaying the other ones
Show only '[object name]'	Shows the current object while hiding the other ones
Toggle filter	Inverts the filtering – shows hidden objects and vice versa
Clear filter	Removes the filtering to display all objects
Configure [objects] filter	Appears only to the folder or parent nodes of database objects - like 'Tables', 'Indexes', etc. Allows the creation of a complex filter with multiple filtering criteria, see Configure Filters .

The third way of filtering is to use the **Filter** item on the context menu on several objects:

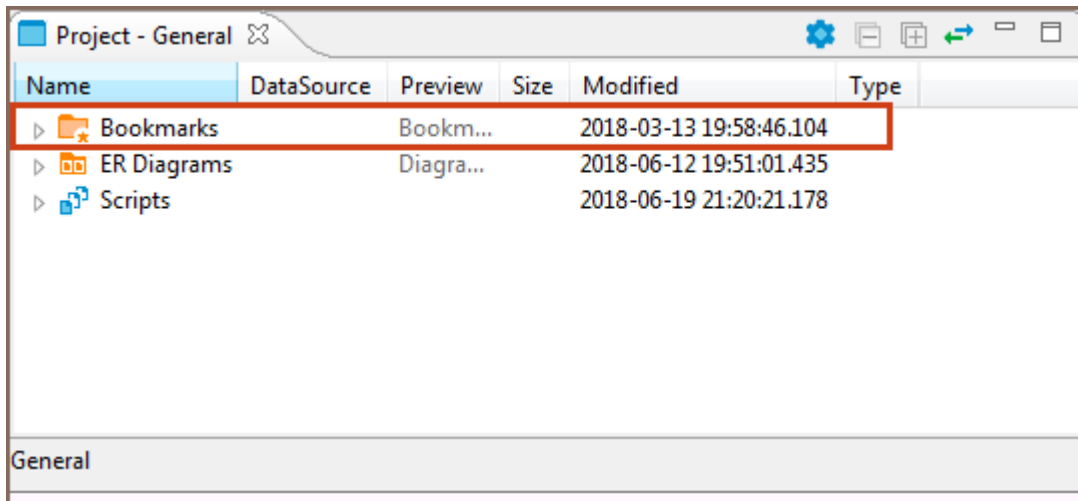
1. Select several objects of the same type using Ctrl or Shift keys.
2. Right-click the selection, then click **Filter**, and then choose one of the options on the submenu:

Filter submenu item	Description
Hide N objects	Hides the selected objects while displaying the rest
Show only selected objects	Shows the selected objects while hiding the rest

To reset such filters, right-click the parent (folder) node displaying the dots sign **(...)**, and then click **Filter -> Clear filter**.

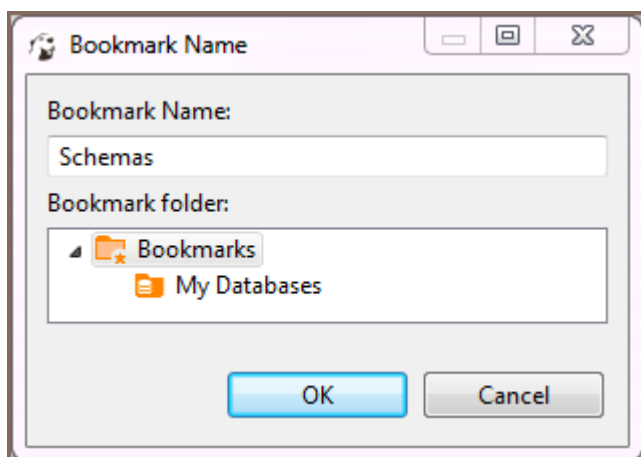
Bookmarks

Bookmarks are quick access links to objects of a database. They appear in the project tree inside the [Projects](#) or [Project Explorer](#) views.



To create a bookmark:

1. In the [Database Navigator](#) or under **Connections** node of the Projects view, click the database object of interest to focus on it.
2. Press CTRL+ALT+SHIFT+D . The **Bookmark Name** dialog box appears.
3. In the **Bookmark Name** field, enter the bookmark name, then in the **Bookmark folder** field, click the folder, and then click **OK**:



The bookmark appears in the selected folder of the related project.

To open an object using its bookmark, double-click the bookmark or right-click it and click **Open Bookmark** on the context menu. You can rename and delete bookmarks using the context menu as well.

Projects overview

Table of contents

[Creating Project](#)

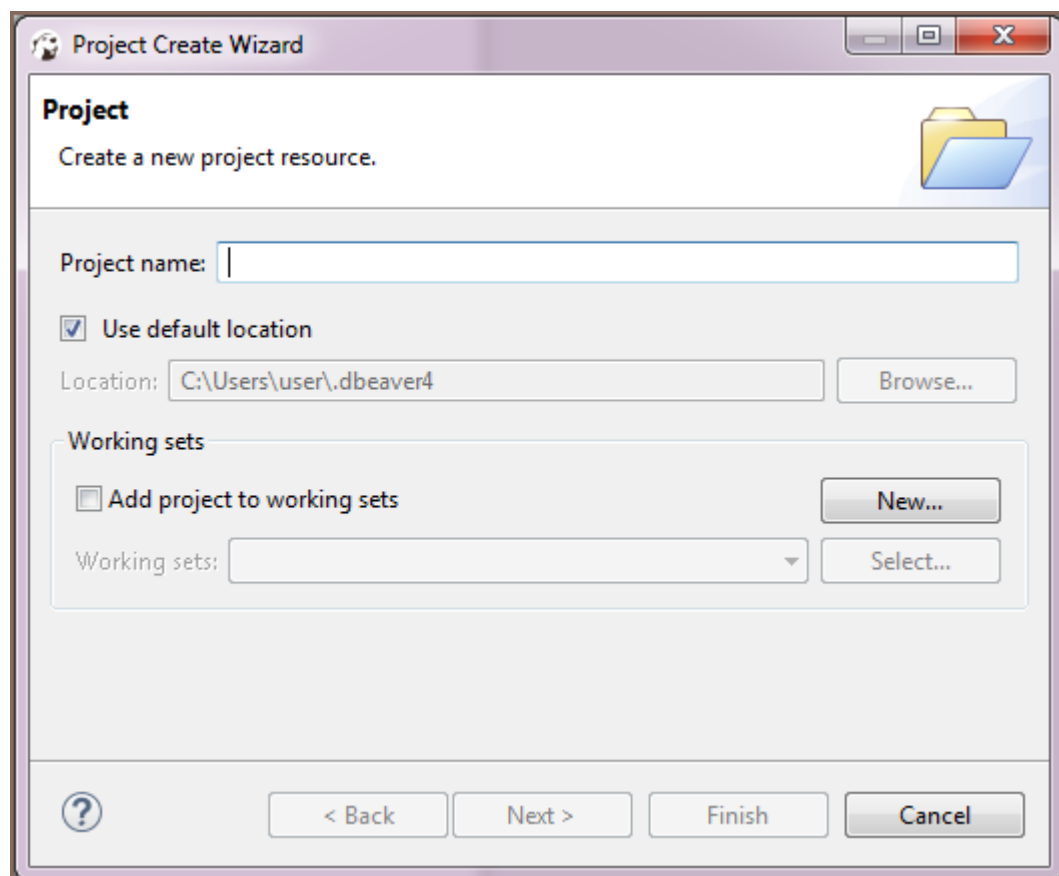
[Deleting Project](#)

The [Projects view](#) allows the creation of new projects as well as renaming and deleting projects that are not active.

NOTE: You cannot rename or delete a project that is set as active.

Creating Project

To create a project, in the Projects view, in the toolbar, click **Create Project** (). The Project Create Wizard opens.



The image shows the 'Project Create Wizard' dialog box. It has a title bar with a question mark icon and standard window controls. The main area is titled 'Project' with the subtitle 'Create a new project resource.' and a folder icon. It contains a 'Project name:' text field. Below it is a checked checkbox 'Use default location' and a 'Location:' text field showing 'C:\Users\user\.dbeaver4' with a 'Browse...' button. A section titled 'Working sets' contains an unchecked checkbox 'Add project to working sets', a 'Working sets:' dropdown menu, a 'New...' button, and a 'Select...' button. At the bottom are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

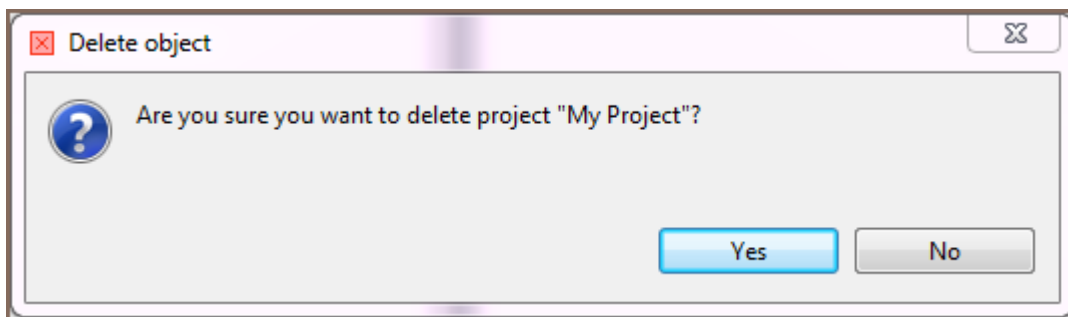
1. In the Project screen, in the **Project name** field, specify the name of the project.

2. To keep the default location to store the project, leave the **Use default location** checkbox selected. If you want to change the location, clear the checkbox and enter the name of the new directory into the **Location** field or click **Browse** and select the directory in the folder tree.
3. Click **Finish**. The new project appears in the projects tree.

Deleting Project

To delete a project, in the Projects view, right-click its name in the tree and click **Delete** on the context menu. Two confirmation dialog boxes will appear one after another:

1. **Delete object** dialog box is to confirm the deletion of the project itself. Click **Yes** if you are sure you want to delete it. Otherwise, click **No**.



2. **Delete project** dialog box is to confirm the deletion of the project's contents. These are the project configuration files and scripts stored in the file system. Click **Yes** if you want the contents to be deleted as well. To keep the contents, click **No**.

NOTE: If you have deleted a project and then re-create it with the same name, the new project picks up all the database connections of the deleted project.

Projects View

Table of contents

[Overview](#)

[Project Structure](#)

[Toolbar and View Menu](#)

[View Menu items](#)

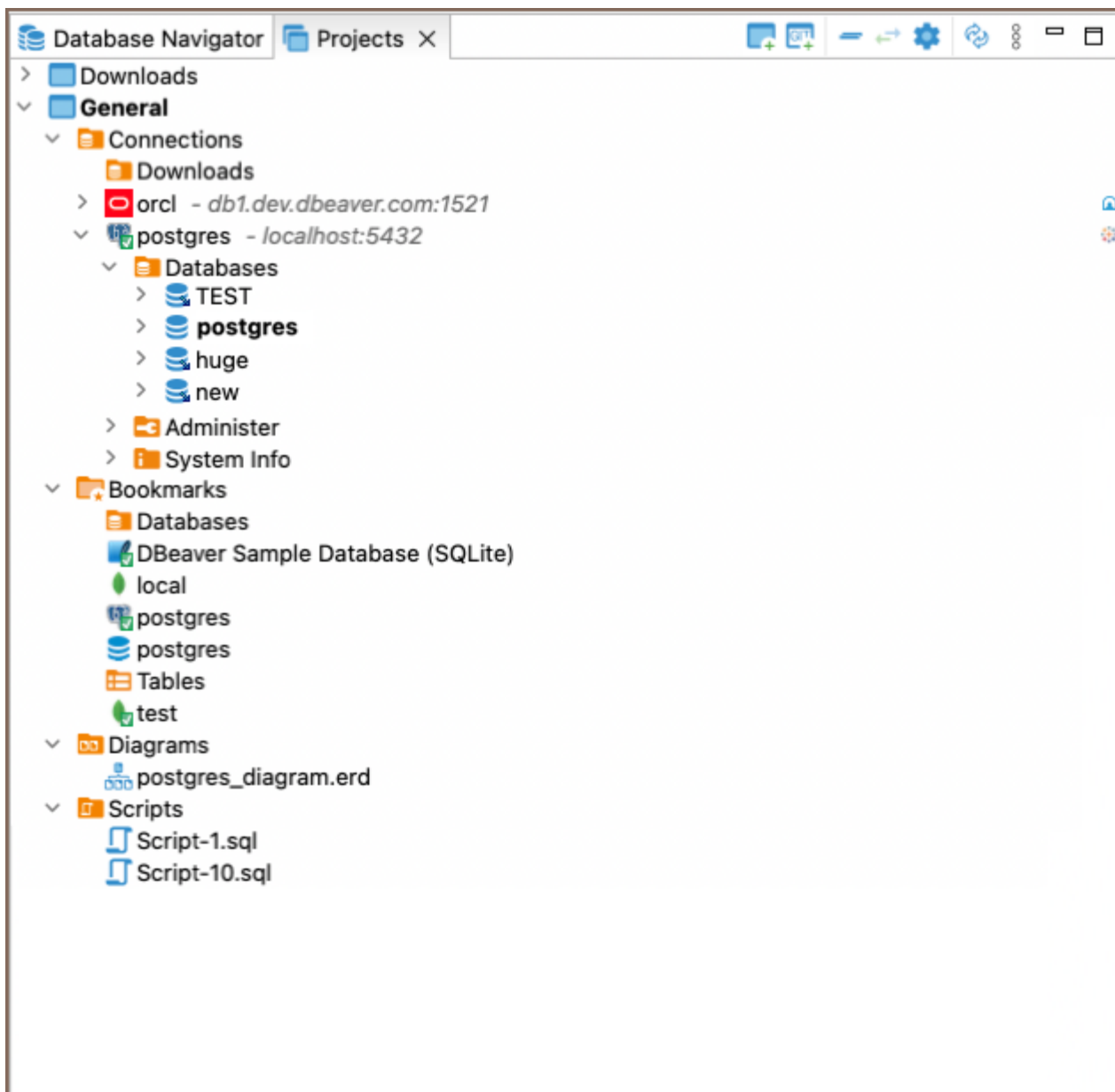
[Context Menu](#)

[Special Actions](#)

Overview

You might need to classify and group database connections into projects. Projects store objects related not to a particular database but to all database connections. These are usually files stored on the file system.

The Projects view displays all projects created in the system and provides tools to manage them. To open the Projects view, on the **Window** menu, click **Projects**.



For information on how to change the view layout, please see the [Application Window Overview](#) article.

Project Structure


Projects in the system are displayed in a tree format and all have the same high-level structure:







Section	Description
Connections	Repeat the content of the Database Navigator view for this project. Perform the same actions over the objects of the databases as in the Database Navigator.
Bookmarks	Contains bookmarks – shortcuts to database objects.
Diagrams	Contains ER diagrams that you can drag-and-drop here from other folders.

Toolbar and View Menu

The toolbar is integrated into the title bar at the top of the window and contains buttons that mirror items in the view menu.

View Menu items

To open the view menu, click the **View Menu** button  located in the upper-right corner of the window. The view menu contains the following items:

Icon	Item	Description
	Create Project	Opens the Project Create wizard.
	Create Project from Git	Opens the Import Projects from Git wizard. Learn more about Git Team work in our article
	Refresh Projects	Refreshes the projects tree to display changes caused by creating, modifying or deleting projects.
	Collapse All	Collapses the tree to the root level.
	Link with editor	Highlights the object in the tree that has its editor open Note: Button enabled when at least one editor is open, otherwise disabled.
	Preferences	Opens the Preferences window, allowing you to adjust resource locations settings specific to the Projects view.

Context Menu

Right-click an object in the tree to access its context menu. For objects under the **Connections** node, see the [Database Navigator](#) documentation. Context menus for other nodes offer basic functionalities such as copy-pasting, renaming, and deleting objects, as well as property management and folder creation.

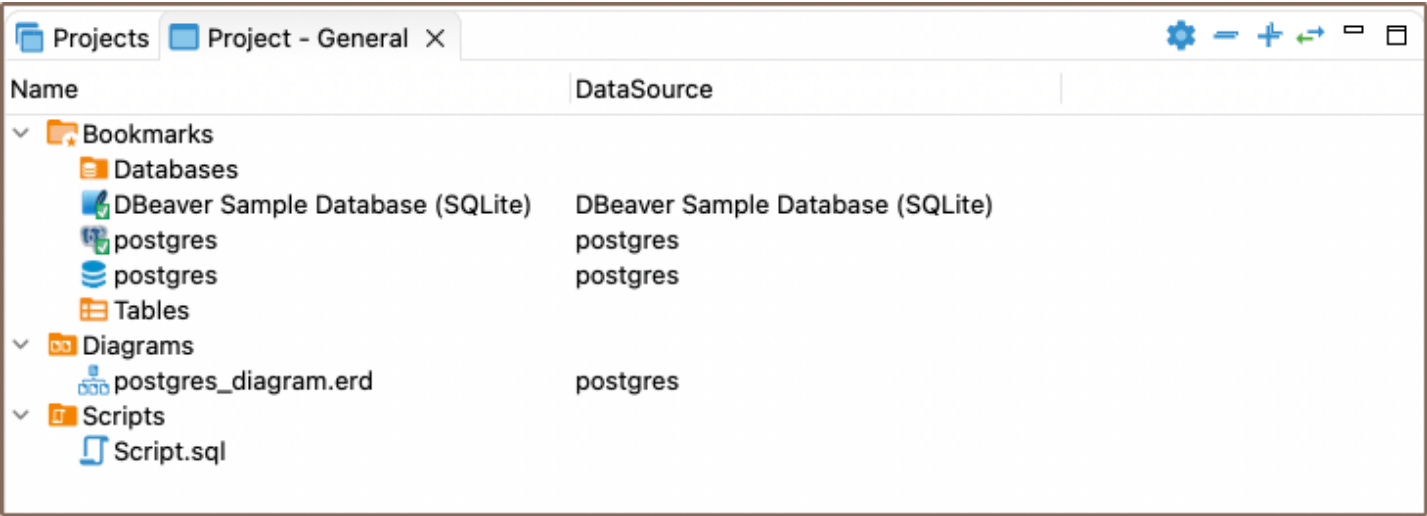
Special Actions

- **Set Active Project:** Makes the selected project root node active and visible in the Database Navigator.
- **Link File** and **Link Folder:** Create links to files and folders within the file system.
- **Show Resource in Explorer:** Opens the project's directory in the user's file system.
- **Restore from Local History...:** Opens the **Restore from Local History** window where you can select a file and its version (if multiple versions exist) for restoration.

For more details on project management, consult the [Projects](#) article.

Project Explorer

The **Project Explorer** view displays detailed contents of the currently active project. To open the **Project Explorer**, click **Window -> Project Explorer**.



For information on how to change the view layout, please see the [Application Window Overview](#) article.

The title of the **Project Explorer** includes the name of the project: Project – [Project name]. **General** is a project that initially exists in the system by default.

The **Project Explorer** displays the content of a project with metadata. The content includes: [Bookmarks](#), [ER Diagrams](#), and [Scripts](#). The metadata appears in columns which you can hide or show.

The **Project Explorer** view provides a toolbar that contains the following buttons:

Button	Name	Description
	Configure columns visibility	Opens a dialog box in which you can select columns to display in the view
	Collapse All	Collapses the tree to the root level
	Expand All	Expands the tree nodes
	Link with editor	<ul style="list-style-type: none">- Enabled when at least one editor is open, otherwise disabled- Highlights the object in the tree that has its editor open

To sort the metadata in the table by a certain column, click the column header.

Project security

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Master password for local configuration](#)

[Use Windows Integration password provider](#)

[Project password](#)

DBeaver supports local storage for connection secure data. It includes:

Database server user credentials

SSH tunnel user credentials

Proxy user credentials

By default, user names and passwords are stored in the `credentials-config.json` file, encrypted with a DES key.

However, this key is not secure as it is publicly available in DBeaver source files, making it possible for unauthorized third parties to decrypt the file using external software. For more secure options, you can read about [security options in DBeaver PRO](#).

Master password for local configuration

It is possible to set a Master password for all projects in a local workspace.

The side effect of this configuration is that you cannot share your connections (with password) between different users because user credentials are stored in a completely separate location, and they are protected by a local user password.

For more information about Master password, refer to the dedicated article [Managing Master Password](#).



Use Windows Integration password provider

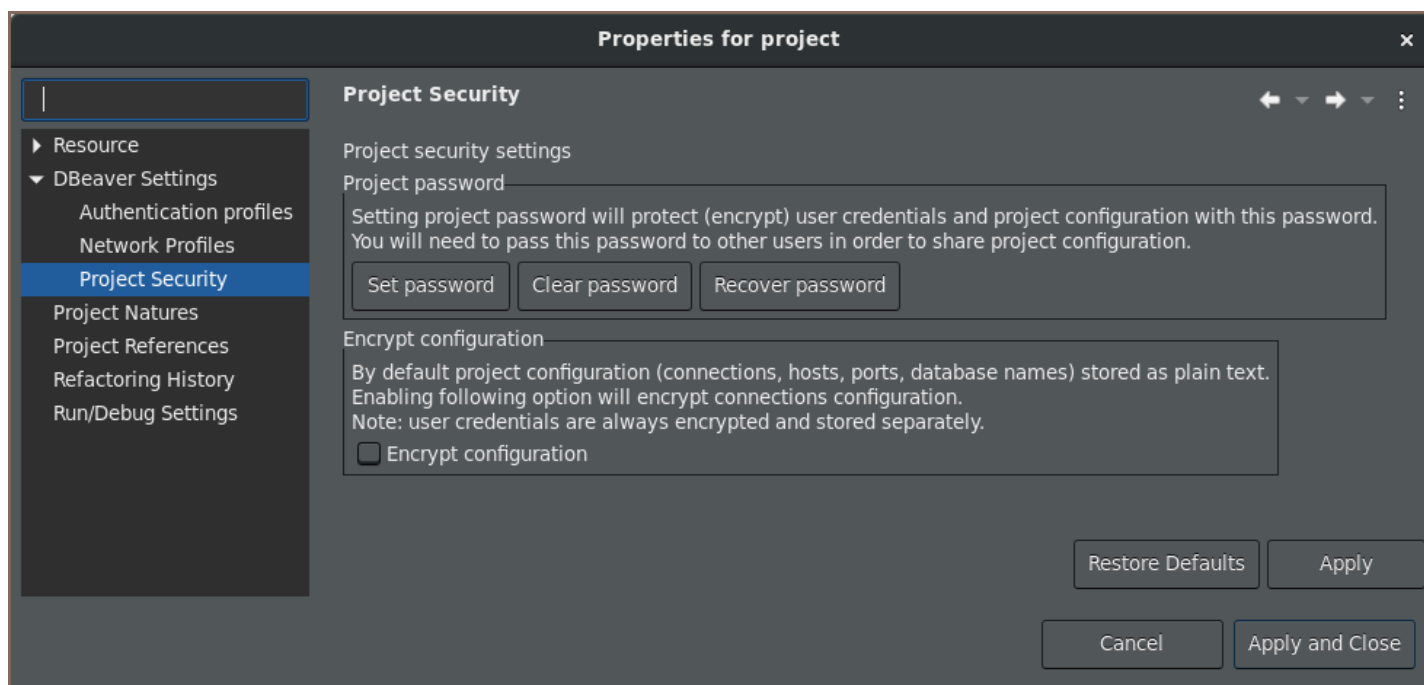
You can disable the default password provider and enable a "Windows Integration" provider. This provider does not need a master password but it uses a randomly generated password stored in a local user secure storage (in Windows). This is easier (as you don't need to remember the master password) but less secure (anybody who has access to your Windows user account will have access to DBeaver's stored credentials).

Project password

You may specify a password for a project. It will encrypt all the project's configurations with this password. Also, you will be able to share your project settings with other users (you will need to pass the project password as well).

In order to enable a project password open the project properties. You can do this by:

- Clicking on main menu File->Project security
- Clicking on "Configure" icon in the project explorer view toolbar and switching to the Project Security tab
- Pressing ALT+Enter on a project element in **Projects** view and switch to Project Security tab



On the project security page click on the "Set Password" button to enable the project password. Click on Clear to disable it (you will need to enter a current project password to clear it).

Editors overview

Table of contents

- [Tabs](#)
- [Database object path](#)
- [Content area and Toolbar](#)

The Database object, or metadata editor is available for multiple database objects such as tables, views and schemas. To open the metadata editor for an object, in the [Database Navigator](#) or in the [Projects](#) view:

- Double-click the database object
- Click the database object and press `Enter` or `F4`

Upon performing these steps, you will see a window where the tabs contain the following common parts:

Tabs

Database object path

film x

PropertiesDataER Diagram

dvdrentalDatabasesdvdrentalSchemaspublicTablesfilm

Table Name: film

Object ID: 17576

Tablespace: pg_default

Owner: postgres

☐ Has Row-Level Security

☐ Partitions

Partition by:

Comment: films table

Extra Options:

Columns

	Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
123	film_id	1	serial4			[v]	nextval('film_film_id_seq'::regclass)	some column
abc	title	2	varchar(255)		default	[v]		
abc	description	3	text		default	[]		
123	release_year	4	year			[]		
123	language_id	5	int2			[v]		
123	rental_duration	6	int2			[v]		
123	rental_rate	7	numeric(4, 2)			[v]	3	4.99
123	length	8	int2			[]		
123	replacement_cost	9	numeric(5, 2)			[v]	19.99	
abc	rating	10	mpaa_rating			[]	'G'::mpaa_rating	
🕒	last_update	11	timestamp			[v]	now()	
🔍	special_features	12	_text		default	[]		
📄	fulltext	13	tsvector			[v]		

13 items

Content area

Toolbar

Tabs

- The **Properties** tab, available for all objects, encompasses the properties of the database object as well as its sub-entities. For more information, please refer to the [Properties Editor](#) article.
- The **Data** tab appears for tables and views and represents the [Data Editor](#).
- The **ER Diagram** tab appears for tables and schemas and displays [ERD \(Entity Relation Diagrams\)](#).

Database object path

The object's path shows the chain of all its parent entities. The entities are clickable: clicking an entity in the path, depending on its nature, either shows its children or opens an editor or a settings window.

Content area and Toolbar

The Content Area and Toolbar are key parts of the Properties tab. The Content Area shows information about a database object and things like its sub-parts or DDL. The Toolbar has different tools on each tab to help with different tasks.

Data Editor overview

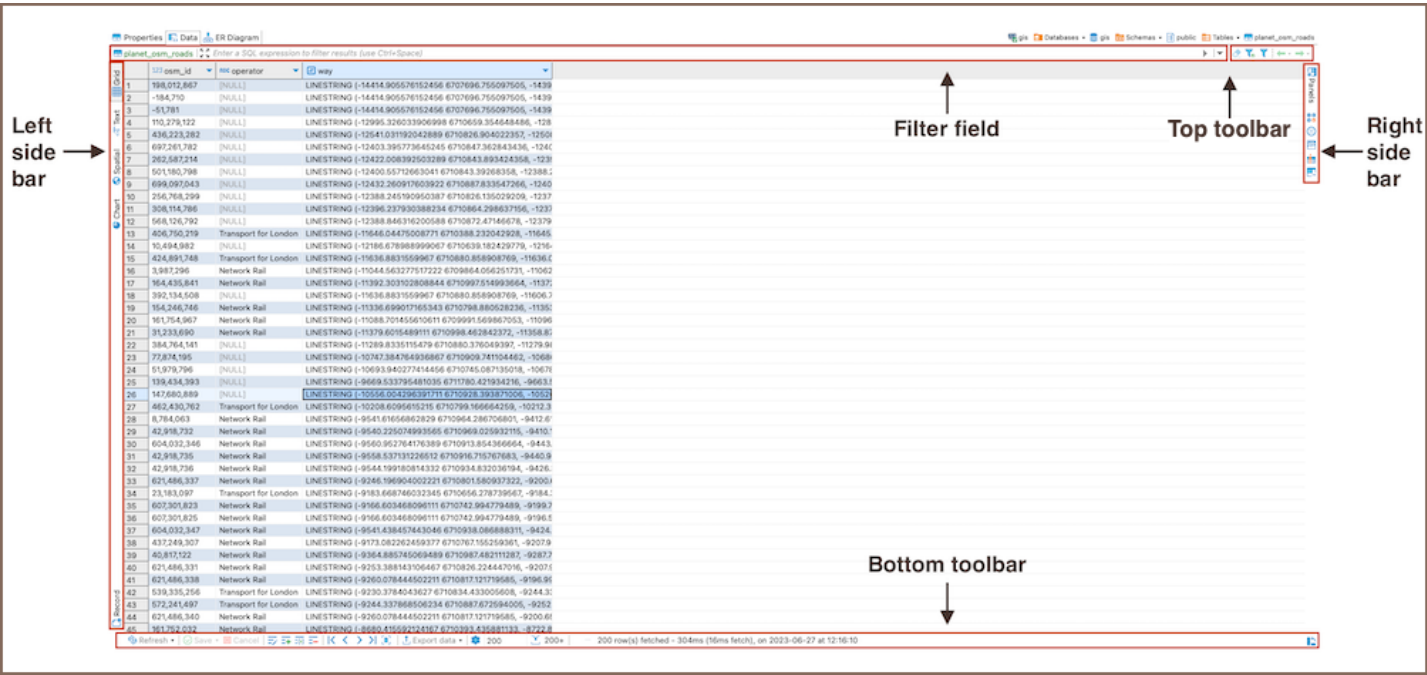
Table of contents

- [Top toolbar](#)
- [Left side bar](#)
- [Right side bar](#)
- [Bottom toolbar](#)
- [Column context menu](#)
- [Cell context menu](#)

The Data editor appears:

- as the **Data** tab of the [Database Object Editor](#), which is only available for tables and views;
- as the **Results** tab when you run a custom SQL query in [SQL Editor](#).

The Data editor allows to view and edit data of a database table or view. The central part of the Data editor is the data grid. The editor also provides [top toolbar](#), [bottom toolbar](#), [left side bar](#), [right side bar](#) and a filter field:







To learn how many rows the table data contains, click the **Calculate total row count** button in the bottom toolbar.

The number of rows appears in a status field next to the button: 203

To learn about ways to navigate data in the table data, see [Navigation](#) article.






Top toolbar

The top toolbar contains the following buttons:

Button	Name	Description
	Remove all filters/orderings	Removes all filters and orderings applied to the data in the filter field.
	Save filter settings for current object	Saves the current filter settings for the database object to apply next time when you reopen it in the editor, see details in the Data Filters article.
	Custom Filters	Opens the Result Set Order/Filter Settings window. See Data Filters article for more information.
	Forward and backward - history navigation buttons	Navigate forward and backward in the Data Editor history, see the history section of our article for more information.








Left side bar

The left side bar contains the following tabs:

Button	Name	Description
	Grid	Switches to the grid view of data.
	Text	Switches to the plain text view of data.
	Spatial	Switches to the spatial view. For more details, see the GIS data article.
	Chart	Switches to the chart view. For more details on charts, see the Managing Charts article.
	Record	<ul style="list-style-type: none">- Same as pressing <code>Tab</code>- Switches the positions of rows and columns so that the columns appear as rows, and the rows hide in one Value column. See details in the Table vs. Record Views section of our article.

Right side bar






The right side bar contains the following tabs (see the [Panels](#) for more information):











Button	Name	Description
	Panels	Opens panels on the right side of the Data Editor.
	Calc	Opens the result cells calculation panel (<code>SUM</code> , <code>MAX</code> , <code>AVG</code> , etc.).
	Grouping	Opens grouping panel window tools.
	Metadata	Opens Metadata panel.
	References	Opens References panel.
	Value	Opens Value Viewer.
	Result Details/Query Trace	Opens the Result Details or Query Trace panel depending on the database.

Note: The **Result Details** panel is compatible with specific databases such as [BigQuery](#), while the **Query Trace** panel is available for [Cassandra](#) and Yugabyte CQL. For guidance on enabling **Query Trace**, see [Panels article](#).

Bottom toolbar


The bottom toolbar provides the following buttons:

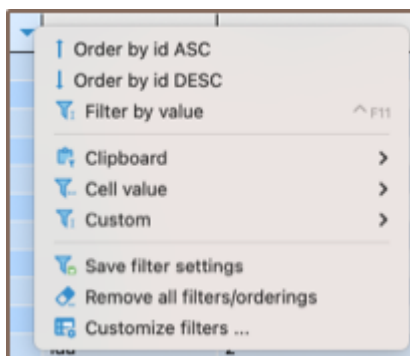
Button	Name	Description
	Refresh	Refreshes the whole results set, including all items that are not visible on the screen, while its dropdown option allows to customize the refresh frequency over a specific period.
	Save	Saves all unsaved changes to the data such as adding, duplicating, deleting rows, inline editing of values. See the Data Viewing and Editing article for information.
	Cancel	Discards all unsaved changes to the data.
	Edit cell value in separate dialog /editor	Opens the cell in focus for editing in a separate editor or dialog box. See details in the Cell Editor section of our article.
	Add new row	Adds a new empty row below the current row, see details in the Adding, Copying and Deleting Rows section of our article.

	Duplicate current row	Copies the current rows and pastes the copy below the current row, see details in Adding, Copying and Deleting Rows section of our article.
	Delete current row	Colors the rows in focus in red to mark them for deletion, see details in the Adding, Copying and Deleting Rows section of our article.
	Move to first row	Moves the focus (highlighting) from the current to the first row of the table.
	Move to previous row	Moves the focus (highlighting) from the current to the previous row of the table.
	Move to next row	Moves the focus (highlighting) from the current to the next row of the table.
	Move to last row	Moves the focus (highlighting) from the current to the last row of the table.
	Fetch all data	Fetches the whole result set making it ready for display. See the Scrolling Results Page section of our article.
	Configure	Opens a menu with settings.
	Result-set fetch size	Displays the initial fetch number of rows in the result set on loading or refreshing.
	Calculate total row count	Calculates the total number of rows in the table.

Note: Some of these buttons may be disabled and may not work if you are using a read-only connection, connecting to a read-only database or if you see the result of a complex query, such as joining two or more tables.

Column context menu

Each column has a context menu, accessed by clicking the downward arrow button , providing different filter options.



Cell context menu

Every cell in the table data also has a context menu – right-click the cell to open the menu. The context menu provides the following items:

Menu Item	Description
Copy	Copies the content of the current cell or column to the clipboard.
Advanced Copy	Opens advanced copy submenu that allows copying data with preset formatting parameters.
Paste	Pastes the copied content to the cells in focus.
Advanced Paste	Pastes several values delimited with a tabulation or line break starting from the selected cell.
Delete	Deletes the row that has the cell in focus. NOTE: In fact, when users click Delete , the system only highlights the red row while the actual deletion happens when users click Save .
Edit	Opens a submenu enabling inline editing. See the Data Viewing and Editing article.
Order	Displays a submenu that allows to specify ordering criteria for the data. The submenu contains the most common ordering options that can be applied to the column in focus. See details in Data Filters article. By default, DBeaver orders data by sending a request to the server (the Server-side results ordering checkbox selected). To order data on the client side using DBeaver's internal algorithm, clear the checkbox.
Filter	Displays a submenu that allows to specify filter criteria for the data. The submenu contains the most common filters that can be applied to the cell in focus. See details in Data Filters article. By default, DBeaver filters data by sending a request to the server (the Server-side results ordering checkbox selected). To filter data on the client side using DBeaver's internal algorithm, clear the checkbox.
View /Format	Opens a submenu that provides options for formatting and modifying the view of data. See the Data View and Format article.
Navigate	Opens a submenu that helps users to navigate through the table data. See the Navigation article.
Layout	Changes the layout of data, see the Data View and Format article.
Export	Opens the Data Transfer wizard that guides you through the steps to export data to a selected format. See the Data Transfer article.

data	Note: The system exports the whole result set including records that are not visible on the screen and preserves all applied data filters and ordering.
Generate SQL	Opens a submenu where you can select the type of SQL query to generate. See the SQL Generation article.
Generate Mock Data	Opens Mock Data Generator. See the Mock Data Generation in DBeaver article.
Logical structure	Opens a submenu allowing you to write virtual column expressions. See the article about Virtual column expressions .
Open with	Opens the data in external applications like Excel or a web browser.
Toggle result panel	Opens Value Viewer panel on the right side of the Data Editor.
Refresh	Refreshes the whole results set including all items that are not visible on the screen.

For more information about using the Data Editor, please see the subsections of this article - open them via the contents tree on the right.

Data View and Format

Table of contents

[Grid vs Plain Text Views](#)

[Table vs Record Views](#)

[Rows Coloring](#)

[Value range and gradient](#)

[Using regex](#)

[Coloring by Data Types](#)

[Transforming Data Presentation](#)

[Structuring Complex Data Types](#)

[Configuring Numeric and Time Data Formats](#)

[Configuring Boolean presentation](#)

[Text-based presentation](#)

[Icon-based presentation](#)

Grid vs Plain Text Views

You can switch between two data presentations in SE version and four presentations in EE version. Pressing **CTRL+.** switches available presentations in turn.







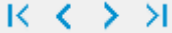
- To see the data in a grid view, similar to an Excel spreadsheet, click the **Grid** button ( **Grid**) on the bottom toolbar of the editor.
- To switch to the plain text view, click **Text** ( **Text**) on the bottom toolbar.
- To switch to JSON view (available in EE version only for MongoDB documents and JSON tables), click **JSON** on the toolbar.
- To switch to XML view (available in EE version only for XML tables), click **XML** on the toolbar.

Table vs Record Views

The table view is a standard table (Excel-like) in which columns are vertical and rows are horizontal. This view is the default one. If you click the **Record** button in the bottom toolbar of the editor ( **Record**), or press `Tab` , or right-click a cell and then click **Layout -> Record** on the context menu, the rows and columns switch positions. The columns will appear as rows, and the rows will be hidden in one **Value** column which will show only one row of data. The column headers will shift from the top of the table to its left side:

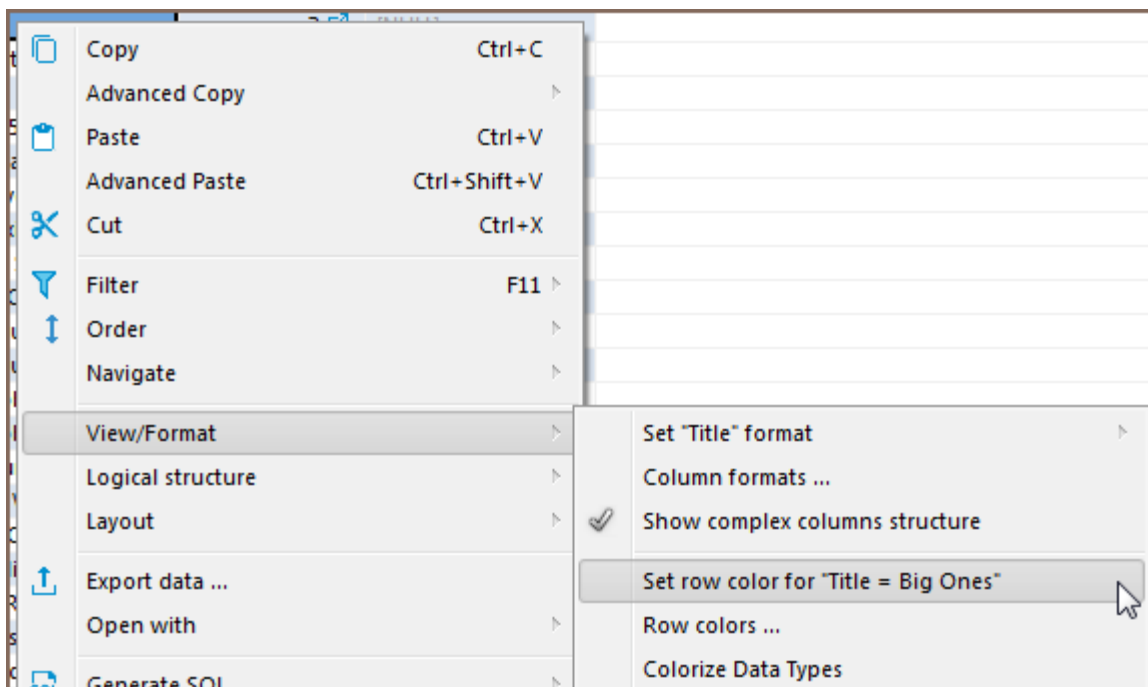
 ?	Value
123 AlbumId	22
ABC Title	Sozinho Remix Ao Vivo
123 ArtistId	 16
 Column1	[NULL]

The Record view is useful if the table contains a big number of columns. To navigate from row to row of data, use the navigation buttons on the bottom toolbar of the editor: 

To return back to the standard table view, click the **Record** button again.

Rows Coloring

In the data editor, you can color all rows that have the same value as a particular cell of a certain column. To do so, right-click the cell and click `View/Format => Set the row color for {column name = value}` on the context menu:

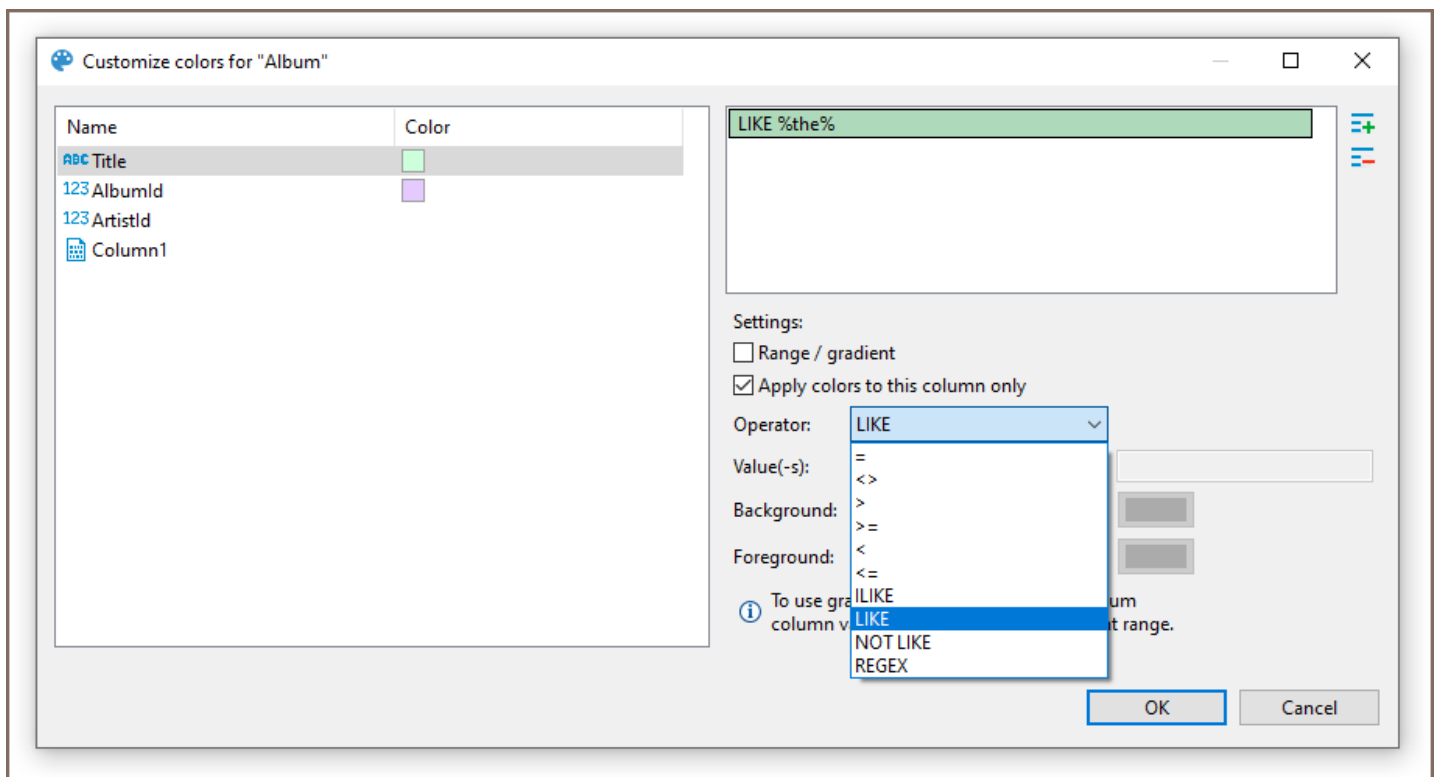


Then choose the color in the palette window that appears and click **OK**. The current row and all other rows that contain the same value change their color to the one you have selected:

	123 AlbumId	ABC Title	123 ArtistId	Column1
3	3	Restless and Wild	2	[NULL]
4	4	Let There Be Rock	1	ewqewq
5	5	Big Ones	3	[NULL]

To remove the coloring by a particular column, right-click the cell again and click **View/Format** ⇒ **Clear color** on the context menu.

By choosing **View/Format** ⇒ **Row colors ...** from the context menu, you can gain more precise control of coloring conditions:



Here, you can define multiple conditions for single column using rich set of predefined operators, change background color and define a range between two values.

Operators work as you may expect. Note that they're executed on the client-side, that means no extra queries are made in order to apply colors.

Value range and gradient

Value range allows you to paint your rows with gradient that fades from first value to second value:

ABC Title	123 AlbumId	123 ArtistId	Column1
For Those About To Rock W	1	1	PNG IHDR ,Ó9...
Balls to the Wall	2	2	PNG IHDR n n I9±.
Restless and Wild	3	2	[NULL]
Let There Be Rock	4	1	ewqewq
Big Ones	5	3	[NULL]
Jagged Little Pill	6	4	[NULL]
Facelift	7	5	[NULL]
Warner 25 Anos	8	6	[NULL]
Plays Metallica By Four Cell	9	7	[NULL]
Audioslave	10	8	[NULL]

In this example we defined a range for column `AlbumId` that fades from `#80c6ff` to `#8000ff` between values `1` and `10`.

Using regex

You can use regular expressions for matching complex values. Otherwise, you can be artistic and, for example, paint rows with odd values in your favorite color:

APC Title	123 AlbumId	123 ArtistId	Column1
For Those About To Rock W	1	1	PNG IHDR ,Ó9...
Balls to the Wall	2	2	PNG IHDR n n I9±.
Restless and Wild	3	2	[NULL]
Let There Be Rock	4	1	ewqewq
Big Ones	5	3	[NULL]
Jagged Little Pill	6	4	[NULL]
Facelift	7	5	[NULL]
Warner 25 Anos	8	6	[NULL]
Plays Metallica By Four Cello	9	7	[NULL]
Audioslave	10	8	[NULL]

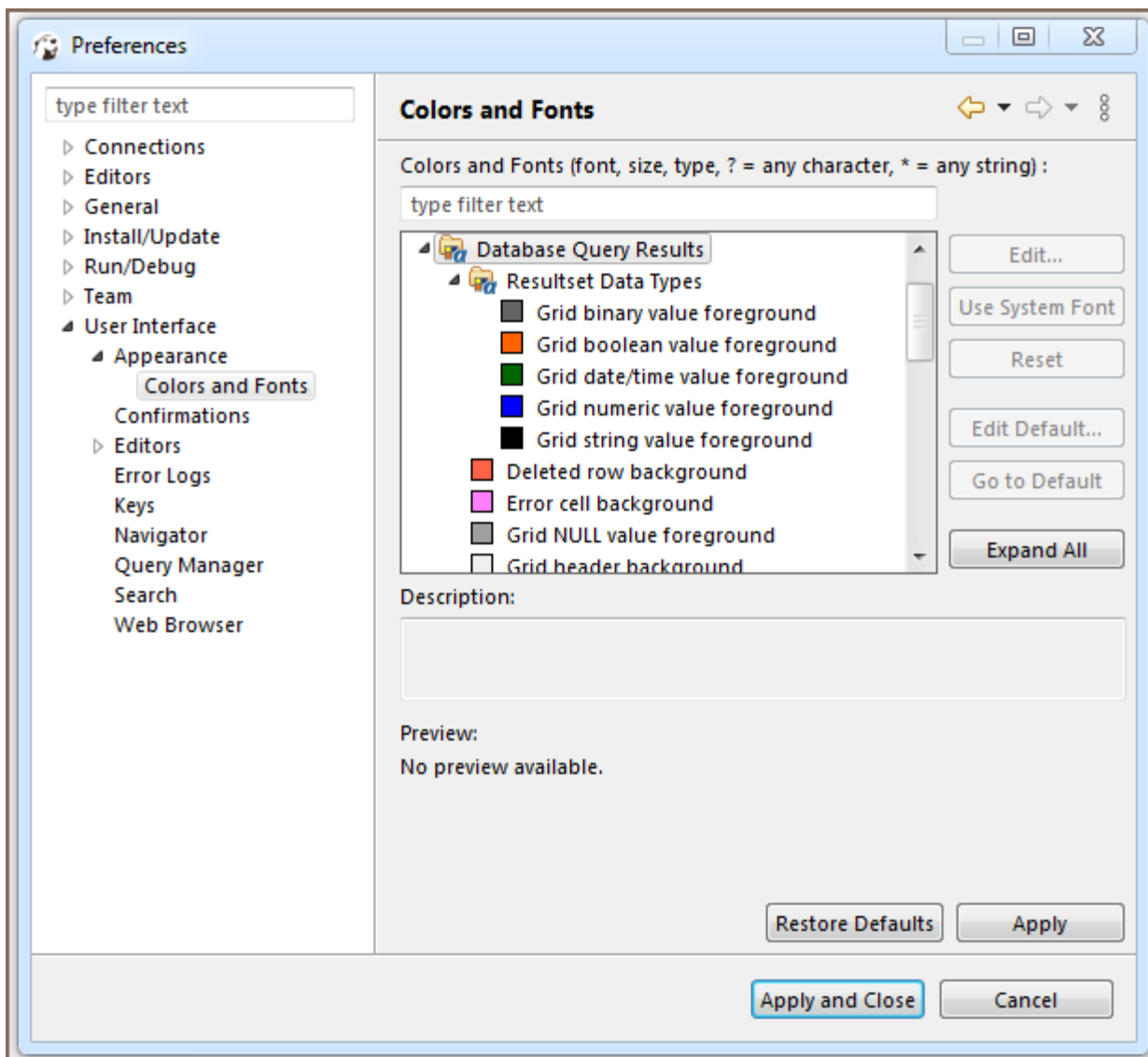
Snippet for coloring odd rows: `^\d*[13579]$` ; even rows: `^\d*[02468]$`

Coloring by Data Types

Besides coloring rows by a value, you can color the values in the columns by data types. To do so, right-click any cell in the table and, on the context menu, click **View/Format -> Colorize Data Types**. The values in the cells will be colored in different Colors according to the current color preferences:

	123 rental_rate	123 length	123 replacement_cost	rating	last_update	special_features	fulltext
1	0.99	86	20.99	PG	2007-09-10 17:46:03	{Deleted Scenes,Behind the Scenes}	'academi':1
2	4.99	48	12.99	G	2007-09-10 17:46:03	{Trailers,Deleted Scenes}	'ace':1 'adn
3	2.99	50	18.99	NC-17	2007-09-10 17:46:03	{Trailers,Deleted Scenes}	'adapt':1 'a:
4	2.99	117	26.99	G	2007-09-10 17:46:03	{Commentaries,Behind the Scenes}	'affair':1 'ch
5	2.99	130	22.99	G	2007-09-10 17:46:03	{Deleted Scenes}	'african':1 'i
6	2.99	169	17.99	PG	2007-09-10 17:46:03	{Deleted Scenes}	'agent':1 'ai
7	4.99	62	28.99	PG-13	2007-09-10 17:46:03	{Trailers,Deleted Scenes}	'airman':1 'i

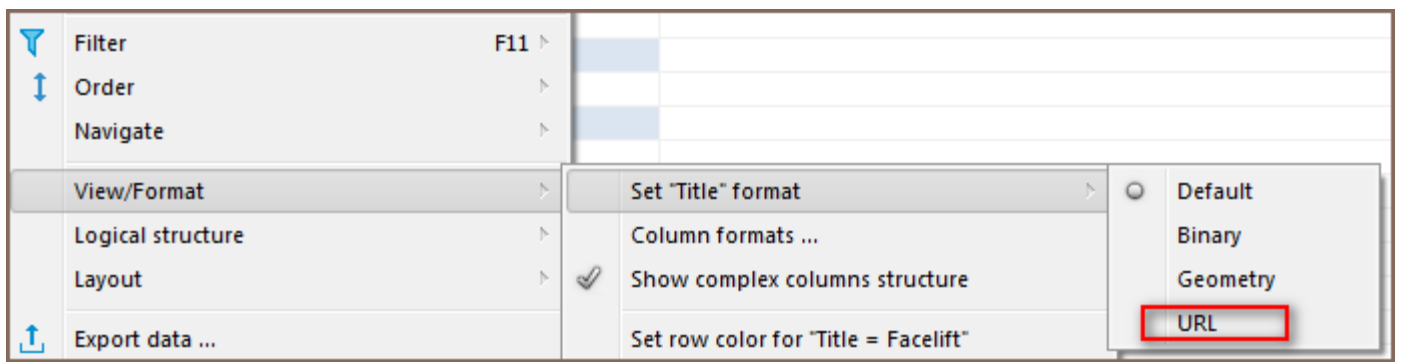
You can change the color preferences in the Preferences window by: clicking **Window -> Preferences** on the main menu. In the window of the navigation pane on the left, expand **User Interface** and then **Appearance**, and then click **Colors and Fonts**:



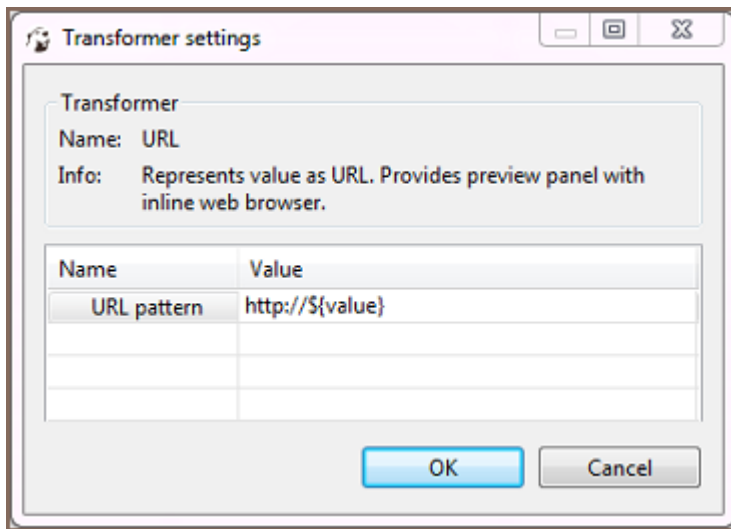
To remove the coloring by data types, on the context menu, click **View/Format -> Colorize Data Types** again.

Transforming Data Presentation

For string and numeric data types, DBeaver provides tools to transform the data presentation into a number of formats, such as URL and Binary for strings and Epoch Time, Number Radix, etc. for numbers. To change the data presentation in a certain column, right-click a cell in the column. Then, on the context menu, click **View/Format -> Set {column name} format** and click the presentation type name:



The Transformer settings window opens showing the value in the chosen format. Click **OK** to apply the change:



The values in the column appear in the new format.

NOTE: For URL format, the resulting cell provides a link to the URL in a browser window.

To roll back the changes to the default format, right-click any cell in the column, and on the context menu, click **View/Format -> View as -> Default**.

Structuring Complex Data Types

For complex data types (that themselves represent a structure), such as objects, structures and arrays, DBeaver provides a tool for breaking them into columns:

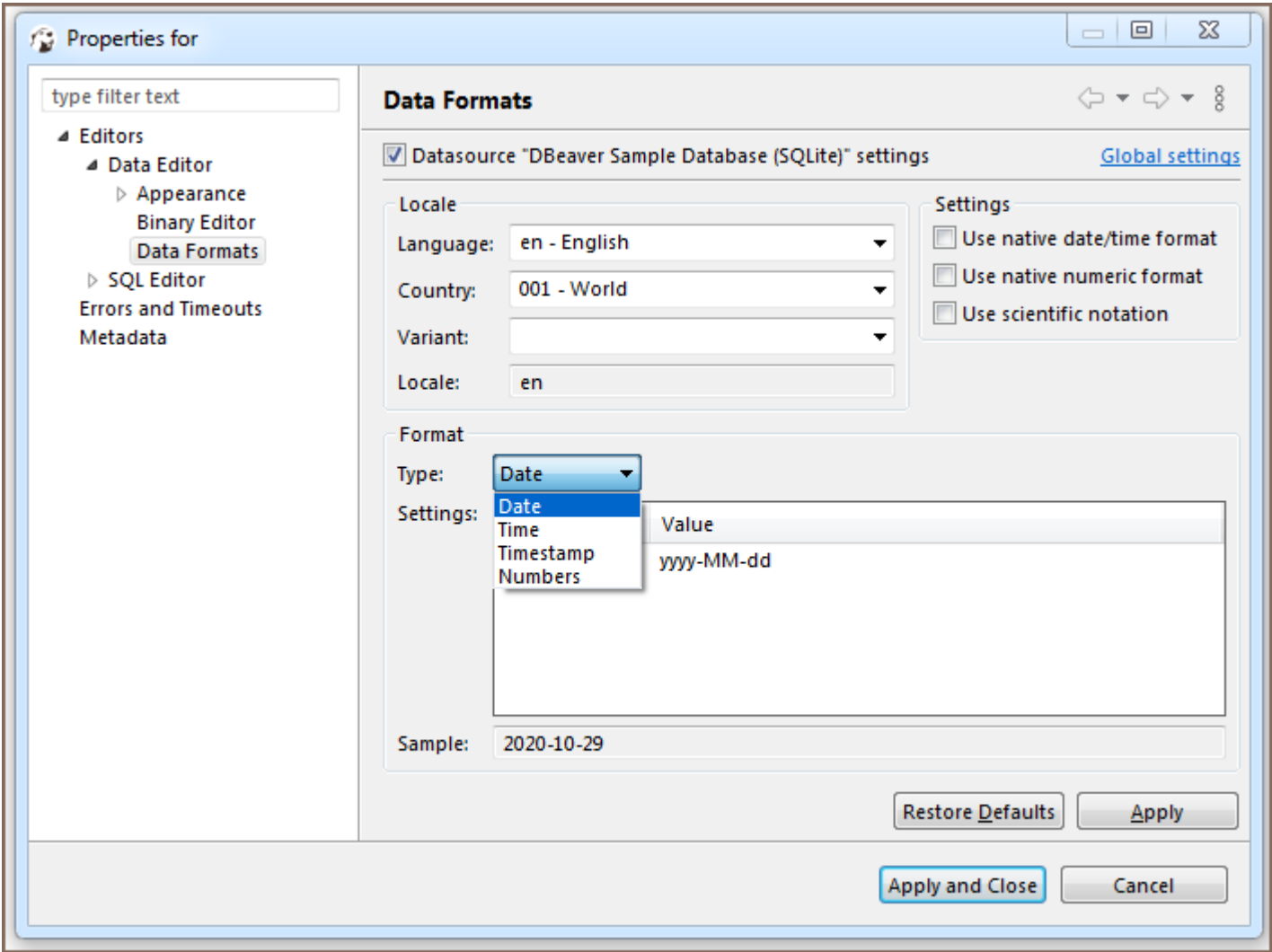
Properties Data ER Diagram									
restaurants Enter a SQL expression to filter results (use Ctrl+Space)									
	ABC zipcode	ABC borough	ABC cuisine	grades	date	ABC grade	123 score	ABC name	ABC restaurant_id
1	10462	Bronx	Bakery	2014-03-03 04:00:00	A			2 Morris Park Bake Shop	30075445
2	11225	Brooklyn	Hamburgers	2014-12-30 03:00:00	A			8 Wendy'S	30112340
3	10019	Manhattan	Irish	2014-09-06 04:00:00	A			2 Dj Reynolds Pub And Restaurant	30191841
4	11224	Brooklyn	American	2014-06-10 04:00:00	A			5 Riviera Caterer	40356018

To do so, right-click a cell in the column and, on the context menu, click **View/Format -> Visualize complex columns**

.

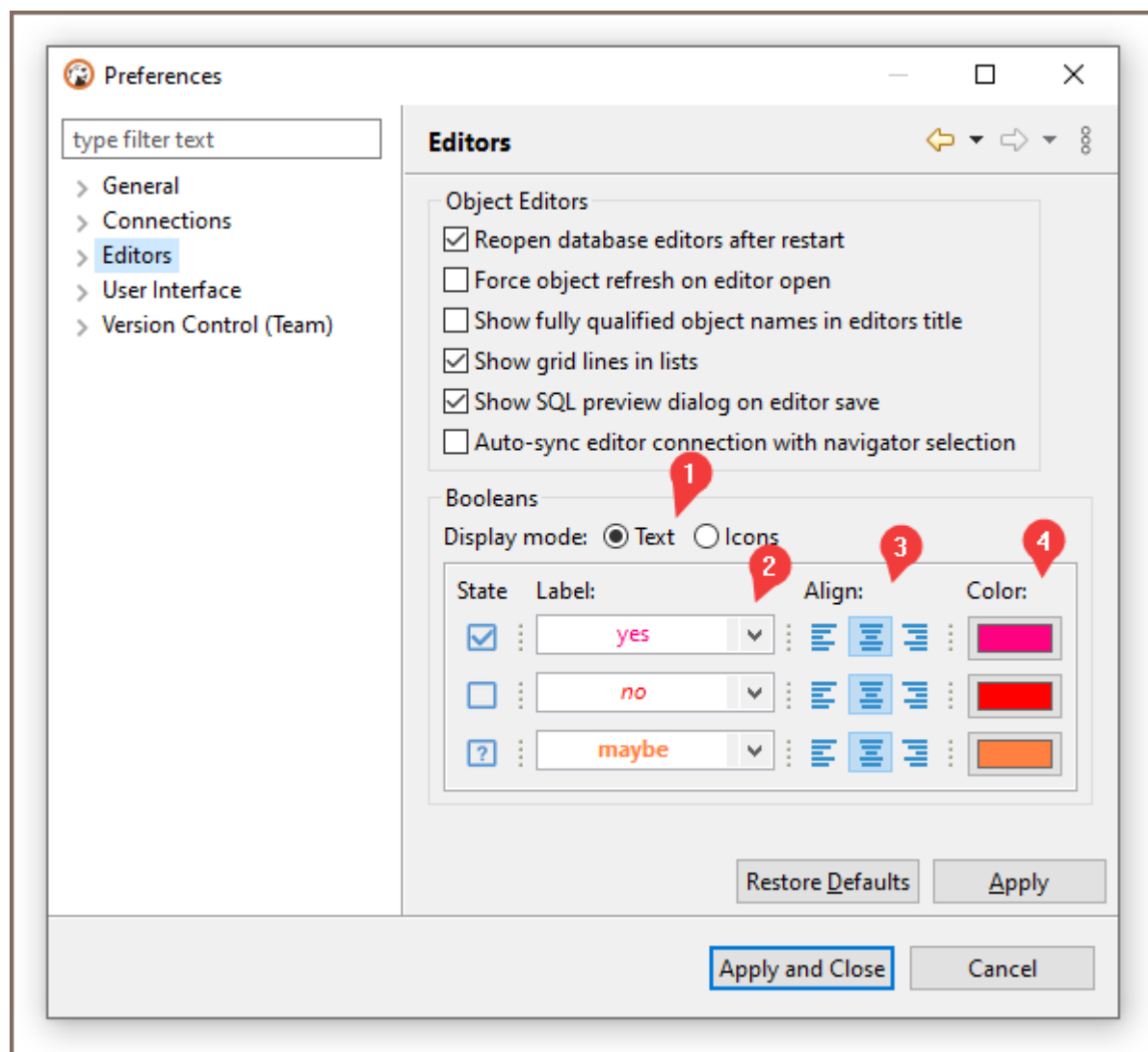
Configuring Numeric and Time Data Formats

You can specify the exact format of Time, Timestamp, Date, and Number data used in the currently open database or globally. To specify a format, right-click any cell in the table and, on the context menu, click **View/Format -> Data formats**. The Properties window opens displaying the **Data Formats** page:



To configure only the format for the current database, select the **Datasource "[Connection name]" settings** checkbox. To configure the settings globally, to all databases that you have in DBeaver, click **Global settings**. You can specify the locale for the data format in the **Locale** area. In the **Type** dropdown list, click the name of the data type and in the **Settings** table, click the required format. To apply the changes and make them visible in the table, click **Apply and Close** and then refresh the window (F5).

Configuring Boolean presentation



You can choose between two presentation modes¹:

- Text-based
- Icon-based

Text-based presentation

This is the most customizable mode. You can:

1. Change **labels** under **Label** column.

Otherwise, you can use presets available in **Drop-down Menu**²

2. Change **alignment** of value inside grid cell.³

Following variants are available: **left**, **center**, and **right**

3. Change **color** of value using color picker under **Color** column⁴.

You can reset color to match current theme's contrast color in **Drop-down Menu**² ⇒ **Colors** ⇒ **Use the**

4. Change font **style** in **Drop-down Menu**² ⇒ **Styles** .

Following variants are available: normal, **bold**, and *italic*

Icon-based presentation


This presentation only supports alignment changing.

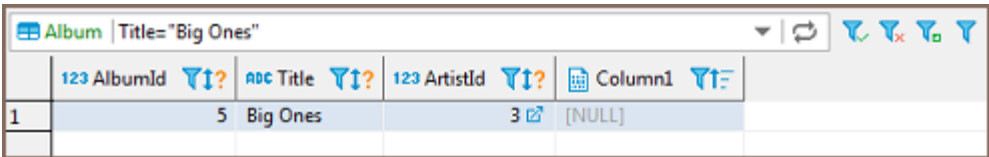
Data Filters

Table of contents

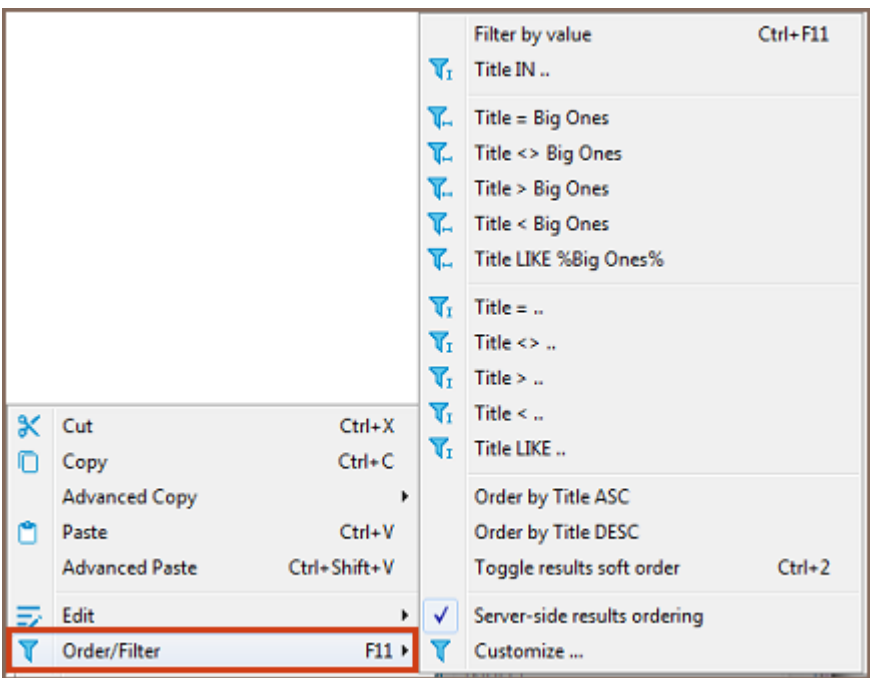
- [Advanced filters configuration](#)
- [Ordering Data in Columns](#)
- [Managing Display of Columns in Data Table](#)
- [Sorting Columns in Data Table](#)

You can apply custom filters to table contents or query results. There are several ways in which you can filter data in the table.

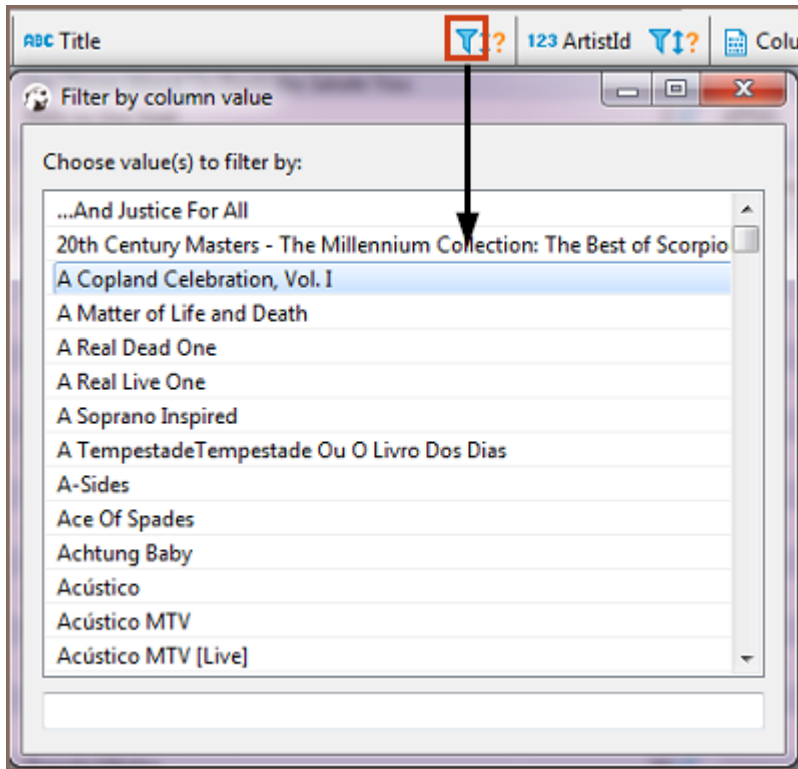
One of the ways is to use the filter field above the table next to the top toolbar. To filter data, enter an SQL expression into the field and click the Apply filter criteria button () next to the field or press Enter .




You can apply ready-to-use SQL expressions or SQL expression templates via the context menu. To select a ready SQL expression or a template, press F11 or right-click the cell, then click **Order/Filter** on the context menu and then click one of the expressions.



The third way is to filter data by a cell value using the filter icon in the column header. To filter data this way, click the filter icon in the column header and then double-click the cell value in the Filter by the column value dialog box:



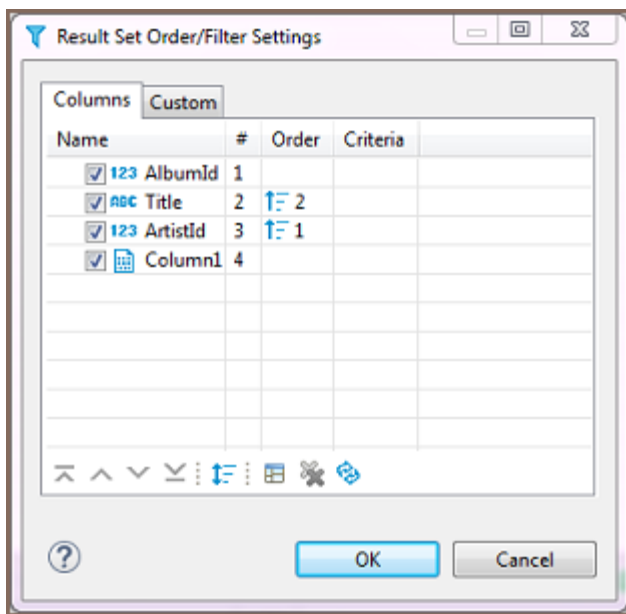
The data updates dynamically. To remove a filter, click the Remove All Filters/Orderings button () in the top toolbar of the editor.

You can save the current filter settings for the database object to apply the next time you reopen it in the editor.

To save the current filter settings, click the Save filter settings for current object button () in the top toolbar.

Advanced filters configuration

The main tool for managing the appearance of the data table is the Result Set Order/Filter Settings window.

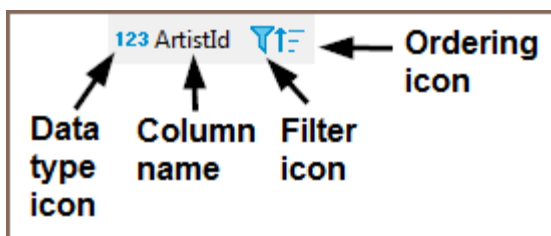


To open this window, click the Custom Filters button (🔍) in the top toolbar of the editor or click the Configure button (⚙️) and then click **Order/Filter** on the dropdown menu.

The Result Set Order/Filter Settings window provides tools to:

- Order data inside columns
- Manage the display of columns in the table
- Manage the order of columns in the table
- Filter data in the table using an SQL expression

Another tool for managing the data appearance is the column headers. In the data table, every column header contains three elements which each has its own function: Data (column) type icon, column name, filter icon, and ordering icon.




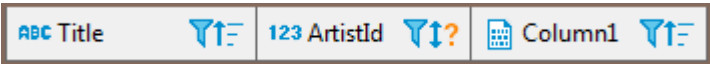
- By simply clicking the column name or column type, the icon highlights the whole column.
- You can click the column type icon and then drag and drop the column to a different position in the table.
- You can click the column name and then drag the cursor to the right or left to highlight multiple columns.

- Clicking the ordering icon allows you to order the data in the column in ascending or descending order - see the 'Ordering Data in Columns' section, further in this article
- Clicking the filter icon allows you to filter the data by a cell value, see [TBA]




Ordering Data in Columns

You can order data in columns in one of the following ways:


1. Click the ordering icon () in the header of the column.





The icon has three states:

- Clicking once establishes ascending order ()
- Clicking a second time changes the order to descending ()
- Clicking a third time removes the ordering from the column ()





To order data by several columns, go column by column, setting the order with the Ordering icon, starting from the column by which you want to order data first.

2. Click the Custom Filters button () in the top toolbar of the editor to open the Result Set Order/Filter Settings window (see above):
 - a) Next to the column by which you want to order data in the first turn, set the ascending or descending order using the same three-state principle as described above.
 - b) Set the ordering in other columns by which you want to sort the data in the second, third, etc. turn. The **Order** column indicates the order in which the sorting will happen.


NOTE: The number (#) column indicates the initial order of columns.

Columns		
Custom		
Name	#	Order
<input checked="" type="checkbox"/> 123 AlbumId	1	
<input checked="" type="checkbox"/> 123 ArtistId	3	 2
<input checked="" type="checkbox"/> Column1	4	 1


- c) To easily move the ordering setting from column to column, you can use the Move up/down/to top/to

bottom/ buttons:    

To reset the data ordering to its initial state, click the Reset button () in the same window.

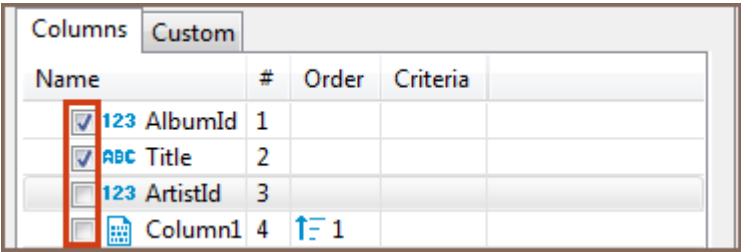
Also, to remove all ordering settings, click the Remove All Filters/Orderings button () in the top toolbar of the Data Editor.


Managing Display of Columns in Data Table

To hide a single column, right-click the column or any cell in it and click **View/Format -> Hide column** on the context menu. To unhide a hidden column, open the Result Set Order/Filter Settings window (see the image at the beginning of this article) and select the checkbox next to the column name, or click the Reset button ().

To display or hide columns in the data table, in the Result Set Order/Filter Settings window:



1. Select the checkboxes next to the columns that you want to see in the table and clear the checkboxes next to those that you want to hide.







2. Use the Show All () and Show None () buttons at the bottom of the window.

Sorting Columns in Data Table

You can modify the order of columns in the data table in two ways:

1. Click the icon in the column header and drag-and-drop the column to a new position.
2. To sort the column alphabetically, in the Result Set Order/Filter Settings window (open by clicking the Custom Filters button () in the top toolbar of the editor), click the Sort button ()

3. In the Result Set Order/Filter Settings window, click the column to set the focus to it and then move it using the navigation buttons: (   )

Data viewing and editing

Table of contents

[Overview](#)

[Inline editing](#)

[Cell Editor](#)

[Data format handling](#)

[Specialized handling of CLOB and BLOB data types](#)

[Setting cell value to NULL](#)

[Setting cell value to default](#)

[Rows manipulation](#)

[Copying and pasting cells](#)

[Basic copy](#)

[Advanced copy](#)

[Basic paste](#)

[Advanced paste](#)

[Saving and previewing changes](#)

Overview

DBeaver provides tools for data editing and viewing within its [Data Editor](#). It supports inline editing in the grid view for quick data changes and a cell editor for more intricate entries, including a multi-line text. Users can manage rows by adding, copying, or deleting them and can utilize copy-paste functions for data transfer. Detailed information on these features and the steps for saving and reviewing changes are provided in the following sections.

Inline editing


Inline editing refers to the process of modifying content directly within a cell in a table. To initiate inline editing, users have several options:

- Double-click the desired cell.
- Click the cell to focus on it, then press `Enter` .
- Right-click the cell and select **Inline edit** from the context menu.

Once in inline editing mode, the cell becomes editable, allowing the user to modify its value.

Cell Editor

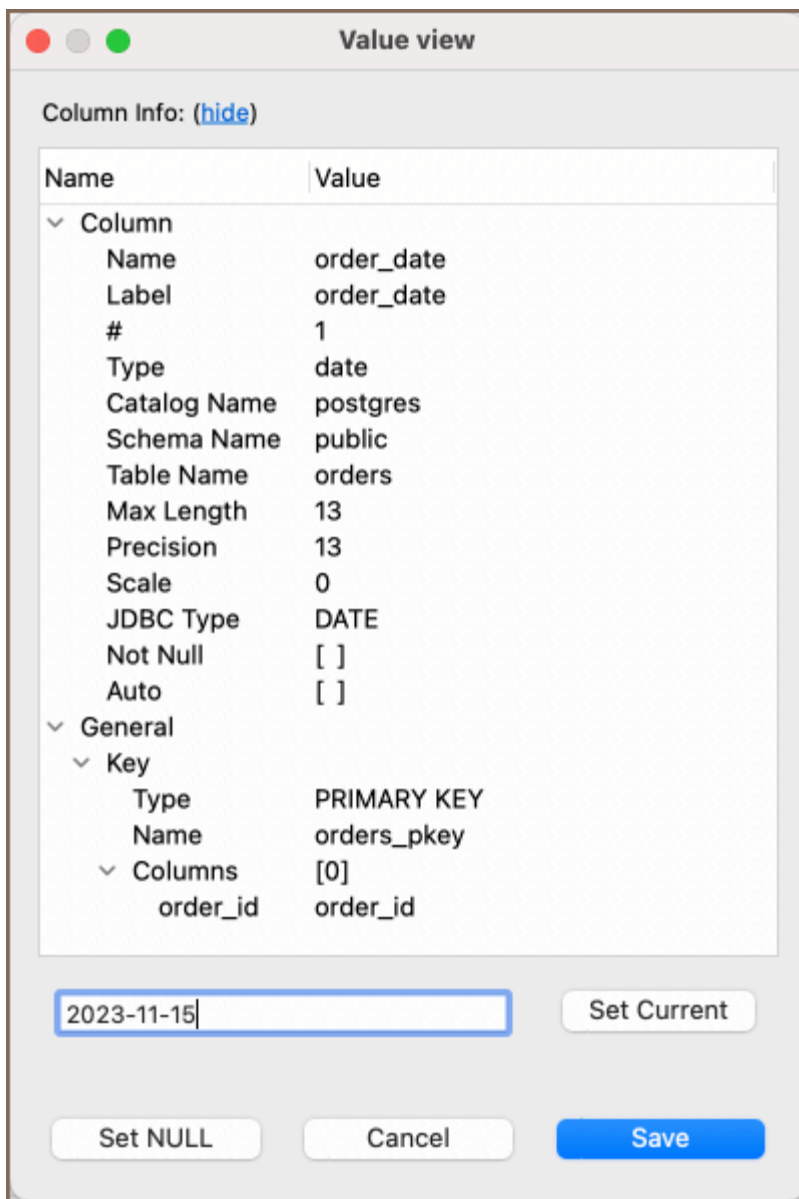
The Cell Editor in DBeaver is a tool designed for detailed data editing within individual cells. To access the Cell Editor, follow either of these methods:

- Right-click on the desired cell and select **Edit -> Edit cell** from the context menu.
- Click the cell to focus, then either press `Shift+Enter` or click the **Edit cell** button () located in the bottom toolbar.

Note: The toolbar is customizable. See [Toolbar Customization](#) article.

Data format handling

DBeaver manages a variety of data formats, and for most cells, activating the Cell Editor displays a property window:



This window is divided into two main sections:

- **Column Info:** Provides detailed properties of the column.
- **Value:** Enables users to modify the cell's value. After making changes, click **Save** to apply them. For extended editing, use the **Open Editor** option for a separate editor tab.

Specialized handling of CLOB and BLOB data types

In addition to general data formats, DBeaver offers specialized support for **CLOB** / **BLOB** data types:

- **Editing CLOB/BLOB data:** Editing cells containing **CLOB** / **BLOB** data in the Cell Editor opens the content in a new tab, allowing for comprehensive editing, particularly useful for large data objects.

- **Saving/Loading BLOB values:** Users can save and load **BLOB** values to and from external files, aiding in the management of binary data like images and large text files.
- **Image display in BLOB Columns:** DBeaver automatically renders images stored in **BLOB** columns (formats like **gif** , **png** , **jpeg** , **bmp**) within the interface, streamlining the process of viewing image data.
- **Value View panel:** To efficiently browse images in **BLOB** columns, users can activate the value view panel by pressing **F7** .

Setting cell value to NULL

To set a cell's value to **NULL** , right-click on the cell and choose **Edit -> Edit cell -> Set to NULL**.




Setting cell value to default

To set a cell's value to its default, right-click on the cell and select **Edit -> Set to default** from the context menu.

The default value for the field, if predefined, will be displayed in parentheses next to this option.

Rows manipulation

Refer to the table below for instructions on adding, copying, and deleting rows in DBeaver's Data Editor.

Action	Description	Button /Image
Adding rows	To add a new, empty row, click the Add row button in the bottom toolbar. This inserts a row below the focused row. Populate the new row via inline editing or separate cell editor.	
Copying rows	To duplicate rows, select the rows and click the Duplicate row button. The duplicates will appear below the original rows.	
Deleting rows	For deleting rows, select them and click the Delete current row button. Rows marked for deletion turn red and are removed upon saving.	

Copying and pasting cells

This section outlines the procedures for copying and pasting cell data within DBeaver.

Basic copy

To copy the contents of one or more cells to the clipboard in a TAB-delimited format, you can use the shortcut `Ctrl+C` . Alternatively, right-click on the cell or selection of cells and select **Copy** from the context menu. This copied content can then be pasted into spreadsheet editors like Excel.

Advanced copy

For more control over the copy process, DBeaver offers an advanced copy option. This feature allows users to include column names and row numbers, configure delimiters, and choose value formats during the copy operation. Activate this feature by pressing `Ctrl+Shift+C` or by selecting **Advanced Copy -> Advanced Copy** from the context menu after right-clicking on the selected cells. The Advanced Copy settings window provides various options:

Field	Description
Copy header	Includes the column names as headers in the copied data.
Copy row numbers	Adds the row number for each row in the copied data.
Quote cell values	Encloses each cell value in quotes.
Always quote values	Forces quoting of all cell values, regardless of the need.
Copy as HTML	Formats the copied data as HTML, suitable for web publishing.
Value display format	Determines how the copied data will be displayed.
Column Delimiter	Character or sequence used to separate columns in the copied data.
Row Delimiter	Character or sequence used to separate rows in the copied data.
Quote Character	Character used to quote cell values if quoting is enabled.

In addition to these options, users can perform a **Save as** operation in various formats by selecting **Advanced Copy** from the context menu after right-clicking on the selected cells, which can be further configured for each format. Here are the additional options available:

--	--

Option	Description
Copy as CSV	Copies the data in Comma-Separated Values format.
Copy as DbUnit	Copies the data in the DbUnit dataset format.
Copy as HTML	Copies the data in Hypertext Markup Language format for web use.
Copy as JSON	Copies the data in JavaScript Object Notation format.
Copy as Markdown	Copies the data in Markdown format suitable for documentation.
Copy as SQL	Copies the data as SQL INSERT or UPDATE commands.
Copy as Source code	Copies the data in a format ready to use in source code.
Copy as TXT	Copies the data as plain text.
Copy as XML	Copies the data in eXtensible Markup Language format.

To further refine how each format is copied, select **Configure 'Copy as' commands...** from the context menu. This opens a configuration window where you can set the specific copying behavior for each format, such as delimiter choice, quoting options, and more, ensuring the copied data meets the requirements of different use cases and target applications.

Basic paste

To paste copied content, press `Ctrl+V` when focusing on a cell. DBeaver intelligently handles the pasting process, applying necessary data type conversions to fit the destination cell's format.

Advanced paste

DBeaver's **Advanced Paste** feature caters to scenarios requiring the insertion of multiple cells or complex data structures. It can be accessed through the context menu or by using the shortcut `Ctrl+Shift+V`. When activated, the following options are presented in a dialog:



Feature	Description
Insert multiple rows	Allows for the pasting of data into multiple rows at once.
Ignore quotes	Disregards quotation marks when parsing pasted values.

Insert NULLs	Allows for pasting the <code>NULL</code> values.
NULL value mark	Defines the symbol or text that represents a <code>NULL</code> value.

These settings provide additional control over how data is pasted, ensuring that data is inserted correctly and efficiently, especially in bulk operations or when special data considerations are necessary.


Saving and previewing changes

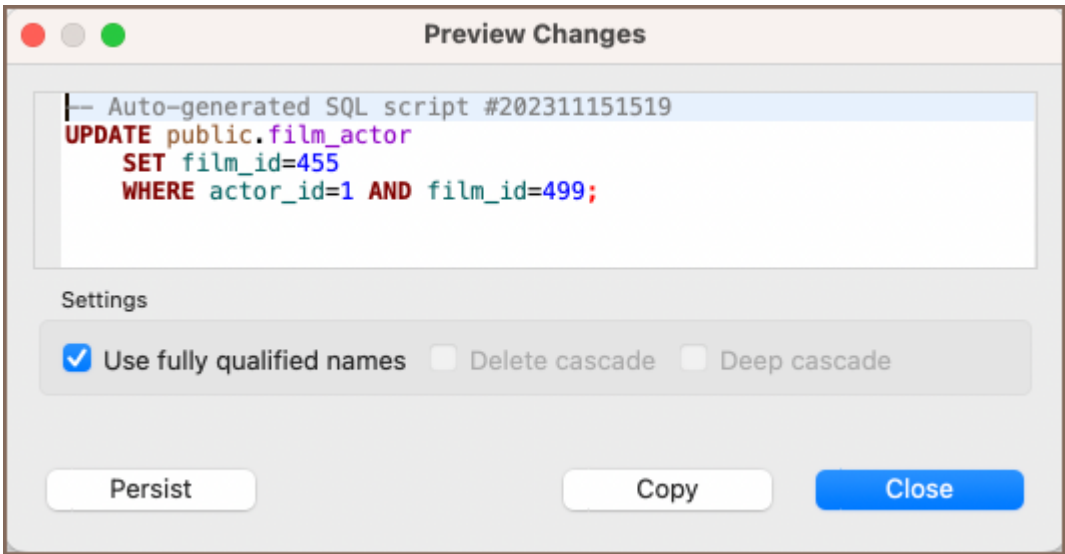
In DBeaver, data modifications are committed to the database when saved. To confirm changes:

- Click the **Save** button  **Save** in the bottom toolbar.
- To revert modifications, click the **Cancel** button .

Tip: The **Save** and **Cancel** buttons are enabled only after making edits and moving to another cell.

To validate changes before saving, preview the SQL script:

- Click the dropdown arrow next to the **Save** button .
- Choose **Generate Script**. The Preview Changes window will display the SQL script for review and copying if required.



Data refresh

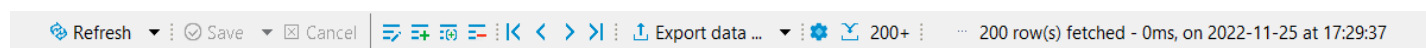
Table of contents


[Auto refresh](#)

[1 - Use a predetermined duration](#)

[2 - Set your duration](#)


In the bottom left part of the Data editor, you can find the toolbar with actions:



Refreshing is necessary if the database contains changes made by other users working on it simultaneously with you, and you want to see them in your DBeaver window. To refresh data manually, right-click anywhere in the data table and click  **Refresh** on the context menu or press F5 .

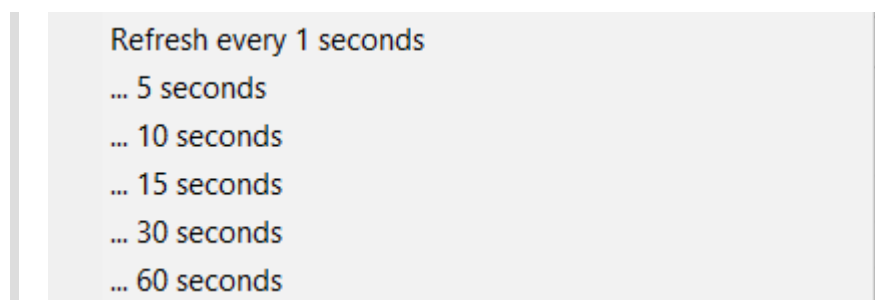
Auto refresh

You can also schedule auto-refresh to happen regularly. To auto-refresh the database on schedule:

Click on the arrow in the auto-refresh button  **Refresh** . Now you can configure auto refresh in the two following ways:

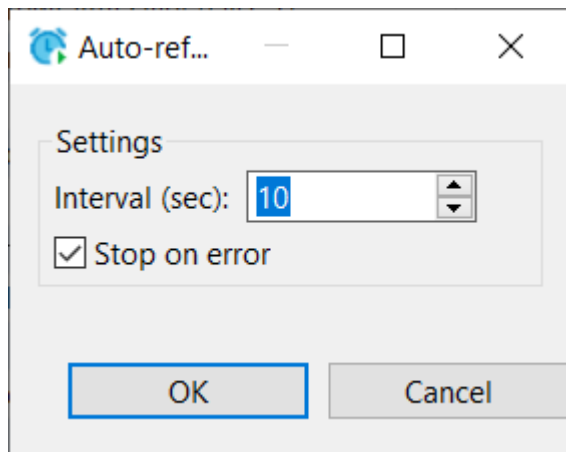
1 - Use a predetermined duration

You can select via choosing an available predetermined period in the context menu.




2 - Set your duration

You can set your time duration between refreshes. To do that, select **Customize ...**. The Auto-refresh configuration dialog box will open:



- a) Set **Interval** in seconds.
- b) Select the **Stop on error** checkbox if you want the refresh to stop when it encounters an error or clear it if the refresh should ignore mistakes.
- c) Click **OK**.

When you perform either of these two alternative options above, the system starts refreshing the data as scheduled, and the Configure auto-refresh button changes to the **Stop auto-refresh** button  **Refresh** ▼. To stop the auto-refresh, click the **Stop auto-refresh** button or the arrow next to it and click **Stop** on the auto-refresh menu.

Navigation

Table of contents

[Scrolling Results Page](#)


[Data Rows](#)

[History](#)

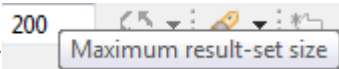
[Navigate Foreign Keys and Referencing Tables](#)


[Navigation Links](#)

Scrolling Results Page

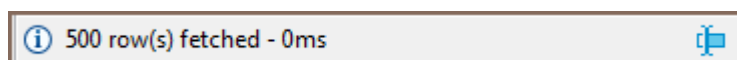
If the result set has many rows, you can scroll the results page. To learn how many rows the data table contains, click the **Calculate total row count** button in the bottom toolbar. The number of rows appears in a status field next to the button:  203 . Alternatively, you can right-click a cell in the table and then click **Navigate -> Row Count** on the context menu.

By default, DBeaver limits the number of rows fetched to **200** (you can change this value in the main toolbar or in preferences). The maximum number of rows that DBeaver fetches to display in the Data tab is specified in the


Maximum result-set size field in the main toolbar: 

Once you scroll to the last row of the current result portion, DBeaver fetches the next portion (next N rows). You can disable this behavior in preferences. You can also manually fetch the next portion of data equal to the maximum result set size. To do so, click the **Fetch next page of results** button () in the bottom toolbar or right-click the table and click **Navigate -> Fetch next page** on the context menu.

The number of rows fetched is visible in the status field under the data table:



To see the details, click the details button in the status field.

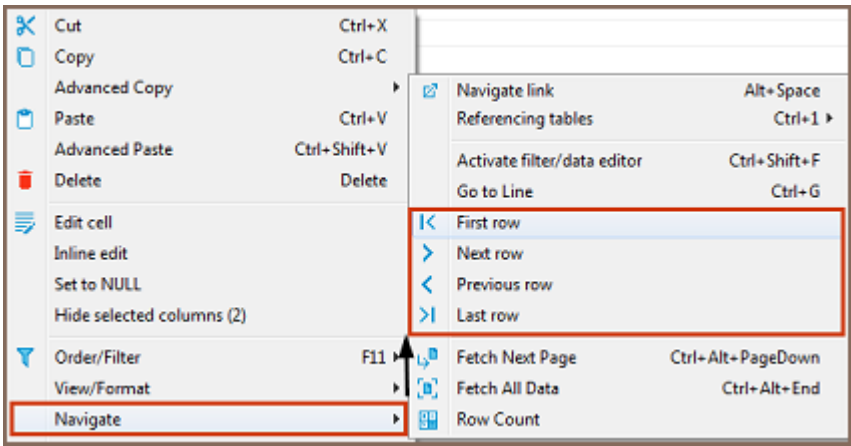
To fetch the whole result set, click the **Fetch all rows** button () in the bottom toolbar or right-click the table and click **Navigate -> Fetch All Data** on the context menu.

NOTE: Be careful when fetching the whole result set. If it is huge, it might cause program hangup or out-of-memory errors.

You can navigate through the result set using standard shortcuts Home , End , PgUp , PgDown , Ctrl+Home , (

Data Rows

To jump to the first or last row or move one row forward or backward, use the navigation buttons in the bottom toolbar or on the context menu:



To jump to a specific line, right-click anywhere in the table and click **Navigate** -> **Go to Line** on the context menu. Then in the Go to Row dialog box, enter the row number and click **OK**.

History

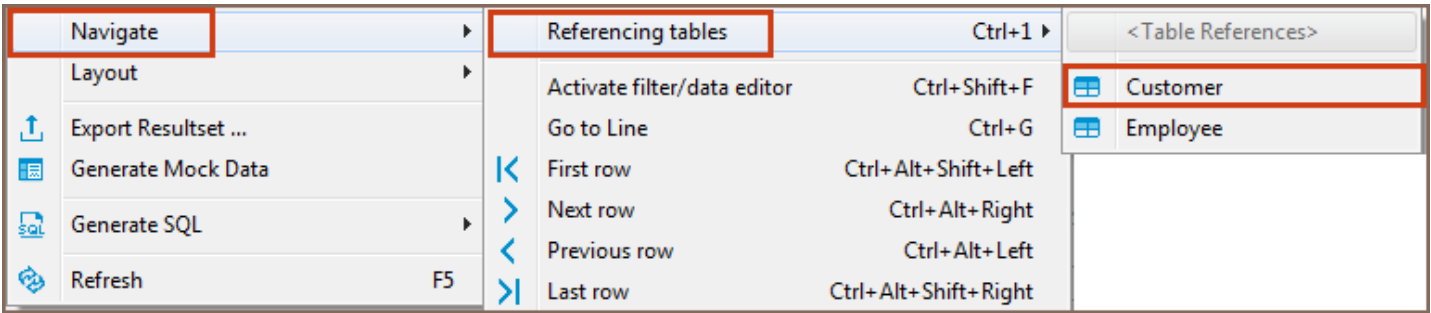
DBeaver remembers the history of actions such as applying filters to data, opening reference tables and other tables via links. You can navigate among such tables and filtered views:

- Use the forward and backward buttons in the top toolbar:
- Click Ctrl+Left or Ctrl+Right

Hovering over these buttons displays the names of the tables or filtered views saved in the history.

Navigate Foreign Keys and Referencing Tables

You can navigate with foreign keys or reference tables – those that reference the current table. To open a referencing table, press `Ctrl+1` or right-click the cell and click **Navigate->Referencing tables->[table name]**:



The referencing table opens in the same editor. To navigate back and forth between the initial and referencing tables, use the history navigation buttons (← →) in the top toolbar of the editor. In order to open a referencing table in a new window use the `Ctrl+Shift+1` shortcut to the show menu.

Navigation Links



In the data editor, you can navigate to linked tables – the ones that the current table references. To open a linked table, click the Navigate link icon in a cell that contains it:


	123 AlbumId	ABC Title	123 ArtistId
7	7	Facelift	5
8	8	Warner 25 Anos	6
9	9	Plays Metallica By Four Cel	7

Another way is to right-click such a cell and click **Navigate -> Navigate link** on the context menu. The linked table opens in the same editor, filtered by the cell value:

Artist ArtistId = 6		
	123 ArtistId	ABC Name
1	6	Antônio Carlos Jobim

NOTE: The table name in green above the table indicates which table is currently open in the editor.

To navigate back and forth between the initial and linked tables, use the history navigation buttons ( ) in the top toolbar of the editor.

You can open a linked table in a separate editor. To do so, simultaneously hold the `Ctrl` key (or `#` command key on macOS) and click the Navigate link icon () in the cell.

Managing Panels

Table of contents

[Value Viewer](#)

[Dictionary handling in Value Viewer](#)

[Working with dictionaries](#)

[Preferences Settings for dictionaries](#)

[Metadata Panel](#)

[Calc Panel](#)

[Grouping Panel](#)

[Function management in the Grouping panel](#)

[Adding a function](#)

[Removing a function](#)

[Column management in the Grouping panel](#)

[Adding a column](#)

[Removing a column](#)

[Additional settings](#)

[Default sorting](#)

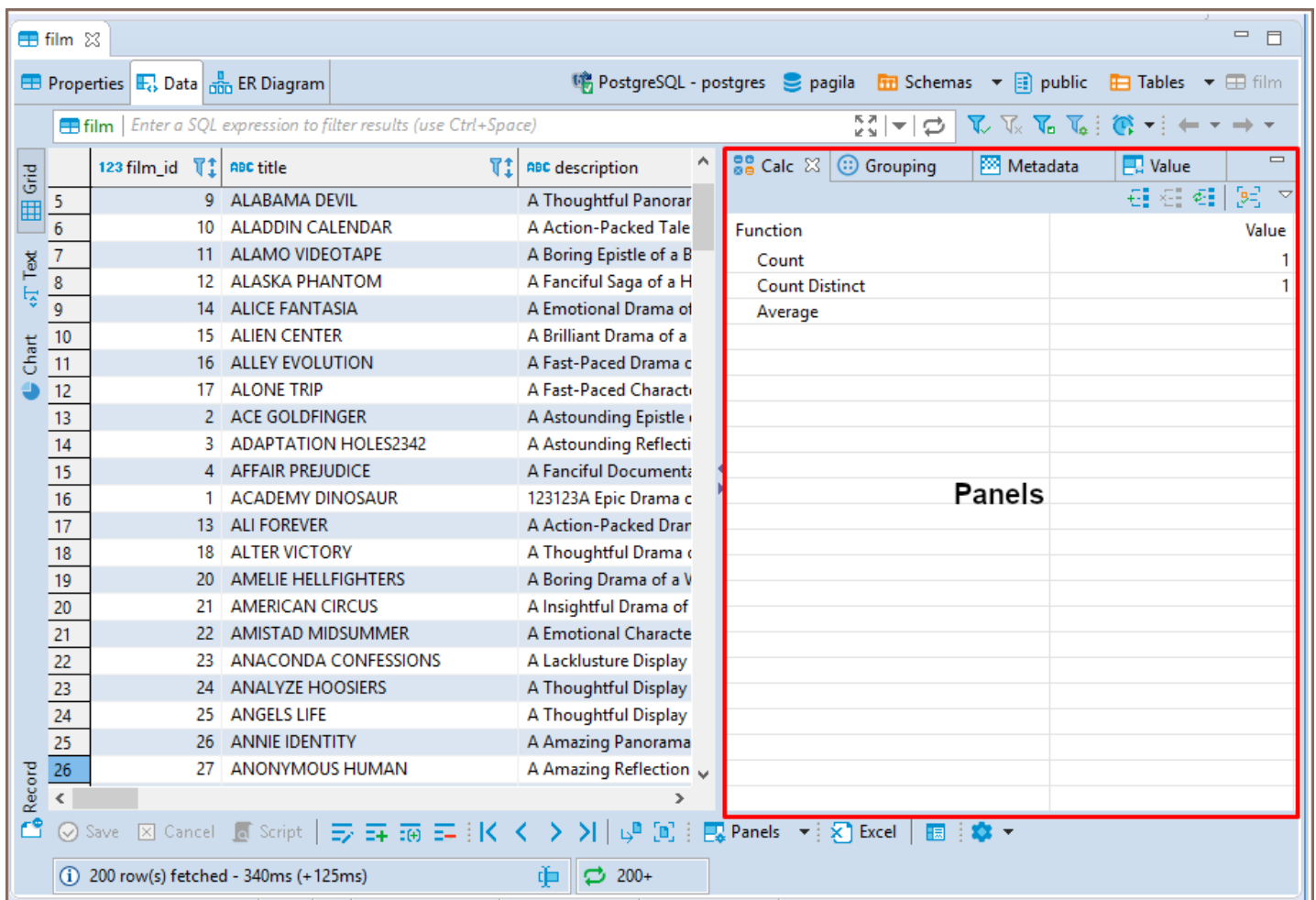
[Show duplicates only](#)

[References panel](#)

[Result details panel](#)

[Query Trace panel](#)

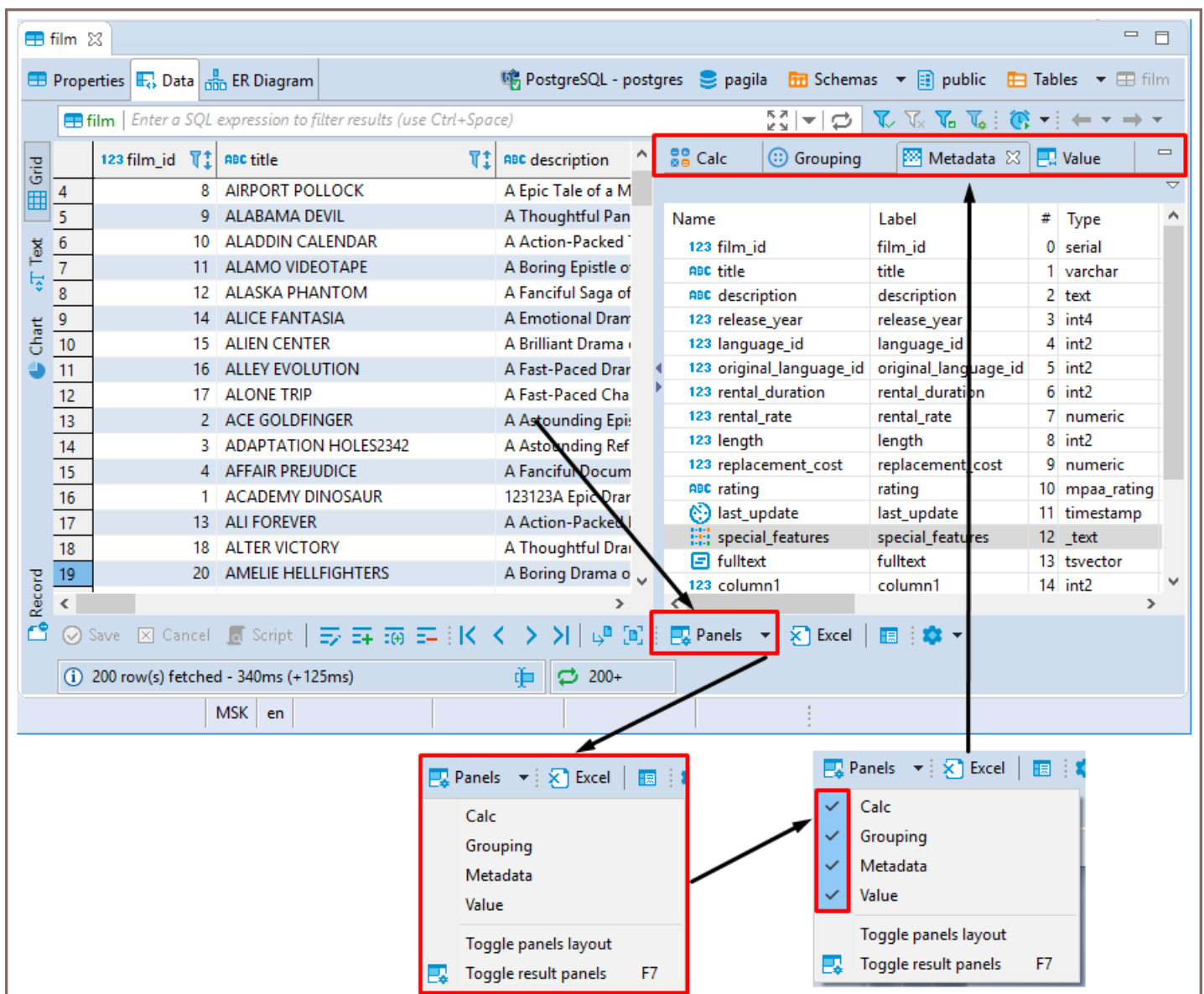
Panels provide additional space in the [Data editor](#) in which you can manipulate data. The panels are handy if you work with complex types (structures, arrays), long text data, or BLOBs. Panels appear as tabs in an additional pane in the right hand side of the Data tab:



This additional pane appears only when you open one of the four panels:

- Calc
- Grouping
- Metadata
- Value viewer (default)

To open the panels, click **Panels** on the bottom toolbar. By default, the Value viewer panel opens. Alternatively, you can open the Value panel by pressing **F7** on a cell. To open the other panels, click the down arrow next to the **Panels** button and click the name of the panel on the menu:

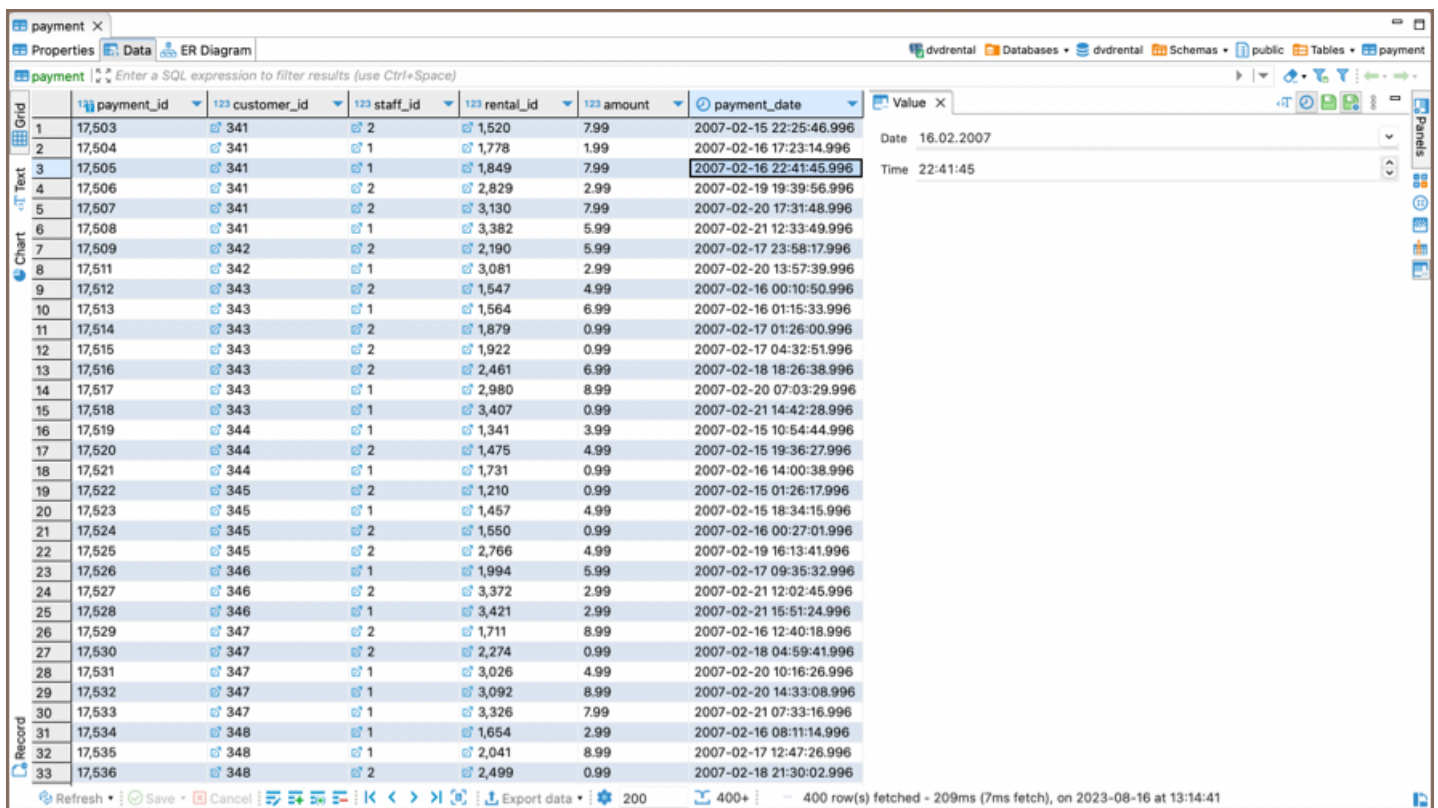


Panels will also open if you try to inline-edit a cell with a complex data type.



To close the panels, click the **Panels** button again or click the standard Close (cross) icon in the upper right corner of each panel.

You can also show and hide panels by clicking the **Configure** button (⚙️) on the bottom toolbar and then **Toggle result panels** on the Configure dropdown menu.

Value Viewer



The **Value Viewer** panel displays the value currently in focus and allows editing. The toolbar of the **Value Viewer** contains the following common buttons for all data types:

Button	Name	Description
	Apply cell changes	Displays the data table changes in the Value viewer.
	Auto-apply value	Enables the automatic display of changes made in the Value viewer in the data table. When auto-saving is enabled, the changes appear in the data table at the same time when they are made in the Value viewer.

Note: This buttons does not save changes made to the database. To save the changes in the database, you need to use the **Save** button on the bottom toolbar of the [Data Editor](#).

Additional buttons may appear depending on the specific data type being viewed (e.g., **Arrays** , **JSON** , **datetime** , Binary content).

Dictionary handling in Value Viewer

The **Value Viewer** provides specific handling for dictionaries when dealing with foreign keys. If the table contains foreign keys, the **Value Viewer** will display a dictionary with values. For data types that are not `string`, `numeric`, or `datetime`, only one value appear in the dictionary panel.

Working with dictionaries

When working with dictionaries in the **Value Viewer**, the following functionalities are available:

- **Edit Value:** Displays the current value, which can also be edited within the **Value Viewer**.
- **Dictionary** field: Displays the table referenced by the foreign key. Clicking on it will open a new tab with that table.
- **Define Description** field: Clicking on this field will open a window where you can:
 - Choose dictionary description columns or set custom criteria.
 - Define a custom expression for description.
 - Set a column delimiter.
- **Search:** Allows you to type dictionary value or part of description string to search within the dictionary entries.

Preferences Settings for dictionaries

You can further customize the behavior of dictionaries in the Value Viewer by accessing the preferences settings.

Navigate to **Window -> Preferences -> Editors -> Data Editor -> Dictionaries**.

Within these settings, you can configure the following options:

- **Keywords for dictionary description columns:** Allows you to specify keywords that will be used for dictionary description columns.
- **Maximum amount of elements on a dictionary panel page:** This enables you to set a limit for the number of elements displayed on a single page within the dictionary panel.

These preferences provide additional control over how dictionaries are handled and displayed within the **Value Viewer**.

Metadata Panel

The Metadata panel displays metadata for each cell in the row containing the cell currently in focus. You can just view the metadata.

Metadata

Name	Label	#	Type	Catalog Name	Schema Name	Table Name	Max Length	Precision	Scale	JDBC Type	Not Null	Auto
123 film_id	film_id	0	serial	pagila	public	film	11	10	0	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
ABC title	title	1	varchar	pagila	public	film	255	255	0	VARCHAR	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ABC description	description	2	text	pagila	public	film	2,147,483,647	2,147,483,647	0	VARCHAR	<input type="checkbox"/>	<input type="checkbox"/>
123 release_year	release_year	3	int4	pagila	public	film	11	10	0	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>
123 language_id	language_id	4	int2	pagila	public	film	6	5	0	SMALLINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>
123 original_language_id	original_language_id	5	int2	pagila	public	film	6	5	0	SMALLINT	<input type="checkbox"/>	<input type="checkbox"/>
123 rental_duration	rental_duration	6	int2	pagila	public	film	6	5	0	SMALLINT	<input checked="" type="checkbox"/>	<input type="checkbox"/>
123 rental_rate	rental_rate	7	numeric	pagila	public	film	6	4	2	NUMERIC	<input checked="" type="checkbox"/>	<input type="checkbox"/>
123 length	length	8	int2	pagila	public	film	6	5	0	SMALLINT	<input type="checkbox"/>	<input type="checkbox"/>
123 replacement_cost	replacement_cost	9	numeric	pagila	public	film	7	5	2	NUMERIC	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ABC rating	rating	10	mpaa_rating	pagila	public	film	2,147,483,647	2,147,483,647	0	OTHER	<input type="checkbox"/>	<input type="checkbox"/>
🕒 last_update	last_update	11	timestamp	pagila	public	film	29	29	6	TIMESTAMP	<input checked="" type="checkbox"/>	<input type="checkbox"/>
📄 special_features	special_features	12	_text	pagila	public	film	2,147,483,647	2,147,483,647	0	ARRAY	<input type="checkbox"/>	<input type="checkbox"/>
📄 fulltext	fulltext	13	tsvector	pagila	public	film	2,147,483,647	2,147,483,647	0	OTHER	<input checked="" type="checkbox"/>	<input type="checkbox"/>
123 column1	column1	14	int2	pagila	public	film	6	5	0	SMALLINT	<input type="checkbox"/>	<input type="checkbox"/>
✅ column2	column2	15	bool	pagila	public	film	1	1	0	BOOLEAN	<input type="checkbox"/>	<input type="checkbox"/>
🕒 time_name	time_name	16	time	pagila	public	film	15	15	6	TIME	<input type="checkbox"/>	<input type="checkbox"/>
🕒 date_name	date_name	17	date	pagila	public	film	13	13	0	DATE	<input type="checkbox"/>	<input type="checkbox"/>
🕒 dateTime_name	dateTime_name	18	timestamp	pagila	public	film	29	29	6	TIMESTAMP	<input type="checkbox"/>	<input type="checkbox"/>


Calc Panel


The Calc panel is useful for getting basic statistics across data in several columns and rows:

The screenshot shows the DBeaver interface with a table of film data. The table has columns: length, replacement_cost, rating, and last_update. The aggregate panel on the right shows the following statistics:

Function	Value
Count	200
Count Distinct	21
Maximum	29.99
Average	20.305



You can select several columns and rows in standard ways - by pressing and holding the left mouse button or by clicking cells while holding the `Ctrl` or `Shift` keys. The panel updates dynamically to show statistics for the selected data.

To see the data grouped by columns, click the Group by columns button () . To remove the grouping by columns and see the summary values for all columns, click the same button again.

By default, the panel applies and displays results for two functions – **Count** and **Count Distinct**. To add other functions, click the **Add function** () button on the toolbar of the panel or right-click one of the rows in the Aggregate panel and click **Add function** on the context menu and then click the name of the function. The following functions are available:

- Sum
- Average
- Minimum
- Maximum
- Median

● Mode

To remove an individual function, click the function and then click **Remove function** () on the toolbar of the panel, or right-click the function and click **Remove function** on the context menu. To remove all functions, click **Reset** () on the toolbar or on the context menu.

You can copy the value of a particular function to the clipboard - right-click the row and click **Copy Value** on the context menu.

You can also copy all functions with their values - right-click in the table and click **Copy All** on the context menu.

Grouping Panel

The Grouping panel provides tools to calculate statistics based on a table of a custom SQL query. It uses **GROUP BY** queries to extract unique values for **COUNT** (default), **SUM**, **AVG**, **MIN**, **MAX**, and other analytics functions, displaying the results in dedicated columns.

To obtain the grouping results for one or more columns of a data table, open the **Grouping panel**, then, in the results table, put the cursor onto the data type icon of the table header, and drag-n-drop the column(s) into the panel:

Album X

Properties

Data

ER Diagram

DBEaver Sample Database (SQLite) Tables Album

Album

Enter a SQL expression to filter results (use Ctrl+Space)

Grid

Text

Chart

Record

	AlbumId	Title	ArtistId
1	1	For Those About To Rock We Salute You	1
2	2	Balls to the Wall	2
3	3	Restless and Wild	2
4	4	Let There Be Rock	1
5	5	Big Ones	3
6	6	Jagged Little Pill	4
7	7	Facelift	5
8	8	Warner 25 Anos	6
9	9	Plays Metallica By Four Cellos	7
10	10	Audioslave	8
11	11	Out Of Exile	8
12	12	BackBeat Soundtrack	9
13	13	The Best Of Billy Cobham	10
14	14	Alcohol Fueled Brewtality Live! [Disc 1]	11
15	15	Alcohol Fueled Brewtality Live! [Disc 2]	11
16	16	Black Sabbath	12
17	17	Black Sabbath Vol. 4 (Remaster)	12
18	18	Body Count	13
19	19	Chemical Wedding	14
20	20	The Best Of Buddy Guy - The Millenium C	15
21	21	Prenda Minha	16
22	22	Sozinho Remix Ao Vivo	16
23	23	Minha Historia	17
24	24	Afrociberdelia	18
25	25	Da Lama Ao Caos	18
26	26	Acústico MTV [Live]	19
27	27	Cidade Negra - Hits	19
28	28	Na Pista	20
29	29	Axé Bahia 2001	21
30	30	BBC Sessions [Disc 1] [Live]	22
31	31	Bongo Fury	23
32	32	Carnaval 2001	21
33	33	Chill: Brazil (Disc 1)	24
34	34	Chill: Brazil (Disc 2)	6
35	35	Garage Inc. (Disc 1)	50

Grouping X

Enter a SQL expression to

123 ArtistId

No Groupings

Drag-and-drop results column(s) here to create grouping

Press CONTROL to configure grouping settings

Refresh

Save

Cancel

Export data

200

200+

If you add several columns to the panel, DBEaver groups data in the order the columns go and calculates statistics based on the grouping.

Grouping X				
BillingCountry, BillingPostalCode Enter a SQL expression to filter results (use Ctrl. ▶)				
	RBC BillingCountry	RBC BillingPostalCode	123 count_	
1	Argentina	1106	7	
2	Australia	2010	7	
3	Austria	1010	7	
4	Belgium	1000	7	
5	Brazil	01007-010	7	
6	Brazil	01310-200	7	
7	Brazil	12227-000	7	
8	Brazil	20040-020	7	
9	Brazil	71020-677	7	
10	Canada	B3S 1C5	7	
11	Canada	H2G 1A7	7	
12	Canada	K2P 1L7	7	
13	Canada	M6J 1V1	7	
14	Canada	R3L 2B9	7	

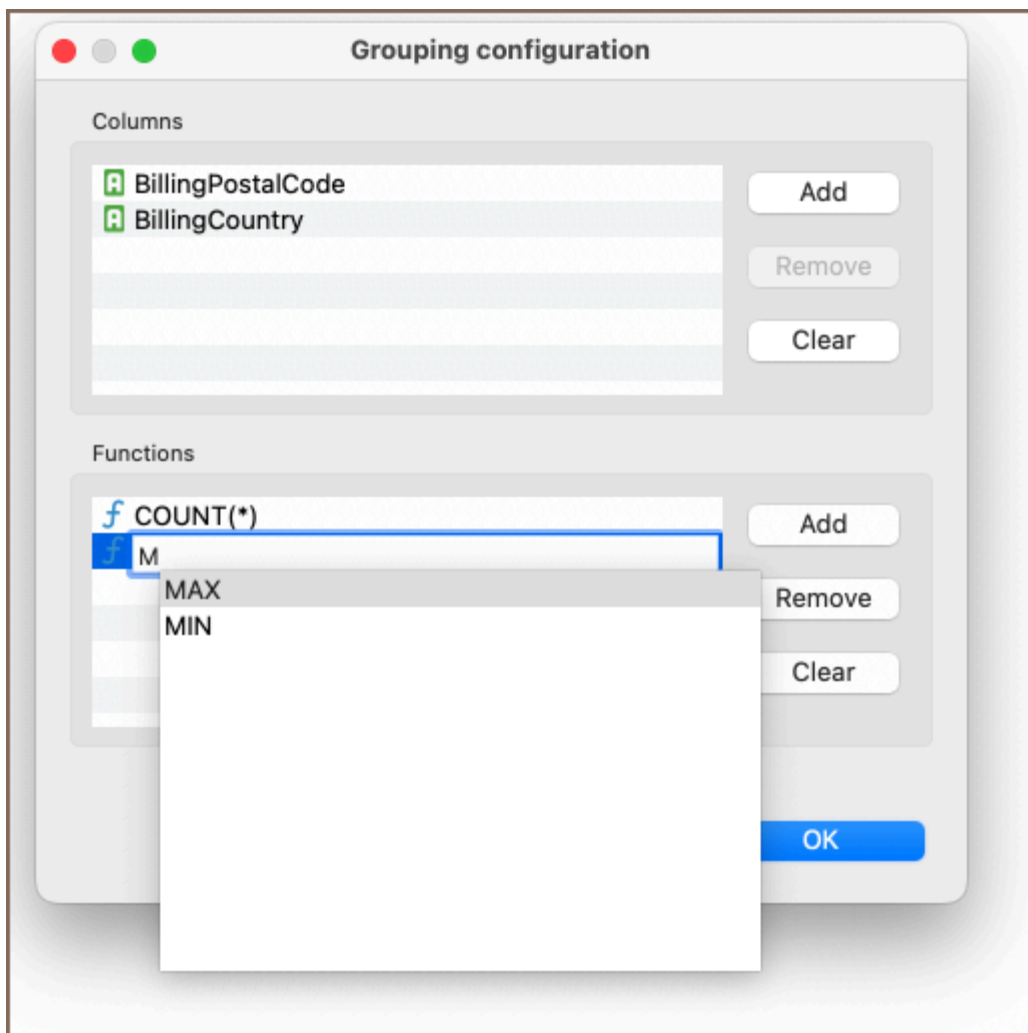
If you want to see the entire query, click on the **Query** button located in the top-left corner of the **Grouping** panel to open it.

Grouping X				
BillingCountry, BillingPostalCode Enter a SQL expression to filter results (use Ctrl. ▶)				
<pre>SELECT BillingCountry, BillingPostalCode, COUNT(*) AS count_ FROM Invoice x GROUP BY BillingCountry, BillingPostalCode</pre>				
1				
2				
3				
4	Belgium	1000	7	
5	Brazil	01007-010	7	
6	Brazil	01310-200	7	
7	Brazil	12227-000	7	
8	Brazil	20040-020	7	
9	Brazil	71020-677	7	
10	Canada	B3S 1C5	7	
11	Canada	H2G 1A7	7	
12	Canada	K2P 1L7	7	
13	Canada	M6J 1V1	7	
14	Canada	R3L 2B9	7	

Function management in the Grouping panel

Adding a function

By default, the **COUNT** function is used. You can add other functions as well.



To add a function, follow these steps:

1. Click the **Edit grouping columns** button on the panel's toolbar. Alternatively, right-click the **Grouping panel** window and navigate to the **Edit grouping columns** option.
2. In the **Grouping Configuration** window, locate the **Functions** area. Click **Add** and type the function into the new row.
 - DBeaver provides auto-complete options for your convenience.
 - Indicate the column name in brackets. Note that **COUNT** is the only function that allows using ***** instead of specifying a column name.
3. To complete the process, click **OK**.

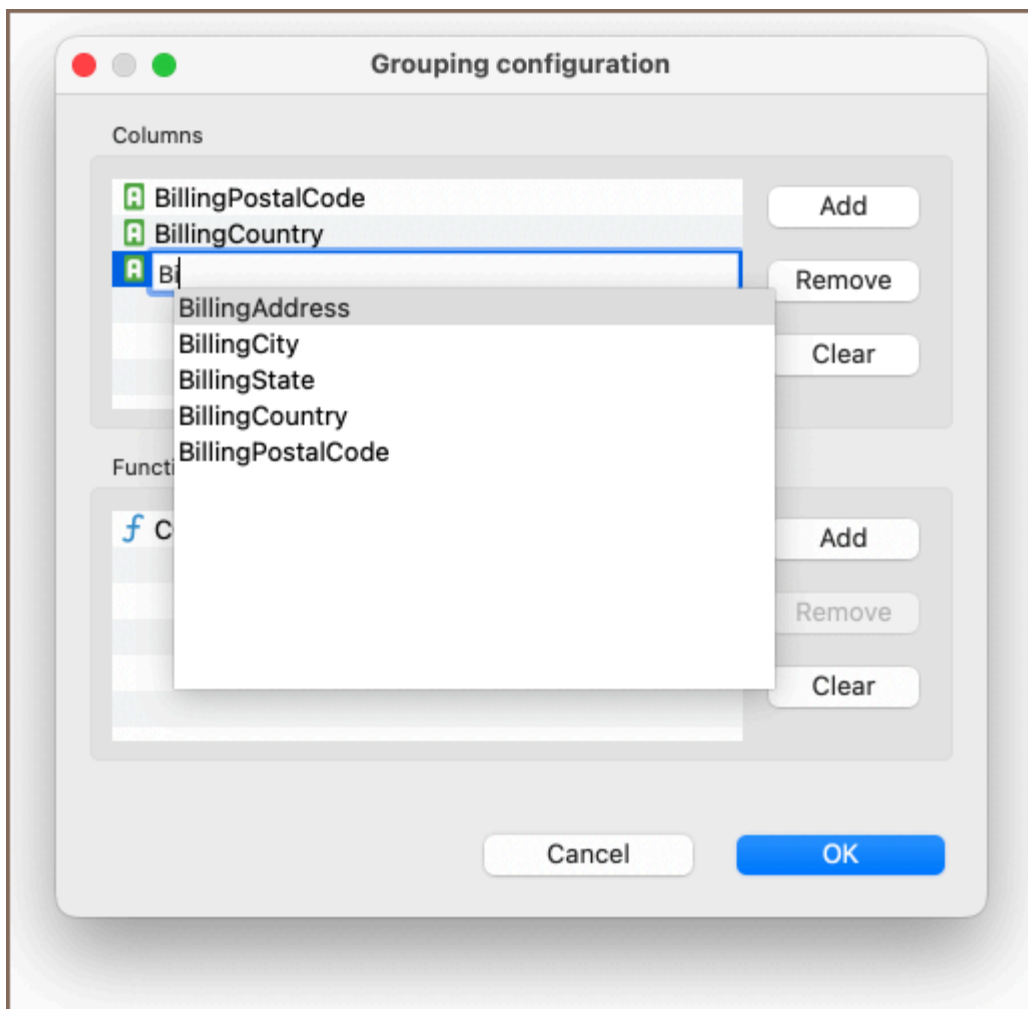
Removing a function

To remove a function, in the same **Grouping Configuration** window, select the function and click **Remove**, then click **OK**. To remove all functions, click **Clear** and then **OK**.

Column management in the Grouping panel

Adding a column

You can also manage columns in the same **Grouping Configuration** window.



To add a column:

1. Click the **Edit grouping columns** button on the panel's toolbar.
2. In the **Grouping Configuration** window, go to the **Columns** area. Click **Add** and type the column name into the new row.
 - DBeaver provides auto-complete options for your convenience.


3. To complete the process, click **OK**.

Tip: You can add a column with an expression for MySQL/MariaDB databases. The expression will be calculated in the resulting column:

The screenshot shows the 'Grouping configuration' dialog on the left and a data grid on the right. The dialog has two sections: 'Columns' and 'Functions'. In the 'Columns' section, 'rating', 'length', and 'length+20' are listed. In the 'Functions' section, 'COUNT(*)' is listed. The data grid on the right shows columns: 'rating', 'length', 'length+20', and 'count_'. The 'length' and 'length+20' columns are highlighted with red boxes, and a red arrow points from 'length' to 'length+20'.

	rating	length	length+20	count_
1	G	[NULL]	[NULL]	2
2	G	47	67	2
3	G	48	68	2
4	G	49	69	1
5	G	50	70	1
6	G	51	71	1
7	G	52	72	3
8	G	53	73	2
9	G	54	74	1
10	G	56	76	2
11	G	57	77	3
12	G	58	78	1
13	G	59	79	2
14	G	60	80	1
15	G	61	81	3
16	G	62	82	3
17	G	63	83	3


Removing a column

- To remove a column, in the same **Grouping Configuration** window, select the column and click **Remove**, then click **OK**. To remove all columns, click **Clear** and then **OK**.
- Alternatively, you can remove a column by clicking the column name and then clicking the **Remove grouping column** button  in the panel's toolbar.

To clear all results from the **Grouping panel**, click the **Clear grouping** button .

Additional settings

Default sorting


Click the **Default Sorting** button  to access sorting options for the grouped data. The dropdown menu provides the following sorting criteria:

Option	Description
Unsorted	Leaves the data in its original order.
Ascending	Sorts the grouped data in ascending order.

Descending	Sorts the grouped data in descending order.
------------	---

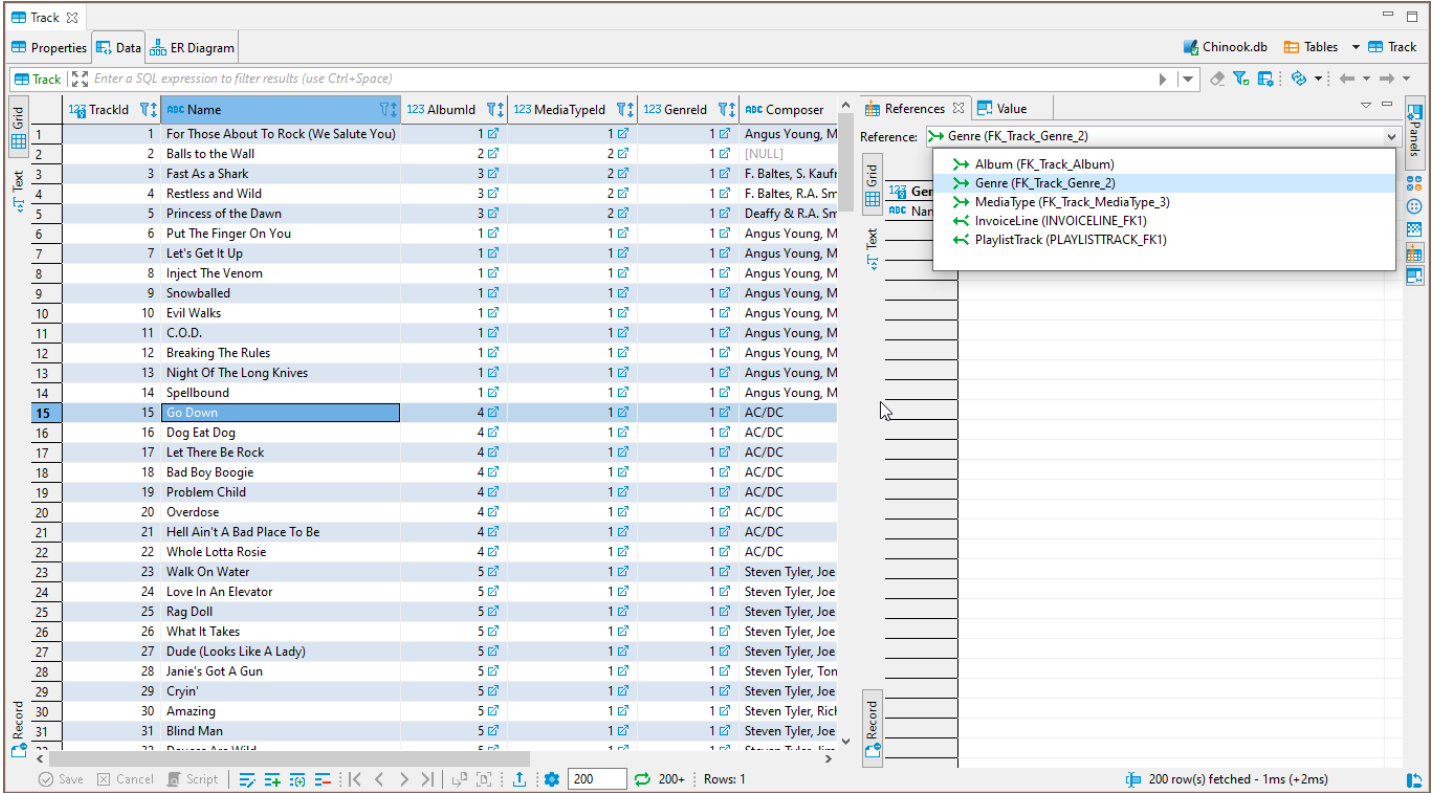
Select the desired option to apply the corresponding sorting criteria to the grouped data.

Show duplicates only

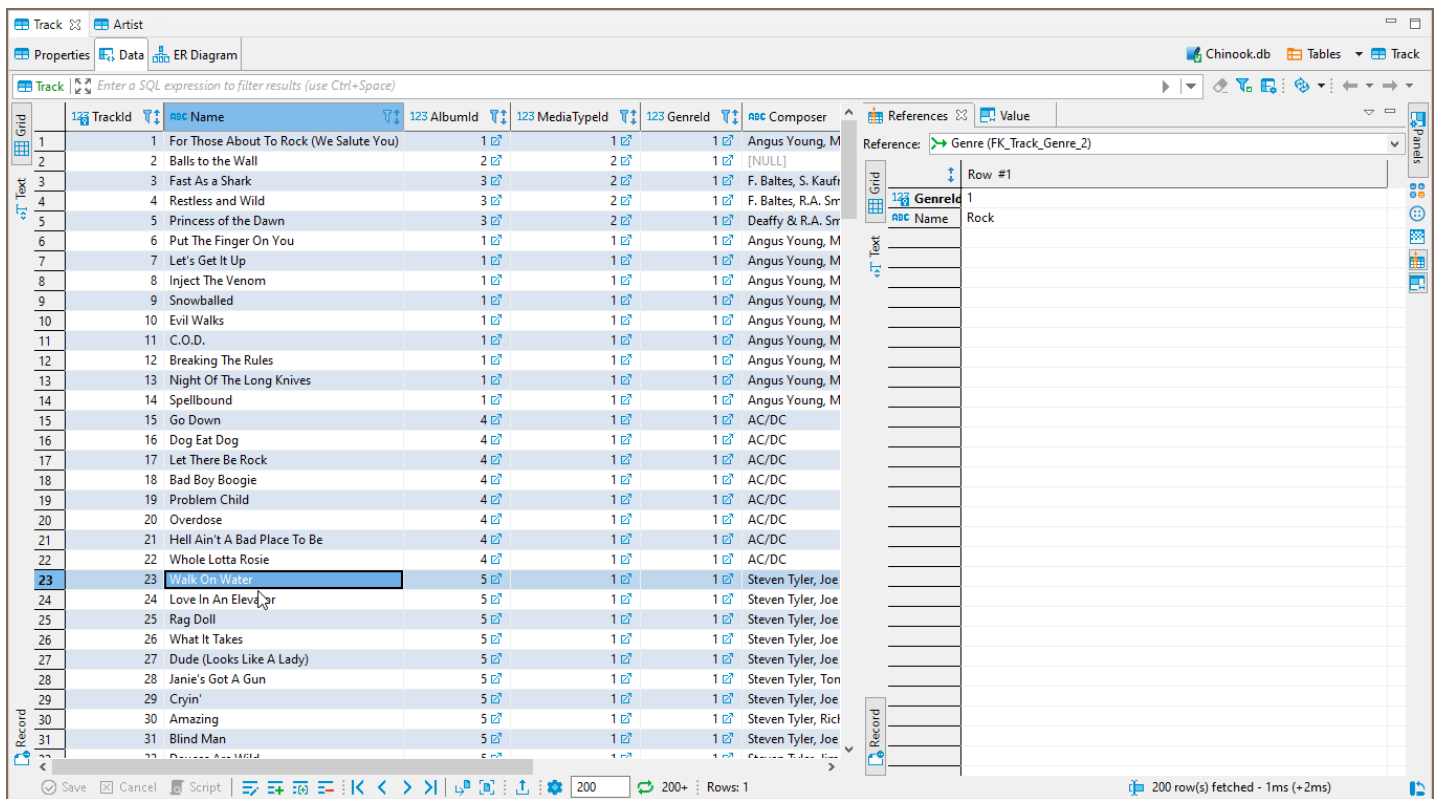
Click the **Show duplicates only (COUNT > 1)** button  to filter the results and display only those rows where the **COUNT** is greater than 1.

References panel

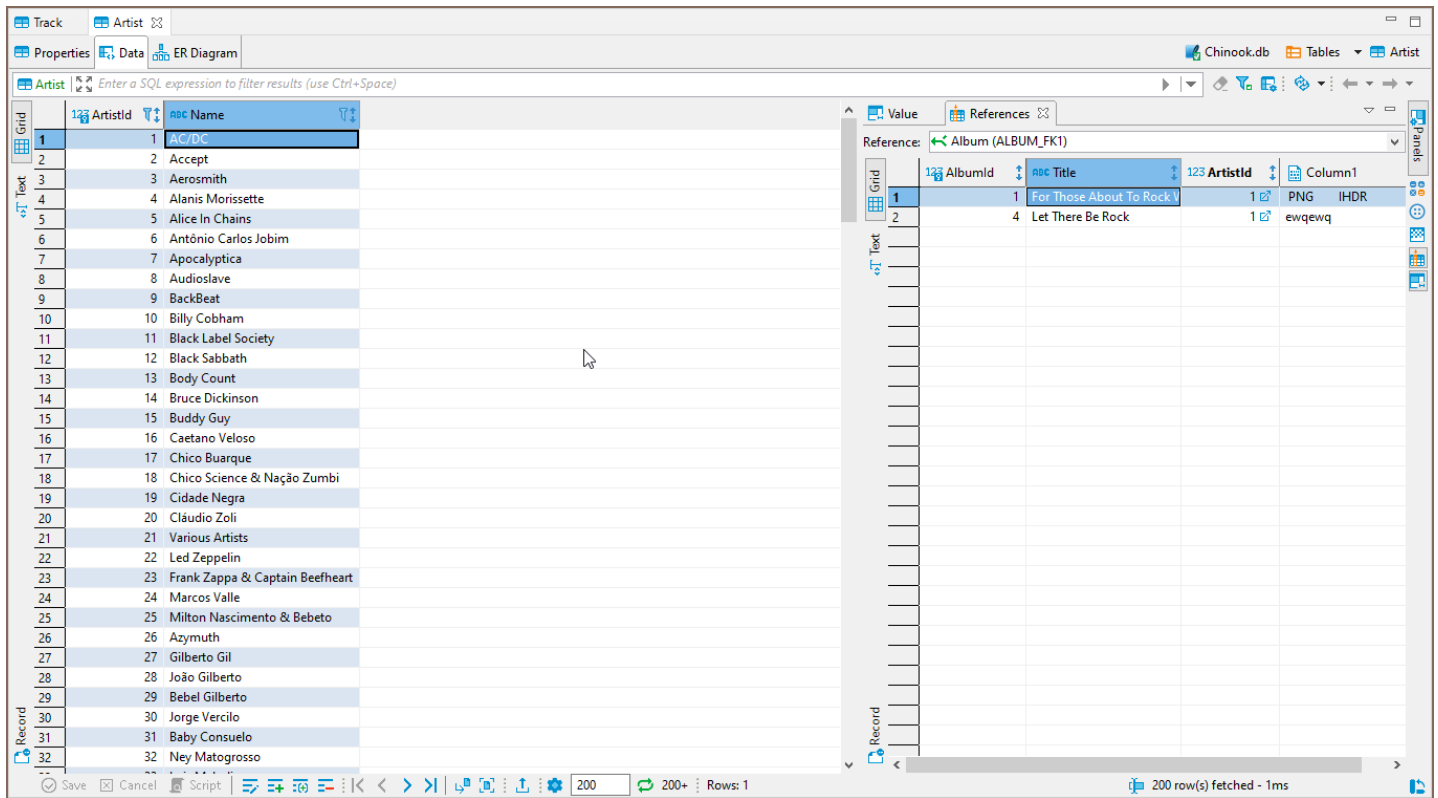
The references panel allows you to see all the related information for the chosen row from other connected tables. The information is presented in an additional data viewer window, filtered to show the information related to the currently selected row. If a table opened in data viewer has a foreign key referencing another table, or it is referenced with a foreign key by another table, all of those connected tables can be picked from a dropdown list.



When a table that is referenced by a foreign key in the current table is chosen, the information from the row corresponding to a referenced key will be shown, in this situation the record mode is enabled by default, but it can be turned off like in a normal data viewer.



When a table that references the current table is chosen, the references panel will show all the rows that refer a selected primary key in the current table.



Result details panel

The **Result details** panel provides you with information about the execution of a **SELECT** query.

Results 1 X

SELECT first_name, last_name, last_u | Enter a SQL expression to filter results (use Ctrl+Space)

Grid

Text

Chart

Record

	ABC first_name	ABC last_name	last_update	ABC column1
1	Leonard	Joanne	[NULL]	
2	Claire	Oscar	[NULL]	
3	Drew	Amanda	[NULL]	
4	Vivian	Loretta	[NULL]	
5	Brandy	Emma	[NULL]	
6	Cliff	Sibyl	[NULL]	
7	Denise	Jesse	[NULL]	
8	Neil	Tallulah	[NULL]	
9	Eliza	Elliot	[NULL]	
10	Samuel	Winnie	[NULL]	
11	Phillipps	Salma	[NULL]	
12	Caleb	Patience	[NULL]	
13	Chas	Myra	[NULL]	
14	Richie	Viola	[NULL]	
15	Lester	Dirk	[NULL]	

Refresh

Save

Cancel

Export data

200

200+

Result details X

Name	Value
Job	job_dS3ldtkZW-D58m\
Location	europa-west1
Project	dbeavergcp
Fetchd from cache	false
Total rows	200
Total bytes processed	1,952
DML affected rows	none

Note: This panel is specific to [BigQuery](#) and provides details that help in monitoring query costs.

Result details panel includes the following metrics:

Name	Description
Job	The unique identifier of the job.
Location	The regional location of the data you have queried.
Project	The project within which the query was executed.
Fetchd from cache	Indicates if the results were retrieved from the cache. true means no additional costs are incurred for data processing.
Total rows	The number of rows returned by the query.
Total bytes processed	The amount of data processed by the query. This is important for cost calculations as costs are incurred based on the amount of data processed. If the Total bytes processed is zero, it indicates that the query results were served from the cache, which does not incur costs.
DML affected rows	The number of rows affected by a Data Manipulation Language (DML) statement.

Query Trace panel

Query trace panel allows you to monitor the progression of reads and writes across the database, providing a detailed breakdown of how a query is executed internally.

Note: This feature is specific to [Cassandra](#) and Yugabyte CQL.

Name	Description
Event	Describes the specific action or stage of the query being logged.
Time	The timestamp or duration (in milliseconds) indicating when the event occurred or its length.
Source	The IP address or identifier of the node where the event originated.
Thread	The thread or process within the source that executed the event.

Managing Data formats

Table of contents

[Data Format Profiles](#)

[Changing Data Formats](#)

[Locale](#)

[Native Date and Time Mode](#)

[Data Type Format](#)

[Date](#)

[Time](#)

[Timestamp](#)

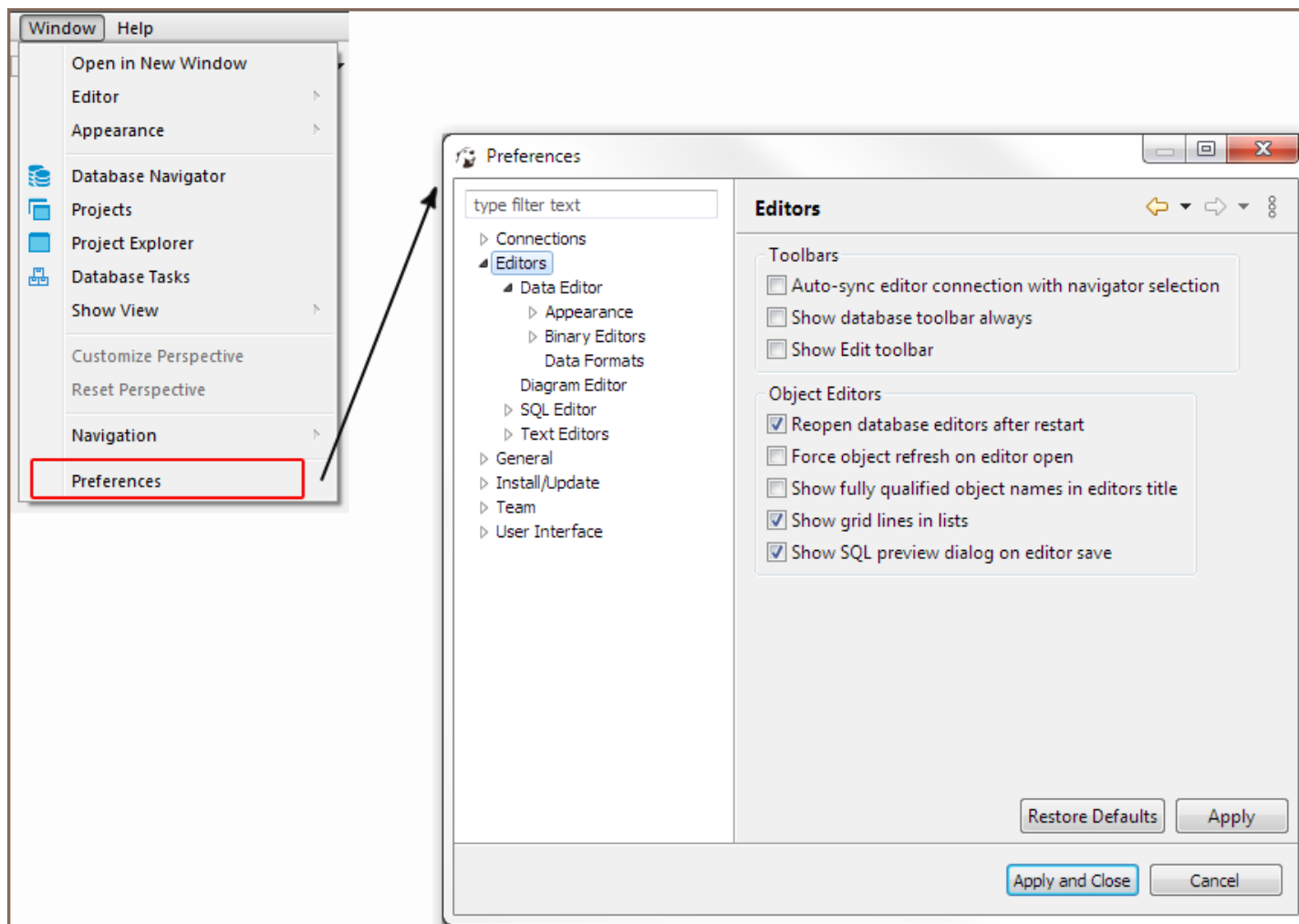
[Numbers](#)

[Restoring Default Data Formats](#)

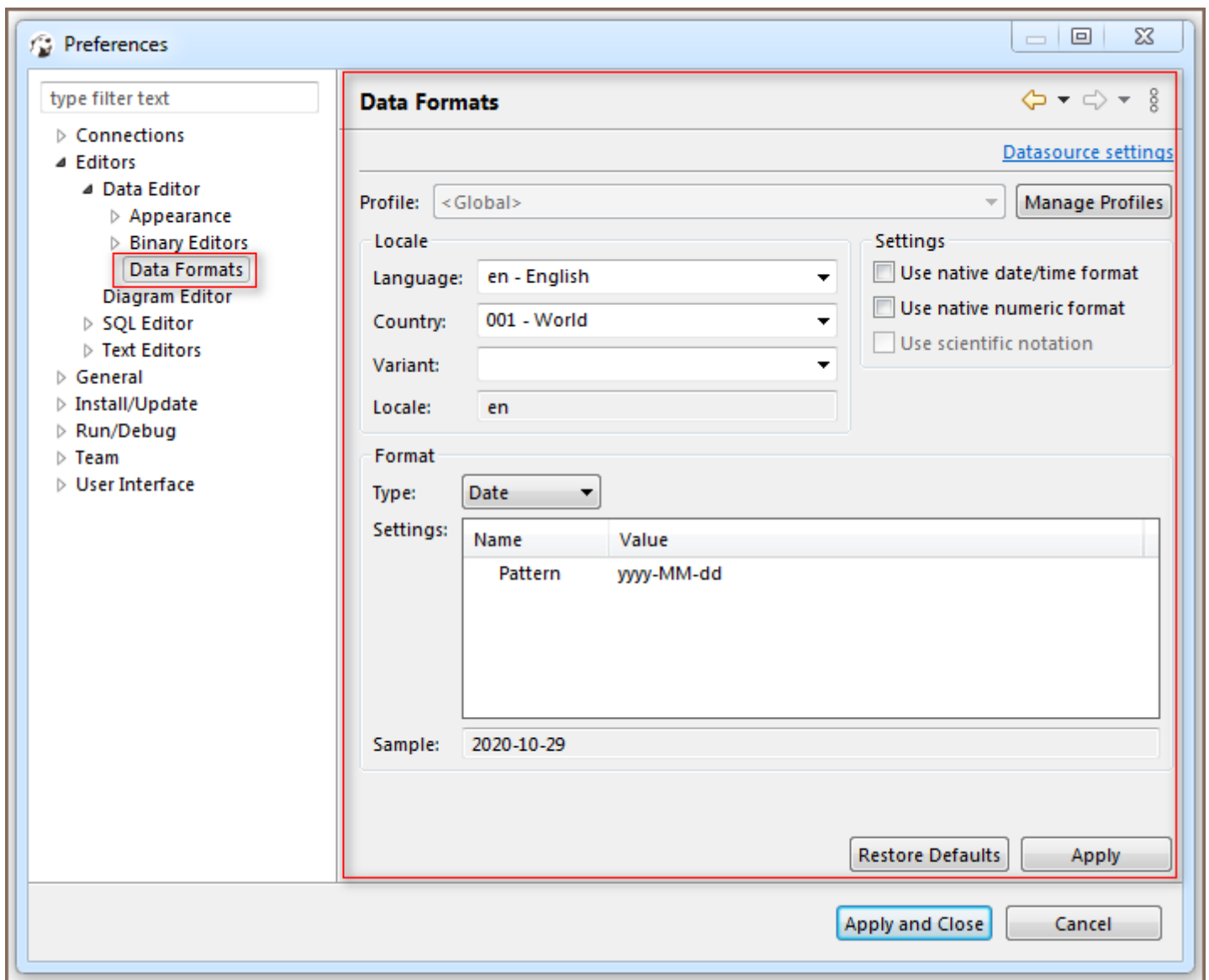
[Datasource Settings](#)

The DBeaver formatting functions allow you to set up database locale and change datasource format settings. This feature can be very useful, for example, for database migration.

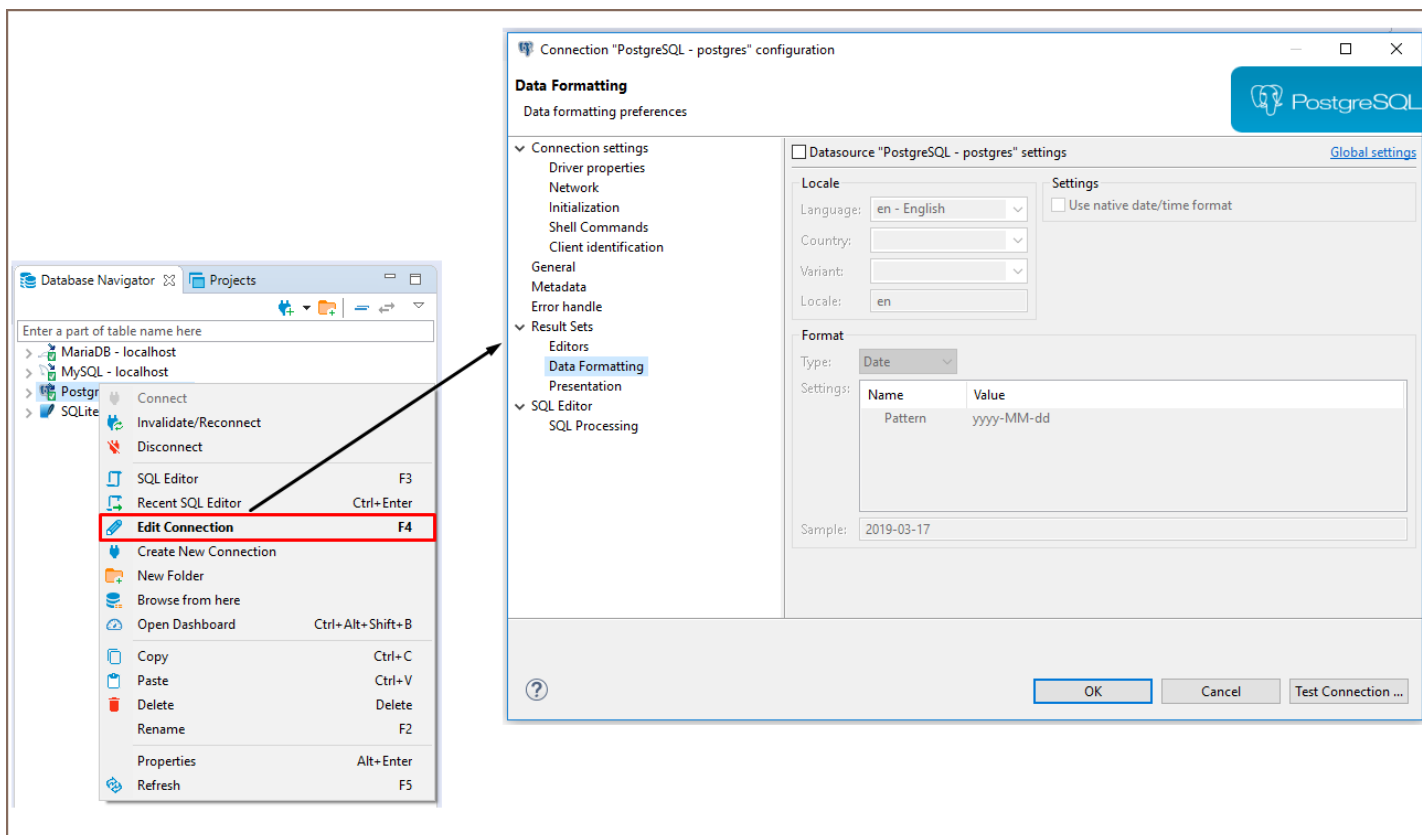
To change the data format settings use the option Window -> Preferences in main menu.



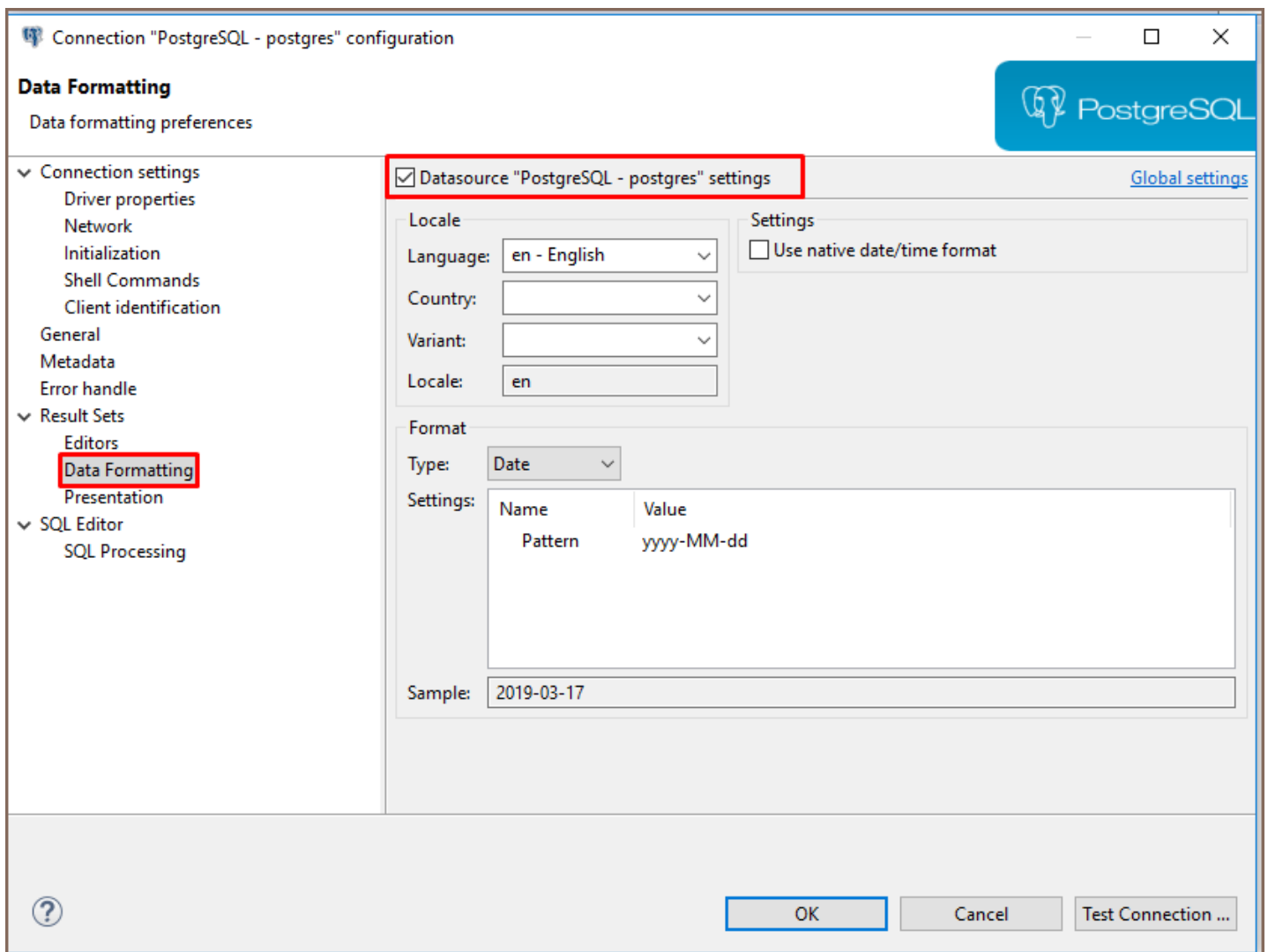
In the Preferences dialog box go to Editors -> Data Editor -> Data Formats.



Or, in the [Database Navigator](#) right-click a connection and select **Edit Connection** menu option.



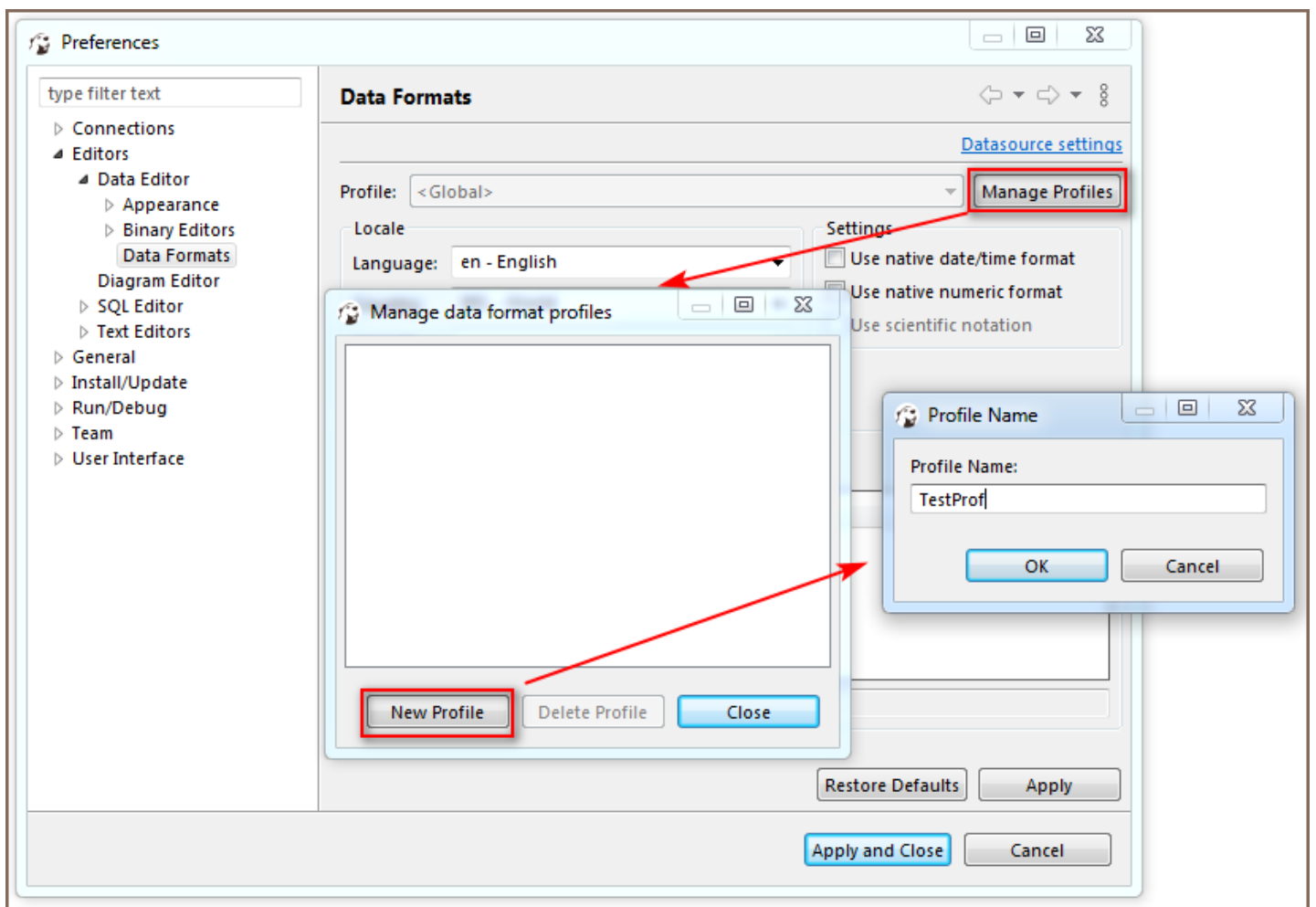
In the right area of the opened **Data formatting preferences** dialog window go to Data editor -> Data Formats and select the **Datasource settings** check box in the left area to customize the data format settings.



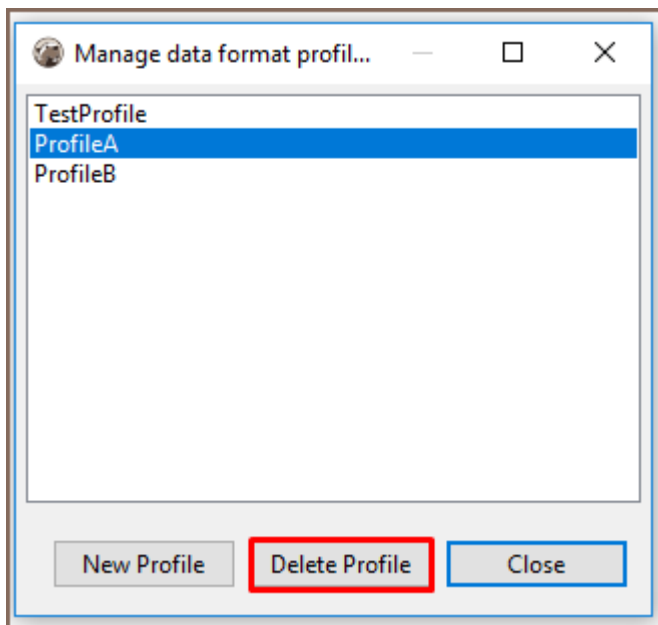
Data Format Profiles

Data format profiles allow you to apply a set of data format preferences to the whole current project by one click.

To create a data format profile press the **Manage Profiles** button. In the opened dialog window press button **New Profile**, define the name and press **Create**.



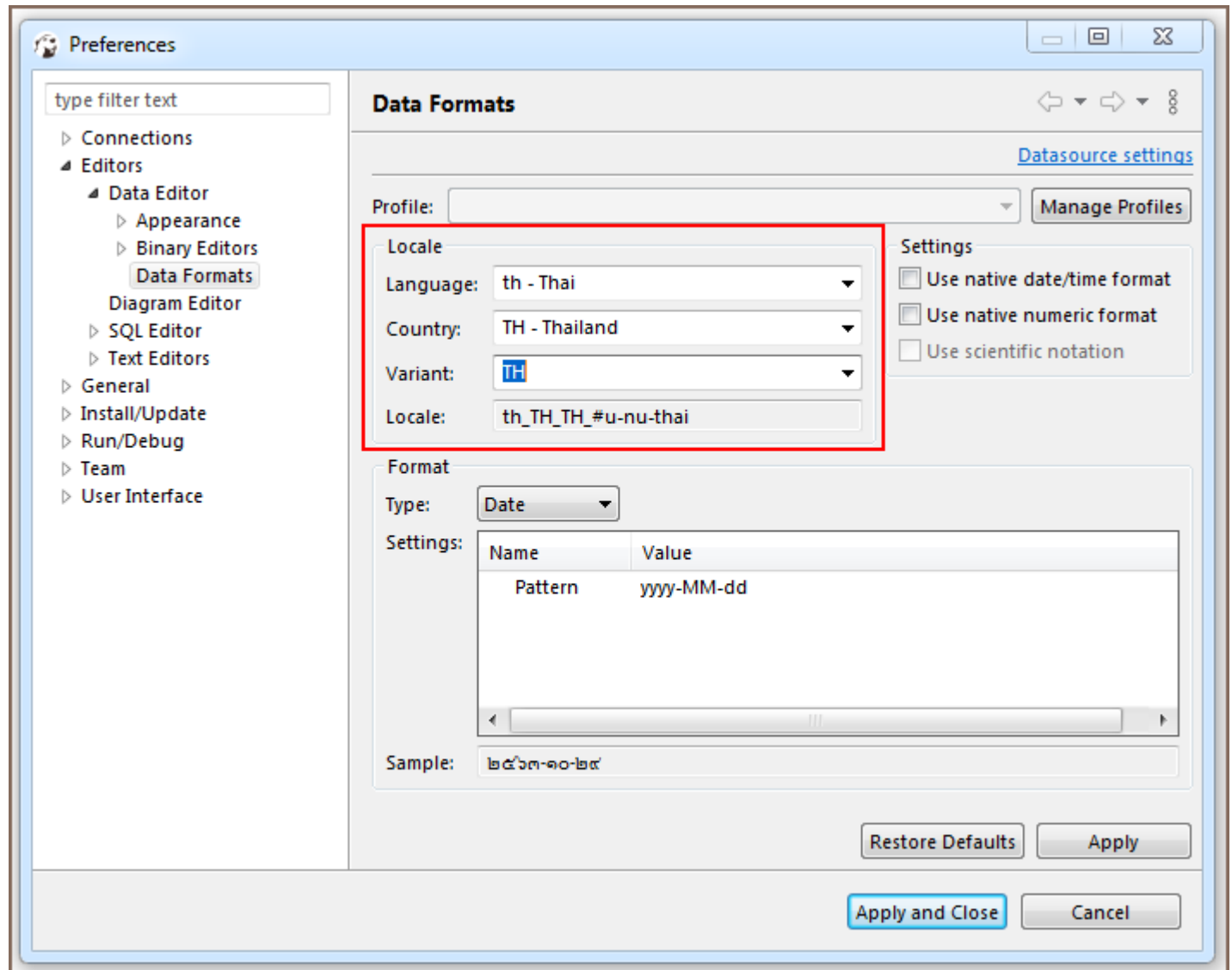
To delete a data format profile press the **Manage Profiles** button, then in the opened dialog window select the profile you want to delete and press the button **Delete Profile**.



Changing Data Formats

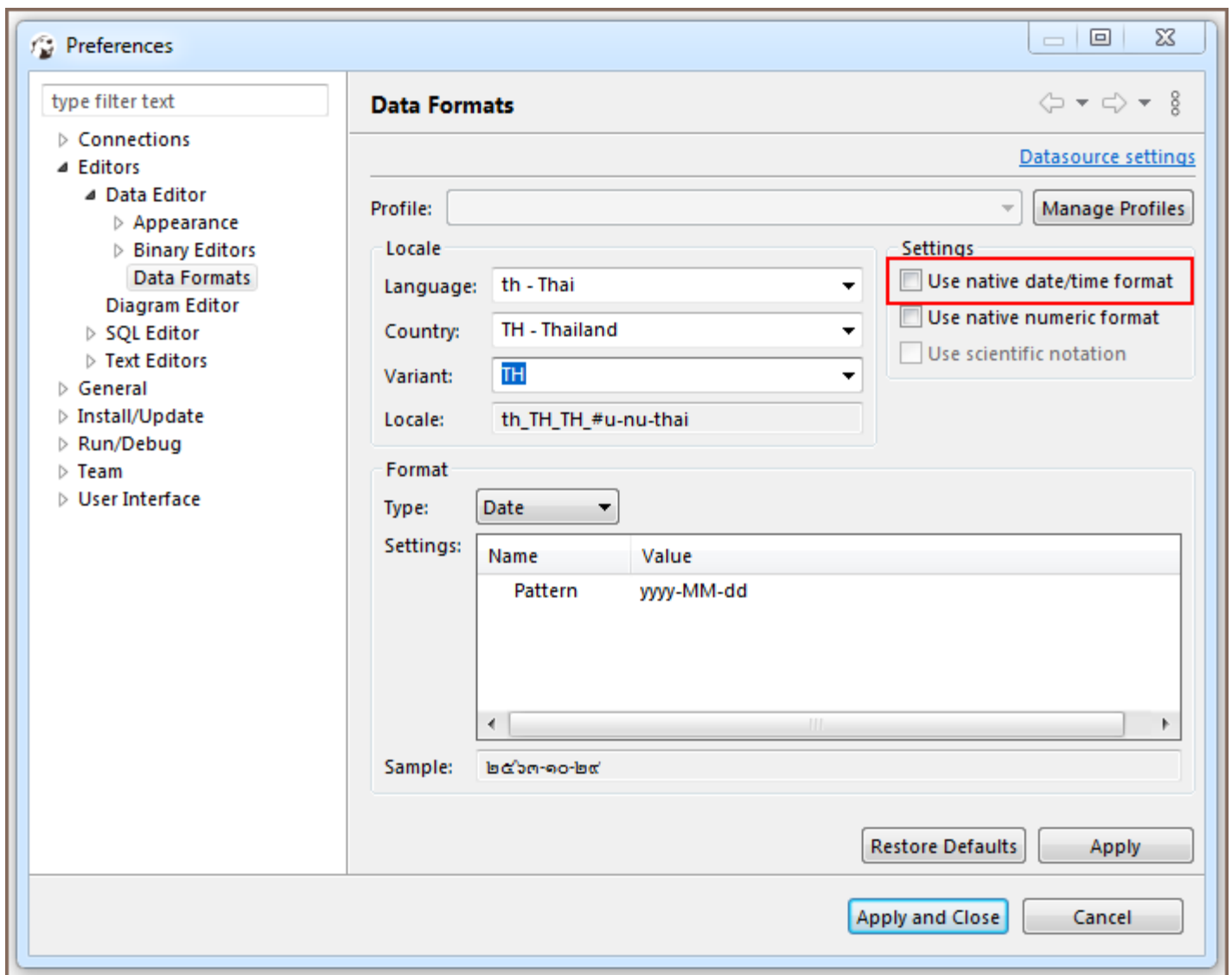
The following groups of data format settings can be adjusted:

Locale



To define this setting select a language, country and variant if available.

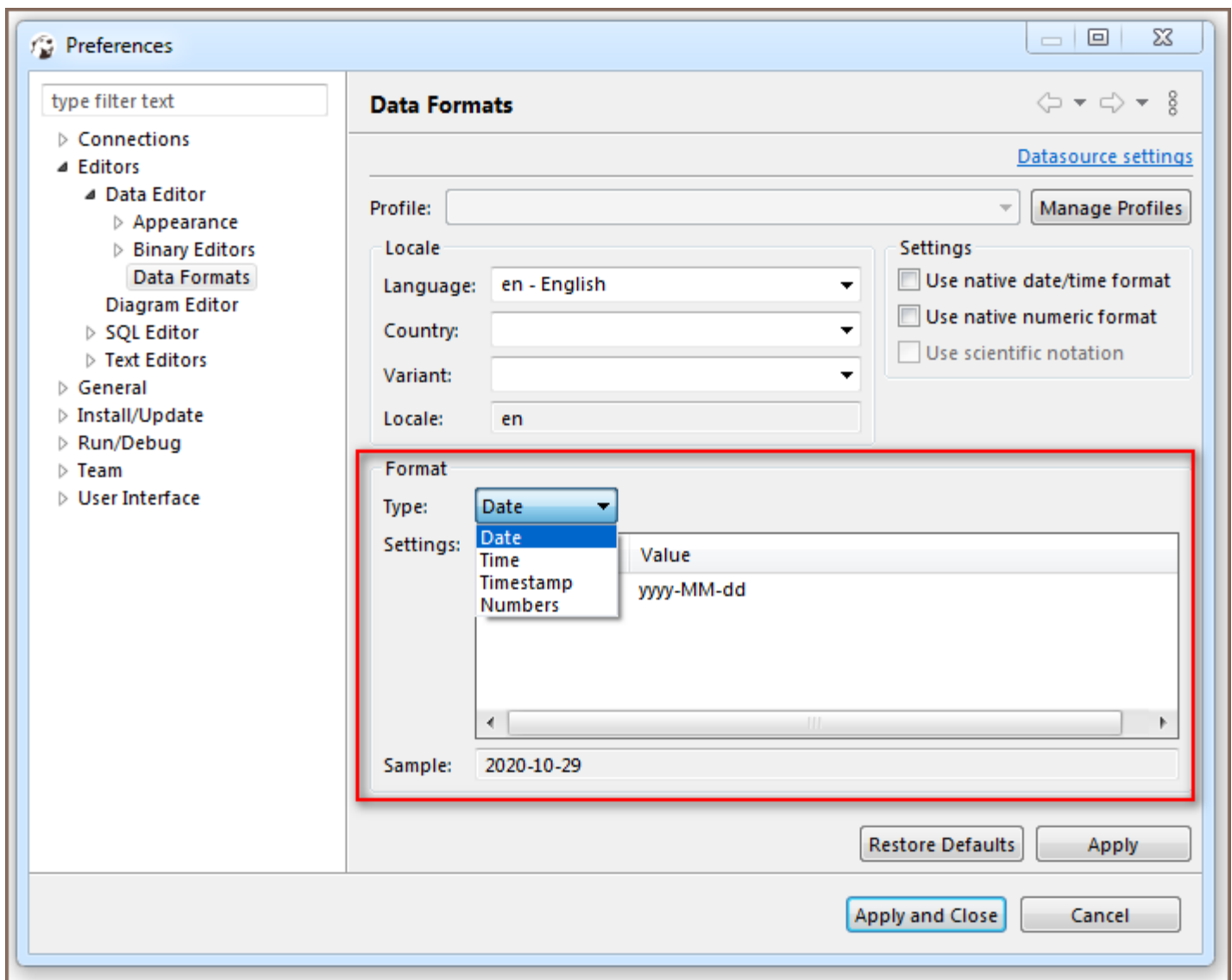
Native Date and Time Mode



Select **Use native date/time format** check-box and the data format originally built-in to the datasource will be used.

You can change the format of the following data types:

Data Type Format



The format of the following data types can be customized:

- Date
- Time
- Timestamp
- Numbers

Date

The default value for this data type is *yyyy-MM-dd*.

Pattern	Description
yyyy or y	Year of era (4 digits)

yy	Year of era (2 last digits)
YYYY	Week year
M	Month in year without leading zeros
MM	Month in year
MMM	Short month name in year
MMMM	Month name in year
D	Day in year
d	Day in month without leading zeros
dd	Day in month
E	Day name in week
G	Era designator
z	General time zone
Z	RFC 822 time zone

Time

The default value for this data type is *HH:mm:ss*

Pattern	Description
H	Hour in day (0-23) without leading zeros
h	Hour in day in am/pm (1-12) without leading zeros
HH	Hour in day (0-23)
hh	Hour in day in am/pm (1-12)
a	Am/pm marker
m	Minute in hour without leading zeros
mm	Minute in hour
s	Second in minute without leading zeros

ss	Second in minute
S	Millisecond
ffffff	Microseconds

Timestamp

The default value for this data type is *yyyy-MM-dd HH:mm:ss*

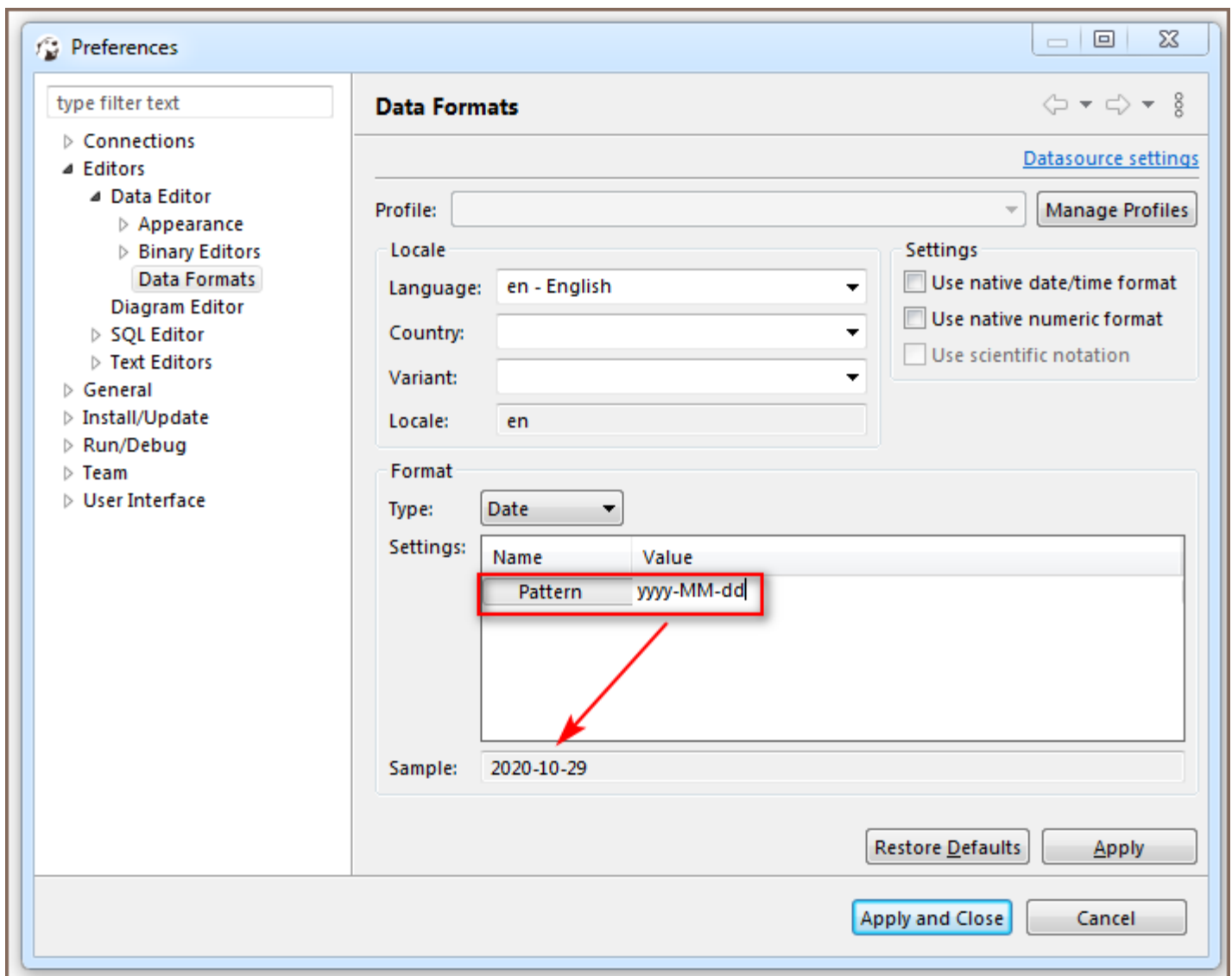
[Link on Java date pattern documentation](#)

Numbers

The following parameters can be configured from this type of data:

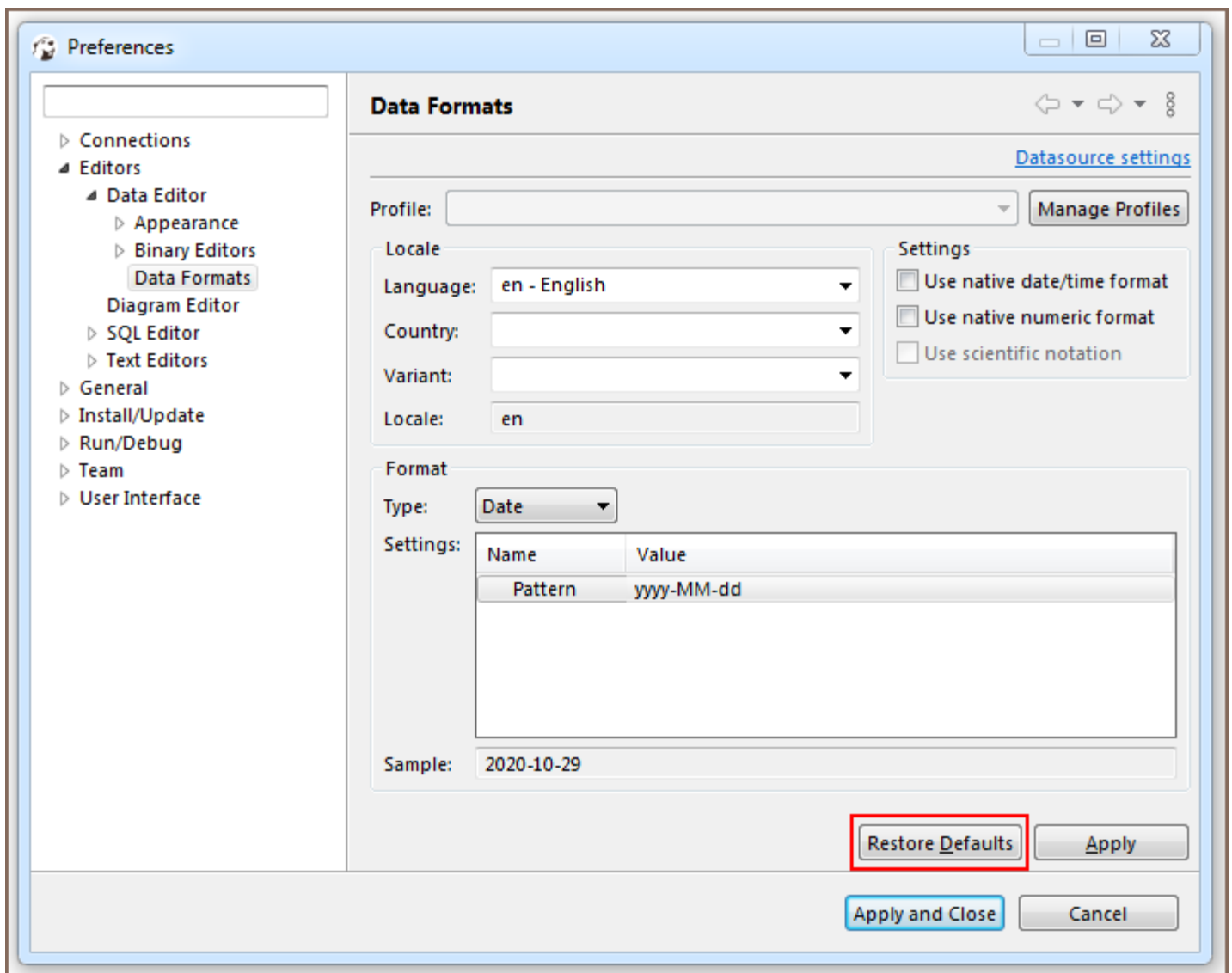
- **Use Grouping** - Long numbers can be hard to read if they have too many digits. For example, the factorial of 30 is 33 digits long! Select this check-box to enable Grouping mode, in which digits are displayed in clumps of 3 or 4 (depending on the current radix) separated by commas.
- **Maximum integer digits** - Defines the maximum number of digits to the left of the decimal point.
- **Minimum integer digits** - Defines the minimum number of digits to the left of the decimal point.
- **Maximum fraction digits** - Defines the maximum number of digits to the right of the decimal point.
- **Minimum fraction digits** - Defines the minimum number of digits to the right of the decimal point.
- **Use data type scale for fraction digits** - Some numeric columns or parameters may have a predefined scale, that is the maximum number of digits to the right of the decimal point. Select this check-box if you want the predefined precision to be used.
- **Rounding mode** - Specifies a rounding behavior for numerical operations capable of discarding precision. Each rounding mode indicates how the least significant returned digit of a rounded result is to be calculated. To learn more, please refer to [Oracle documentation](#).

To change the data type format, change the value displayed in the **Pattern** area, save the changes made by pressing the **Apply** button and observe the expected result in the **Sample** field.



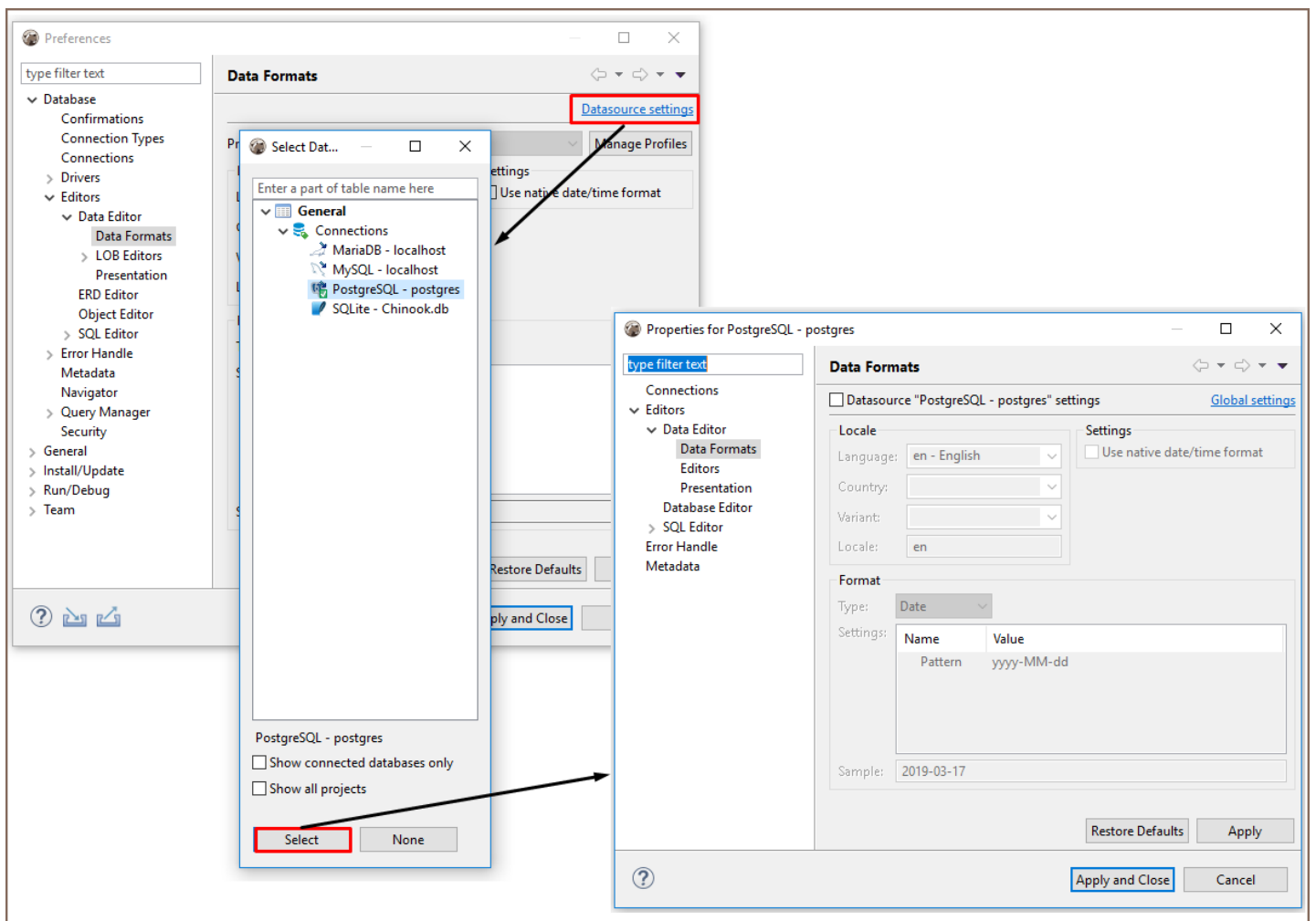
Restoring Default Data Formats

To restore the default data format settings, press the **Restore Defaults** button.



Datasource Settings

Press **Datasource settings** link to change data format settings for a particular datasource, then adjust the settings in the opened dialog box.



To save changes made press **Apply**.

Managing Charts

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Creating Charts In SQL Editor](#)

[Creating Charts In Data Editor](#)

[Creating Charts In Grouping Panel](#)

[Editing Chart Settings](#)

[Setting Axis X](#)

[Setting Axis Y](#)

[Setting Other Options](#)

[Copying to clipboard](#)

[Exporting Charts](#)

[Printing Charts](#)

[Zooming and Viewing Options](#)

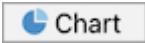
The default grid view of query results may look unremarkable. Use DBeaver's Charts feature to transform your

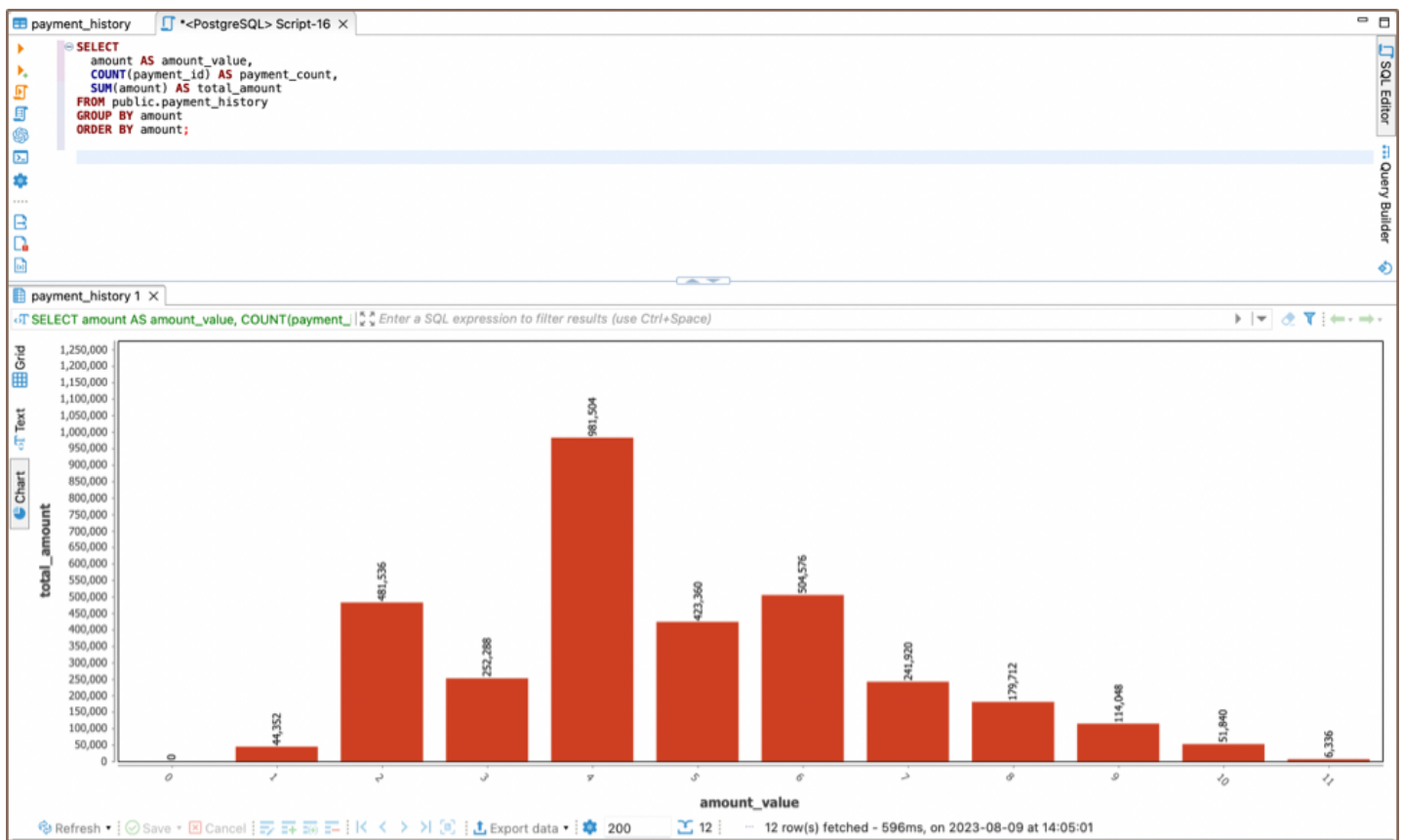
SELECT queries into colored charts. You can visualize data in **Bar**, **Line**, or **Pie** formats.

Create a chart directly in the [SQL Editor](#), [Data Editor](#) or [Grouping Panel](#). It's a quick and simple way to bring your data to life.

Creating Charts In SQL Editor

Visualizing large data sets makes your analysis faster and more precise. Charts let you grasp important details quickly.


To create a chart in the **SQL Editor**, click the **Charts** button  on the left vertical toolbar of the query results area. Your data instantly transforms into a visual representation.

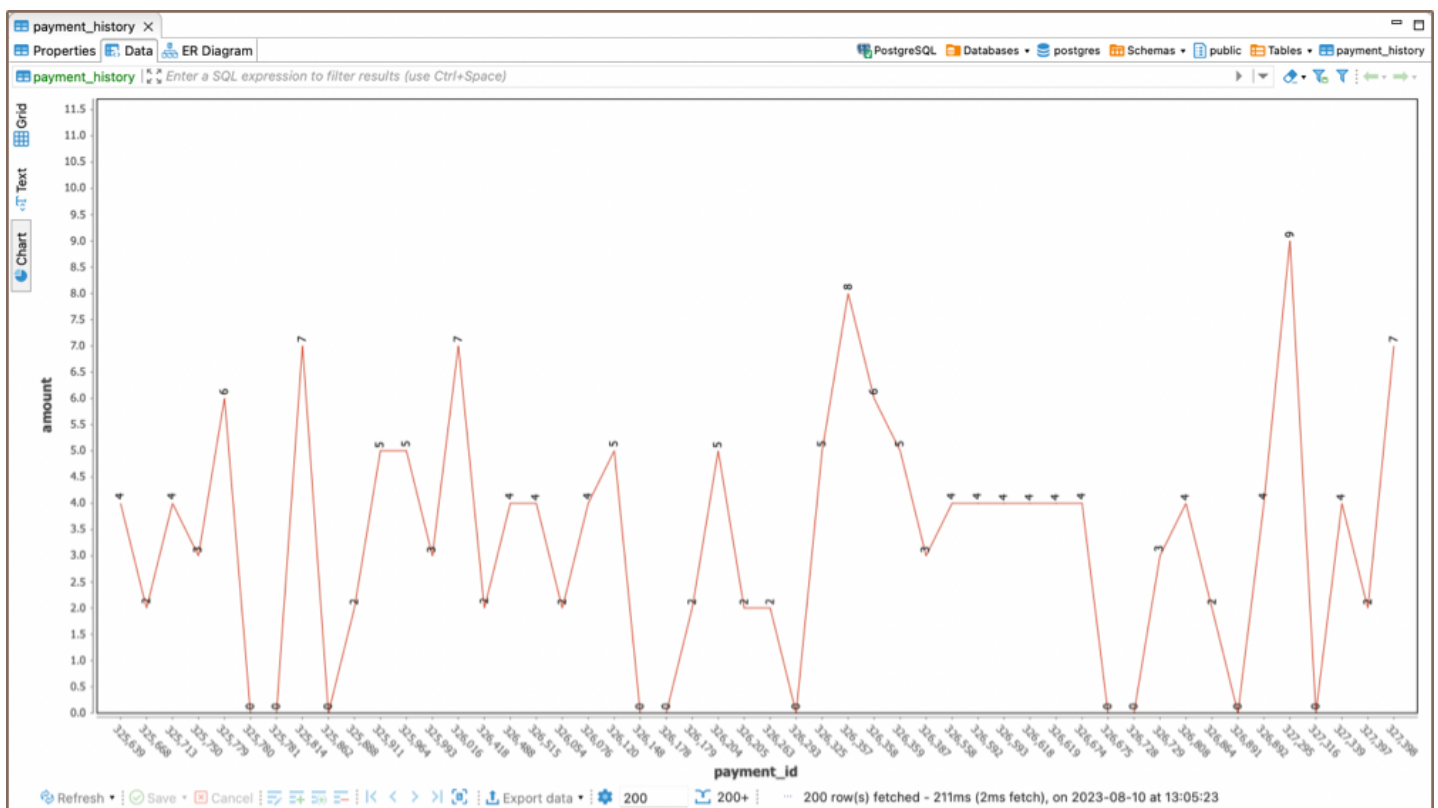


Creating Charts In Data Editor

Charts are a powerful tool for visualizing structured analytical data, such as those stored in Views.

If you want to create a chart for a table, it requires some preparation. First, sort and apply various [filters](#) to the table's columns to structure the data. These changes directly affect the appearance of the chart, enabling you to customize the visual representation to fit your needs.

To build a chart using the **Data Editor**, simply press the **Charts** button  located in the left vertical toolbar. A chart based on your structured data will be created.

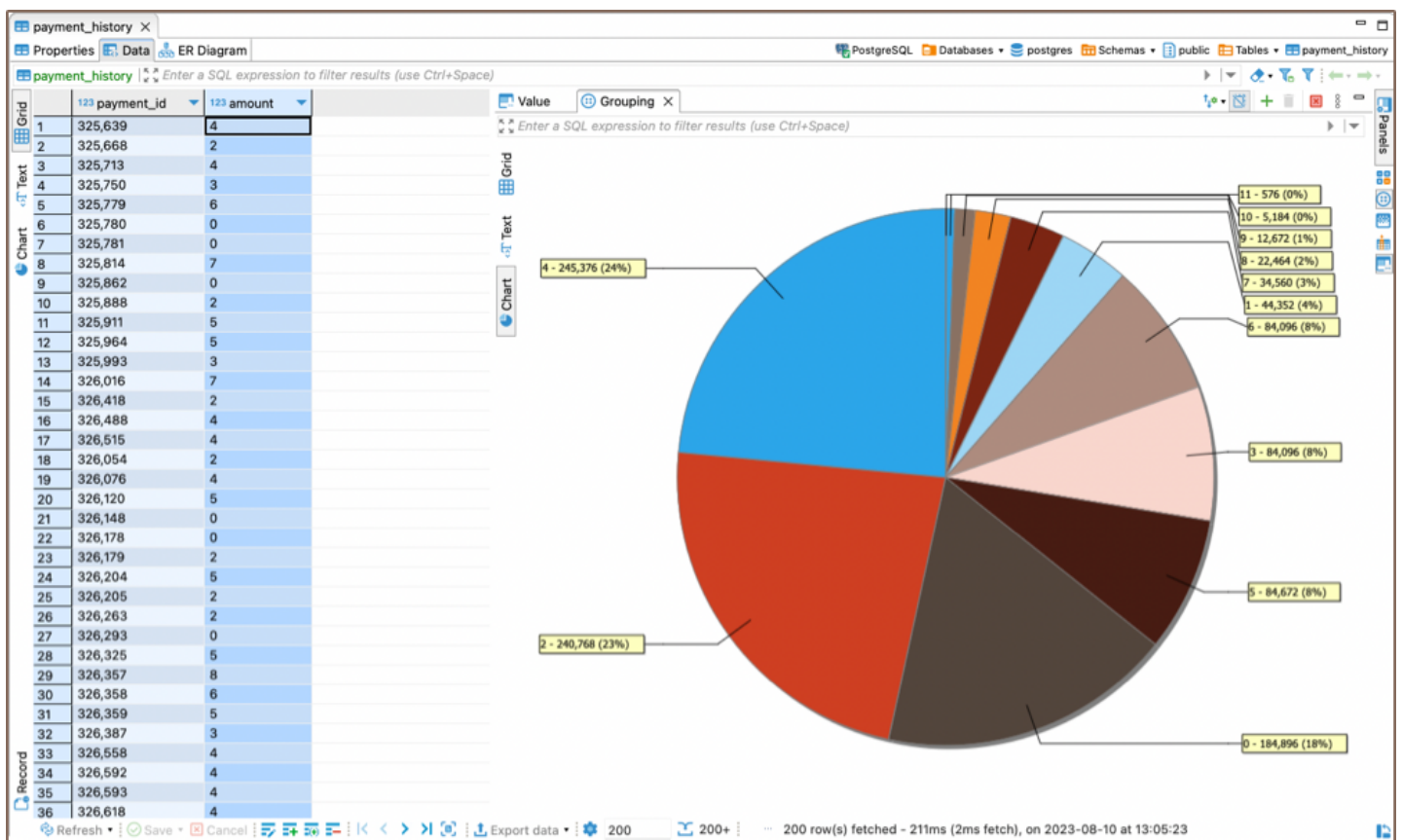


Note: By default, the data for Y-axis is taken from the first column of the table containing numeric values.

Creating Charts In Grouping Panel

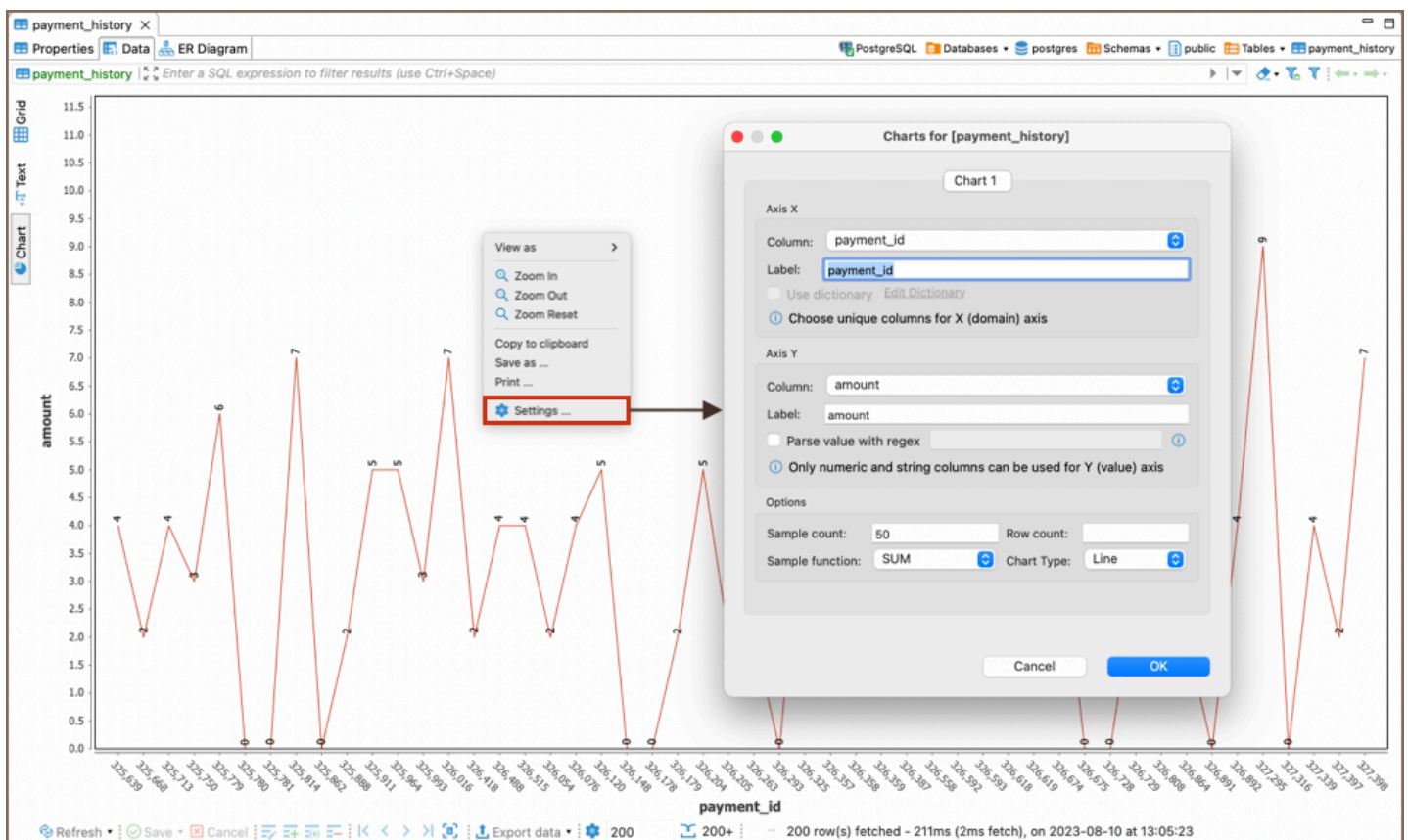
The **Charts** feature also works with analytical tools like the **Grouping Panel**. When building a chart for a table that contains grouping results for one or more columns, you can easily switch the source data for the X and Y axes by changing the columns in the Charts Editor.

To create a chart in the **Grouping Panel**, simply press the **Charts** button  on the left vertical toolbar



Editing Chart Settings

To edit the chart settings, right-click inside the chart area to open the chart's context menu, and then select the **Settings** option. The **Chart Editor** will appear, allowing you to make adjustments.



The following chart settings can be adjusted:

Setting Axis X

- In the **Column** drop down list of available columns select a column whose data will be used on X-axis of the bar chart. Make sure you choose unique columns for X-axis.
- Define a user-friendly axis name in the **Label** text field.

Setting Axis Y

- In the **Column** drop down list of available columns, select a column whose data will be used on Y-axis of the bar chart.

Note, that only numeric and string columns can be used for Y-axis.

- Define a user-friendly axis name in the **Label** text field.
- For advanced customization, you can use the **Parse value with regex** option. Regular expressions (regex) allow you to filter, match, or manipulate the column values according to specific patterns or criteria. This

functionality offers precise control over the data displayed on the Y-axis, enabling you to create more refined visual representations. If you're unfamiliar with regular expressions, the [Regular Expressions Tutorial](#) provides comprehensive guidance and examples.

Setting Other Options

You can also set the following chart options:

- **Sample count** - maximum number of columns used for building a chart.
- **Row count** - maximum number of rows used for building a chart.
- **Chart Type** - you can choose how the data will be visualized in **Bar**, **Line**, or **Pie** formats.
- **Sample function** - an aggregate function where the values of multiple rows are grouped together to form a single summary value displayed on Y-axis.

The following sample functions are supported:

Name	Description
AVG	Average value
SUM	The sum of all values
FIRST	The first value
LAST	The last value
COUNT	Total count of all values
MIN	The minimum value
MAX	The maximum value

Copying to clipboard

You can copy a chart to a clipboard by selecting the **Copy to clipboard** option in the chart's context menu.

Exporting Charts

You can export a chart into **PNG** format by selecting the **Save as...** option in the chart's context menu.

Printing Charts

You can print a chart by selecting the **Print...** option in the chart's context menu.

Zooming and Viewing Options

In the chart's context menu, you will find additional options for interacting with the chart:

- **Zoom in:** Enlarge the view of the chart.
- **Zoom out:** Reduce the view of the chart.
- **Zoom Reset:** Return to the default zoom level.
- **View as:** Change the visualization format of the chart. A submenu will appear, allowing you to select between **Bar**, **Line**, or **Pie** formats.

Working with Spatial GIS data

Table of contents

[Spatial data viewer](#)

[Differentiating data on the map](#)

[Tile layer management](#)

[Defining custom tile layer](#)

[Viewing string or binary data from any Database on a map](#)

[Labels on a map](#)

[Copying the viewport area](#)

[Additional features](#)

[Copying coordinates](#)

[Miscellaneous buttons](#)

[Viewing Curve Geometries](#)

Spatial data, often represented as either geometric or geographical values, can be visualized on a map or graph. A geometric object is typically composed of a sequence of points that define its shape. For a more comprehensive understanding of spatial data, you can refer to this [detailed explanation](#).

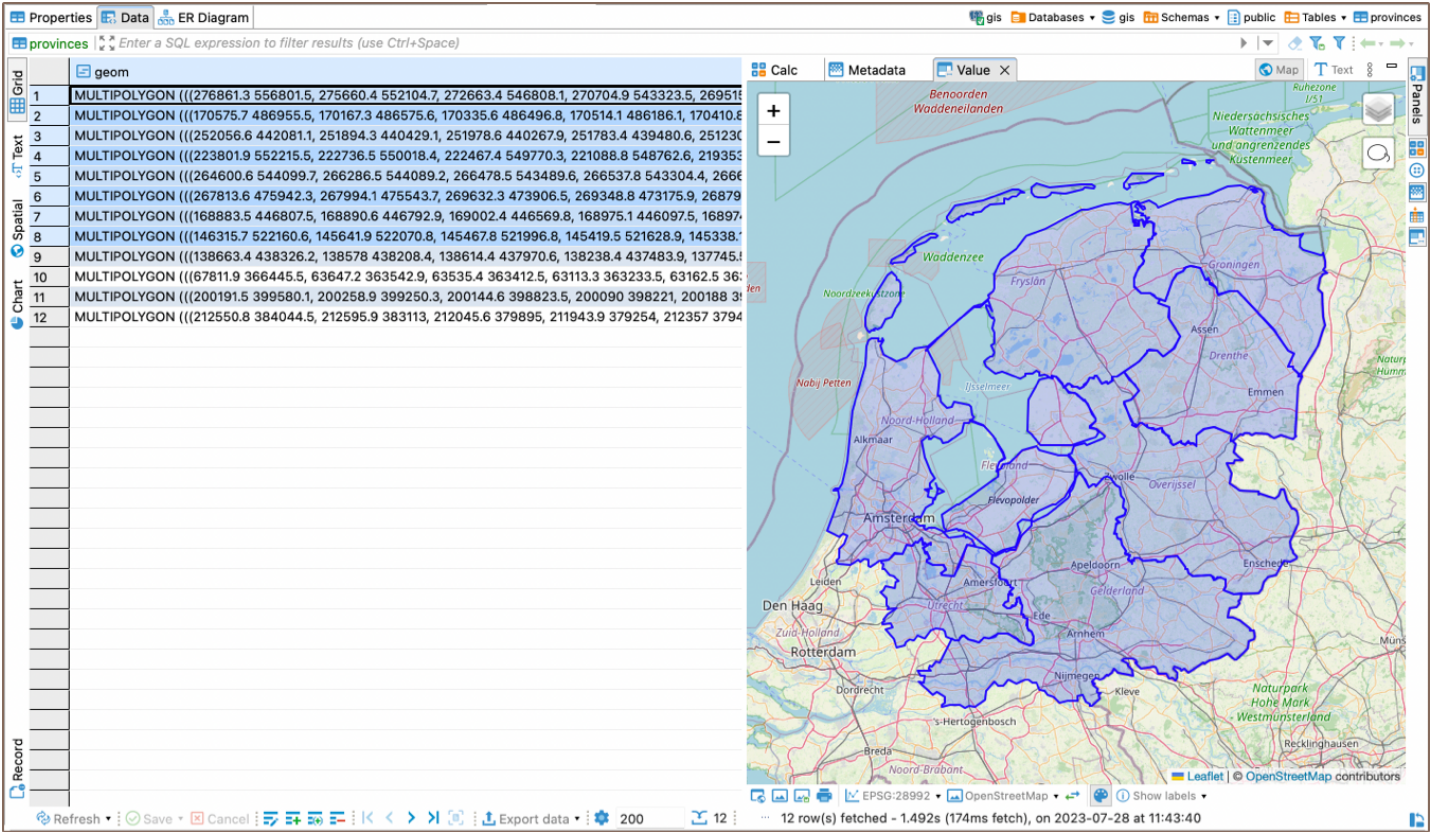
DBeaver's support of spatial data covers the following databases:

- PostgreSQL (PostGIS)
- MySQL
- SQLite (GeoPackage)
- H2GIS
- SAP HANA
- Oracle ★
- SQL Server ★

Each of these databases has its own set of features and capabilities for handling spatial data, and DBeaver is equipped to work effectively with each of them.

Spatial data viewer

The Spatial data viewer in DBeaver is a tool designed to visualise and interact with spatial data. It provides a graphical representation of your spatial data, enhancing your ability to understand and analyze it.



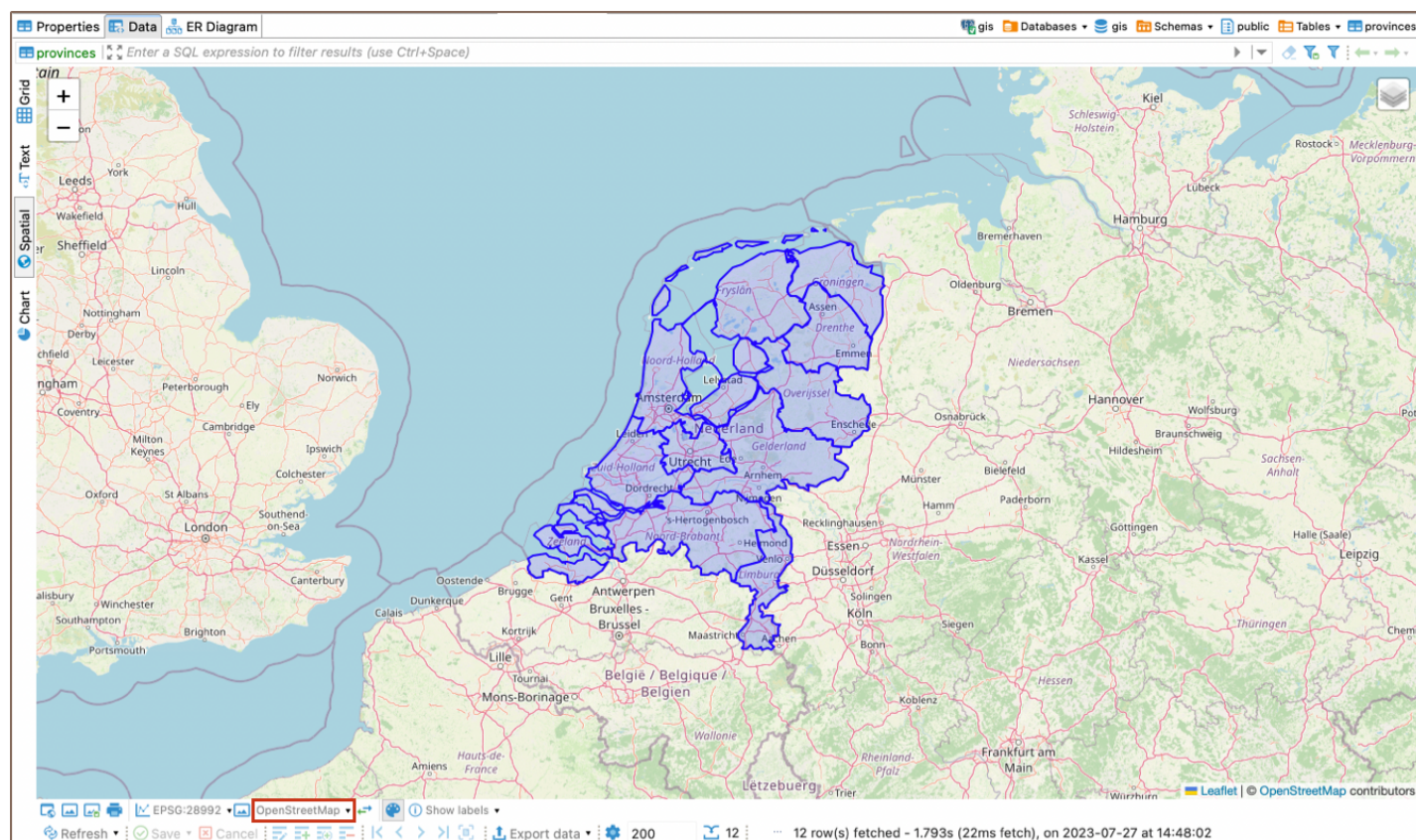
Differentiating data on the map

Every table column has its own color on the map. This helps you find the information you need on the map if you know which column it belongs to. If you click on an object on the map the following information will be displayed:

1. Name of the column in the header
2. Displayable data (strings, numbers, dates etc.) from every other column in the corresponding row

Tile layer management

DBeaver has several predefined map tiles. The tiles can be chosen with the combo below the viewer:



You can select which tile layers you want to see using the layers button. This button includes a drop-down menu where you can either select a layer or click on the **Manage** option.

The **Manage** option provides the ability to upload your own layers and hide pre-installed layers from the menu.

Defining custom tile layer

At this point, you may be wondering what to put in the Layers definition box. Here is a brief explanation.

DBeaver's spatial data viewer uses Leaflet under the hood. To define a custom tile layer, you will need to provide arguments for the `L.tileLayer()` function in the Layers definition box. This function is responsible for loading and displaying the tile layers on the map.

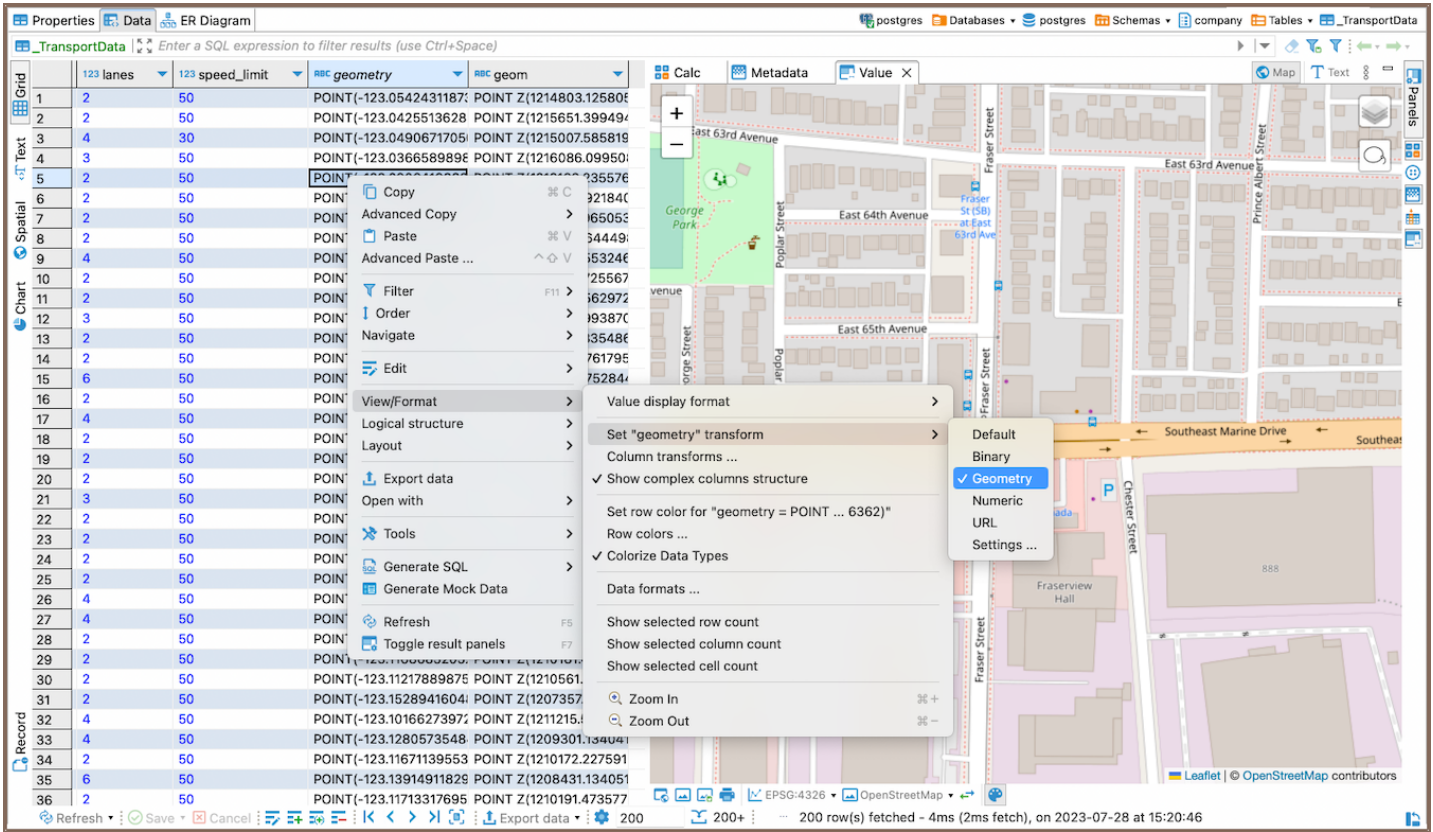
For more detailed information on how to use the `L.tileLayer()` function, refer to the [official Leaflet documentation](#). To get started, you can also look at the definition of predefined tiles in DBeaver.

Viewing string or binary data from any Database on a map

In DBeaver, you can visualize your geodata on the map, regardless of whether it is stored as string or binary data. To do this, select the data cell and navigate to the **View/Format** setting. From there, choose the **Set columnName format** option and select **Geometry** from the available formats.

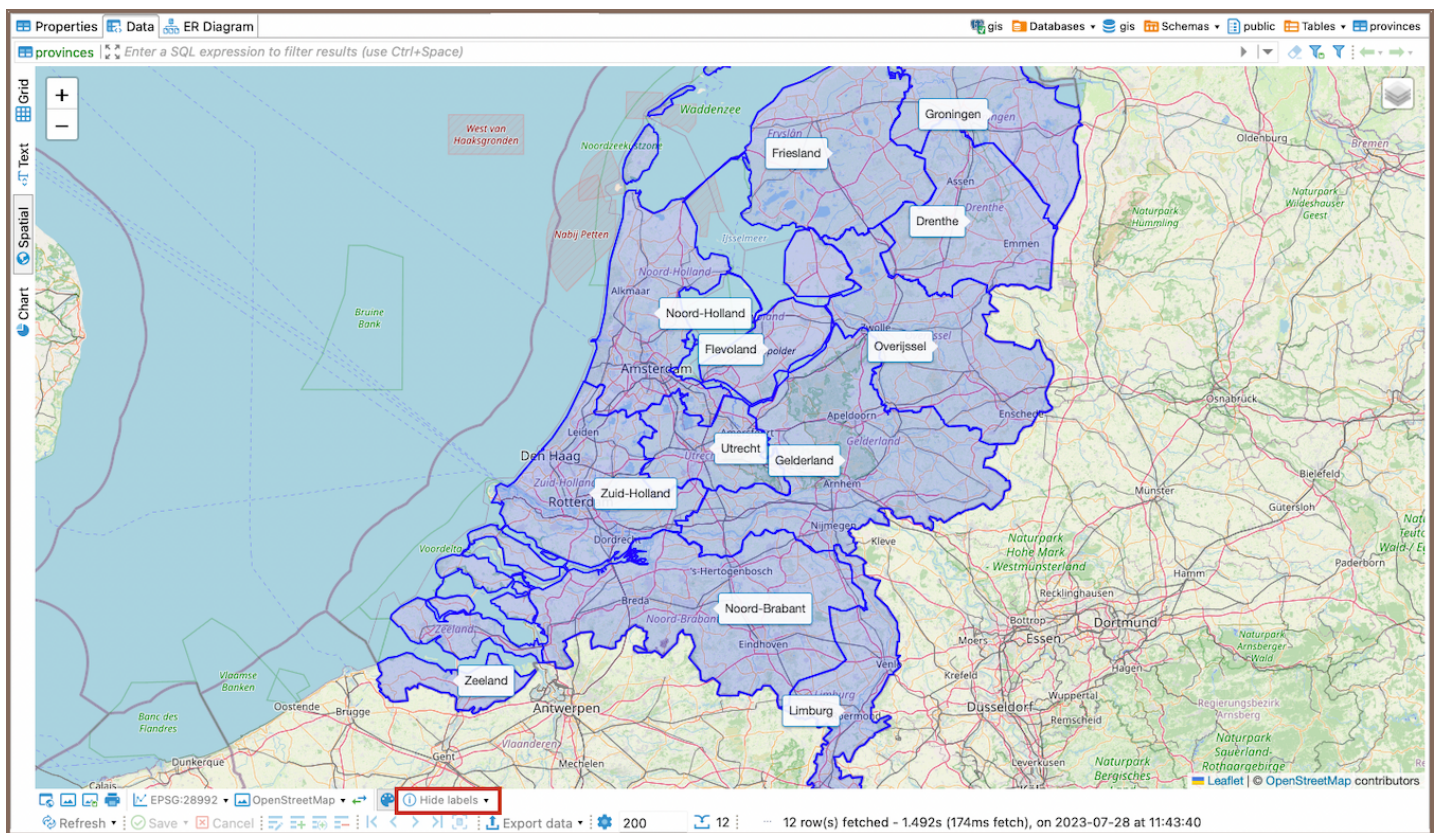
This feature allows you to transform and visualize your data in a spatial context, enhancing your understanding of the data.

Tip: It is particularly useful when working with geospatial datasets, as it enables you to see the geographical distribution and relationships of your data.



Labels on a map

You can also enable labels on the map using the **Show labels** button. This feature allows you to visualize specific data on the map, providing a more detailed view of your spatial data.



The **Show labels** button includes a drop-down menu with an option to **Configure labels**. Selecting this option opens a window where you can customize what data will be visualized in the labels, the order of the data, and the delimiter between the data.

Tip: Once the labels are enabled, the button's name changes to **Hide labels**, allowing you to easily toggle the display of labels on the map.

Copying the viewport area

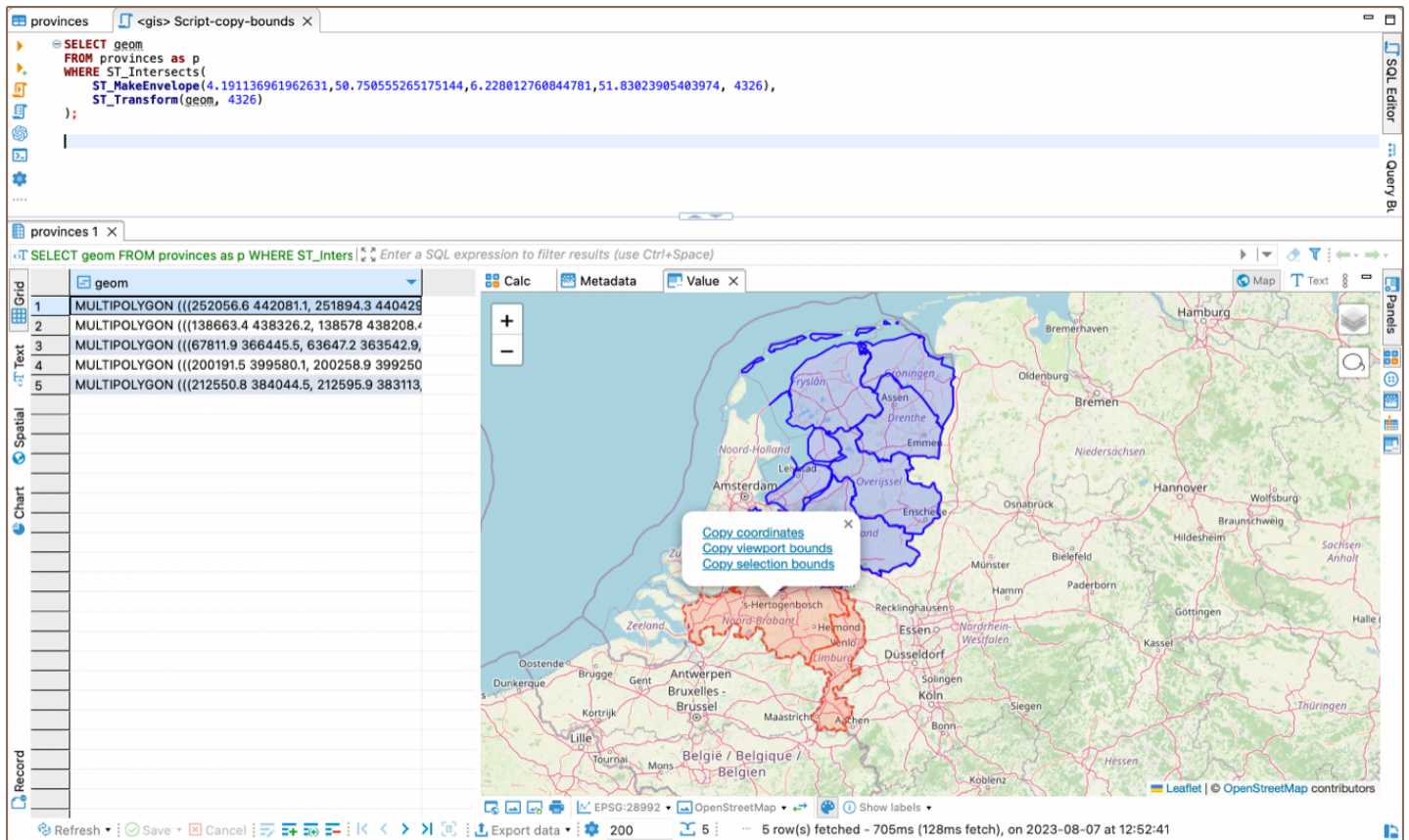
DBBeaver's spatial data viewer also includes a feature that allows you to copy the extent of the current viewport area.

To use this feature, adjust the viewport to the area of interest using the zoom and drag functions. Then, select the **Copy viewport bounds** option. This action will copy the coordinates that represent the extent of the current view area. You can then paste these coordinates into your SQL query or any other place where you need to use them.

Tip: This feature can be useful when you need to examine all spatial data within a specific area or when you want to check their attributes in relation to a particular spatial object.

Here's an example of how you can use these coordinates in an SQL query:

```
SELECT geom
FROM your_table
WHERE ST_Intersects(
    ST_MakeEnvelope(<paste bounds here>, 4326),
    ST_Transform(geom, 4326)
);
```



In this query, you replace `<paste bounds here>` with the coordinates you copied from the viewport.

- The `ST_MakeEnvelope` function creates a rectangular polygon using these coordinates.
- The `ST_Intersects` function then finds any spatial objects in the provinces table that intersect with this rectangle.
- The `ST_Transform` function is used to ensure that the geometries are in the same coordinate system (in this case, `EPSG:4326`).

Tip: If you want to copy the extent of a specific selection, you can use the **Toggle lasso** tool to **Copy selection bounds**, and copy the coordinates.

Additional features

Copying coordinates

You can copy coordinates to the clipboard from any point on the map: just right-click anywhere and select **Copy coordinates**. It copies the coordinates formatted as `latitude`, `longitude` to the clipboard. **NOTE:** The coordinates are copied according to `EPSG:4326 CRS` and are just raw numbers. You may need to remove a comma and switch the `latitude` and `longitude` places to correctly insert it into a database.

Miscellaneous buttons

You can use buttons at the bottom of the view for additional features:

- Open the temporary generated `HTML` file in your default browser.
- Copy the current map to the clipboard as a picture.
- Save the current map as a picture into the selected folder.
- Print the current map.
- Flip the `latitude` and `longitude` coordinates in the source data. This can be useful if the data in your table is saved in (`latitude` / `longitude`) format while the Leaflet reads it as (`longitude` , `latitude`). This button does not change anything in the source data, it just changes how this data is read to show the accurate information on the map.

Viewing Curve Geometries

You can now visualize the curve geometries in DBeaver's spatial data viewer. This enhancement allows you to work with complex spatial data types such as `CIRCULARSTRING`, `COMPOUNDCURVE`, `CURVEPOLYGON`, `MULTICURVE`, and `MULTISURFACE`.

You can use SQL functions like `st_geomfromtext` and `st_curvetoline` to create and view curve geometries.

Tip: This feature can be specifically useful when dealing with complex spatial datasets in PostGIS that include curved geometries. It allows for more precise representation and analysis of geographical features that are not adequately represented by simple polygons or lines, such as circular areas of interest, roads that follow curved paths, or areas defined by a radius from a certain point.

GIS examples

Table of contents

[SQL Queries](#)

[Test databases](#)

SQL Queries

PostgreSQL queries reading GIS data types: (can be executed in SQL editor)

```
SELECT ST_MakePoint(112.123,243.123);  
SELECT GeomFromEWKT('SRID=4326;POINT(111.1111111 1.1111111)');  
SELECT GeomFromEWKT('SRID=4326;POLYGON((0 0,0 1,1 1,1 0,0 0))');
```

Test databases

<https://knowledge.safe.com/questions/70649/fme-basic-training-postgis-database-connection.html>

Parameter	Value
Host	postgis.train.safe.com
Port	5432
Database	fmedata
Username	fmedata
Password	fmedata

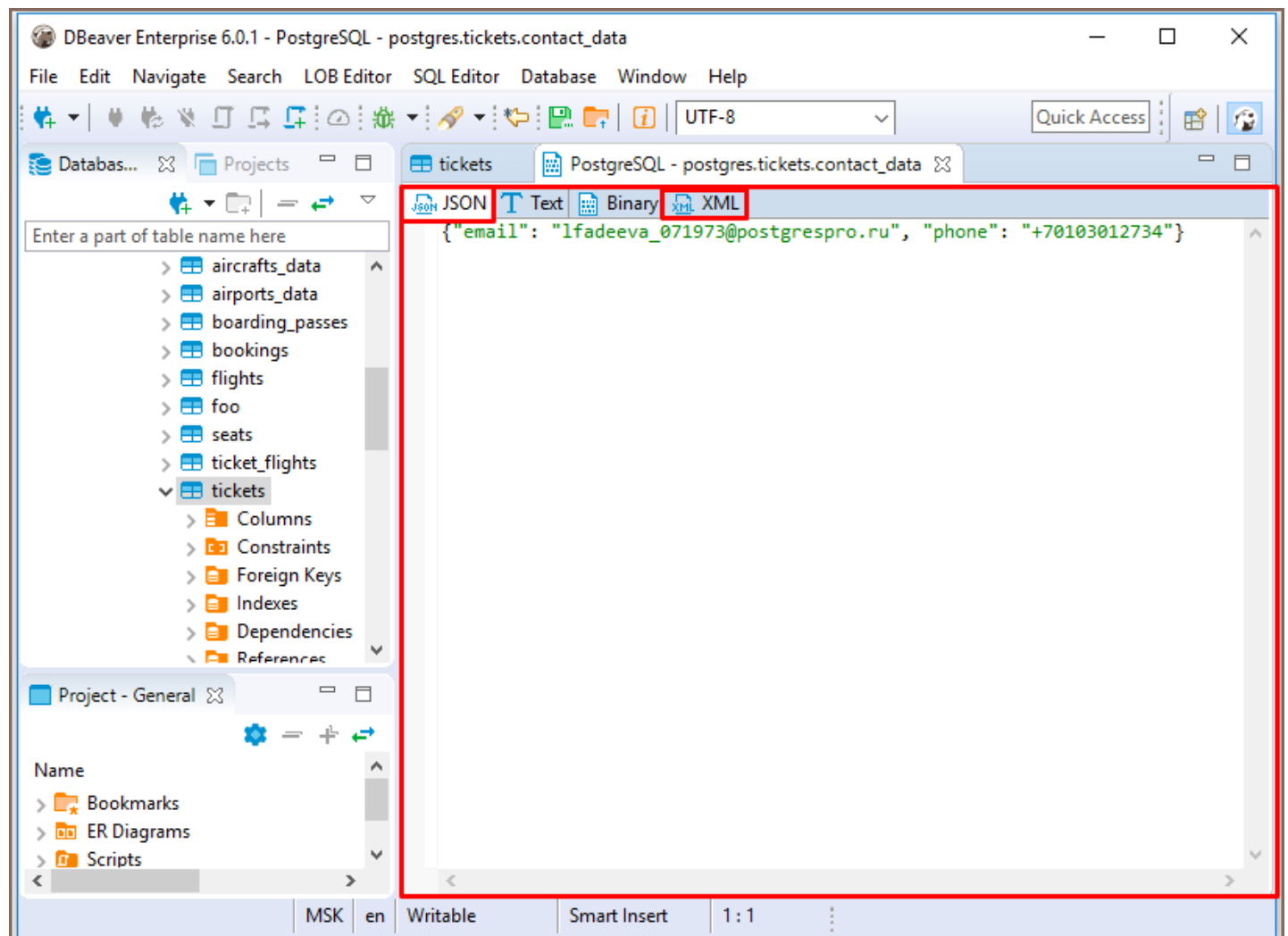
Working with XML and JSON

DBeaver supports XML and JSON column types (in relational databases) by using standard JDBC interfaces. This feature was added in JDBC4 so you will need JDBC4 compliant driver for your database.

In the Data Editor, you can edit XML/JSON data right in the table cells/ However, a huge amount of data may require a larger editor so you might want to save XML/JSON scripts to a local file or upload this type of data from a local file.

To open the full-size XML/JSON editor click the cell containing data in XML/JSON format and press **Shift + Enter**

By default the editor opens on JSON tab, open XML tab to modify XML data.

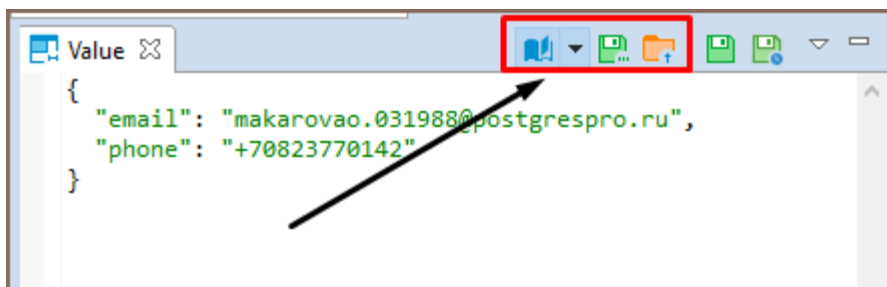


To auto-format XML/JSON script press **Ctrl + Shift + F** keyboard buttons.




Use **Ctrl + S** keyboard shortcut to save the changes made.

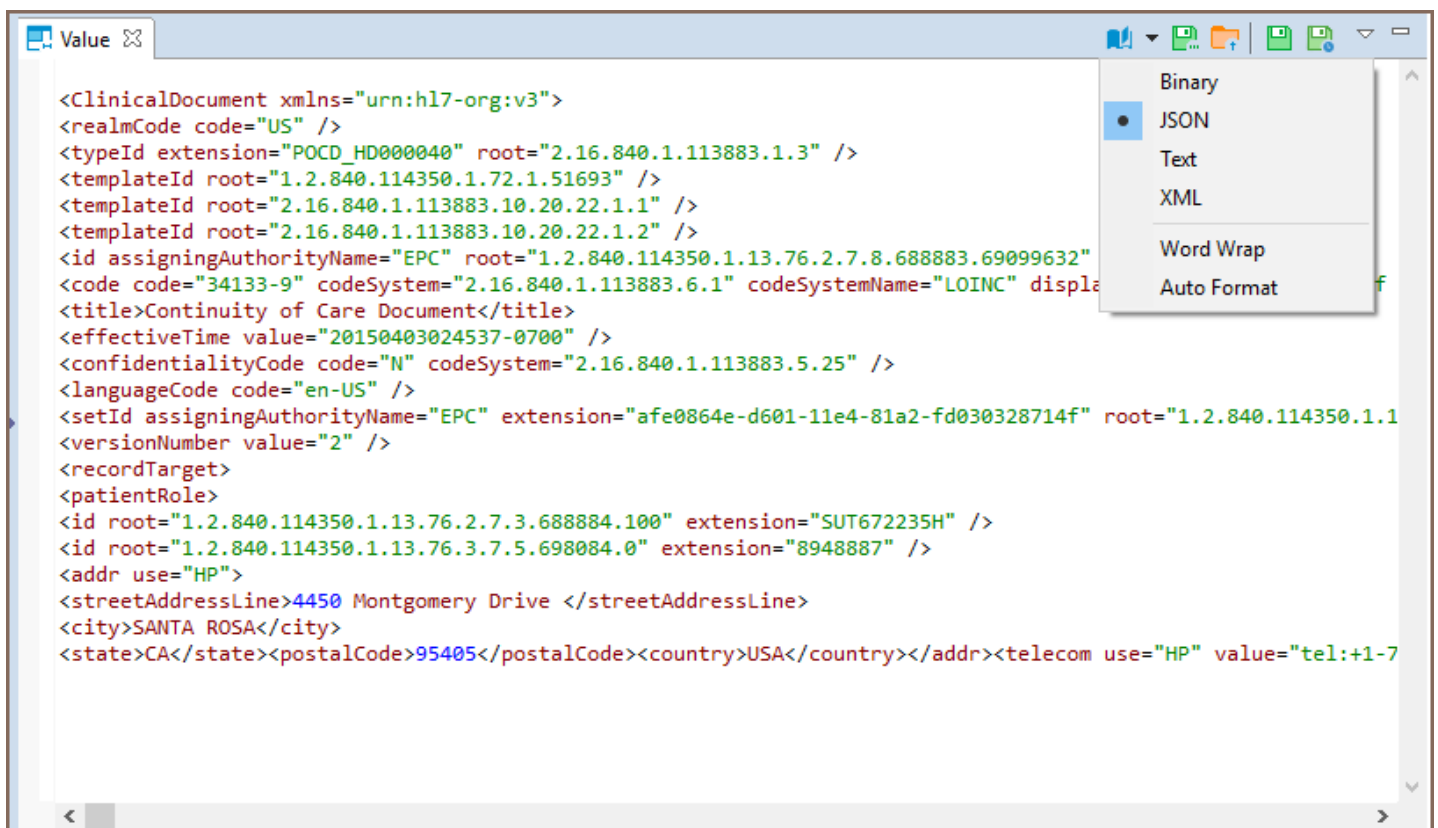
You can also edit XML/JSON content, save it locally and upload it from a local file with the help of **Value** panel toolbar.



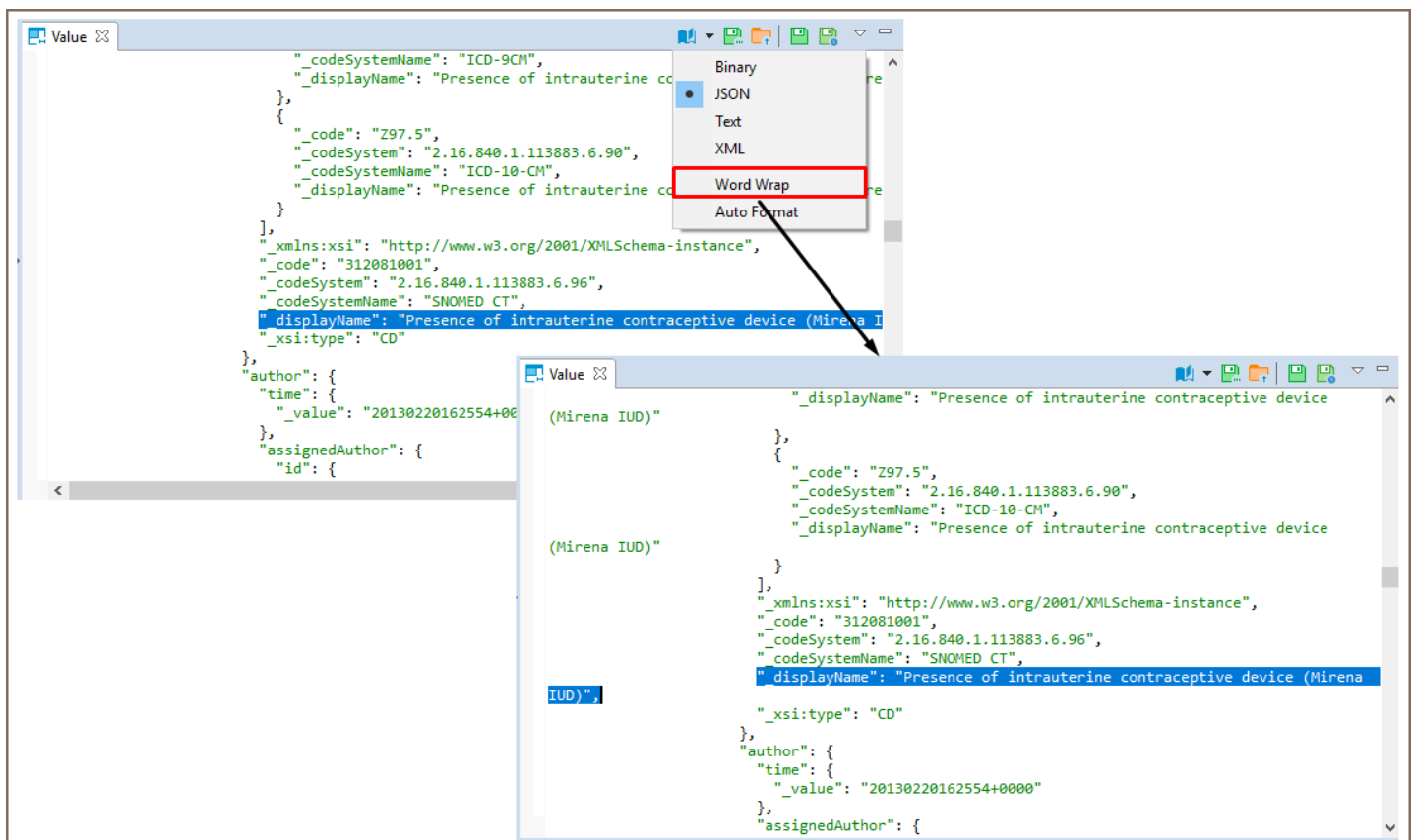
To upload data from a local file, press the **Load from file...** button .

To save the content to a local file, press the **Save to file...** button .

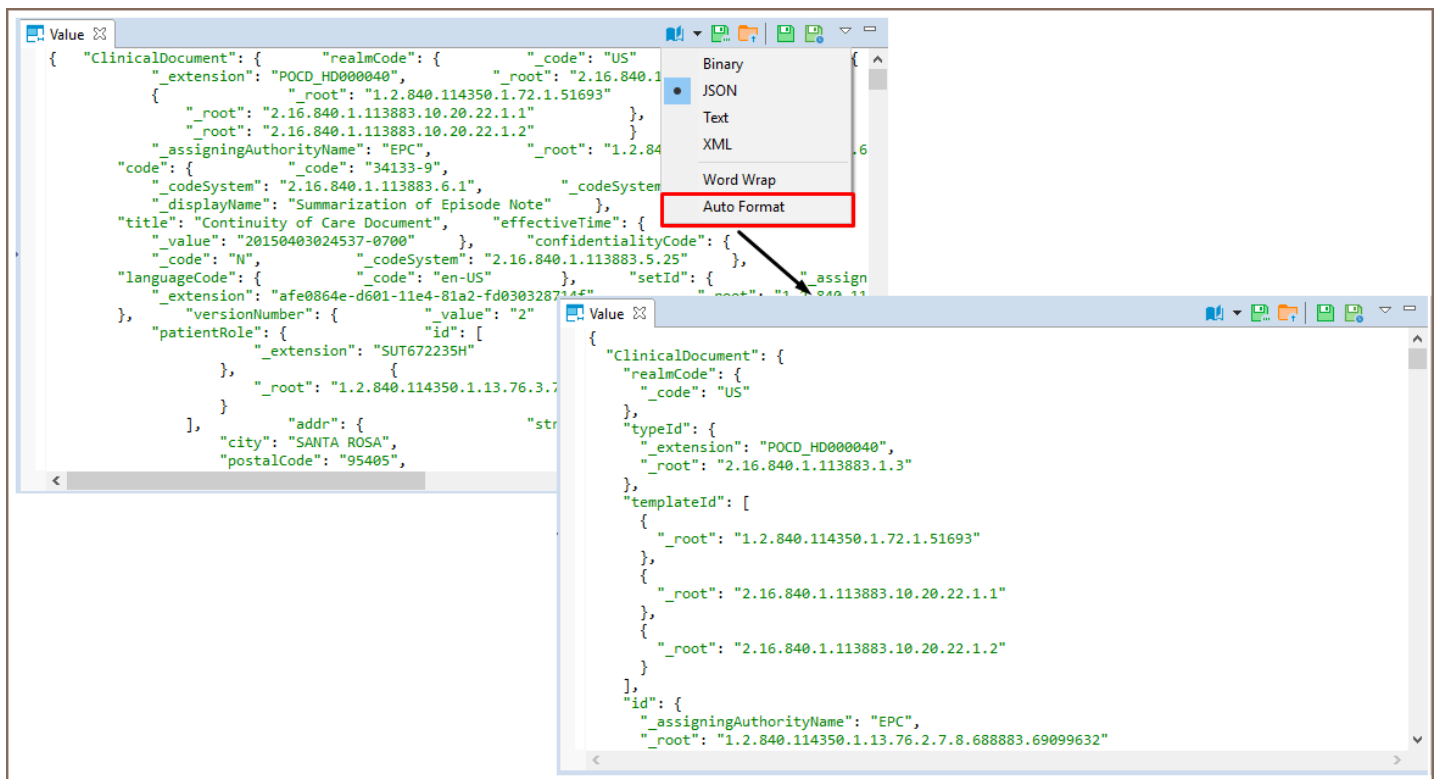
To switch between the formats, press the **Content viewer settings** button  and select the format.



Use **Word Wrap** feature that wraps the text within a screen.



Use **Auto Format** feature to automatically change the appearance of XML/JSON script (fix spaces around operators / commas, fix indentation, etc) and make it more readable.



To learn more about **Value** panel, see [Panels](#).

Virtual column expressions

Table of contents

- [Expression language](#)
- [Column values](#)
- [Standard functions](#)
- [math](#)
- [geo](#)
- [content](#)
- [Defining Virtual Keys](#)

Expression language

user	123 Milliseconds	123 Bytes	123 UnitPrice
ng, Malcolm Young, Brian Johnson	343,719	11,170,334	0.9900000000
	342,562	5,510,424	0.9900000000
Kaufman, U. Dirksneider & W. Hoffmar	230 619	3 990 994	0.9900000000
A. Smith-Diesel, S. Kaufman, U. Dirksne			
A. Smith-Diesel			
ng, Malcolm Young, Brian Johnson			
ng, Malcolm Young, Brian Johnson			
ng, Malcolm Young, Brian Johnson			
ng, Malcolm Young, Brian Johnson			
ng, Malcolm Young, Brian Johnson			
ng, Malcolm Young, Brian Johnson			
ng, Malcolm Young, Brian Johnson			
ng, Malcolm Young, Brian Johnson			
ng, Malcolm Young, Brian Johnson			
ng, Malcolm Young, Brian Johnson			

Add virtual column

Add virtual foreign key

Define virtual unique key

Edit ...

Copy

Advanced Copy

Paste

Advanced Paste ...

Filter

Order

Navigate

Edit

View/Format

Logical structure

Layout

Export data ...

Open with

You can use standard JavaScript-like expression language. DBeaver uses the Jexl engine to process expressions. Language references and examples can be found here: <http://commons.apache.org/proper/commons-jexl/reference/syntax.html>

Column values

All columns' values in the current result set can be referred to by name. Expression `column1 + column2` will produce the sum of two numeric columns or concatenation of two string columns `column` and `column2`.

Standard functions

Standard functions are declared in namespaces.

You can refer to the functions in the namespaces as variables - `nsName.functionName(parameters)`.

math

You can access all math functions as `math.function(parameters)`.

You can find all supported math functions here: https://docs.oracle.com/cd/E12839_01/apirefs.1111/e12048/functmath.htm

geo

Function	Parameters	Description
wktPoint	(longitude, latitude)	Produces WKT (geometry) point out of two coordinates. Default SRID is 4326.
wktPoint	(longitude, latitude, srid)	Produces WKT (geometry) point out of two coordinates and SRID

content

If you have JSON or XML columns in your table, you can add a virtual column with an expression for these columns.

Use `content.json()[parameter1][parameter2]` pattern to create expression for JSON column.

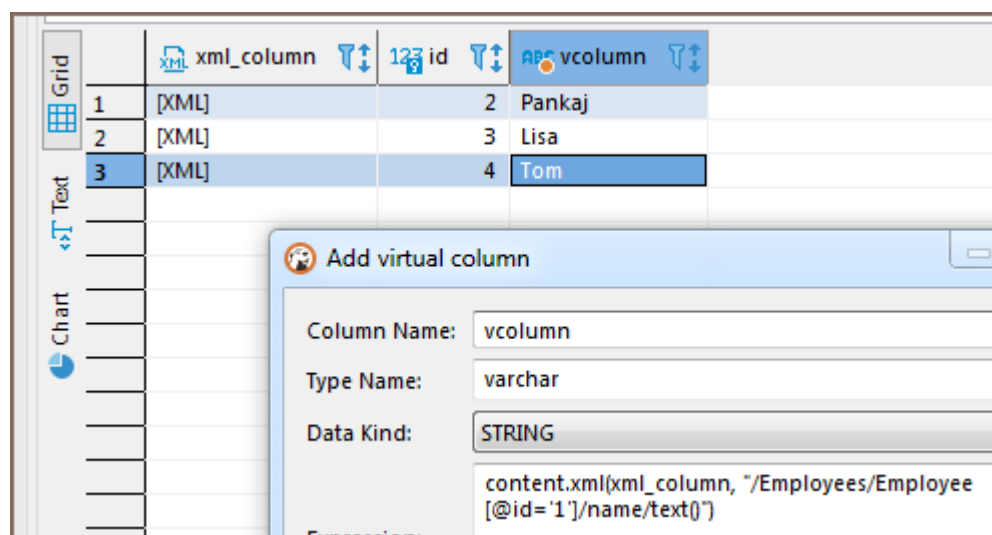
Expression example: `content.json(column1)['glossary']['GlossDiv']['title']`

You can read more about JSON parameters in the link on the top.

Use `content.xml(columnName, "expression")` or `content.xml(columnName, "returnType", "expression")` patterns to create expression for XML column. The quotation is important for parsing processes. XML expression can return types: string, number, boolean, node, nodeset - all these types can be used (!in quotes) for return data type clarification. `content.xml(columnName, "expression")` returns string by default

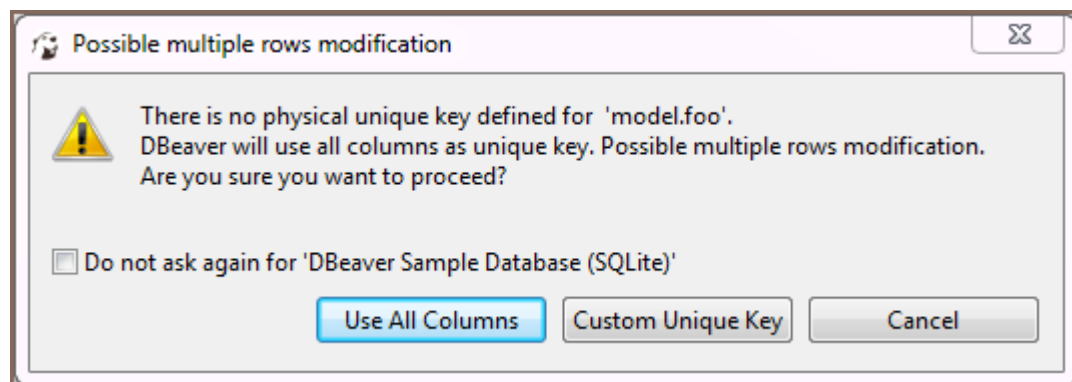
Expression example: `content.xml(column1, "nodeset", "/Employees/Employee[gender='Female']/name/text())`

You can read more about XPath here: <https://en.wikipedia.org/wiki/XPath>

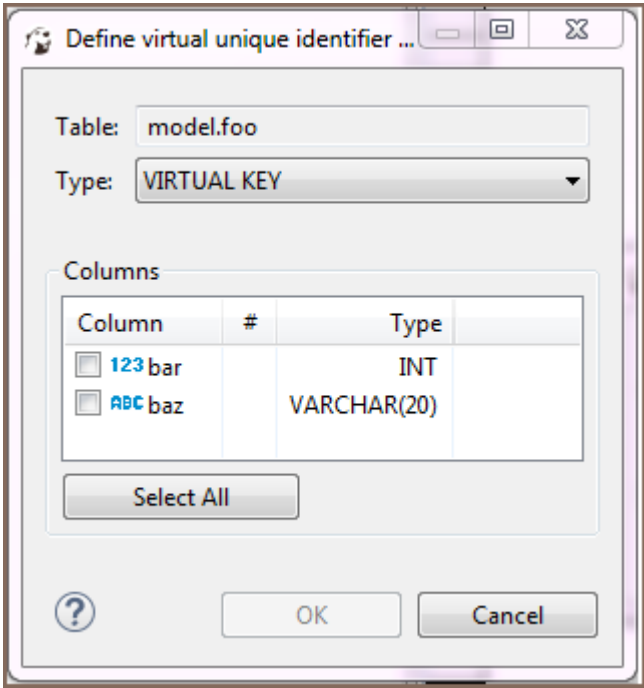


Defining Virtual Keys

To be able to save column value changes, a table must have some unique key (primary key or unique index). Some databases (Oracle, DB2, PostgreSQL) support a special virtual unique column that DBeaver can use to save changes. In other cases, you can define a virtual key – a set of columns that forms a unique combination of values. When you try to save changes in a table without a unique key, DBeaver displays the following error message:



To use all columns as the virtual key, click **Use All Columns**. To create a custom key, click **Custom Unique Key**. Alternatively, to create a unique custom key, you can click the Configure (⚙️) button on the bottom toolbar and then click **Define virtual unique key**** on the Configure menu. The Define virtual unique identifier window opens:

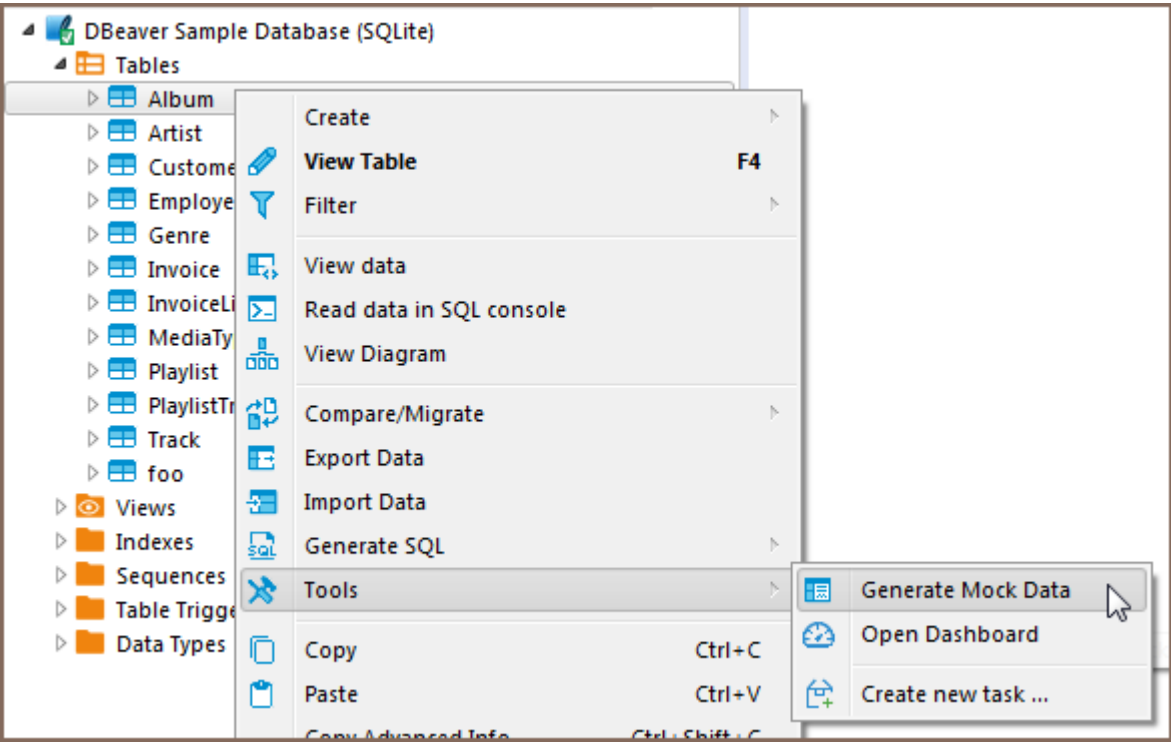


To define the key, select some of the columns or click **Select All** and then click **OK**. To remove a unique key from a table, click the **Configure** button in the bottom toolbar and then click **Clear virtual unique key**.

Mock Data generation

Note: This feature is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Sometimes in software development we need to generate mock, but valid, data for testing. Populating a database manually is a time-consuming and exhausting process. It can be very complicated when you need to generate not just 5–10 users, but thousands of entities of different types. DBeaver Mock Data generator helps you generate test data much easier.



[NULL]	267,014	0.596414983	5,932
[NULL]	291,931	0.58602345	[NULL]
[NULL]	225,651	0.862772703	-1,958
[NULL]	653,269	0.636165559	-3,706
[NULL]	792,738	0.875926256	32,481
[NULL]	[NULL]	0.517556012	[NULL]
[NULL]	599,256	0.652565897	19,923
[NULL]	440,605	0.408405334	-20,854
[NULL]	702,406	0.941095829	22,609
[NULL]	693,320	0.488401949	13,257
[NULL]	[NULL]	0.535391688	17,223
[NULL]	378,120	0.819397271	-17,869
[NULL]	630,360	[NULL]	228

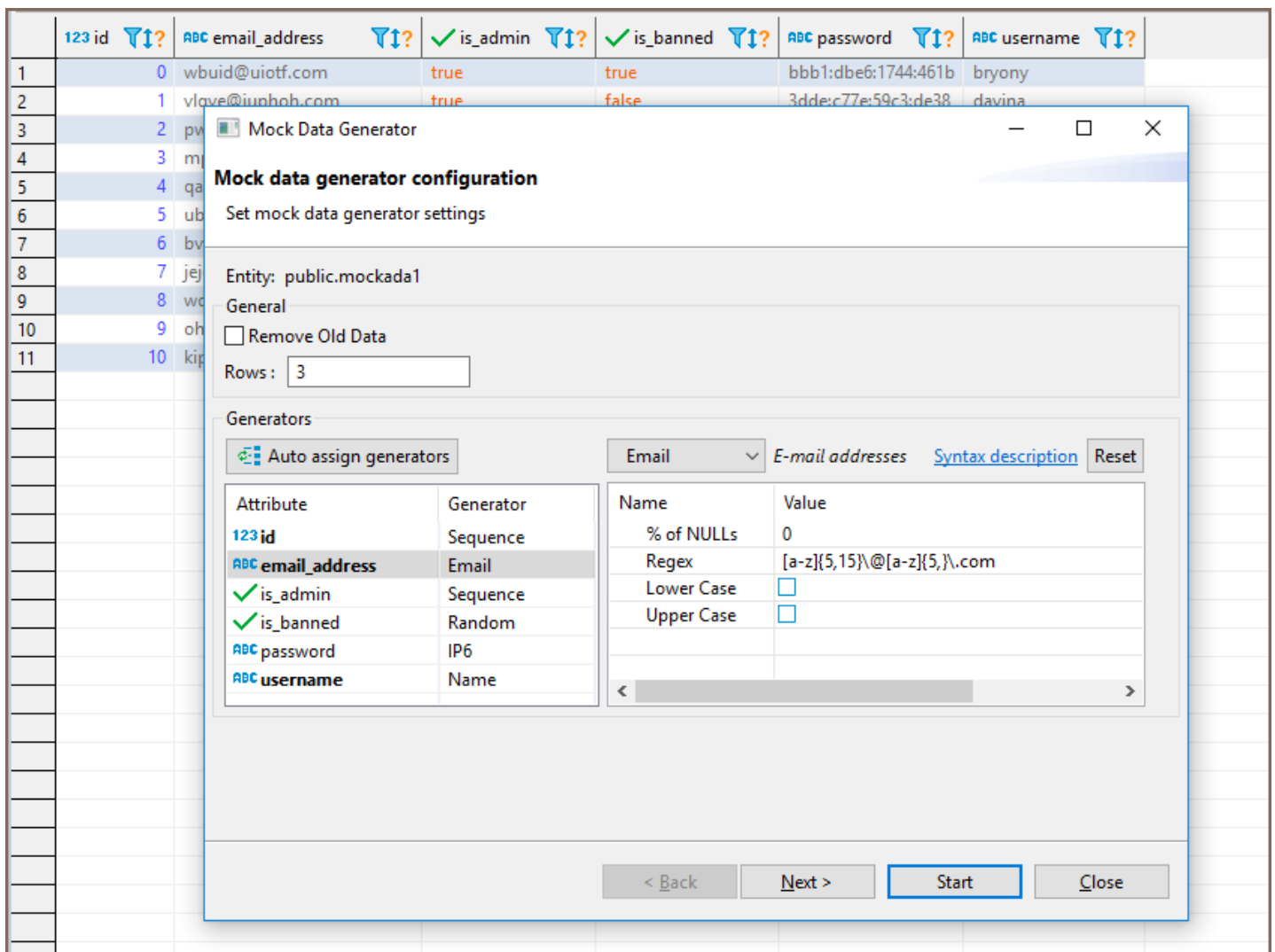
55 row(s) fetched

Generate Mock Data

Disclaimer: The idea behind Mock Data is to generate mock data in a table but it should **NOT TO BE USED IN PRODUCTION ENVIRONMENTS**. Please make sure you have a backup of your database before running the Mock Data generation process.

The following are features of the DBeaver Mock Data generator:

- Works for all the RDBMS that are supported by DBeaver (DB2, MS SQL Server, MySQL, Oracle, PostgreSQL, SQLite, etc.)
- Generates data that matches your database schema:
 - Generated data matches the database column types.
 - All base data types are supported.
 - Constraints (PK, FK, multi-column FK, unique) are supported.
- Supports over 20 configurable data generators (constants, randoms, sequences, names, domains, addresses, prices, regex based, etc.)
- Automatically associates a column with a generator based on the column characteristics
- Saves or overwrites old database data



The following are mock data generators for data types with their configurable parameters:

- Boolean
 - Random
 - Sequence (initial, order)
- Date
 - Random (start, end)
 - Sequence (start, step, reverse)
- Numeric
 - Random
 - Sequence (start, step, reverse)
 - Advanced (min, max, precision, scale) *

- Price preset *
- Coordinate preset *
- String
 - Text (template, min length, max length)
 - UUID
 - Address *
 - City *
 - Country *
 - Domain *
 - Email (gender, with surname, numeric suffix) *
 - Name (gender, with surname) *
 - Price (country, min, max) *
 - Regex based random (regex template) *
 - Credit Card preset *
 - Email preset *
 - Gender preset *
 - HEX Color preset *
 - IP4 address preset *
 - IP6 address preset *
 - Phone Number preset *

- Postal Code preset *
- Price preset *
- Template with parametrized directives for other generators * :
 - address() - US postal address
 - city() - one of the world's largest cities
 - country() - country
 - domain() - one of the top Internet domains
 - email(gender,surname) - e-mail address (gender is ALL|FEMALE|MALE, surname is true|false)
 - name(gender,surname) - personal name (gender is ALL|FEMALE|MALE, surname is true|false)
 - random(minimum,maximum) - random integer
 - regex(pattern) - regex based value for the pattern
 - sequence(start,step) - sequence of integers
- NULL values
- FK - data from the referenced table according to the constraint

123 id	ABC email_address	✓ is_admin	✓ is_banned	ABC password	ABC username
0	wbuid@uiotf.com	true	true	bbb1:db6:1744:461b	bryony
1	vlqe@iuphoh.com	true	false	3dde:c77e:59c3:de38	davina
2	pwwjfw@iitig.com	true	true	6fb7:3c8ea72c:db43	delia
3	mpowgwqrsqq@osndw.com	true	false	042d:b708:e94b:eb55	samantha
4	qaqdv@ucnvs.com	true	true	051d:b48d:47b7:bfed	whitney
5	ubwkehfse@bffkq.com	true	false	35d8:b74b:56b2:57b9	owen
6	bvgoqs@vxzofg.com	true	true	85ef:5ac4:89c3:a3b9	stella
7	jejqt@otstm.com	true	false	d23c:7ed7:bd8e:e26c	eve
8	wqzy@xlkrenfj.com				
9	ohtwr@rbsmx.com				
10	kipyq@kwfkv.com				

Mock Data Generator

Mock Data Generator progress

Mock Data Generator progress log

Removing old data from the 'mockada1'.

Rows updated: 15

Duration: 76ms

Inserting mock data into the 'mockada1'.

Rows updated: 11

Duration: 160ms

< Back

Next >

Start

Close

* These features are available in [Enterprise](#) and [Ultimate](#) editions only.

SQL Editor overview

Table of contents

[Getting started](#)

[SQL Editor overview](#)

[Script panel](#)

[Spelling](#)

[Hyperlinks](#)

[Highlighting](#)

[Toolbar](#)

[Results Panel](#)

[Multiple results in one tab](#)

[Additional features](#)

[Active Database](#)

[Layout Adjustment](#)

[Outline](#)

[Features summary](#)

With the SQL Editor in DBeaver, you can write and execute multiple SQL scripts within a single database connection, save them as files, and reuse them later.

Getting started

To start working with the SQL Editor, you have several options:

1. Via Database Navigator:

- Navigate to your desired database connection in the [Database Navigator](#) view.
- Press **F4** or go to **SQL Editor -> Open SQL script** from the main menu, or right-click on the connection and choose **Open SQL script** from the context menu.
- A **Choose SQL Script** window appears. Click any script to open it in a new tab.

2. Open Recent SQL script:

- Right-click on your database connection and select **SQL Editor** -> **Recent SQL script** from the context menu or go to **SQL Editor** -> **Recent SQL script** from the main menu. Alternatively use the **Ctrl+Enter** shortcut in the **Database Navigator** view.
- A **Choose SQL Script** window appears. Click any script to open it in a new tab.

3. Create a New SQL Script:

- Navigate to **SQL Editor** -> **New SQL Editor** on the main menu.
- Press **F3** and click **New Script** in the **Choose SQL Script** window.

Note: SQL Editor for a connection is different from SQL console for a table or view. Unlike the console, it can save scripts and changes made to them.

You can see all your saved SQL scripts in the [Project Explorer](#) view in the **Scripts** folder.

SQL Editor overview

The SQL Editor contains the Script panel, the Toolbar, and the Result panel.

The screenshot displays the SQL Editor interface with three main components labeled:

- Toolbar:** Located at the top left, it contains various icons for file operations (save, open, print), editing (undo, redo), and execution (run, stop, refresh).
- Script panel:** The top central area where the SQL query is written. It shows a query: `SELECT "CustomerId", "FirstName", "LastName", "Company", "Address", "City", "State", "Country", "PostalCode", "Phone", "Fax", 'Email', "SupportRepId" FROM public."Customer";`
- Result panel:** The bottom area displaying the query results. It includes a search bar, a filter input, and a table of results.

	CustomerId	FirstName	LastName	Company	Address	City
1	1	Luís	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José do
2	2	Leonie	Köhler	[NULL]	Theodor-Heuss-Straße 34	Stuttgart
3	3	François	Tremblay	[NULL]	1498 rue Bélanger	Montréal
4	4	Bjørn	Hansen	[NULL]	Ullevålsveien 14	Oslo
5	5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague
6	6	Helena	Holý	[NULL]	Rilská 3174/6	Prague
7	7	Astrid	Gruber	[NULL]	Rotenturmstraße 4, 1010 Innere	Vienne
8	8	Daan	Peeters	[NULL]	Grétrystraat 63	Brussels
9	9	Kara	Nielsen	[NULL]	Sønder Boulevard 51	Copenhagen
10	10	Eduardo	Martins	Woodstock Discos	Rua Dr. Falcão Filho, 155	São Paulo

Tip: You can open the SQL editor preferences by pressing `Alt+Enter` or right-click and navigate to **Preferences**.

Script panel

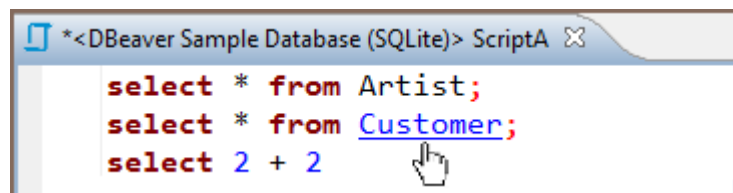
The Script Panel is the primary area where you can write, edit, and manage your SQL scripts. It provides basic text editing features with the added benefits of specialized functionalities tailored for SQL development:

Spelling

The panel includes a spelling checker to identify and highlight misspelled words, assisting you in maintaining the quality of your scripts. For more information check out our [article](#).

Hyperlinks

You can press and hold `Ctrl` and at the same time move the mouse over the SQL text. If DBeaver recognizes some identifier as a table/view name, it presents it as a hyperlink. You can click the hyperlink to open this object's editor:



Highlighting

DBeaver uses SQL syntax highlighting which depends on the database associated with the script. Different databases have different sets of reserved keywords and system functions. For more information, see the relevant section on highlighting settings in our article on [SQL Code Editor](#).

Toolbar

The toolbar is customizable and contains buttons for commonly used commands. For more information on customization, see [our article on toolbar customization](#).

Results Panel

The results panel displays tabs with results in various formats. The tabs resulting from script execution represent instances of the [Data Editor](#). You can create, edit and execute SQL scripts in the script panel and then see the results in the result tabs.

Multiple results in one tab

You can view and manage multiple query results within a single tab.

Employee 2

SELECT EmployeeId, LastName, FirstNi ... Enter a SQL expression to filter results (use Ctrl+Space)

	EmployeeId	LastName	FirstName	Title	ReportsTo	BirthDate	HireDate	Address
1	1	Adams	Andrew	General Manager	[NULL]	1962-02-18 00:00:00	2002-08-14 00:00:00	11120 J...
2	2	Edwards	Nancy	Sales Manager	1	1958-12-08 00:00:00	2002-05-01 00:00:00	825 8 A...
3	3	Peacock	Jane	Sales Support Age	2	1973-08-29 00:00:00	2002-04-01 00:00:00	1111 6 A...
4	4	Park	Margaret	Sales Support Age	2	1947-09-19 00:00:00	2003-05-03 00:00:00	683 10 ...
5	5	Johnson	Steve	Sales Support Age	2	1965-03-03 00:00:00	2003-10-17 00:00:00	7727B ...
6	6	Mitchell	Michael	IT Manager	1	1973-07-01 00:00:00	2003-10-17 00:00:00	5827 B...
7	7	King	Robert	IT Staff	6	1970-05-29 00:00:00	2004-01-02 00:00:00	590 Co...
8	8	Callahan	Laura	IT Staff	6	1968-01-09 00:00:00	2004-03-04 00:00:00	923 7 S...

Refresh Save Cancel Export data 200 8 ... 59 row(s) fetched - 4ms (4ms fetch), on 2023-10-09 at 12:19:20

Customer 2 (2)

SELECT CustomerId, FirstName, LastNi ... Enter a SQL expression to filter results (use Ctrl+Space)

	CustomerId	FirstName	LastName	Company	Address	City
1	1	Luís	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José de
2	2	Leonie	Köhler	[NULL]	Theodor-Heuss-Straße 34	Stuttgart
3	3	François	Tremblay	[NULL]	1498 rue Bélanger	Montréal
4	4	Bjørn	Hansen	[NULL]	Ullevålsveien 14	Oslo
5	5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague
6	6	Helena	Holý	[NULL]	Rilská 3174/6	Prague
7	7	Astrid	Gruber	[NULL]	Rotenturmstraße 4, 1010 Innere	Vienne
8	8	Daan	Peeters	[NULL]	Grétrystraat 63	Brussels
9	9	Kara	Nielsen	[NULL]	Sønder Boulevard 51	Copenhagen
10	10	Eduardo	Martins	Woodstock Discos	Rua Dr. Falcão Filho, 155	São Paulo
11	11	Alexandre	Rocha	Banco do Brasil S.A.	Av. Paulista, 2022	São Paulo

Refresh Save Cancel Export data 200 59 ... 59 row(s) fetched

To use the Multiple results feature in the SQL Editor, follow these steps:



1. **Add Toolbar Item:** Add the **Toggle execution result orientation**  to the SQL Editor toolbar.

Alternatively:

- Right-click in the SQL Editor window and navigate to **Execute -> Toggle execution result orientation**.
- Navigate to **SQL Editor -> Toggle execution result orientation**.

2. **Toggle:** Click the toolbar item to activate multiple query results view.

3. **View results:**

- To view multiple results in a single tab, execute queries using the **Execute SQL Script** button .
- You can collapse some result sets according to your preference by clicking the arrow button .

Important: When using two or more queries in parallel, exercise caution as this may lead to client UI freeze, high database server load, or transaction deadlock.

Additional features

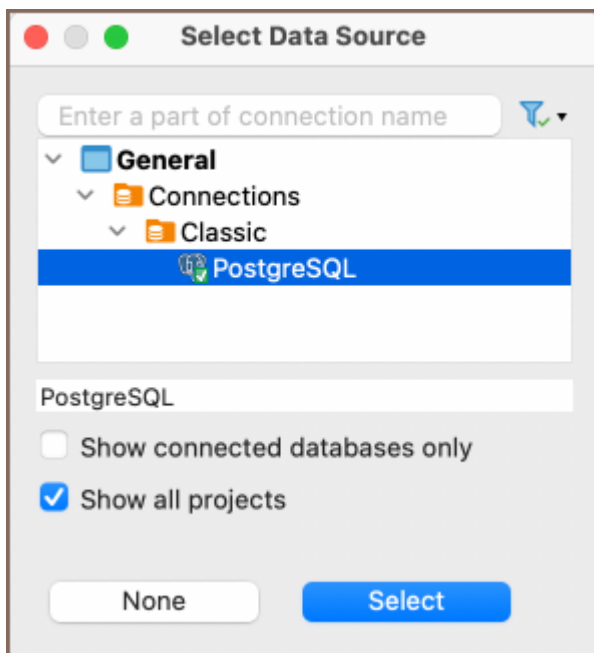
Active Database

You can change the connection associated with the current SQL editor or change the active database/schema, at the same time retaining the SQL text.

To change the connection, press `Ctrl+9` or click the **Active datasource** box on DBeaver's main toolbar:



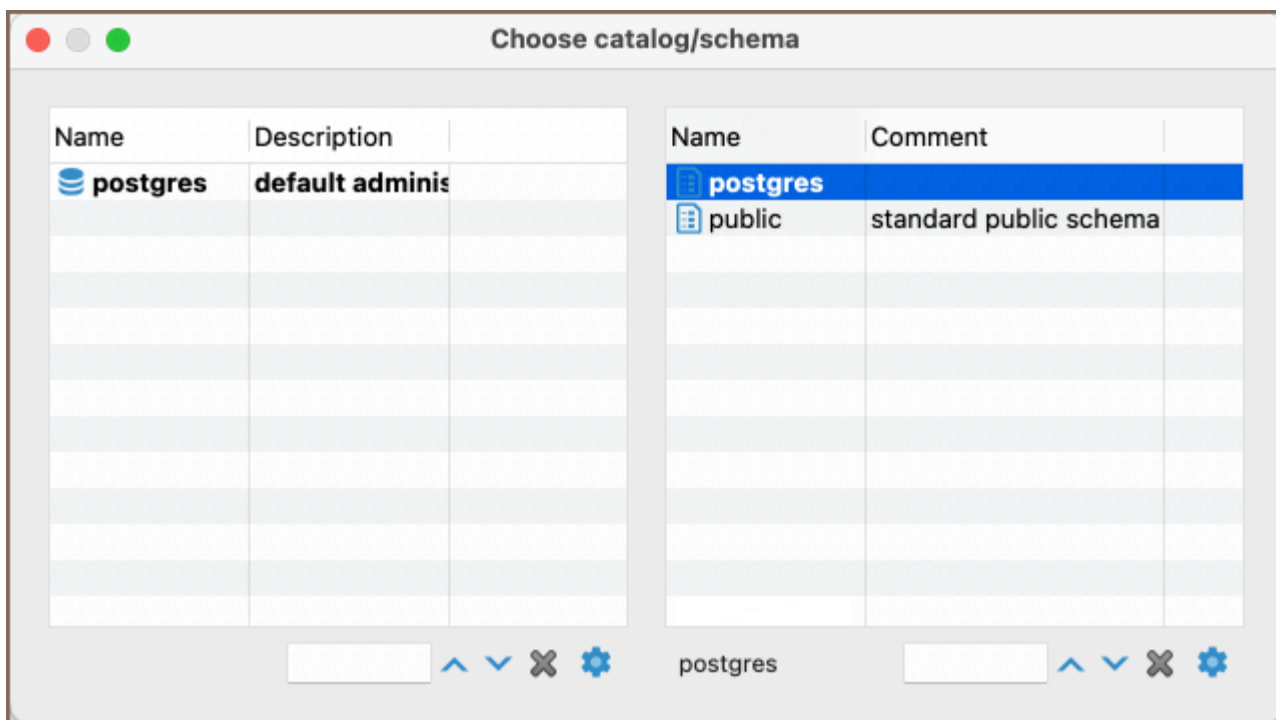
The Select Data Source dialog box opens. In the tree of connections, click the required connection and then click **Select**. To disassociate the SQL Editor with any connection, click **None**:





To change the active schema, press `Ctrl+0` or click the **Active Catalog/Schema** box in DBeaver's main toolbar:




The Choose catalog/schema dialog box opens. In the list of schemas, double-click the required schema:



If there are many schemas, and they do not fit in the dialog box use the search field  to find the schema.

To configure the set of columns to be visible for each schema in the dialog box, click the **Configure columns** button .

Tip: You can easily associate the SQL Editor with the connection that is currently in focus in the Database Navigator (the focus can be on any object of the connection - a table, a folder, etc.) - click the **Link with editor**  or use the shortcut `Ctrl+Shift+,` .

Layout Adjustment

You can modify the layout of the SQL Editor by showing/hiding the results panel and changing the horizontal /vertical position of the panes.

- To toggle (hide/show) the results panel, press `CTRL+6` or right-click anywhere in the script pane and, on the context menu, click **Layout** -> **Toggle results panel**.

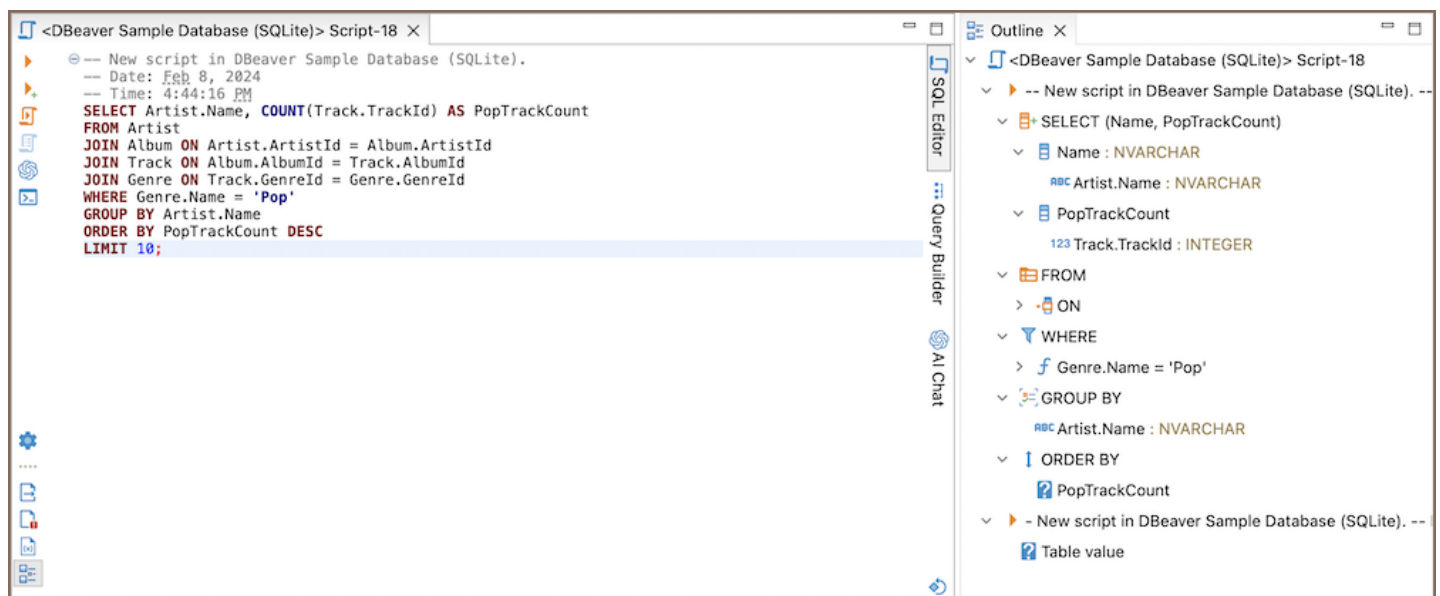
- To maximize the results panel, press **CTRL+Shift+T** , or double-click the results tab name, or right-click anywhere in the script panel and, on the context menu, click **Layout -> Maximize results panel**.
- To switch between the script panel and the results pane, press **Ctrl+Alt+T** or right-click anywhere in the script panel and, on the context menu, click **Layout -> Switch active panel**.

To position both panels horizontally, right-click anywhere in the script panel and, on the context menu, click **Layout -> Horizontal**.


To position both panels vertically, right-click anywhere in the script panel and, on the context menu, click **Layout -> Vertical**.

Outline

You can use the Outline feature in the SQL Editor to get a structured view of your SQL query.



Here's how you can access the Outline:

- Use the shortcut: **Alt+Shift+Q** , 0 (on macOS use **##Q 0**).
- Click on the **Toggle Outline** button  in the SQL Editor bottom toolbar.
- Select **Panels -> Toggle outline** from the [Script panel](#).
- Select **SQL Editor -> Panels -> Toggle outline** from [Menu bar](#).

When you open the Outline, it shows a tree structure of your SQL query. This representation reflects the components of your SQL query and is beneficial for analyzing and moving through large queries. The relationship between the Outline and SQL Editor is interactive:

- The tree selection in the Outline automatically follows the cursor movement in the SQL Editor. This synchronization helps you identify the specific section of the query you are editing within its overall structure.
- Alternatively, clicking an item in the Outline tree will highlight the corresponding fragment in the SQL Editor, facilitating swift navigation to different parts of your SQL query.

Features summary

The SQL Editor offers a variety of advanced features to enhance your scripting experience.

Feature	Description	More Info
SQL Templates	Pre-defined SQL code snippets for quick insertion.	SQL Templates
SQL Assist and Auto-Complete	Helps complete SQL queries by suggesting options.	SQL Assist and Auto-Complete
AI SQL Assistance	Uses AI to suggest optimizations and best practices.	AI SQL Assistance (ChatGPT)
SQL Formatting	Automatically formats SQL queries to improve readability.	SQL Formatting
SQL Execution	Provides various options to execute SQL queries.	SQL Execution
SQL Terminal	A terminal interface for executing SQL commands.	SQL Terminal
Variables Panel	Manage variables within your SQL queries.	Variables Panel
Query Execution Plan	Visualize the execution plan of your SQL queries.	Query Execution Plan
Visual Query Builder	A graphical interface for building SQL queries.	Visual Query Builder
Script Management	Manage multiple SQL scripts within a single editor.	Script Management
Client-Side Commands	Execute client-side commands within the editor.	Client-Side Commands

SQL Execution

Table of contents

[Result tabs](#)

[Naming](#)

[Pinning](#)

[Detaching](#)

[SQL Expression Evaluation](#)

[Row Count](#)

[Query Export](#)

[Parameters and variables](#)

[Variables binding](#)

[Dynamic parameters binding](#)

[Miscellaneous](#)

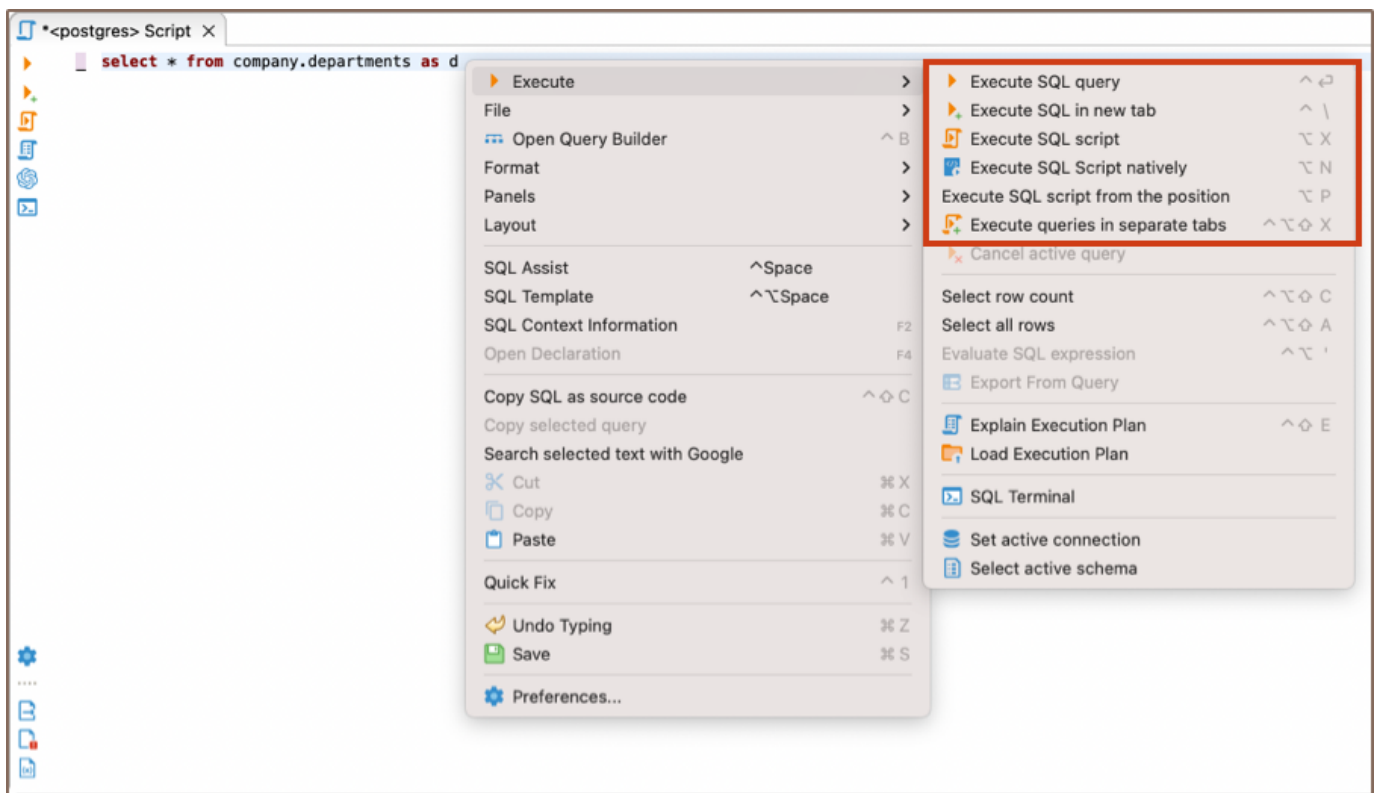
You can execute one query, a highlighted portion of a script, or a whole script. You can execute them using the following:

- Shortcut key combinations (see details further in this article)
- Tools in the main toolbar:

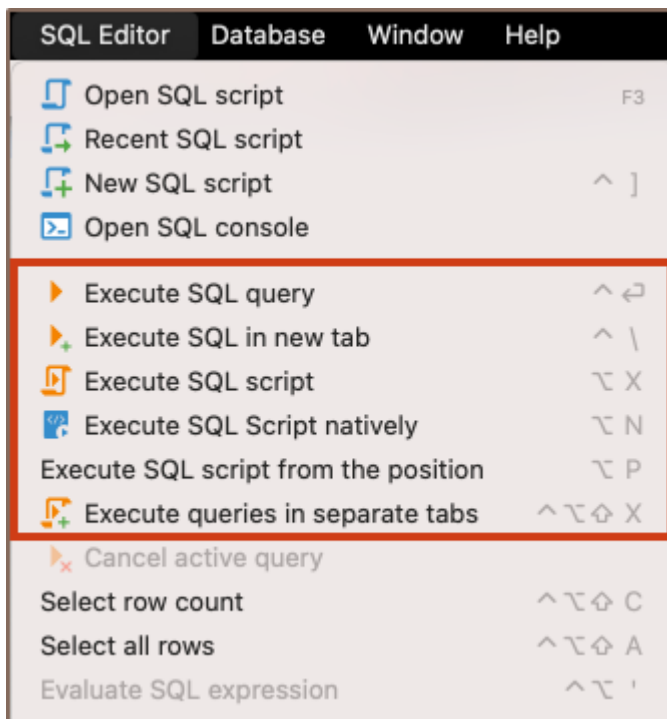


Note: toolbar is customizable. See [Toolbar Customization](#)

- Context menu (right-click the query):



● DBeaver main menu:



To execute a query under the cursor or selected text, press **Ctrl+Enter** or right-click the query and click **Execute** -> **Execute SQL Statement** on the context menu. You can do the same using the main toolbar or main menu: **SQL Editor** -> **Execute SQL Statement**. This executes the SQL query under the cursor or selected text and fills the results pane with the query results.

To execute a query under the cursor in a separate tab, press **CTRL+** or right-click the query and click **Execute -> Execute SQL in new tab** on the context menu. The same can be done using the main toolbar or the main menu: **SQL Editor -> Execute SQL in new tab**. This executes the SQL query under the cursor or selected text and creates a new results tab.

To execute the whole script, press **Alt+X** or click **Execute -> Execute SQL Script** on the context menu or **SQL Editor -> Execute SQL Script** on the main menu or in the main toolbar. This executes all queries in the current editor (or selected queries) as a script. DBeaver parses queries one by one using a statement delimiter (";" by default) and executes them consecutively. You can configure the script execution behavior in the SQL editor preferences (Right-click the script and click **Preferences** on the context menu).

To execute the script natively, press **Alt+N** or click **Execute -> Execute SQL Script natively** on the context menu or **SQL Editor -> Execute SQL Script natively** on the main menu or in the main toolbar. Upon activation, a setup wizard is launched, which allows you to configure the parameters for script execution before the script is launched in the native client like PLSQL, MySQL, or SQLPlus. The results are displayed in a text field in the format of console output. It is handy when functions are not supported by DBeaver drivers and require more specialized clients or when the function is weighty, and a faster client is needed.

Note: This function is available for MySQL/Maria, Oracle, and PostgreSQL and may require additional software installation for each database.

To execute a script opening, each query results in a separate tab, press **Ctrl+Alt+Shift+X** or click **Execute -> Execute Statements In Separate Tabs** on the context menu or **SQL Editor -> Execute Statements In Separate Tabs** on the main menu or in the main toolbar. It executes all queries in the script but opens multiple result tabs. Each script query is executed in a separate thread (that is, all queries are executed simultaneously).

Important: Executing a massive script with numerous queries can result in unforeseen problems.

Result tabs

A single query may generate several result sets represented by tabs. These tabs are linked to the query they are executed from.

- To close an individual tab, press **CTRL+Shift+** or middle-click on a tab header.

- To close all tabs except current, click **Close all result tabs except this** on the context menu of this tab.
- To close all tabs of the desired query, click **Close all result tabs of same query** on the context menu of this tab.

Naming

A tab is often named after the primary table of your query. For example, after executing the following query you will see a single tab called `Album` (assuming that your database has a table called `Album`):

```
SELECT * FROM Album;
```

If a query has joins or, in other words, has multiple source tables, a `(+)` is shown right to the table name. The following query will result in a tab called `Album(+)`:

```
SELECT * FROM Album al, Artist ar WHERE al.AlbumId = ar.ArtistId;
```

Additionally, you can change the name of a given tab via its context menu or by using a special comment:

```
-- title: DBeaver is cool  
SELECT * FROM Album;
```

In other cases, tabs are named in the form of `Results <A> ()`, where:

- **A** is an index of query
- **B** is an index of the result set of this query

Pinning

Tabs can be moved around by dragging them with a mouse and pinned using the **Pin tab** on the context menu of the desired tab. Pinned tabs are stacked on the left. They can be moved among other pinned tabs but can't be mixed with unpinned tabs. Pinned tabs cannot be closed without being unpinned first and cannot be overwritten by executing a query (by making this tab active).

Detaching

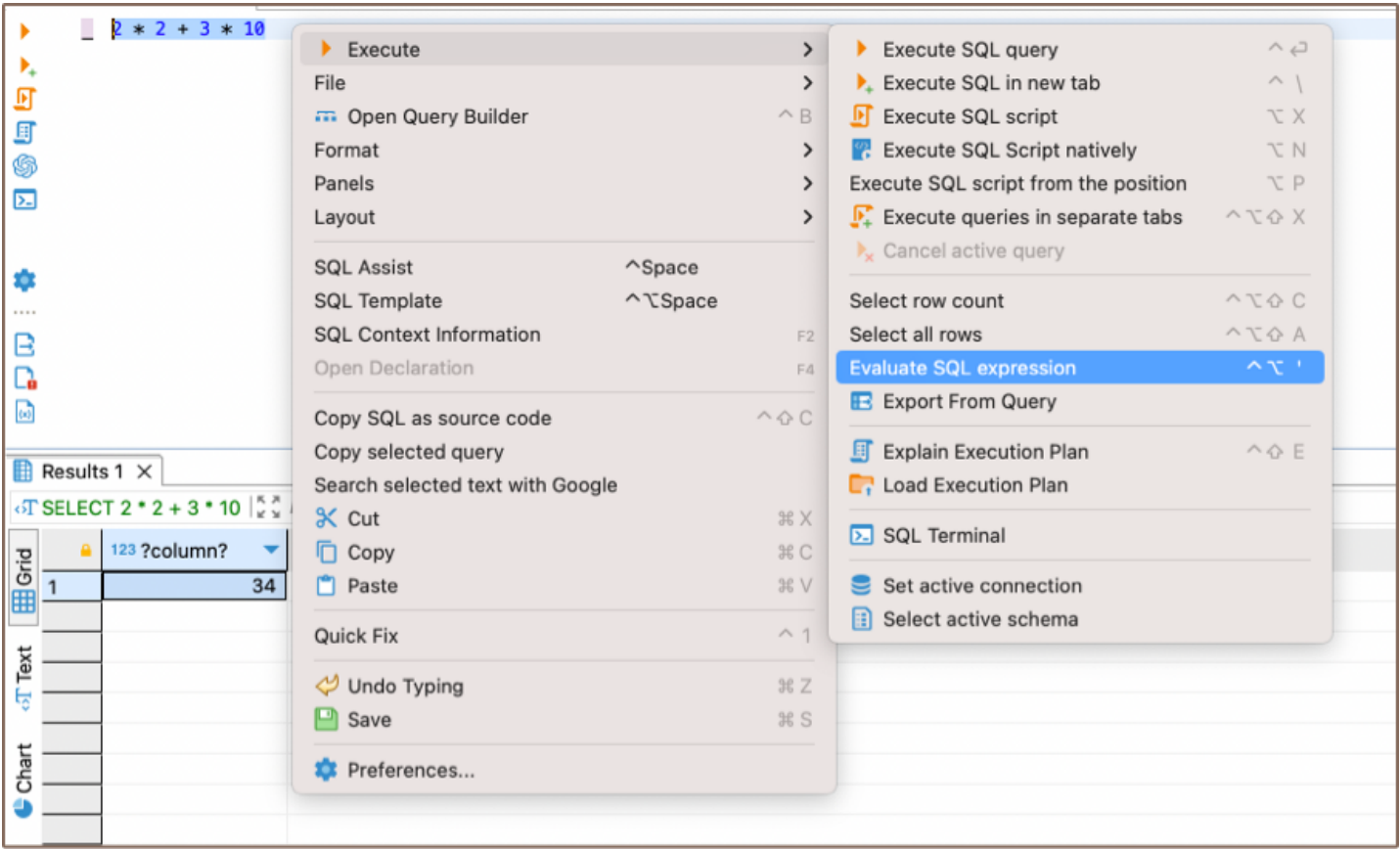
Tabs can be detached from the SQL editor into a separate view using the **Detach Tab** action found in the context menu of the desired tab. After the tab is detached, you can rearrange and move it anywhere you want (for example, you can put two tabs side-by-side for comparison).

Additionally, you can detach it from the application window using **Detach** found in the context menu of an already detached tab.

After the tab is detached, it's still synchronized with the SQL editor, meaning you can edit and refresh data as long as the SQL editor that produced that tab is open. Once you close it, tabs become read-only.

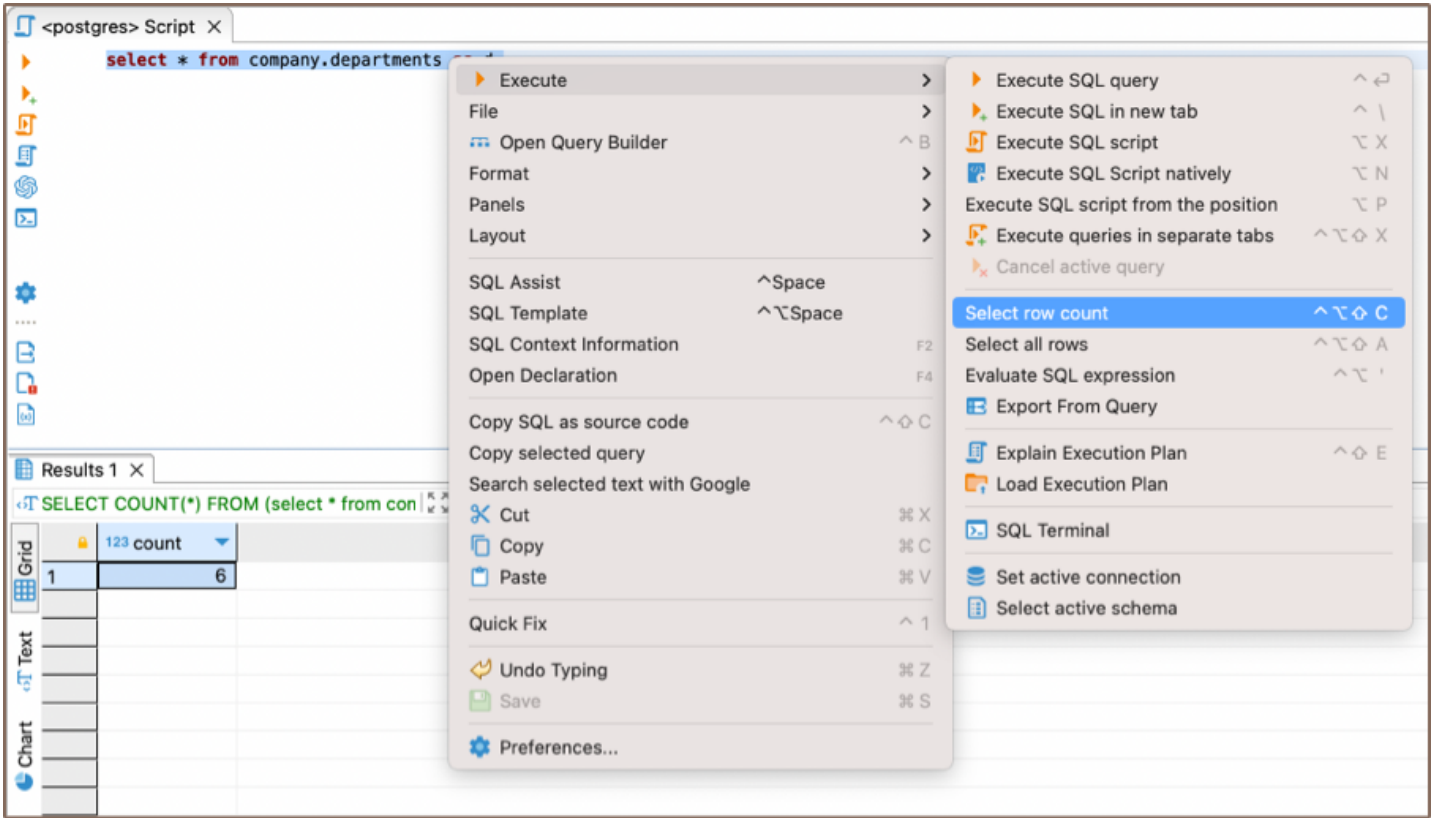
SQL Expression Evaluation

To evaluate an SQL expression, right-click the expression and click **Execute -> Evaluate SQL expression** on the context menu. This command performs a query of **SELECT [expression] FROM DUAL** type:



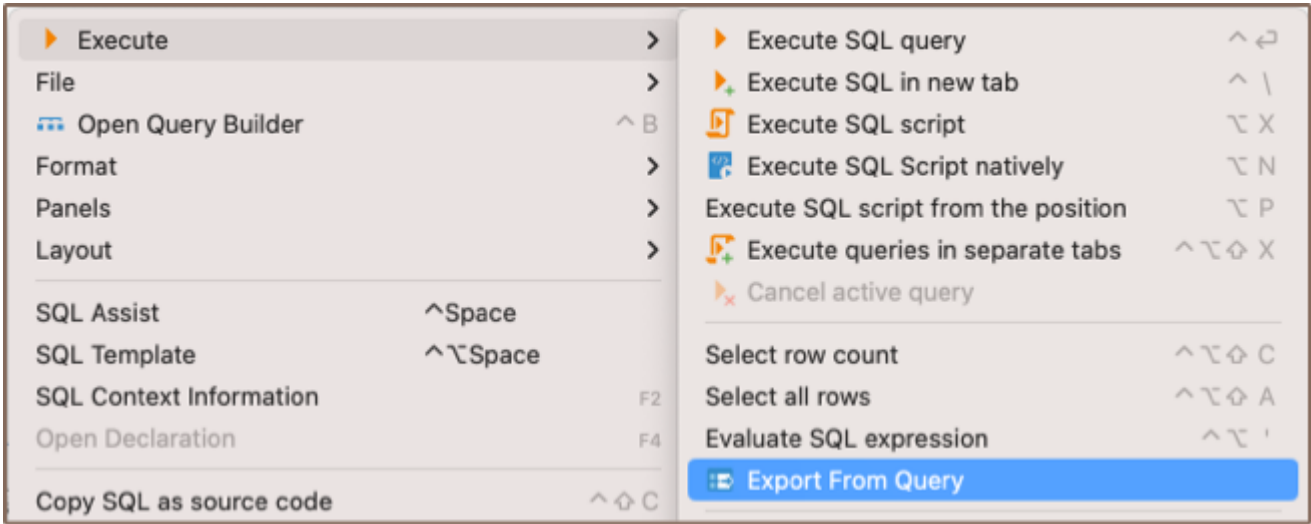
Row Count

If you want to know how many rows an SQL query will produce, you need to apply the Row Count feature – highlight and right-click the SQL text and then click **Execute** -> **Select row count** on the context menu:

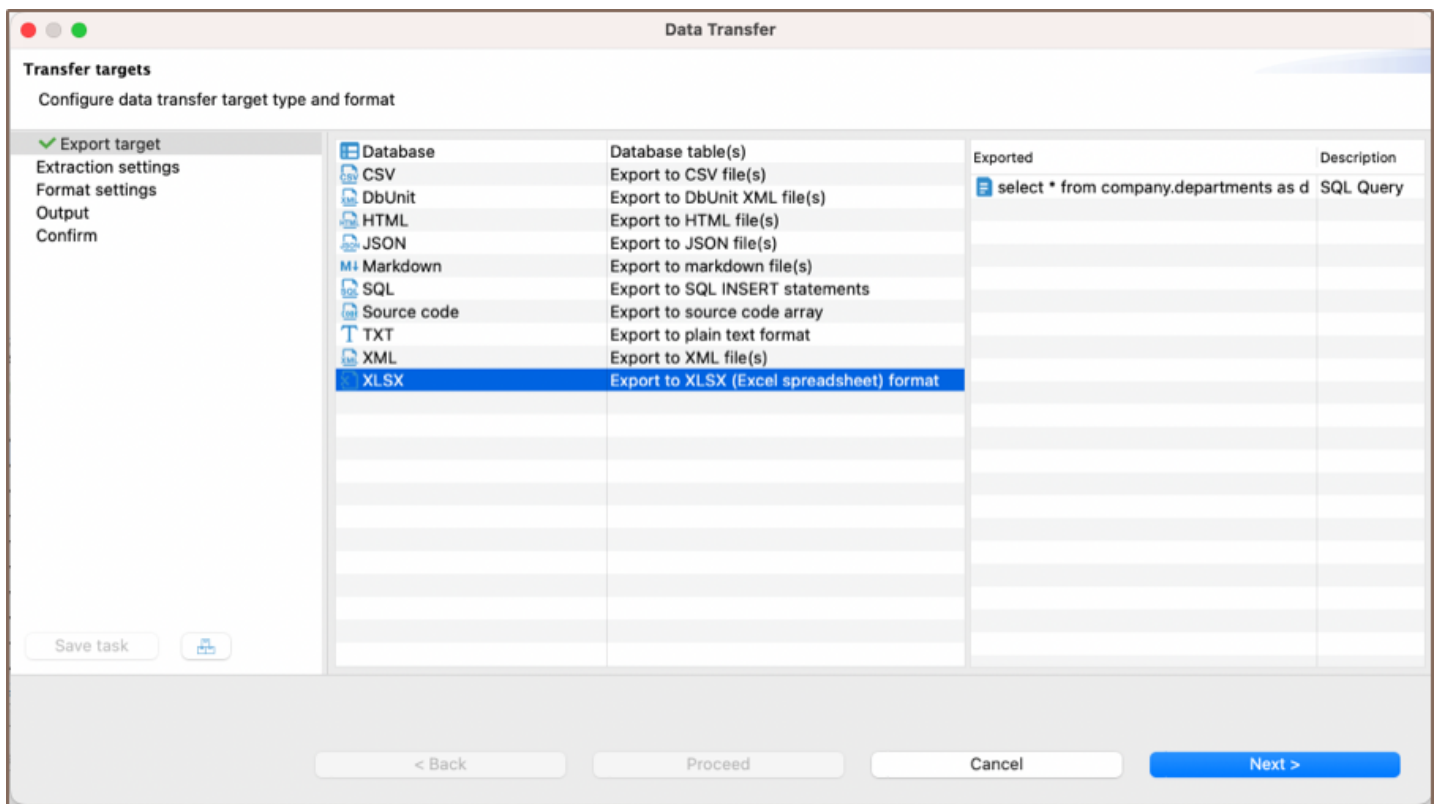


Query Export

It might be useful to export a query if you have a long-running query and you do not need to see its results in the results panel. You can directly export the current query results to a file/table by right-clicking the query and then clicking **Execute** -> **Export From Query** on the context menu:



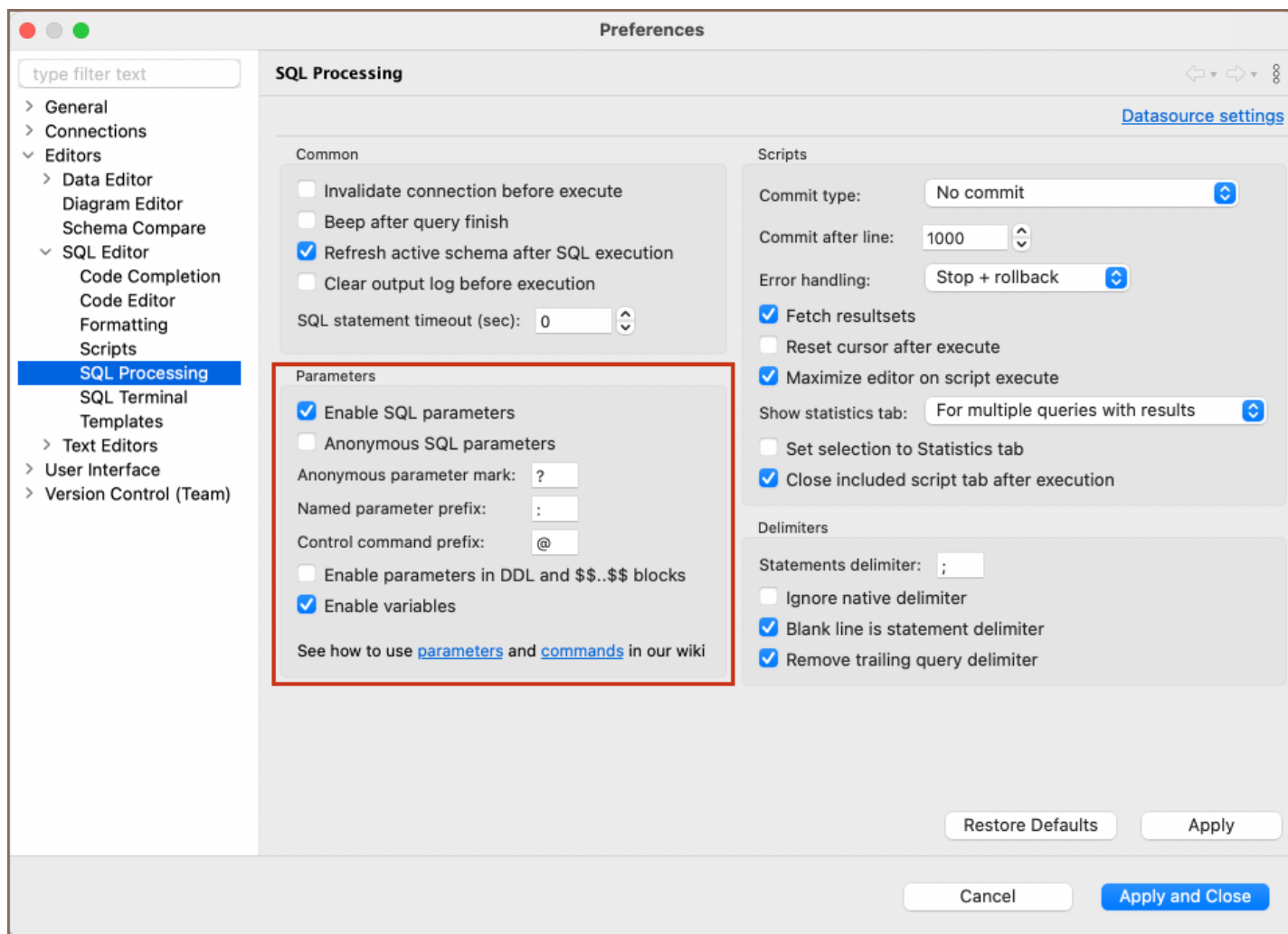
The Data transfer wizard opens. Go through its steps to complete the export of the query.



Parameters and variables

In DBeaver, parameters are used within SQL queries as placeholders, prompting for user input at query execution. Variables, defined with the `@set` command or through the [Variables panel](#), substitute placeholders with predefined values, enabling scripts to run without manual input each time.

To customize parameter and variable behavior in DBeaver, access the settings through **Window -> Preferences -> Editors -> SQL Editor -> SQL Processing**.



Setting	Description
Enable SQL parameters	Allows the use of named parameters within SQL queries.
Anonymous SQL parameters	Permits the usage of unnamed parameters, enabling the SQL editor to recognize placeholders denoted by the character specified in the Anonymous parameter mark field.
Anonymous parameter mark	Sets the symbol for anonymous parameters (default <code>?</code>).
Named parameter prefix	Defines the prefix for named parameters (default <code>:</code>).
Control command prefix	Specifies the prefix for control commands like <code>@set</code> (default <code>@</code>).
Enable parameters in DDL and <code>\$\$..\$\$</code> blocks	Permits the use of parameters within DDL statements and <code>\$\$..\$\$</code> code blocks.
Enable variables	Activates variable substitution within SQL scripts.

For advanced SQL execution, DBeaver extends functionality with context variables that come from:

- **CLI:** Command-line interface variables, which offer a hands-off approach for setting up environments. See the [Command-line guide](#) for usage details.
- **Auth properties:** Authentication-related variables that enhance security for credential handling, detailed in the [Authentication properties documentation](#).

Variables binding

Define the custom variables with the predefined command or add in the [Variables panel](#).

You can create a variable by using the `@set` command followed by the variable name and its value. For instance:

```
@set actor_name = Mark
```

Once defined, you can include variables in your SQL queries. Use the `${varname}` or `:varname` syntax to insert the value of a variable into the script. For example:

```
SELECT * FROM public.actor WHERE first_name = '${actor_name}';
```

Dynamic parameters binding

You can define parameters using the `:parameter` syntax, turning parts of the query into placeholders for values input at execution. For instance:

```
SELECT :first_parameter FROM public.actor WHERE first_name = :second_parameter;
```

Upon every execution of the parameterized query, DBeaver brings up the **Bind parameter(s)** window.

Field	Description
Value	Fill in values for each parameter to ensure correct query execution.
Hide parameters set in script	If checked, parameters that you have already given values with the <code>@set</code> command will not show up.

Tip: To simplify the process, consider using [variables](#) for predefined values before running the script.

Miscellaneous

- To select the current query row count, press `Ctrl+Alt+Shift+C` .
- To open the definition of the database object currently in focus (under cursor) in a viewer/editor, press `F4` .

Query execution plan

Table of contents

[Simple plan view](#)

[Advanced plan view](#)


This feature is supported for the following data sources:

The Execution plan for databases marked with a star is supported only in Lite, Enterprise and Ultimate editions

- MySQL
- PostgreSQL
- Microsoft SQL Server ★
- Oracle
- DB2 LUW
- SAP HANA
- Google Cloud SQL for PostgreSQL ★
- Google Cloud SQL for SQL Server ★
- Google Cloud SQL for MySQL ★
- Couchbase ★
- Firebird
- Exasol
- HSQLDB
- Vertica ★
- ClickHouse ★

- AlloyDB ★
- Netezza ★
- OceanBase
- Ocient

Simple plan view

If a database driver supports the visualization of the execution plan, you can see the execution plan of the current query (under cursor) by pressing `Ctrl+Shift+E` or clicking **Explain execution plan** on the context menu or in the SQL Editor toolbar:  (Note: toolbar is customizable. See [Toolbar Customization](#)) The execution plan command generates a tree of query execution as one of the result tabs and is convenient in estimating if the query /script is quick/optimal enough:

film

Exec. Plan

Log



Output

Node Type	Entity	Cost	Rows	Time	Condition	Name	Value
Hash Join		87.25 - 336...	5462	7.889		General	
Hash Join		77.50 - 231...	5462	3.855		Node Type	Seq Scan
Seq Scan	film_act...	0.00 - 84.62	5462	1.027		Entity	film as f
Hash		65.00 - 65.00	1000	0.593		Cost	0.00 - 65.00
Seq Scan	film as f	0.00 - 65.00	1000	0.341		Rows	1000
Hash		6.00 - 6.00	300	0.171		Time	0.341
Seq Scan	actor as a	0.00 - 6.00	300	0.075		Condition	
						Source	
						Parent-Relationship	Outer
						Parallel-Aware	false
						Relation-Name	film
						Alias	f
						Plan-Rows	1000
						Plan-Width	19
						Actual-Startup-Time	0.002
						Actual-Total-Time	0.341
						Actual-Rows	1000
						Actual-Loops	1

```
select f.title,a.first_name || ' ' || a.last_name
from film f, film_actor fa, actor a
where f.film_id=fa.film_id and fa.actor_id=a.actor_id
```

Type: Seq Scan; Rel: film as f; Cost: 0.00 - 65.00

You can click the rows of the execution plan to see their details (statistics) in the panels below and to the right of the plan.

To reevaluate the plan, click the **Reevaluate** button (). To see the source script on which the plan is based, click the **View Source** button (.

Advanced plan view

Script management

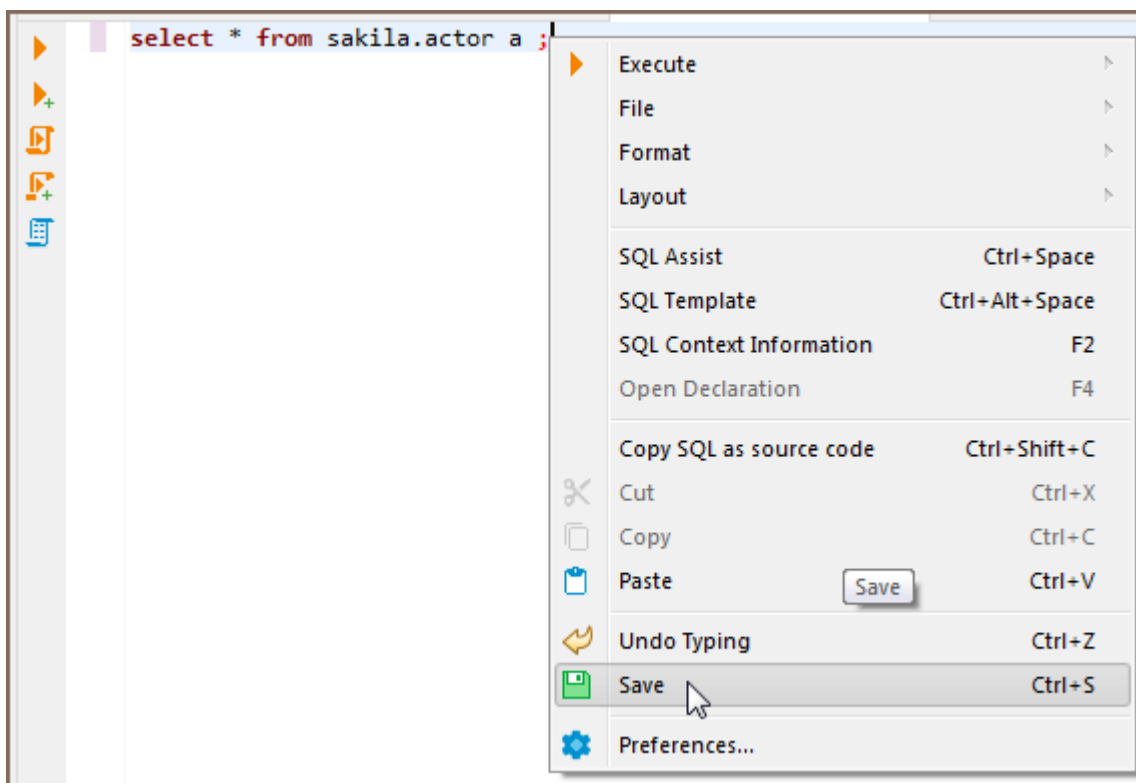
Table of contents

- [Saving Scripts](#)
- [Loading Scripts](#)
- [Renaming Scripts](#)
- [Reverting Changes](#)
- [Changing default scripts directory](#)
- [Adding external directory](#)
- [SQL Console](#)

Saving Scripts

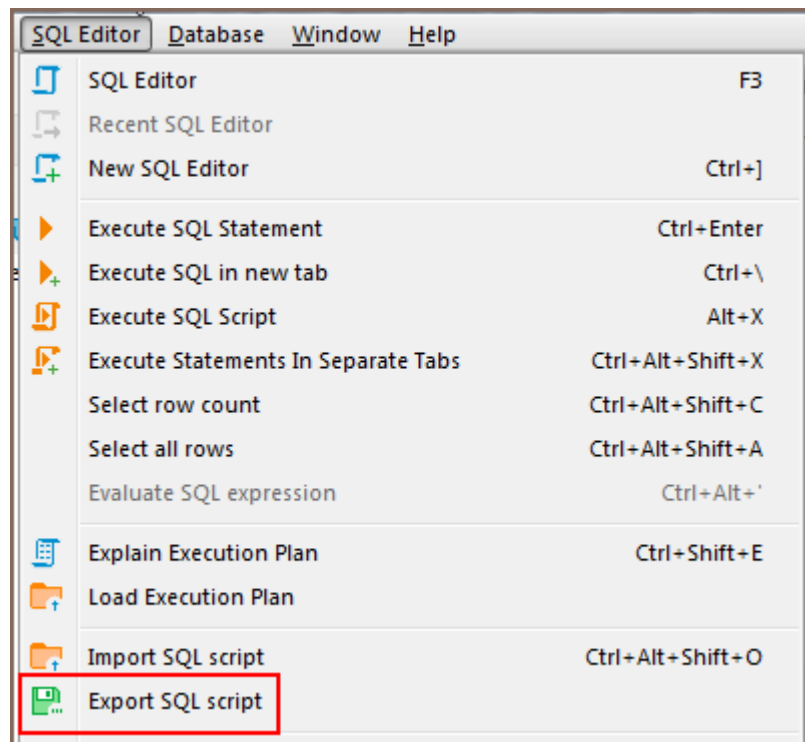
You can save scripts to a predefined space in the currently active project or somewhere in the file system.

To save a script to the current project space, just press `Ctrl+S` or right-click the script and click **Save** on the context menu:

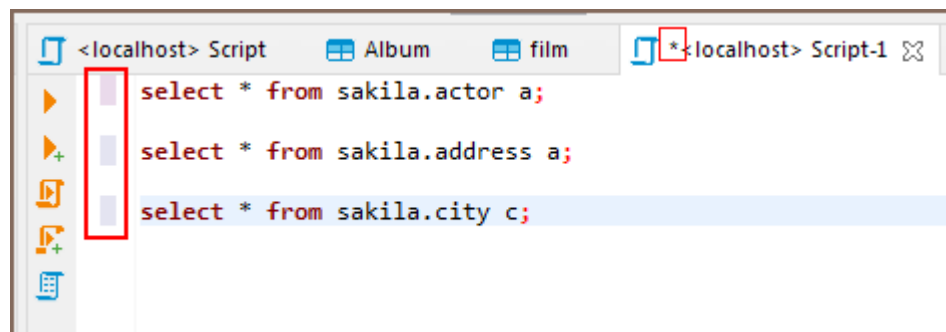


You can find the script saved this way in the [Project Explorer](#) view in the **Scripts** folder.

To save a script to the file system, right-click the script, click **File -> Export SQL script** on the context menu and then select the folder in the file system. You can also click **SQL Editor -> Export SQL script** on the main menu:

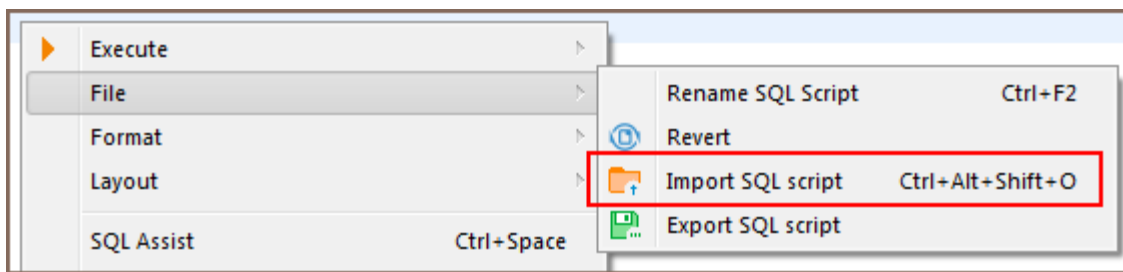


Unsaved data is highlighted in color on the left side of the editor, in addition to having an asterisk in the name of the script.



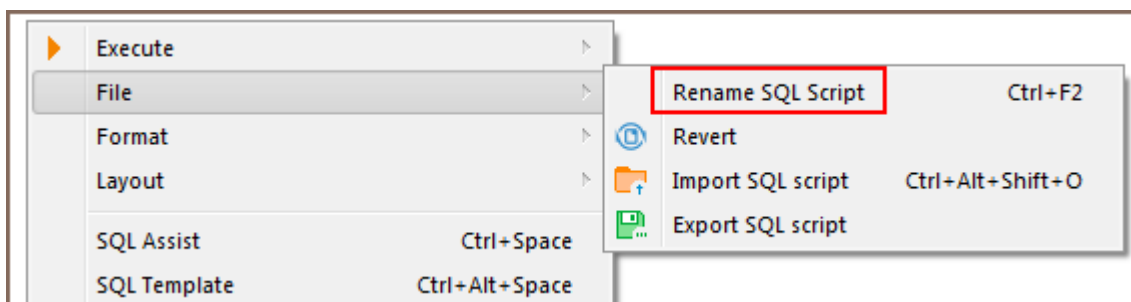
Loading Scripts

To load a script stored in the file system to the SQL Editor, press **CTRL+SHIFT+O**, or click **SQL Editor -> Import SQL script** on the main menu, or right-click the script panel and click **File -> Import SQL script** on the context menu:

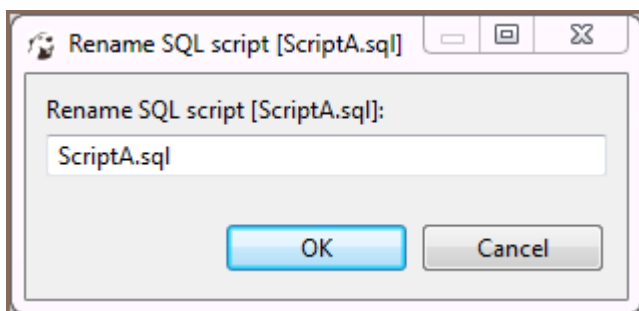


Renaming Scripts

To rename a script, right-click anywhere in the script panel, click **File -> Rename SQL Script** on the context menu or press **CTRL+F2** :



Then enter the new name in the Rename SQL script dialog box and click **OK**:

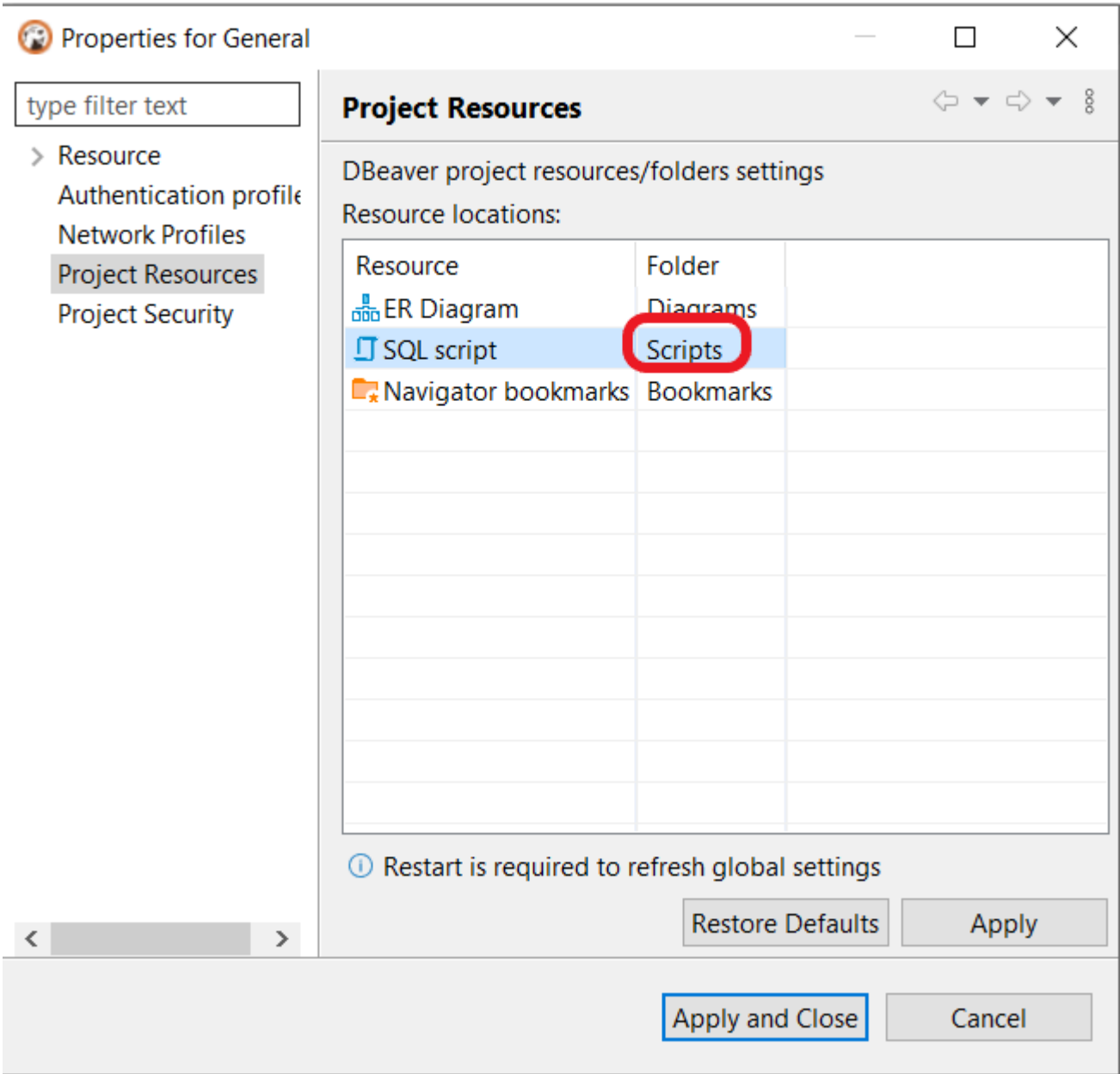
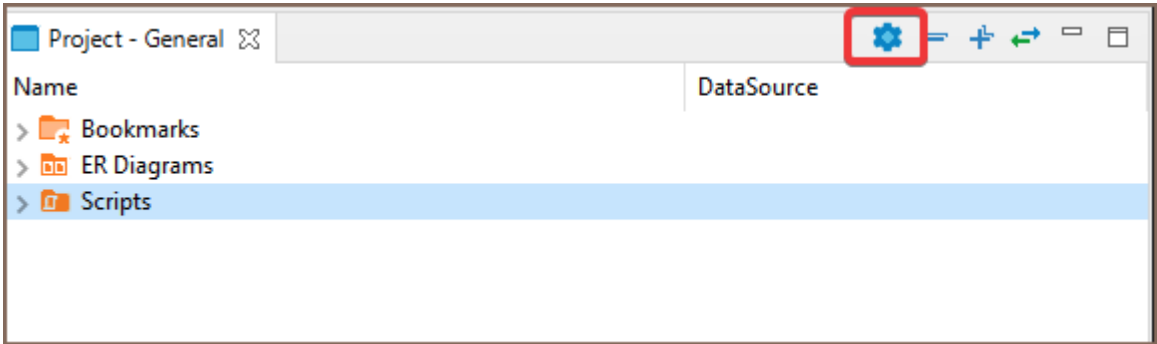


Reverting Changes

If you want to revert all changes made to the current SQL script and return it to its initial state (reload from disk), right-click anywhere in the script panel and click **File -> Revert** on the context menu.

Changing default scripts directory

By default all of the scripts are saved to a "Scripts" folder located in your project inside the workspace directory. This can be changed by clicking the **Configure** button in [Project Explorer](#) view. There you can click on a folder's name and pick any other folder inside the Project.



Adding external directory

You can also link an external directory to your project to either save your scripts into it, or to access scripts that were created outside of DBeaver.

To link an External directory right-click anywhere in the Project Explorer and pick **Create -> Link Folder**. There you can link any directory on your drive to a project. This will allow you to open any externally created scripts through Project explorer and to set this folder as default to save new scripts into.

SQL Console

In some cases you might want to execute a query and not save it in a script. For example, when you read table data using "Read data in SQL console" or open procedure/function source from DDL editor. SQL console does not have an associated .sql file. Its contents will be lost when you close it.

SQL Assist and Auto Complete

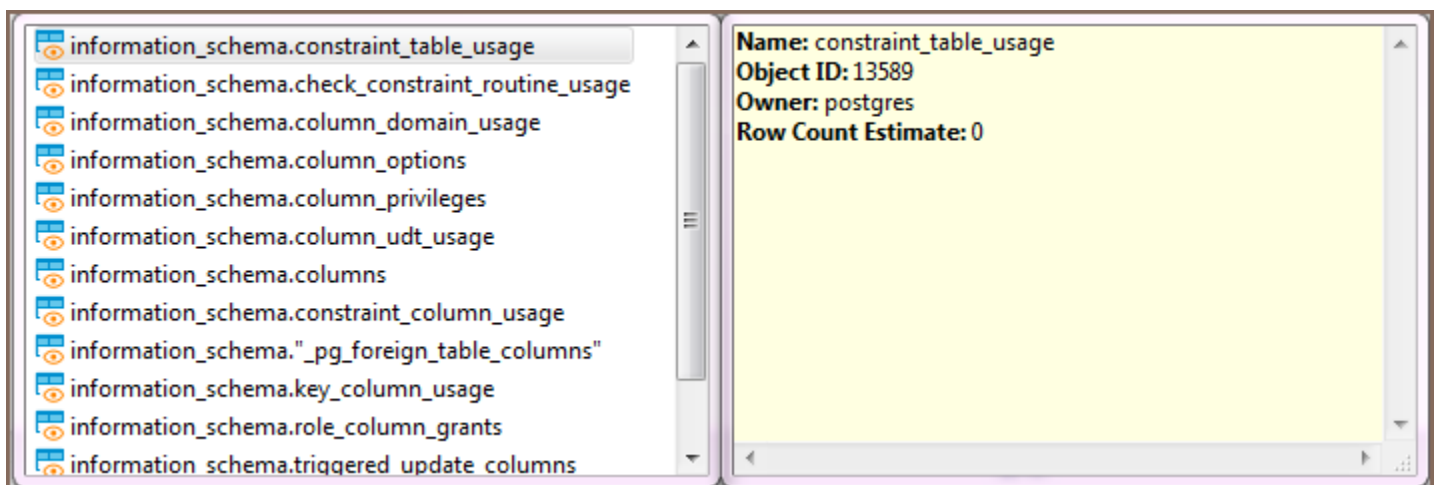
Table of contents

[Hippie Engine](#)

[Customize](#)

The SQL Assist feature provides auto-completion of database object names and SQL commands, and other keywords in queries.

To perform some object name auto-complete, press `Ctrl+Space` or right-click the desired location in the query and click **SQL Assist** on the context menu. DBeaver searches for objects in a database by their names and/or descriptions.



When you start typing an SQL keyword in a statement, DBeaver offers auto-complete options as well.

Another auto-complete function is that it searches for the last entered identifier - type the first letter and press `Ctrl`.

You can also press `Ctrl+Space` after the asterisk in the query similar to `SELECT * FROM tableName` or similar to `INSERT INTO tableName (*)` (brackets are important) (you can use `()[]{}` brackets) - the asterisk will be replaced with a list of all the table columns.

Hippie Engine

Hippie Engine provides autocompletion based on information from the current script file. It scans files looking for words and adds proposals based on similarity with the given string.

It can be toggled on or off in the [Code Completion](#) preferences.

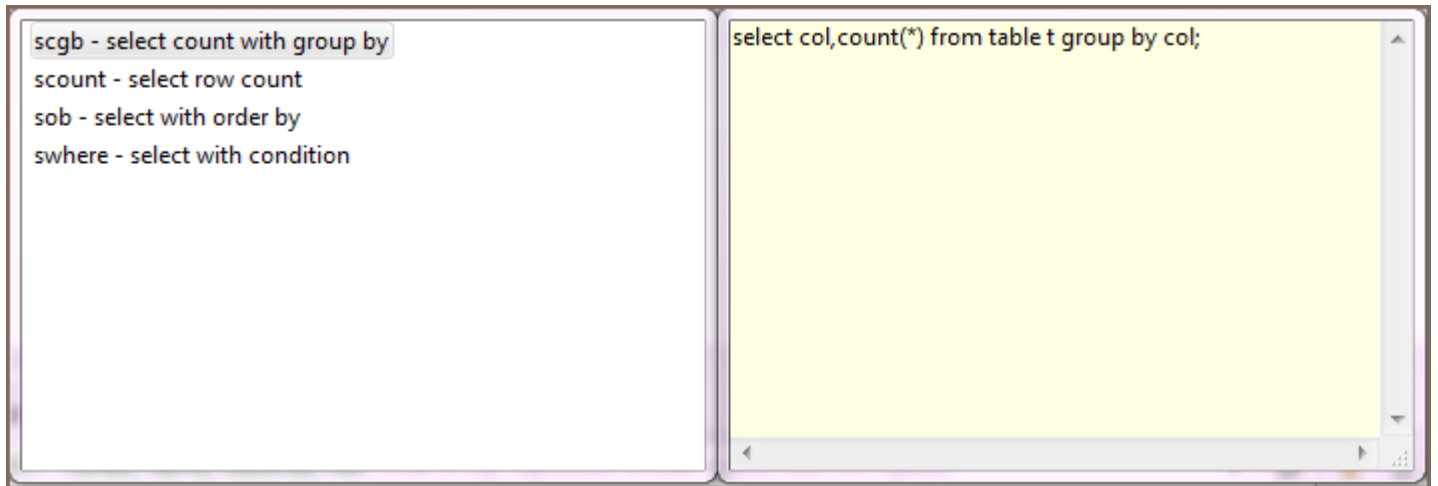
Customize

You can customize your SQL Assist and Auto Complete by navigating to **Window -> Preferences -> Editor -> SQL Editor -> Code Completion**, optimizing the performance of the auto-completion feature.

SQL templates

Templates allow you to insert frequently used SQL statements into an SQL script.

To see available templates, press `Ctrl+Alt+SPACE` or right-click the line in the script pane and click **SQL Template** on the context menu. A box with a list of available templates appears:



To apply a template, in the SQL Editor, in the script pane:

- Type the template name and press `Tab`
- Right-click the line where you want to insert a template expression, click **SQL Template** on the context menu, and then, in the list of templates, double-click the required template name.

The template SQL statement appears in the script.

To edit/add/remove templates, click Configure Icon in the bottom toolbar, then click **Preferences -> SQL Editor -> Templates**. For more information about managing templates, please visit [Eclipse Website](https://www.eclipse.org/).

Standard Eclipse templates:

Variable	Description
<code>\${cursor}</code>	Specifies the cursor position when the template edit mode is left. This is useful when the cursor should jump to a different place than to the end of the template upon leaving the template edit mode.
<code>\${year}</code>	Takes the current year value
<code>\${date}</code>	Takes the current date value
<code>\${time}</code>	Takes the current time value

<code>\${dollar}</code>	Takes the dollar sign \$. Alternatively, two dollar signs can be used: \$\$.
<code>\${user}</code>	Takes the user name
<code>\${word_selection}</code>	Takes the content of the current text selection
<code>\${line_selection}</code>	Takes content of all currently selected lines

DBeaver-specific templates:

Variable	Description
<code>\${schema}</code>	Takes the current schema name
<code>\${catalog}</code>	Takes the catalog name
<code>\${table}</code>	Takes the current table name (from the active catalog/schema)
<code>\${column}</code>	Takes the column name (from the current table)

Visual Query builder

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Opening Visual Query Builder](#)

[Creating Visual Query](#)

[Adjusting Query Settings](#)

[Columns](#)

[Conditions](#)

[Joins](#)

[Sorting](#)

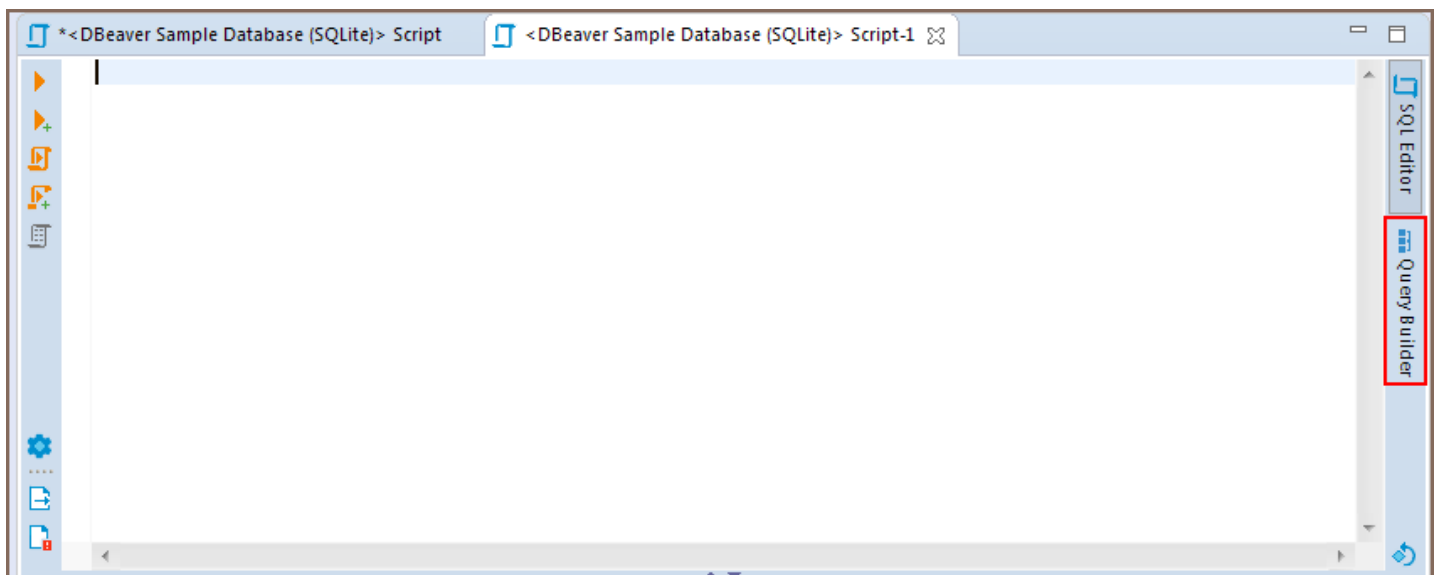
[Miscellaneous](#)

[Executing Visual Query](#)

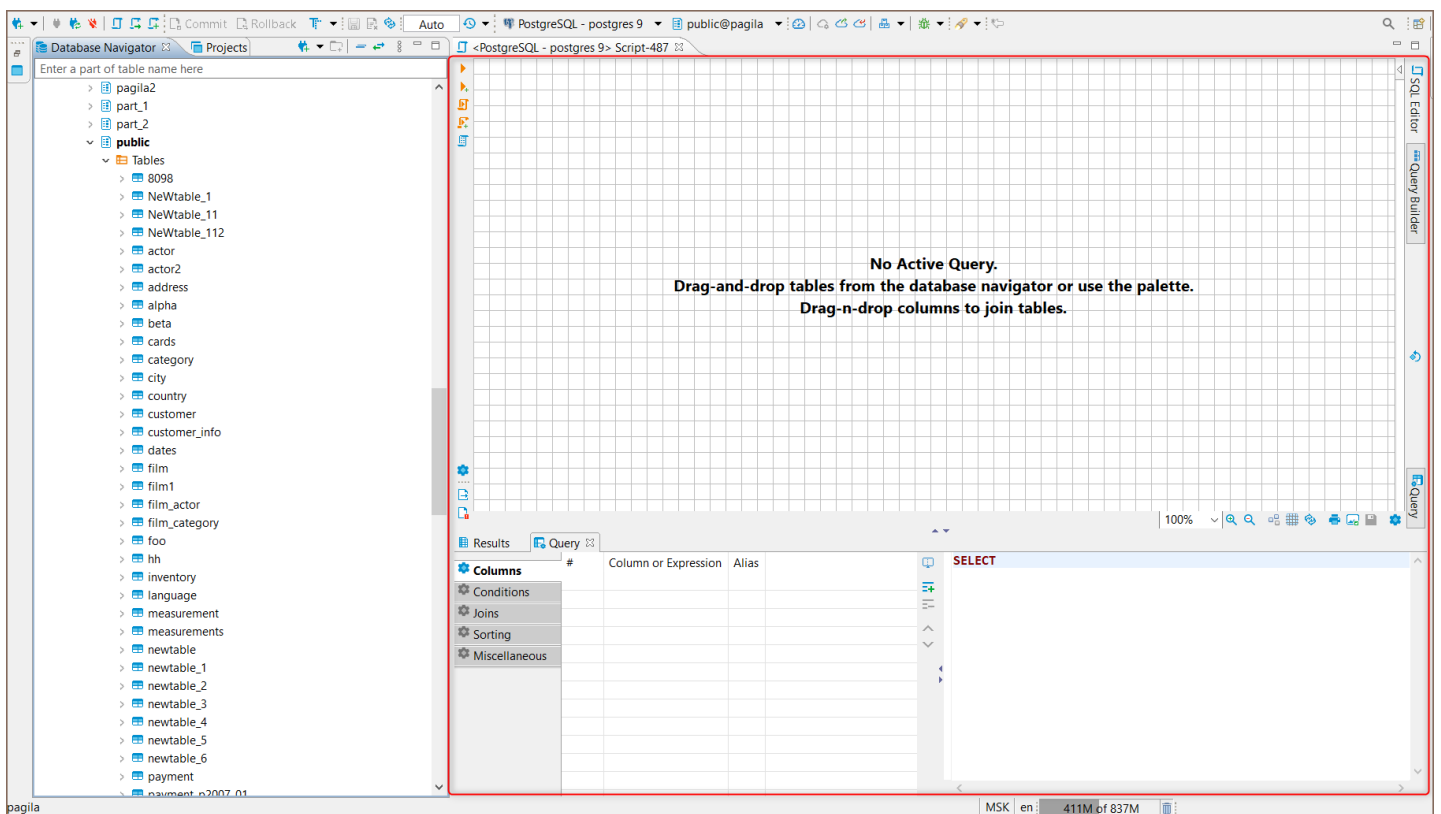
Query Builder is a user-friendly visualization tool that will help you make sense of your complex database designs. It can be useful when you need to understand the various relationships between different tables. Also, it can be helpful for those who are not very familiar with SQL scripting or if you do not want to insert script commands manually. The tool creates SQL scripts automatically based on the visual schema you have created.

Opening Visual Query Builder

To open **Visual Query Builder** click the **Open Query Builder** button in the [SQL Editor](#) tool bar.

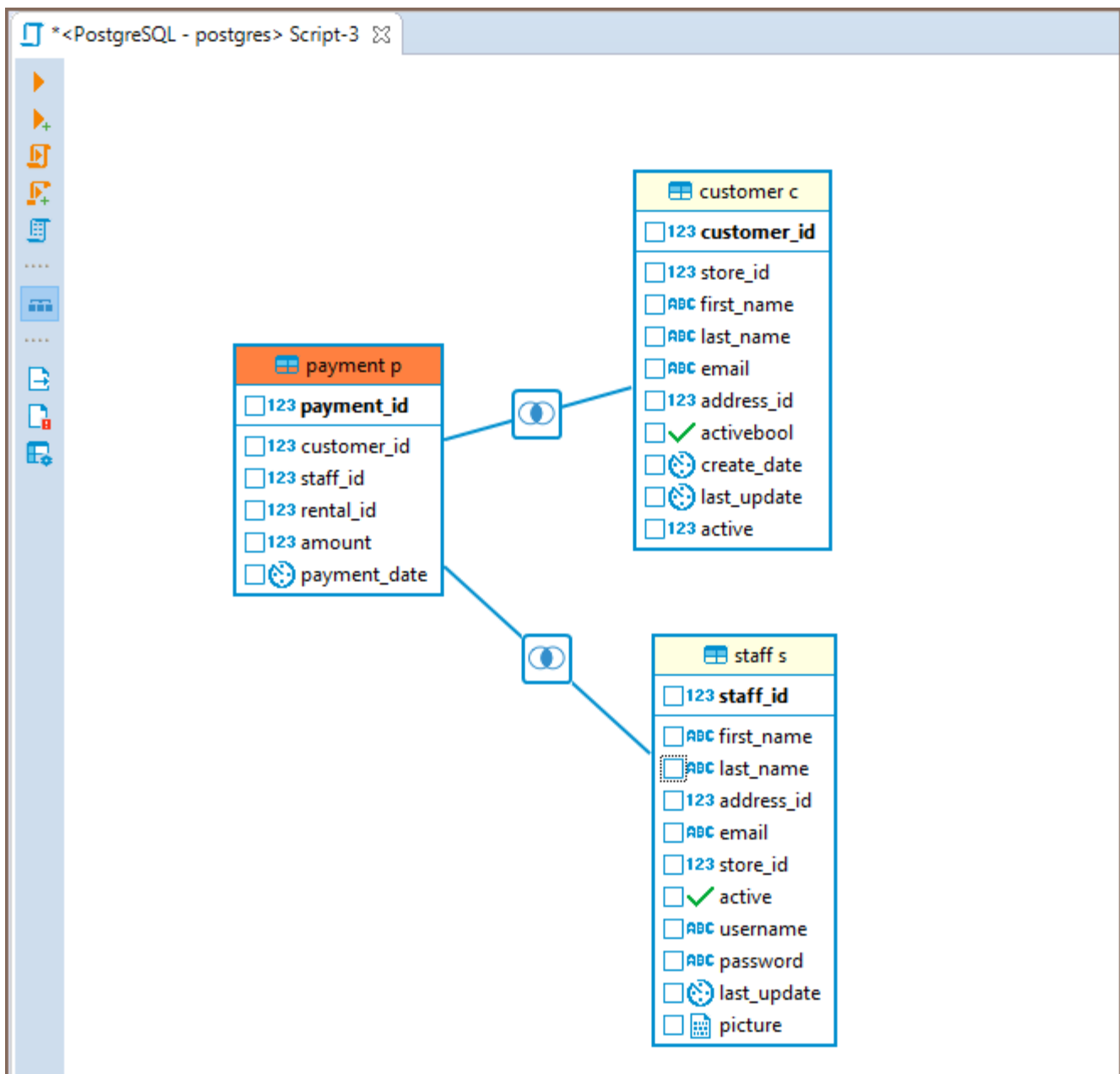


The **Visual Query Builder** will appear on the right.

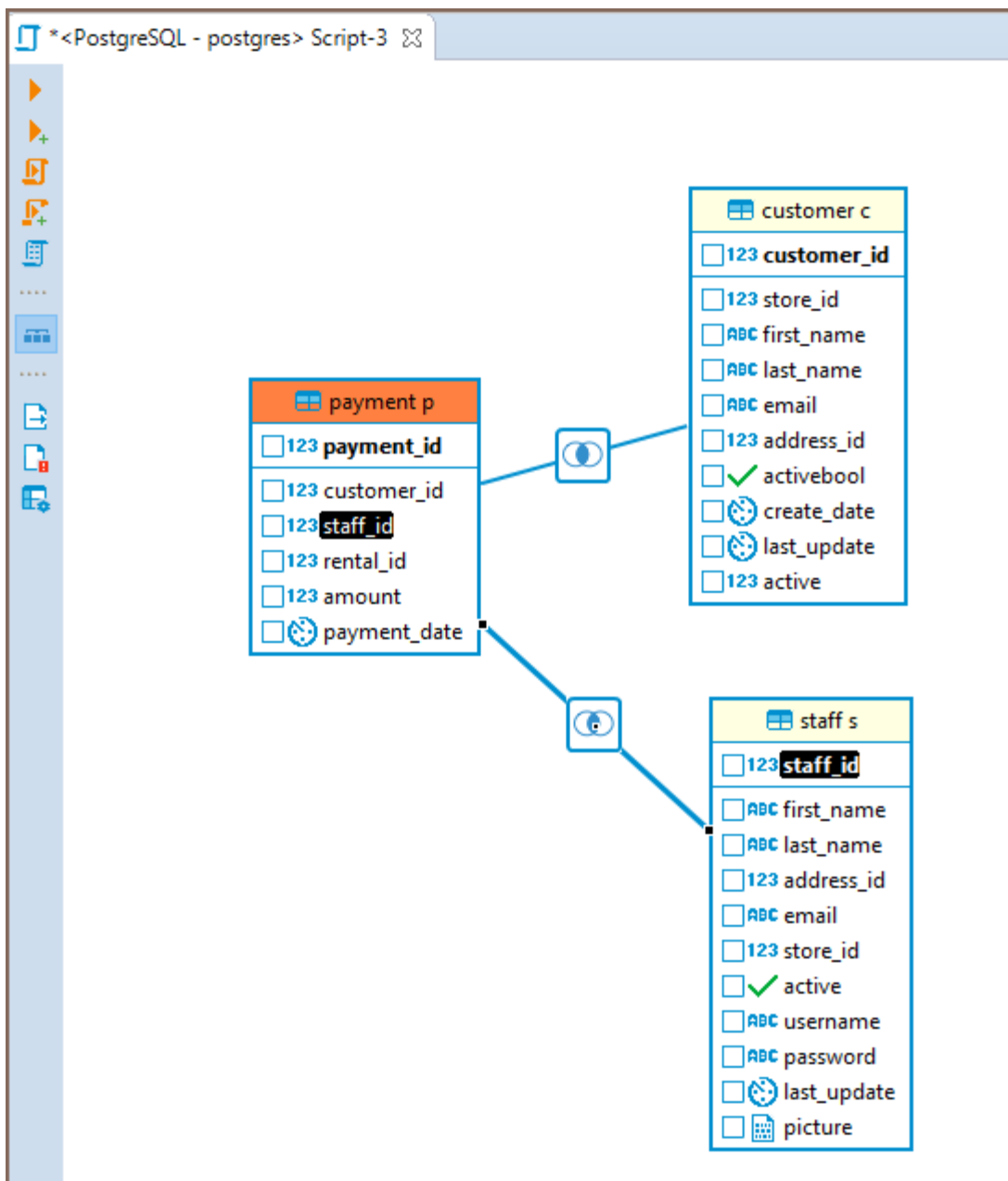


Creating Visual Query

Start creating a query by selecting a query data source: drag-and-drop tables you want to work with from the **Database Navigator** pane into the **Visual Query Builder** area. All the connections existing between the tables will be shown automatically.



To create a new join between the tables, press the left mouse button when the cursor is over the column of one table, holding the right mouse button drag the cursor to the column of another table and release the right mouse button. The connection between the selected columns of the tables will be created visually and in the SQL script a new join will be added.



To remove a join between the tables, click on it. The connection will be highlighted. Then, press **Delete** or use the **Delete** option in the context menu. The visual connection will be removed and the corresponding join will be automatically removed from the SQL script area.

To build a **SELECT** query you need to select columns in the tables you added. To select a column, click the checkbox next to its name - the column will be added to the **Columns** tab of the **Query Settings Editor** and **SELECT** query will be added to the SQL script area automatically.

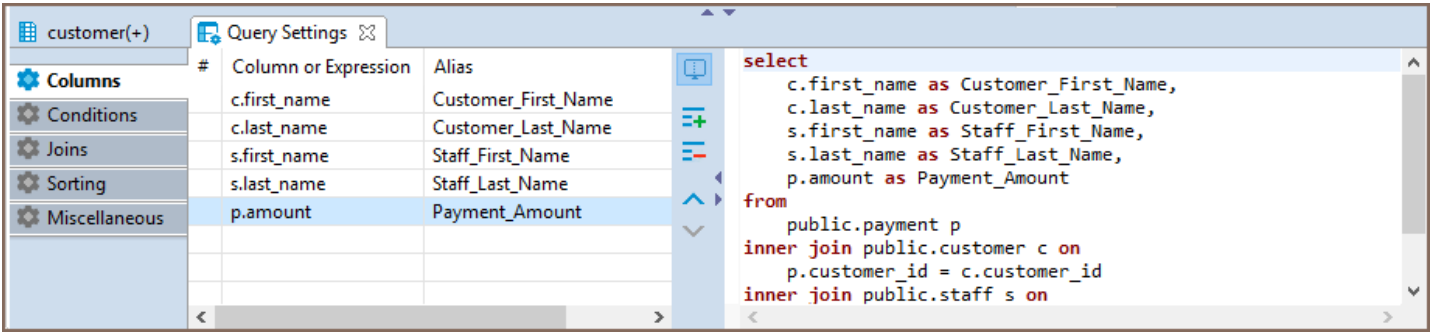
Adjusting Query Settings

Visual Query Builder also allows you to set the query conditions and adjust the representation of query results by means of **Query Settings Editor**.


To open **Query Settings Editor** use **Visual builder query settings** button  in the vertical tool bar on the left.

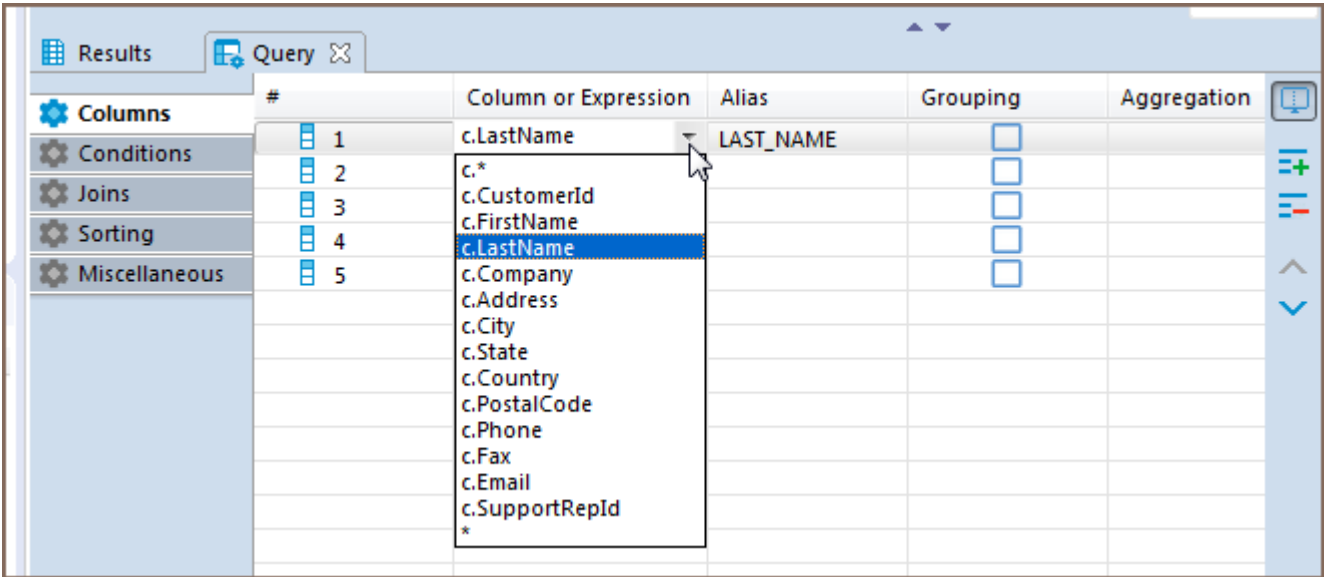
Query Settings window contains five tabs described below.


Columns




Columns tab of the **Query Settings Editor** contains all the columns you added by selecting column names in **Visual Builder** main window. In this tab you can add and remove columns using **Add** and **Remove** buttons correspondingly.

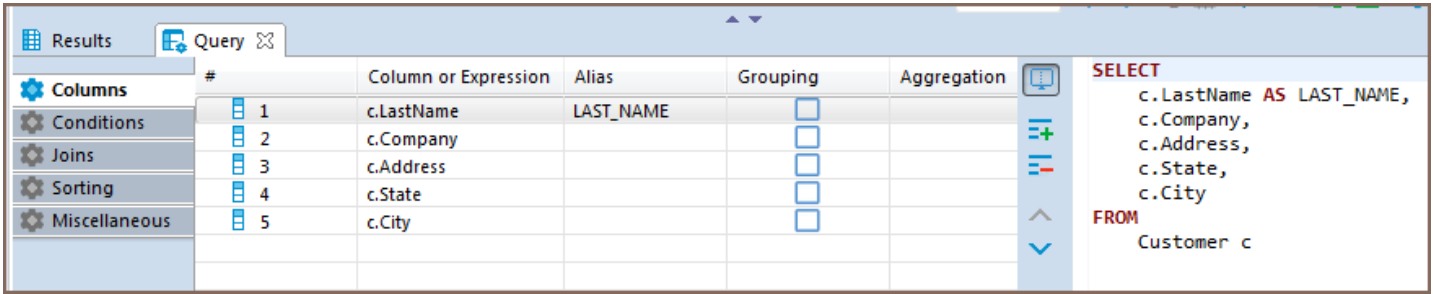
To add a column, press **Add** button  and a new instance will be added to the table. Click on the first cell in **Column or Expression** column and select a column from the list of available columns displayed in the dropdown list appeared.



To remove a column, click on the row containing its name and press the **Remove** button  on the right.

To change the display order of columns in the result table use **Move Up/Down** buttons .

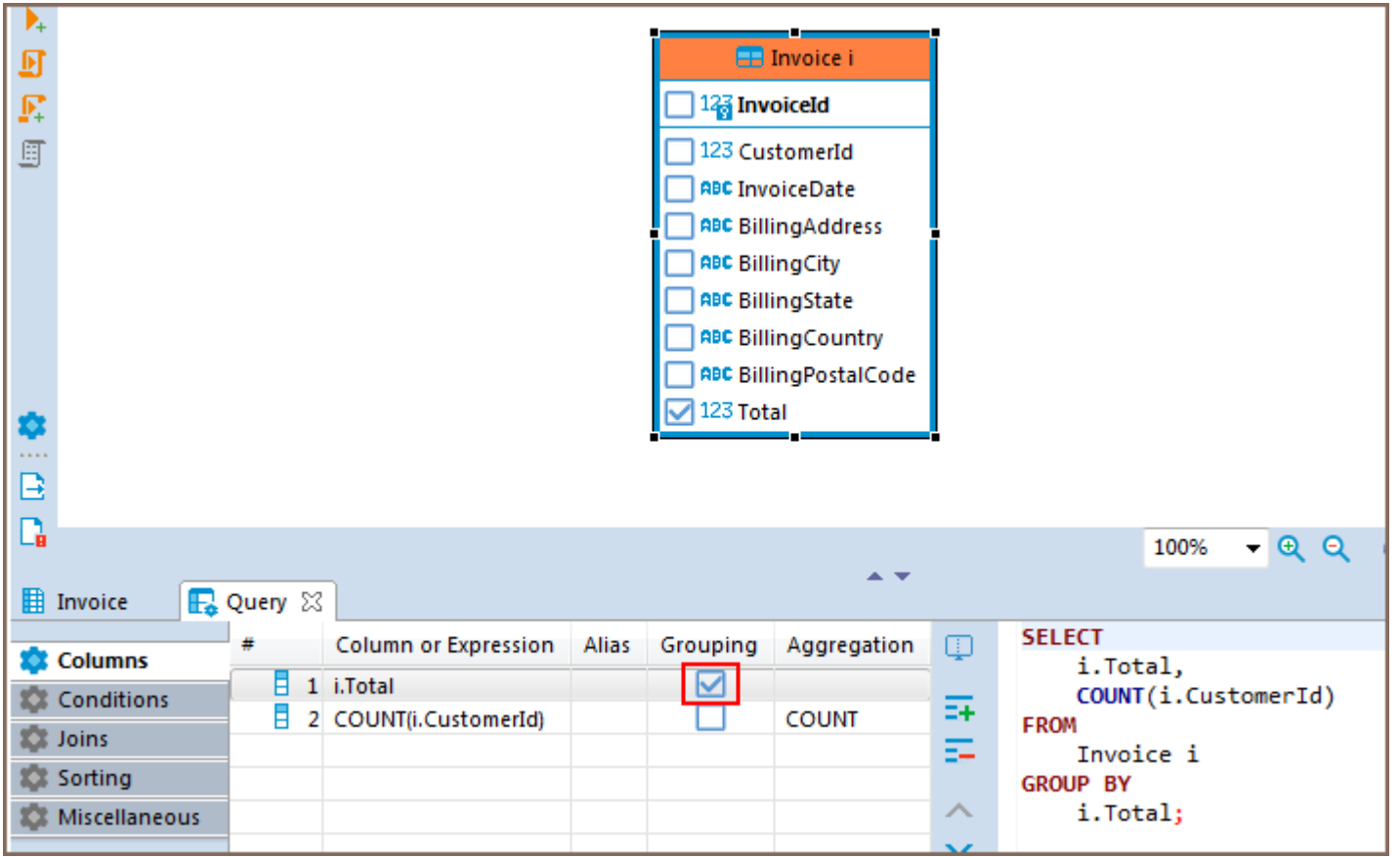
You can also define a user-friendly name of the column to be displayed in the result table. To set a user-friendly name, click on a cell in **Alias** column and insert the name. The change will be immediately displayed in the SQL script area.



#	Column or Expression	Alias	Grouping	Aggregation
1	c.LastName	LAST_NAME	<input type="checkbox"/>	
2	c.Company		<input type="checkbox"/>	
3	c.Address		<input type="checkbox"/>	
4	c.State		<input type="checkbox"/>	
5	c.City		<input type="checkbox"/>	

```
SELECT
  c.LastName AS LAST_NAME,
  c.Company,
  c.Address,
  c.State,
  c.City
FROM
  Customer c
```

If you want a **grouping** condition to be added to your expression, you can click on the checkbox in the column row. The expression will update automatically. The other previously selected columns will become aggregate. If there are no other columns, then the expression COUNT(*) will be automatically added.



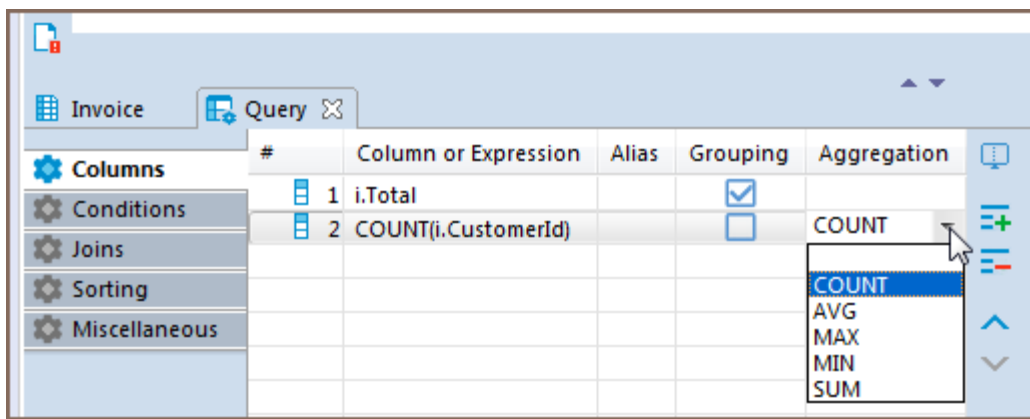
Invoice i

- ☐ 123 InvoiceId
- ☐ 123 CustomerId
- ☐ ABC InvoiceDate
- ☐ ABC BillingAddress
- ☐ ABC BillingCity
- ☐ ABC BillingState
- ☐ ABC BillingCountry
- ☐ ABC BillingPostalCode
- ☒ 123 Total

#	Column or Expression	Alias	Grouping	Aggregation
1	i.Total		<input checked="" type="checkbox"/>	
2	COUNT(i.CustomerId)		<input type="checkbox"/>	COUNT

```
SELECT
  i.Total,
  COUNT(i.CustomerId)
FROM
  Invoice i
GROUP BY
  i.Total;
```

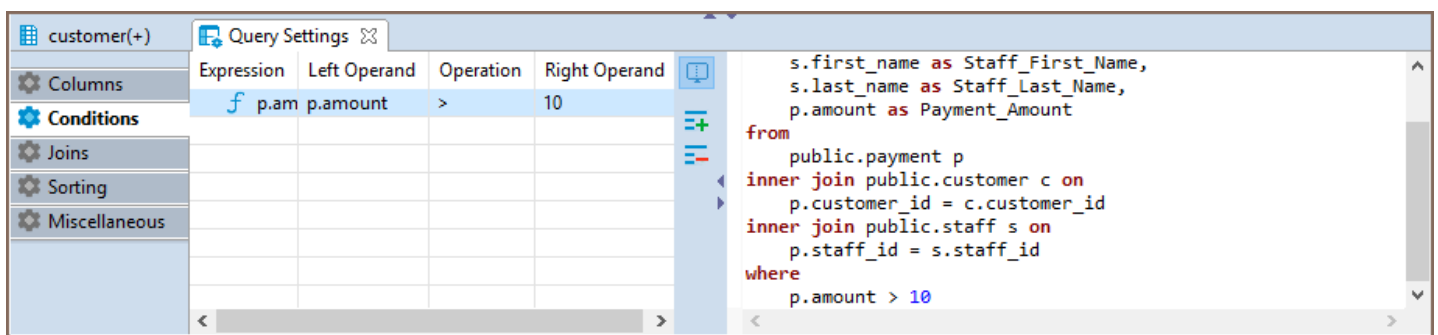
You can select other aggregation functions from the drop-down list. Or enter your own version in the cell.




When removing columns from the list, they will be removed from the grouping expression as well. When adding new columns to the list, it is added to the grouping expression.

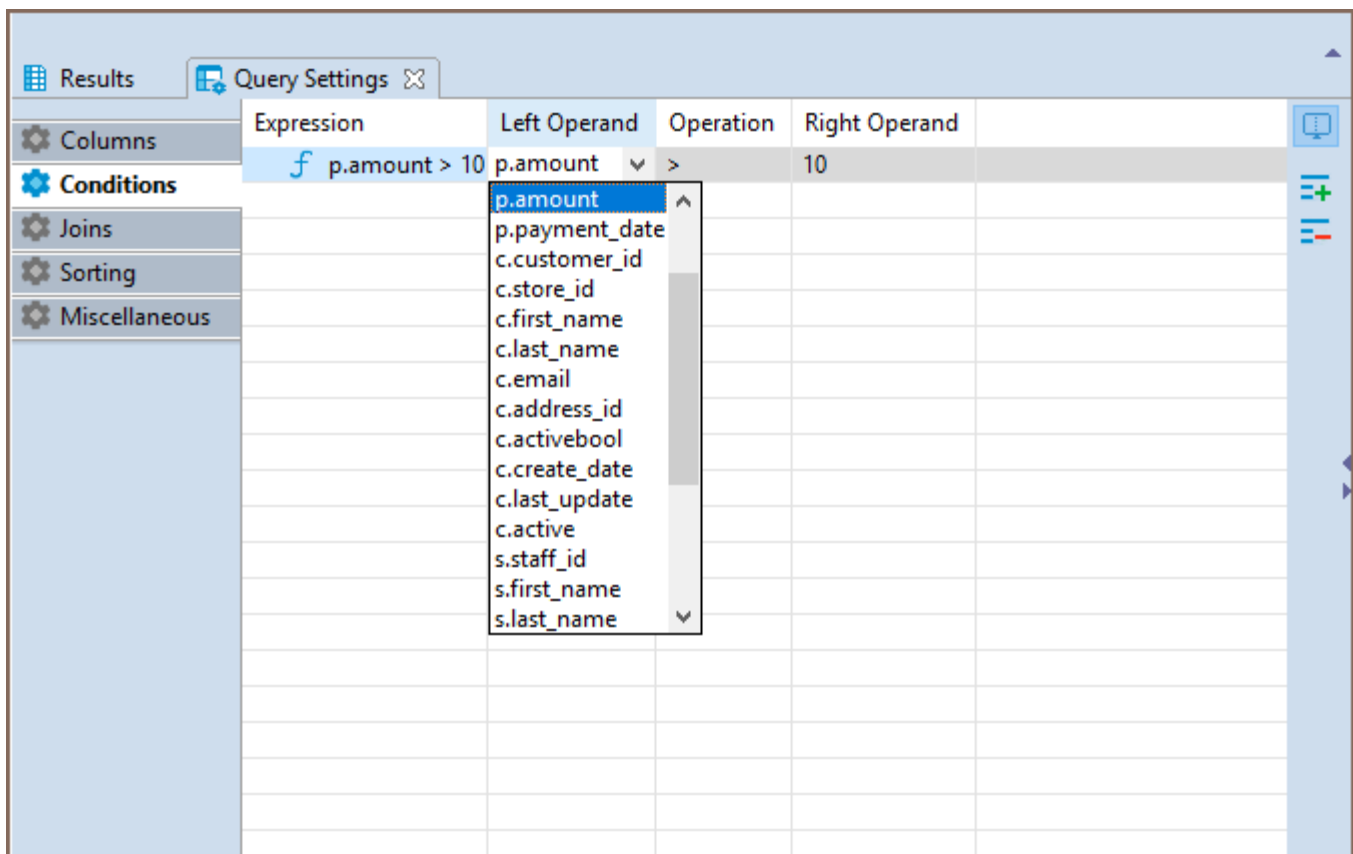
Conditions

The **Conditions** tab is used for managing query conditional expressions.

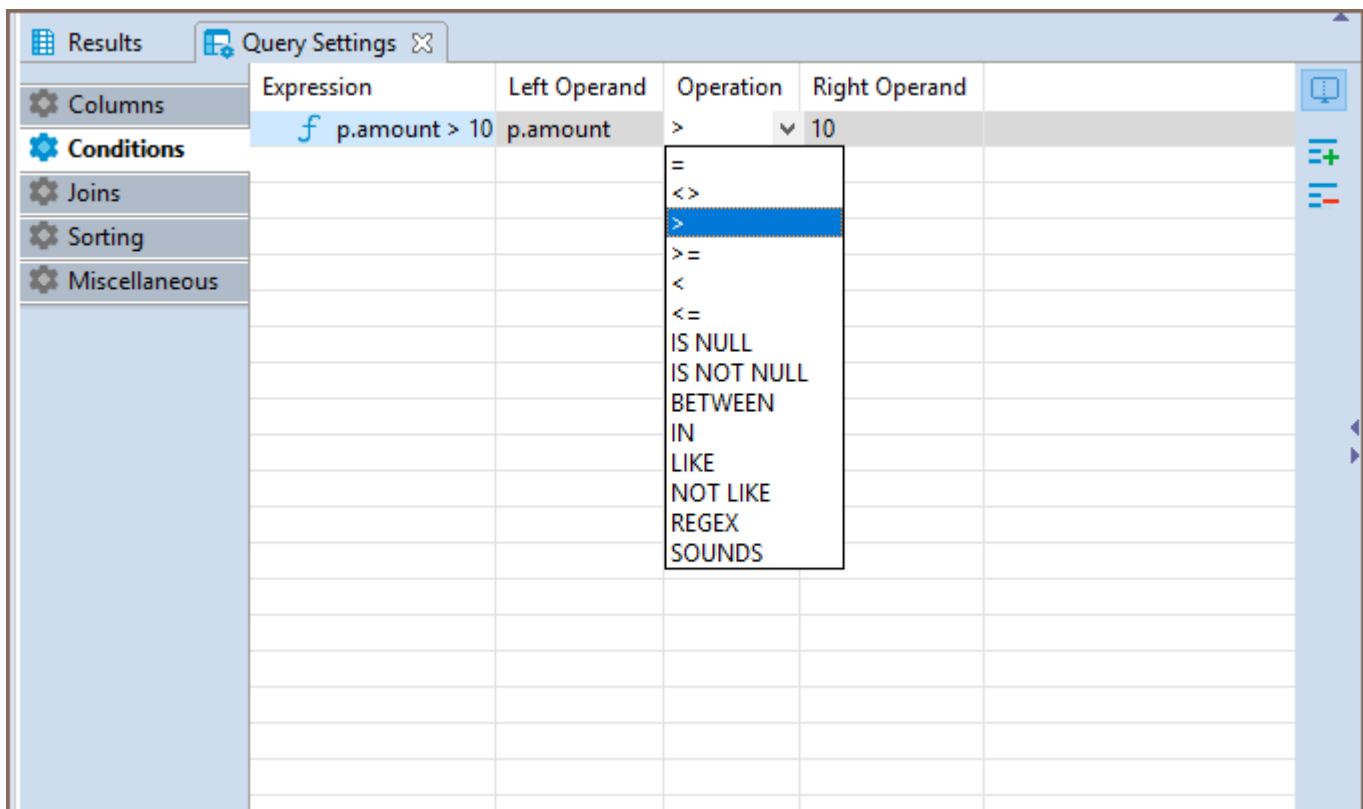


To add a new conditional expression, use the **Add** button  on the right - a new instance will be added and the default conditional expression WHERE will be added to the SQL script area automatically. This default conditional expression can then be adjusted to the one you need:

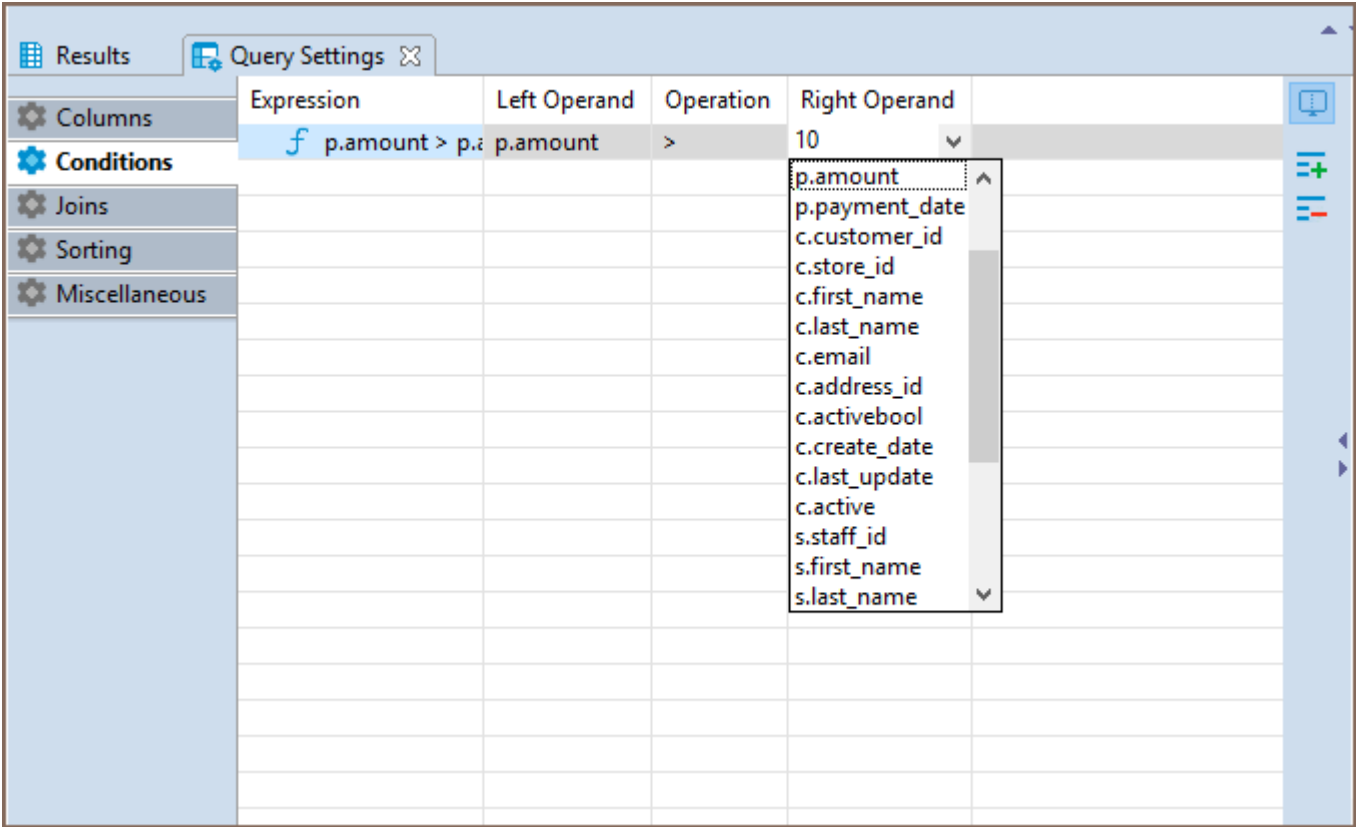
- **Left Operand** setting defines the left operand of the conditional expression. To set the left operand, click the cell in the **Left Operand** column and a drop down list of all available columns will be displayed. Select a column you want to use as the left operand in your conditional expression or insert a digit.




- **Operation** setting defines the comparison rule between the left and the right operands of the conditional expression. To set a comparison rule, click the cell in the **Operation** column and select the rule you need from the drop down list which will appear.



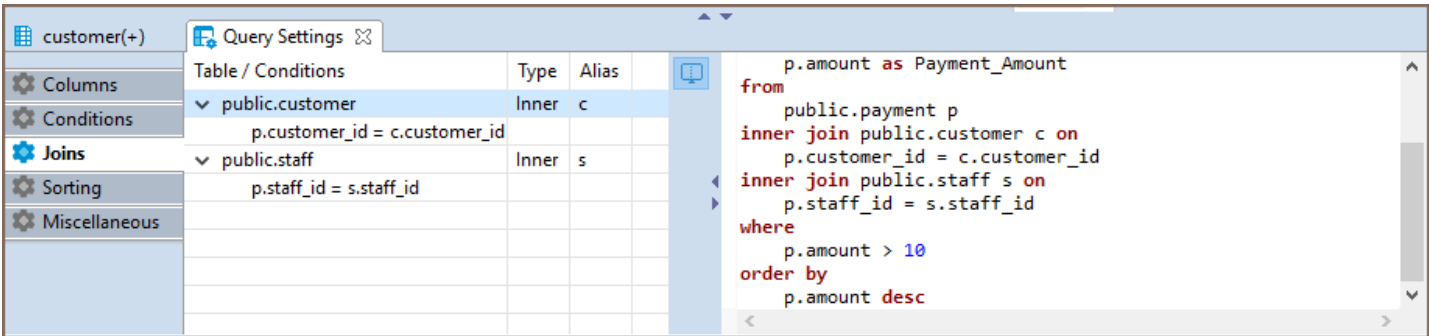
- **Right Operand** setting defines the right operand of the conditional expression. To set the right operand, click the cell in the **Right Operand** column and a drop down list of all available columns will be displayed. Select a column you want to use as the left operand in your conditional expression or insert a digit.



To remove a conditional expression, click on the row containing the expression and press the **Remove** button  on the right.

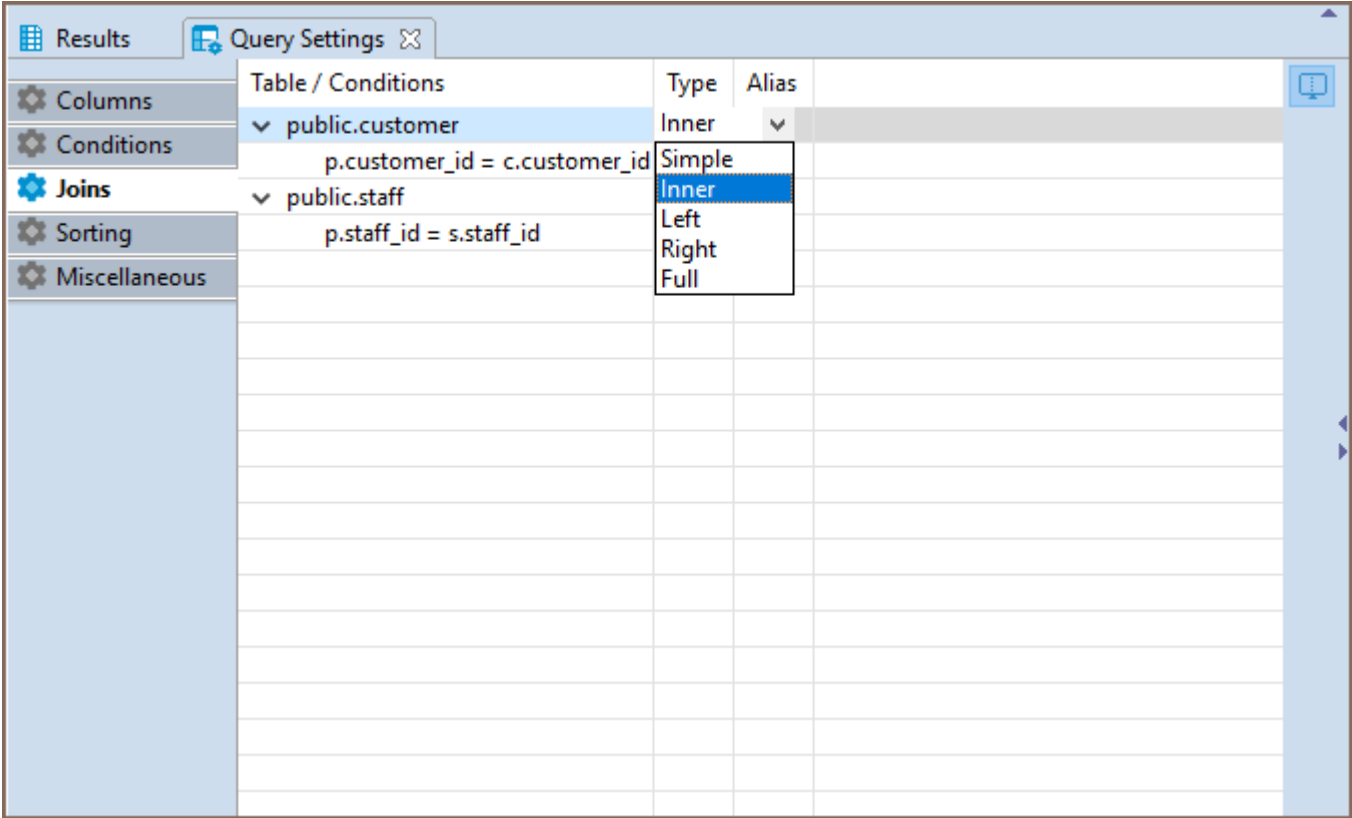
Joins

All the joins existing between the tables in **Visual Query Builder** main window are displayed in the **Joins** tab of **Query Settings Editor**.

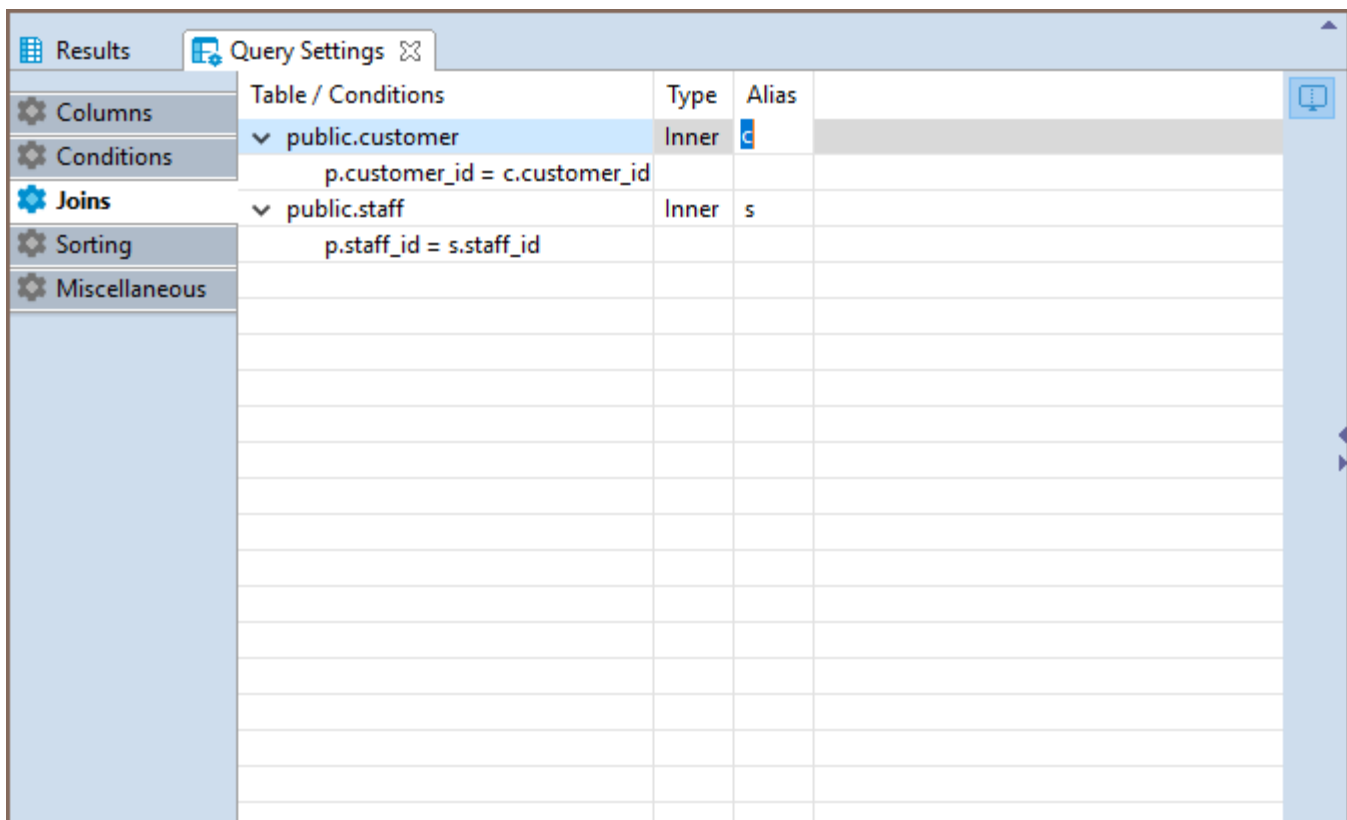


Joins cannot be added or removed by means of **Query Settings Editor**, however, the following join settings can be adjusted here:

- **Type** - defines the type of the join. Click the cell in the **Type** column - a drop down with available join types will be displayed. Select the required option from the list by clicking on it.

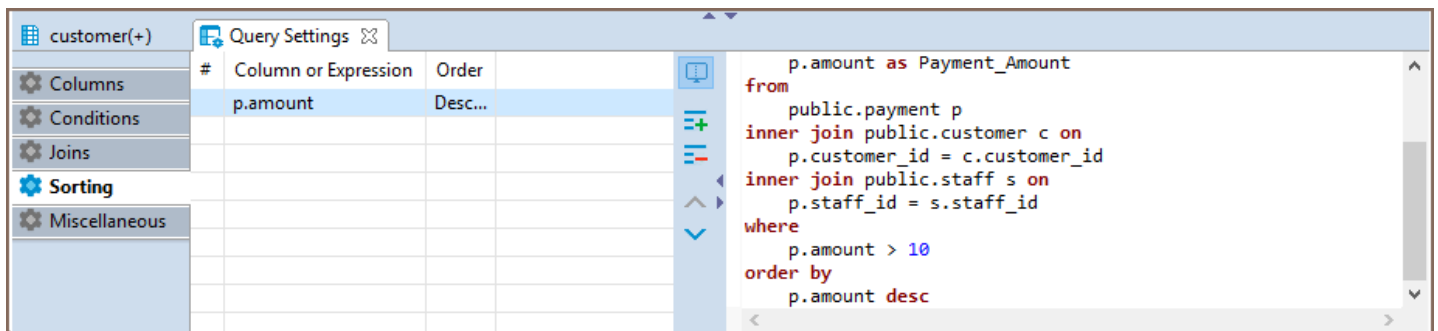



- **Alias** - defines a user friendly name of the join. To define this setting click on the cell in the **Alias** column and input the name.



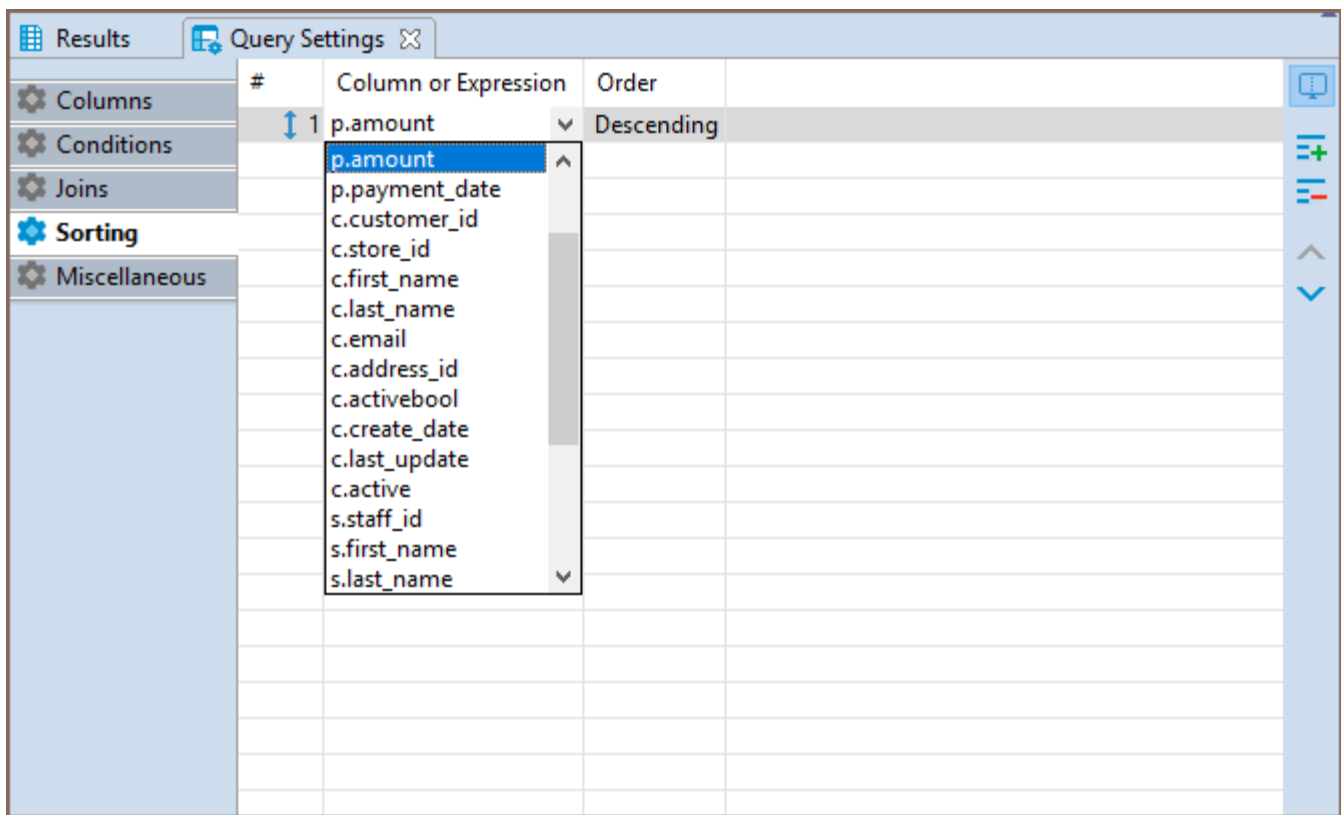
Sorting

In the **Sorting** tab you can set the order of rows in the result table.

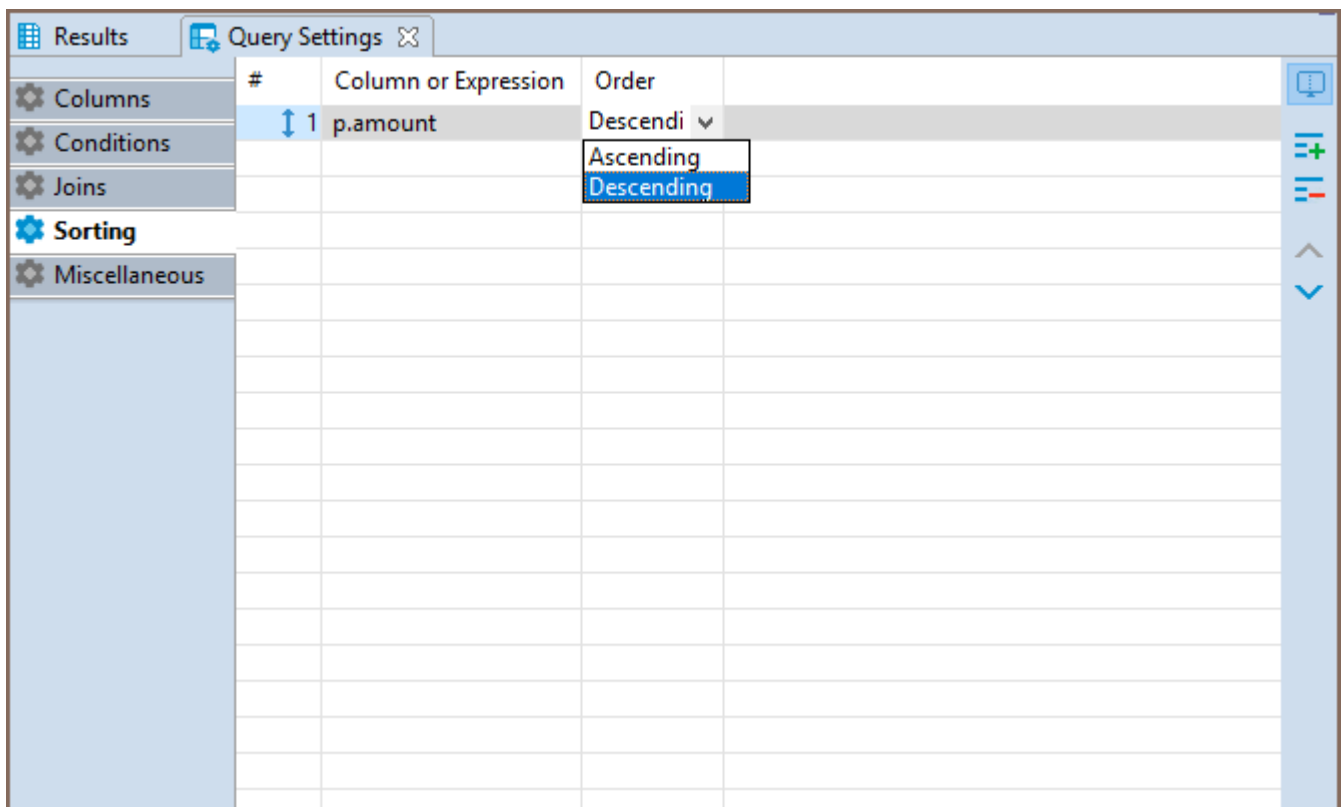


To add a new sorting condition press the **Add** button  on the right and the default conditional expression ORDER BY will be added to the SQL script area automatically. This default conditional expression can then be adjusted to the one you need:

- Once a new condition is added, click the first cell in **Conditions or Expressions** column and a drop down list of all available columns will appear. Select the required column by clicking on its name.



- In the **Order** column you can define whether the rows of the selected column should be sorted in ascending or descending order in the result table. To set the order, click the cell in **Order** column and select the required option from the dropdown. The order by command will be added to the script.

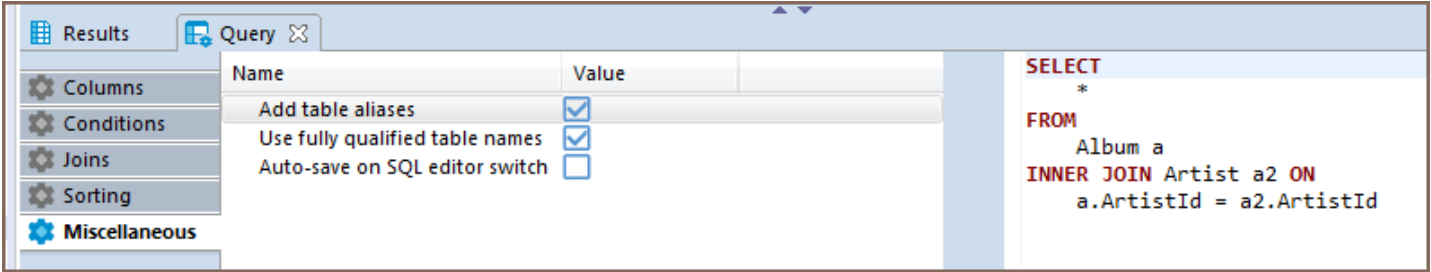


To remove a condition use the **Remove** button  on the right.



Miscellaneous

In the **Miscellaneous** tab it is possible to:

- Enable or disable the automatic generation of aliases for tables by selecting the **Add table aliases** check-box.
- Disable auto-completion for table names by selecting the **Use fully qualified table names** check-box.
- Enable Autosave on SQL-editor switch by selecting the **Autosave on SQL-editor switch** check-box.

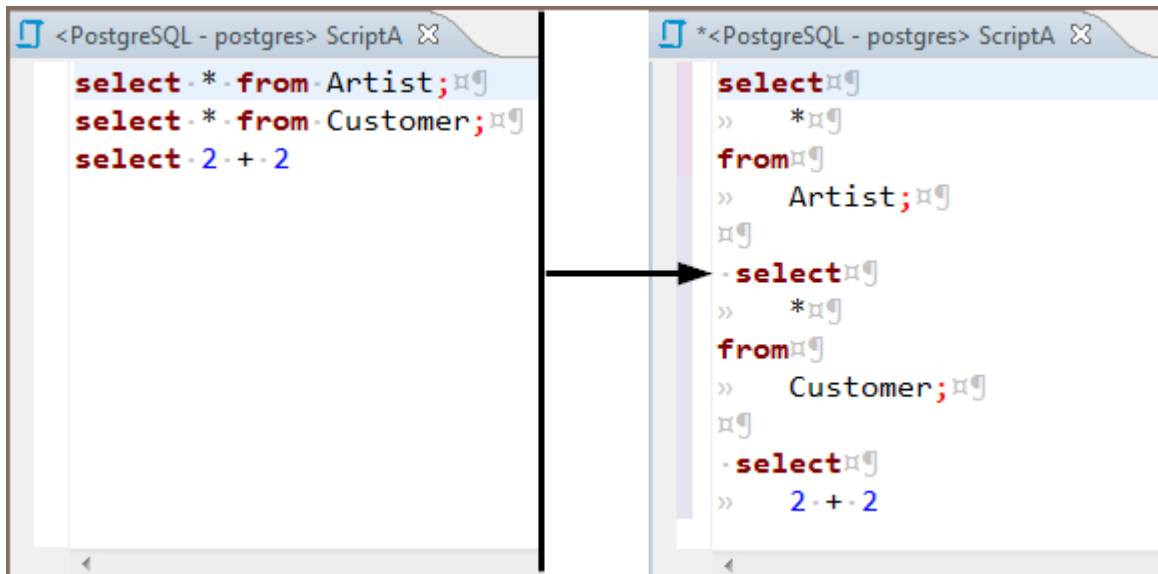


Executing Visual Query

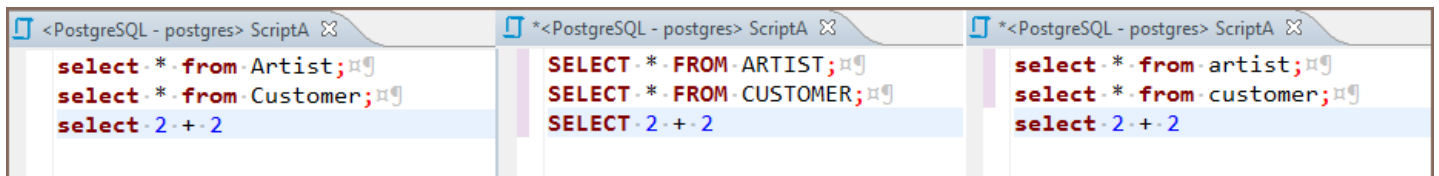
To execute a query, use the **Execute SQL statement** button  to get the results in the same tab or **Execute SQL statement in new tab** button  to get the results in a new tab. Both buttons are located in the **Visual Query Builder** vertical toolbar.

SQL Formatting

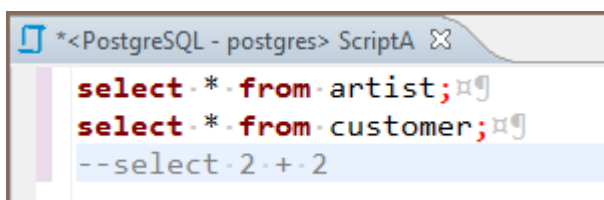
To format an SQL text, select it and press `Ctrl+Shift+F` or right-click the selected text and click **Format -> Format SQL** on the context menu.



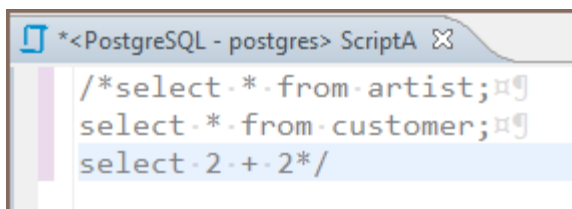
To format a script to upper or lower case, highlight the SQL text, then right-click it and click **Format -> To Upper Case / To Lower Case**, respectively, on the context menu.



To comment out an SQL line, press `Ctrl+/` or right-click the line and click **Format -> Toggle Line Comment** on the context menu. To uncomment a commented line, manually remove the commenting syntax, or press the same button combination, or right-click the line and click the same item on the context menu.



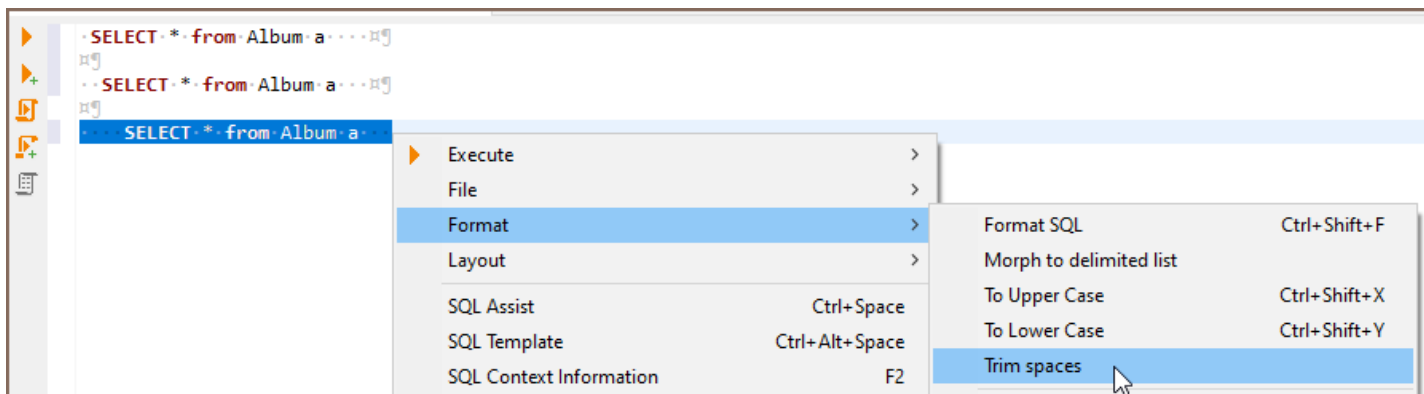
To comment out a block of text, select the text, then press `Ctrl+Shift+/` or right-click it and click **Format -> Toggle Block Comment** on the context menu. To uncomment a commented block of text, you can either manually remove the commenting syntax or select the same block of text, right-click it and click the same item on the context menu or press the same button combination.



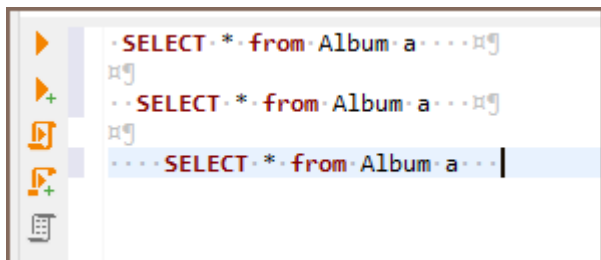
```
*<PostgreSQL - postgres> ScriptA x
/*select *.from artist;
select *.from customer;
select 2.+2*/
```

To trim spaces (leading and trailing) SQL text right-click the selected text or end of string and click **Format -> Trim spaces** on the context menu.

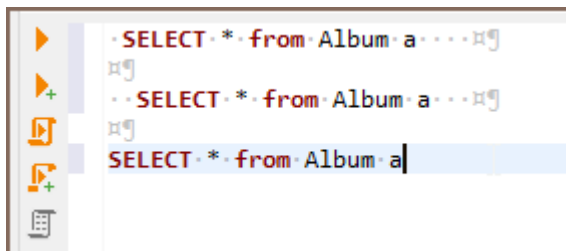
You can choose a part of the text



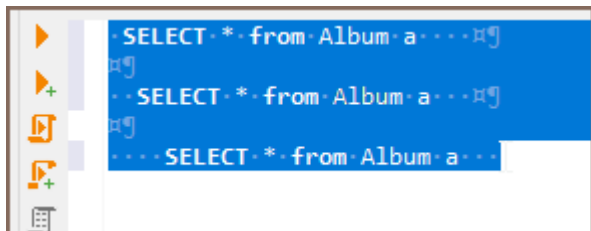
or put the cursor at the end of the row you want to trim



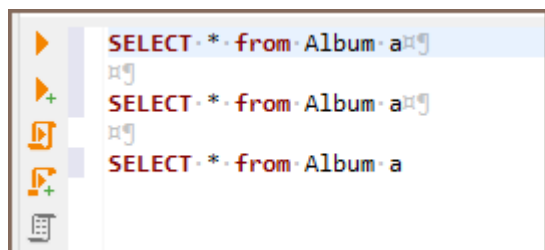
You will get the following result:



Or you can choose a part of the text from many lines



Then each of the selected rows will trim, and the final result will be like this:



Variables panel

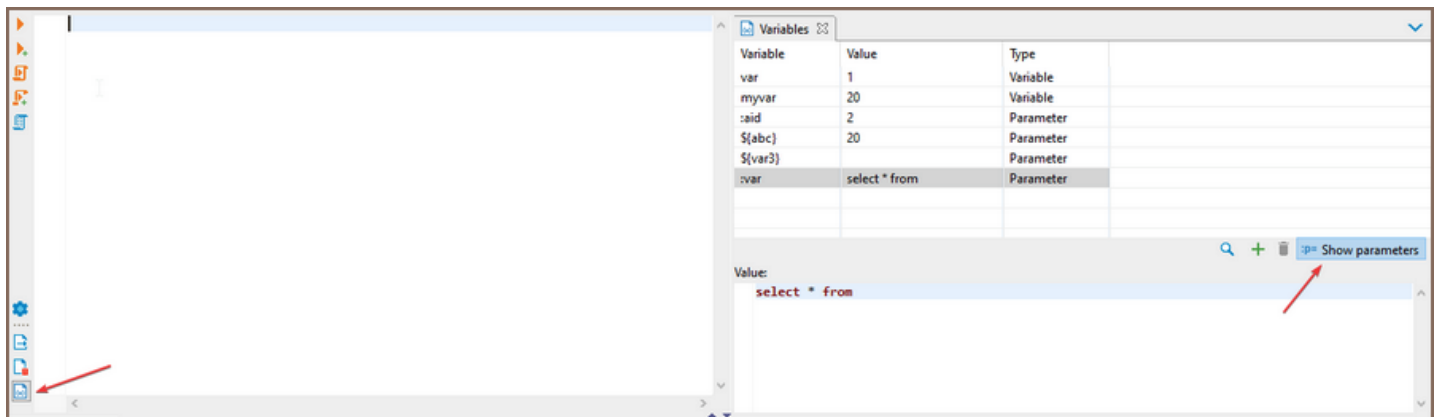
Table of contents

[Manipulating variables](#)

[Moving a panel](#)

You can see all of the currently assigned local variables for SQL Editor. You need to click a "Show SQL variables" button in the SQL editor.

A new tab alongside Output and Execution log panels will be opened with a list of assigned variables. On this panel, you can also show assigned parameters by clicking the corresponding button. (Note: toolbar is customizable. See [Toolbar Customization](#))



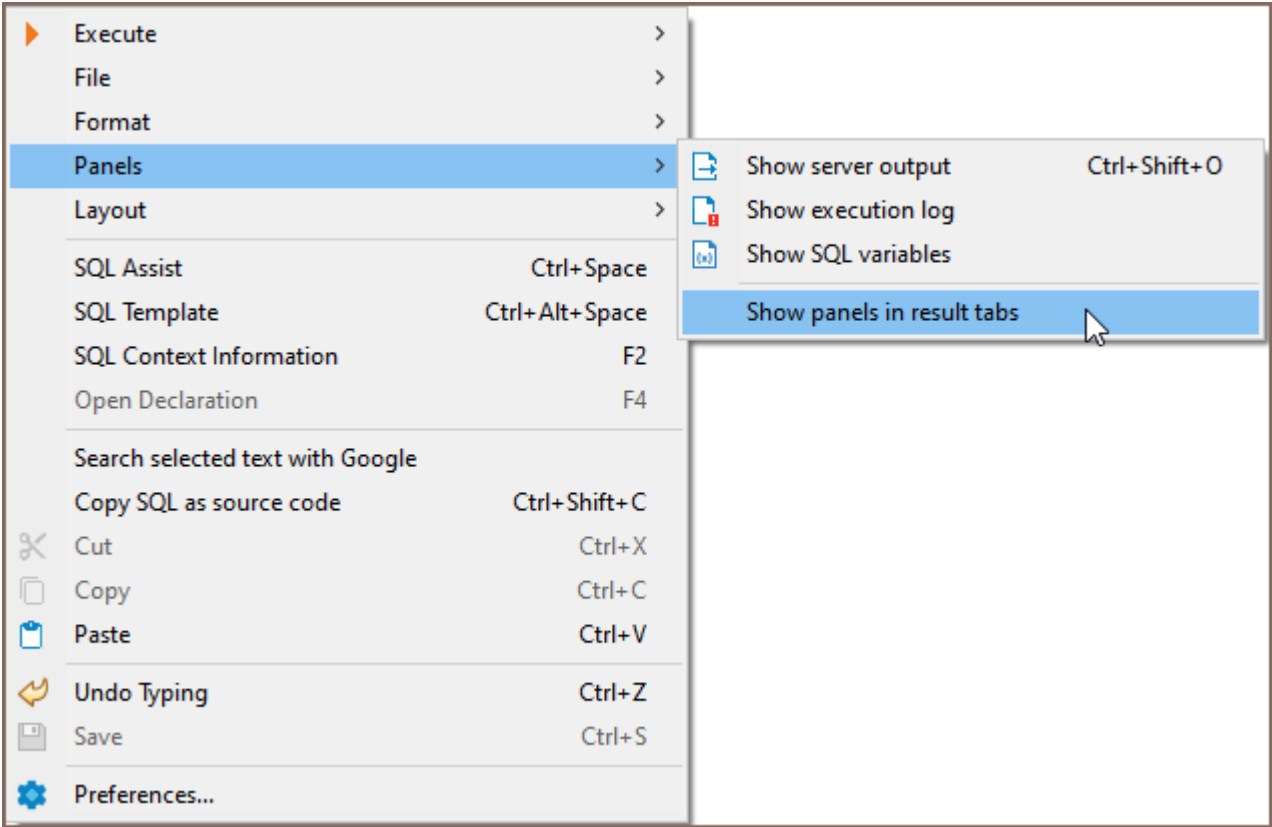
Manipulating variables

You can change values for the current variable using the variables tab. Simply click on a variable's row and edit its value in the window below. To delete or add a variable or parameter, you can use a corresponding button. Instead of typing @set or @unset, you can use these buttons in the script.

If you have a long list of variables, you can click a magnifying glass to initiate a search bar. Start typing either a variable's name or its value to filter the list.

Moving a panel

Variables panel is always connected to output and execution log panels, but this group of tabs can be configured to be shown either at the right side of the editor or at the bottom alongside the results panel. By default, the panels are shown on the right side. To change their location you need to either check or uncheck `Show panels i` in the context menu.



Spelling

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Usage](#)

[Settings](#)

[Activation](#)

[Options](#)

[Custom Dictionaries](#)

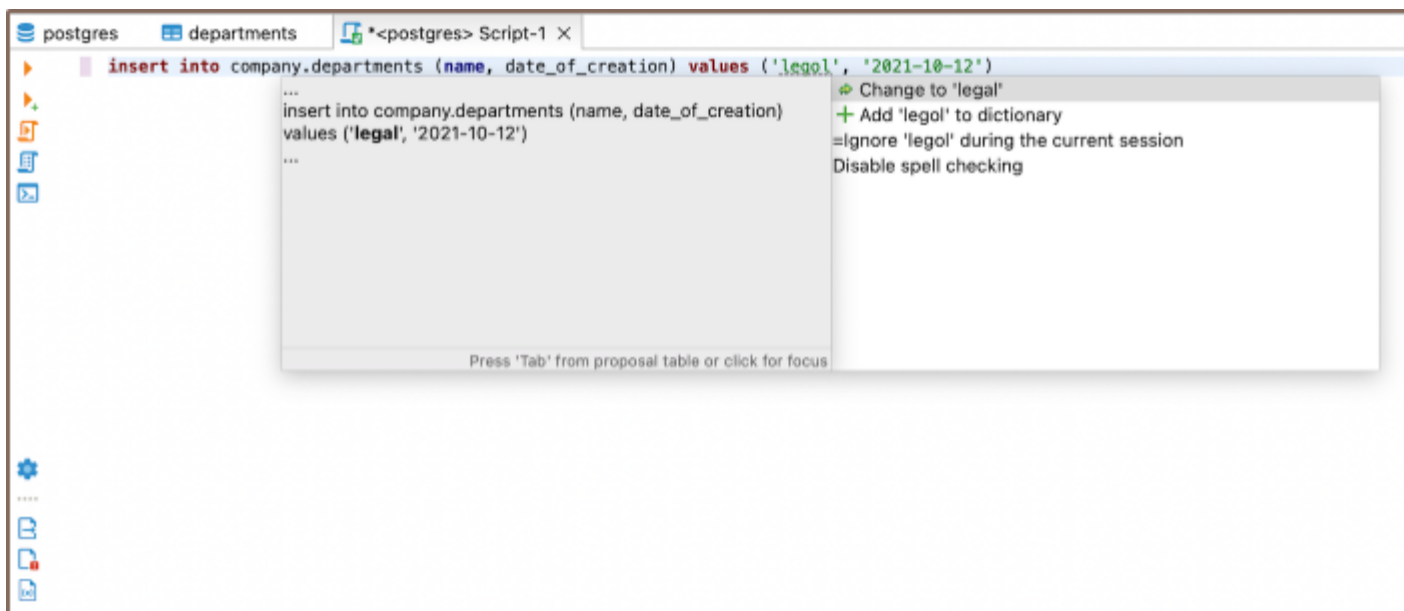
[Advanced Settings](#)

DBeaver's spelling function is a feature designed to detect spelling errors. It allows you to change the incorrect words, upload custom dictionaries in any language and it provides you with advanced options for a more refined text management experience.

Usage

The spelling function is accessible in your SQL script files. Words that are not spelled correctly are underlined, which will alert you to potential inaccuracies.

To change these incorrect words, open the context menu by right-clicking on the word and selecting **Quick Fix** or by using the `Ctrl+1` (or `#1` for Mac OS).



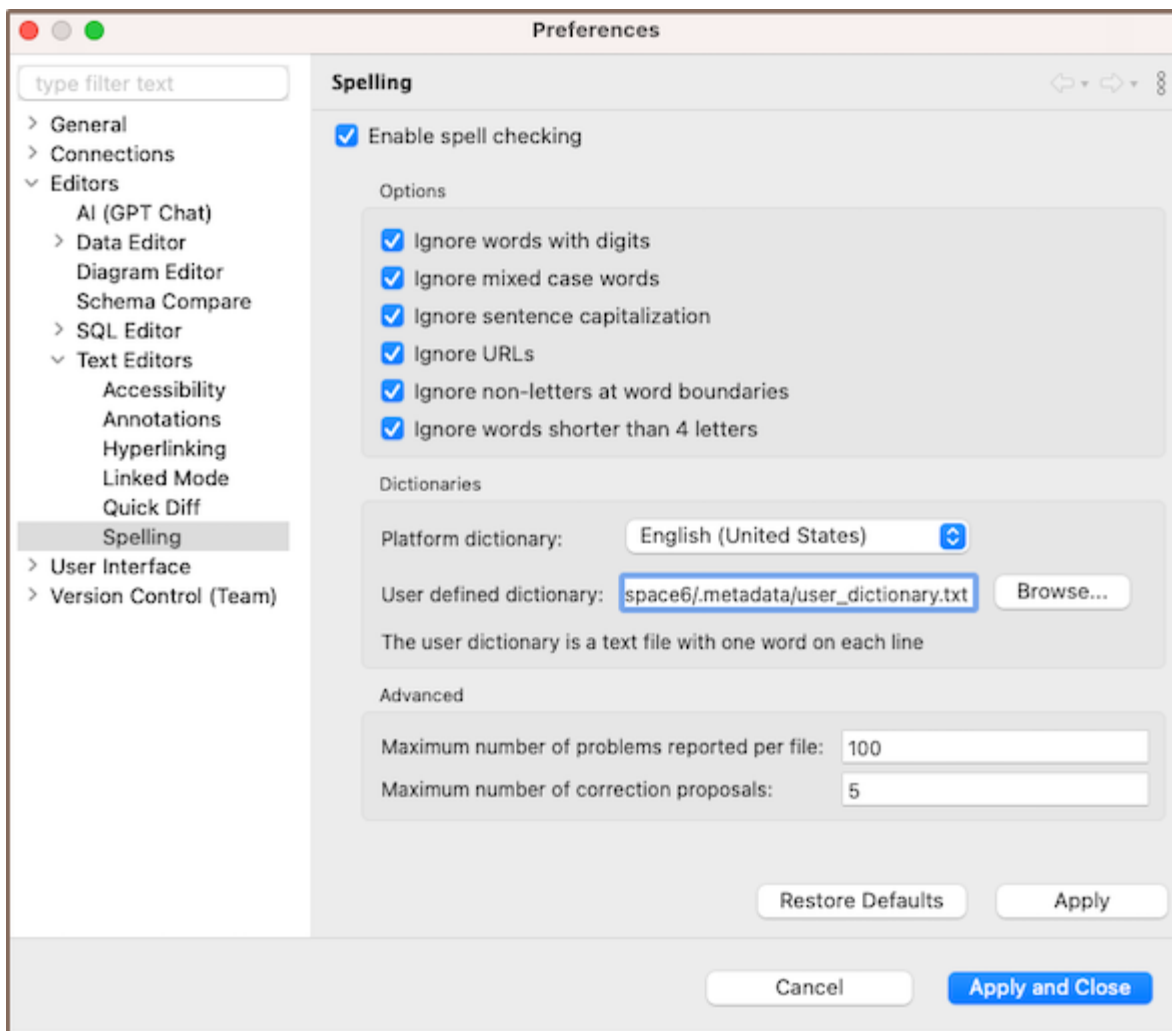
The context menu offers several options:

- Change the incorrect word with to a suggested word.
- Add the word to the dictionary.
- Ignore the word during the current session.
- Disable the spell checking entirely.

To apply any of these changes, you need to double-click on the chosen option with the left mouse button or press the `Tab` key.

Settings

To open the settings for the spelling page, navigate through the following steps: **Window -> Preferences -> Editors -> Text Editors -> Spelling**.



Activation

The spelling function is enabled by default but can be customized to your requirements.

You can disable the spelling function by unchecking the box next to **Enable spell checking** in Preferences.

Options

DBeaver's spelling settings provide specific options for this feature that can be tailored to your needs.

Option name	Description
Ignore words with digits	Skips words containing numbers.
Ignore mix case words	Skips words with combined upper and lower case letters.
Ignore sentence capitalization	Skips words starting with a lowercase letter at sentence beginnings.

Ignore URLs	Excludes web addresses from spell checking.
Ignore words shorter than 4 letters	Disregards shorter words from spell checking.

Custom Dictionaries

DBeaver includes an integrated English dictionary. However, you can choose to disable it and connect your custom dictionary, or use both simultaneously, according to your needs.

In DBeaver's spelling settings you can add your own custom dictionary by specifying the path to it in the **User defined dictionary** field. This dictionary should be a **.txt** file with one word per line.

To disable the custom dictionary, clear the **User defined dictionary** field and apply the changes.

Advanced Settings

You can adjust the **Maximum number of correction proposals**. This setting controls the number of suggested substitutions that appear in the context menu for each misspelled word.

Also, you can specify the **Maximum number of problems reported per file**. This setting determines the total number of words, starting from the first one, that the system will flag for replacement in each file.

SQL Terminal

Table of contents

[Overview](#)

[SQL Terminal](#)

[Context menu](#)

[Preferences](#)

Overview

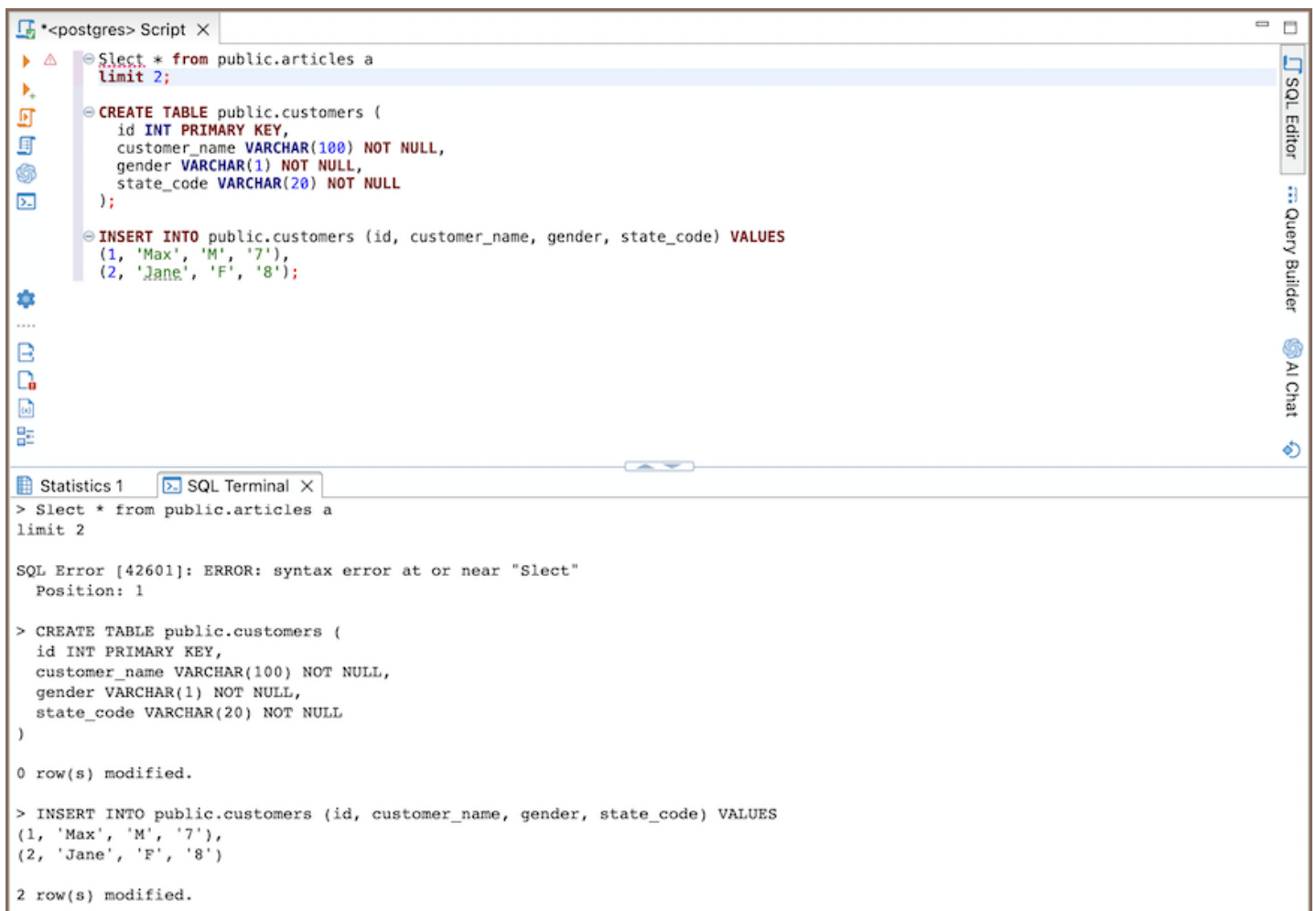
SQL Terminal allows you to display the results of executed SQL queries in a text-based format within the [SQL Editor](#).

SQL Terminal

To open the **SQL Terminal**, press the corresponding button  on the **SQL Editor** toolbar.

Note: The toolbar is customizable. See [Toolbar customization](#) article.

In the **SQL Terminal**, you can view the result set, including data, errors, and statistical information along with the text of the executed query.



Context menu

A context menu is available by right-clicking within the **SQL Terminal** area. This menu provides quick access to various commands.

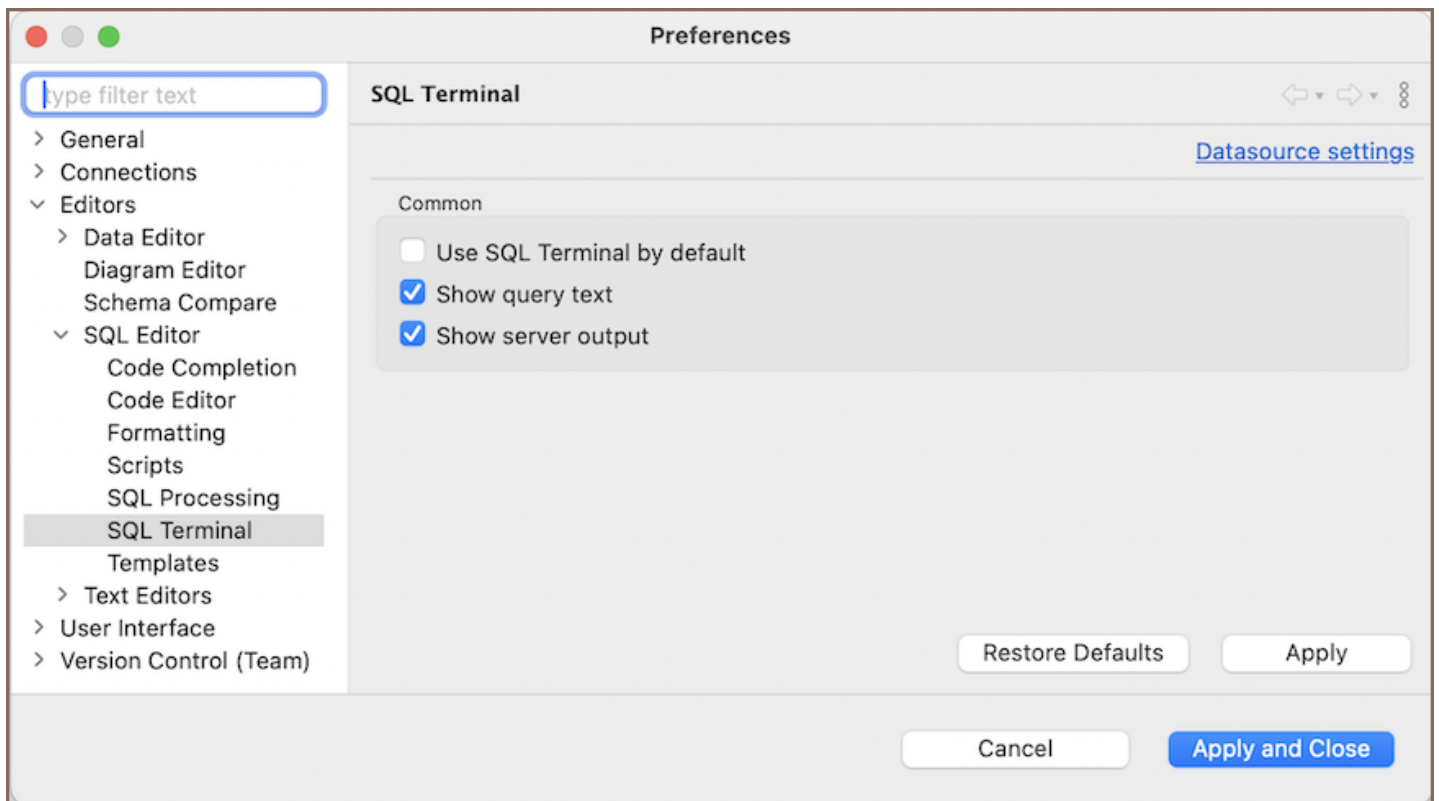


Command	Description
Copy	Copies the selected text to the clipboard.

Select All	Selects all text in the SQL Terminal .
Toggle Word Wrap	Toggles wrapping of long lines of text.
Find/Replace...	Opens the find/replace dialog.
Clear	Clears the contents of the SQL Terminal .

Preferences

To configure SQL Terminal settings, go to **Window -> Preferences -> Editors -> SQL Editor -> SQL Terminal**.



Setting	Description
Use SQL Terminal by default	SQL Terminal will open for every new script.
Show query text	Displays the text of the executed query in the terminal.
Show server output	Shows output messages from the server in the terminal.

AI Smart Assistance

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only. For users of the Community edition to access **AI smart completion** feature, first [install the required extension](#).

Table of contents

[Understanding the AI integration in DBeaver](#)

[Initial setup](#)

[Data privacy](#)

[AI settings and customization](#)

[Basic configuration](#)

[Credentials for OpenAI](#)

[Credentials for Azure AI](#)

[Credentials for Google Gemini](#)

[Credentials for Ollama](#)

[Preferences](#)

[PRO version exclusive settings](#)

[Utilizing AI capabilities in DBeaver](#)

[AI Chat usage](#)

[Resetting query context](#)

[Defining the scope in AI Chat](#)

[AI smart completion usage](#)

[Defining the scope](#)

[Accessing query history](#)

[Disabling AI features](#)

[Best practices for question formulation](#)

[Installing the GPT extension](#)

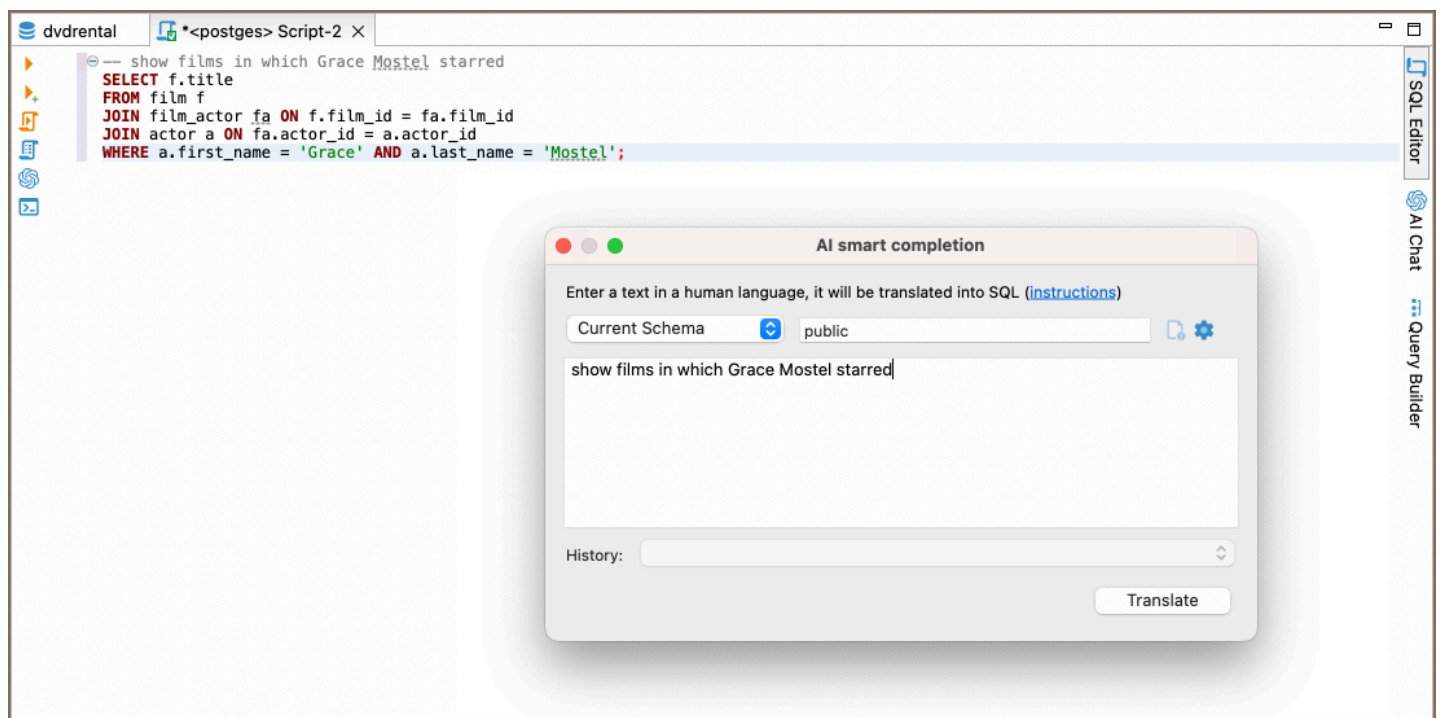
DBeaver offers the ability to construct SQL queries using natural language through our **AI smart completion** and **AI Chat** features. This capability is achieved through integrations with OpenAI's [GPT language models](#), [Azure OpenAI](#) and [Google Gemini](#).

Note: DBeaver is not affiliated with OpenAI, Microsoft Azure, or Google. Integration with AI features is achieved through the public APIs of OpenAI's GPT, Azure OpenAI Service, and Google Gemini.

- To utilize these features, register with the respective service provider and [obtain a secret key](#).
- Users of the DBeaver Community version need to [install the GPT extension](#) to enable this functionality.

Understanding the AI integration in DBeaver

With the **AI smart completion** feature, you can type queries in natural language and DBeaver will convert them into SQL statements. This tool simplifies writing complex queries by interpreting your input and automatically generating the correct SQL code.



The **AI Chat** feature provides a conversational interface, allowing you to communicate with the database in plain language. Type your queries conversationally, and **AI Chat** translates them into executable SQL, streamlining your database interactions.

The screenshot shows the DBeaver SQL Editor interface. The top panel contains two queries with AI-generated prompts. The first query is for films starring Grace Mostel, and the second query adds category names to the results. The bottom panel shows the results of the second query in a grid view.

Query 1: show films in which Grace Mostel starred

```
SELECT f.title
FROM film f
JOIN film_actor fa ON f.film_id = fa.film_id
JOIN actor a ON fa.actor_id = a.actor_id
WHERE a.first_name = 'Grace' AND a.last_name = 'Mostel';
```

Query 2: add the category name of these films

```
SELECT f.title, c.name
FROM film f
JOIN film_actor fa ON f.film_id = fa.film_id
JOIN actor a ON fa.actor_id = a.actor_id
JOIN film_category fc ON f.film_id = fc.film_id
JOIN category c ON fc.category_id = c.category_id
WHERE a.first_name = 'Grace' AND a.last_name = 'Mostel';
```

Results:

ABC title	ABC name
1 Angels Life	New
2 Anonymous Human	Sports
3 Arachnophobia Rollercoaster	Horror
4 Berets Agent	Action
5 Breaking Home	New
6 Command Darling	Foreign
7 Confessions Maguire	Drama
8 Dazed Punk	Games
9 Deceiver Betrayed	Drama
10 Destination Jerk	Drama
11 Excitement Eve	Action
12 Gaslight Crusade	Horror
13 Hellfighters Sierra	Foreign
14 Instinct Airport	Sports

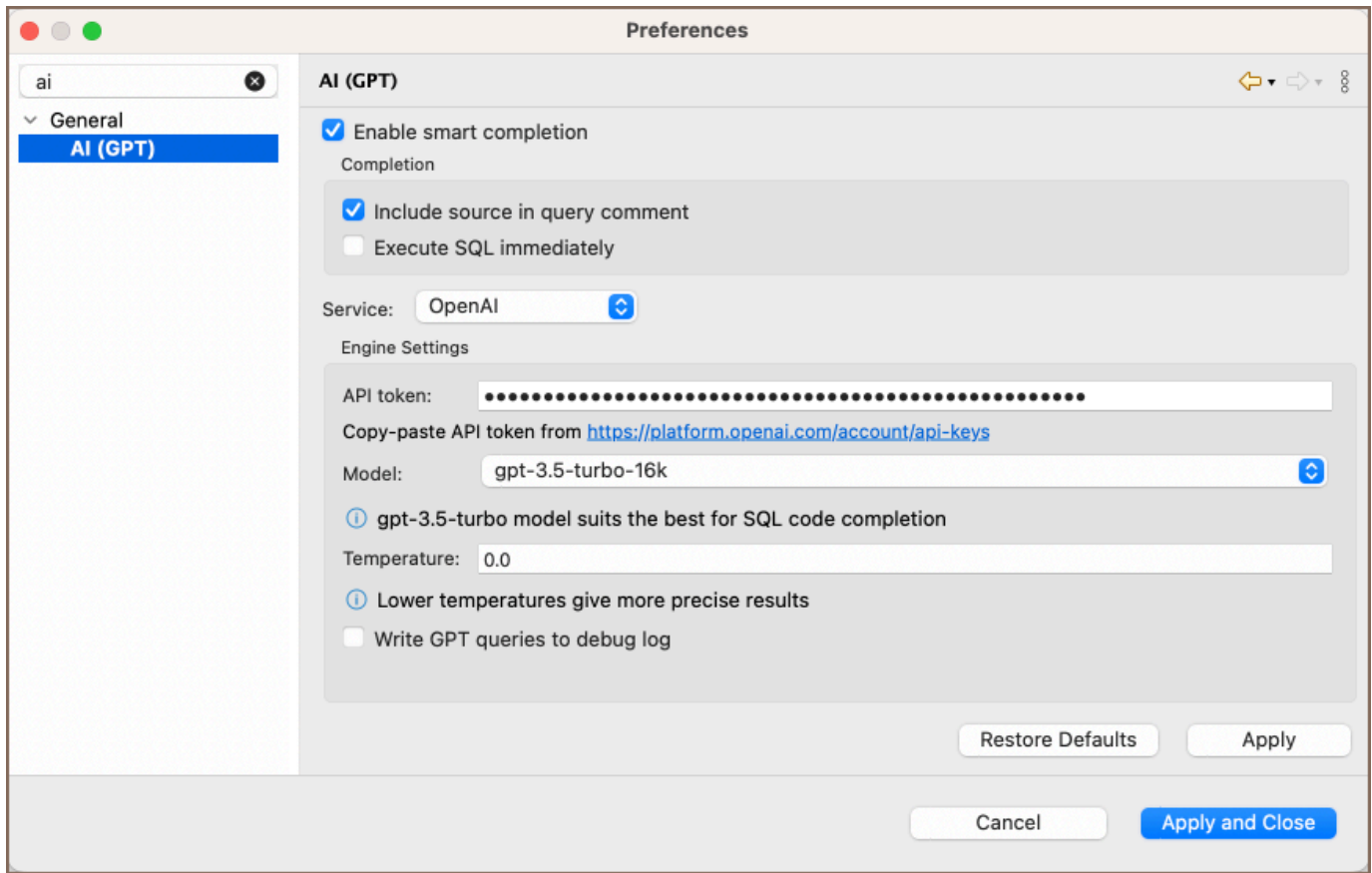
30 row(s) fetched - 6ms, on 2023-11-21 at 12:24:03

Initial setup

To activate the AI features in DBeaver, configure the API token:

1. Navigate to **Window -> Preferences -> General -> AI**.
2. Ensure the **Enable smart completion** option is activated. This option is typically enabled by default in the PRO versions.
3. In the **API token** field, input your AI secret key.

4. Save the changes. See the [Basic configuration](#) for secret key details.



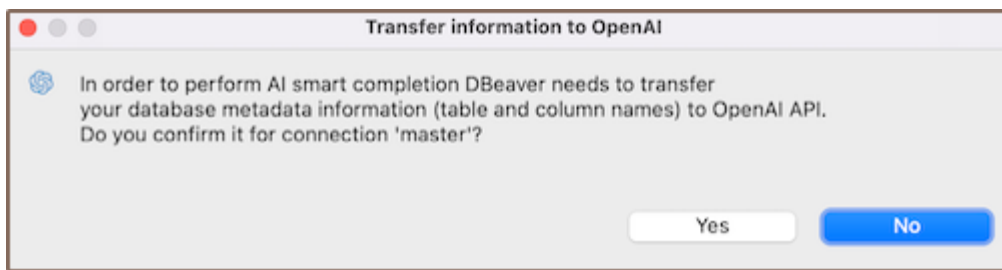
For instructions on utilizing the AI features, visit the [AI Smart completion usage](#) and [AI Chat usage](#) sections.

Data privacy

We prioritize data safety and user privacy. In this section, we outline how data is managed and the measures taken to protect user privacy when using the AI features.

To enable the AI features capabilities, table and column names from the current database schema are transmitted to OpenAI. This step is crucial for accurately translating user requests into SQL queries. Key considerations regarding data privacy are as follows:

- **No Table Data:** Only metadata like table and column names are shared with OpenAI. Actual table data is not transmitted.
- **User Consent:** On the first use of AI completion for a specific connection, DBeaver will prompt for your confirmation to send metadata. This confirmation is mandatory to use the feature.



- **Log Transparency:** The entire request can be logged for your review. To enable this, navigate to **Preferences** and check the **Write GPT queries to debug log** option.
- **Selective Metadata Sharing:** If you prefer not to share information about certain tables, adjust the tables in scope using the **Scope** field.
- **Disabling AI:** PRO version users can [turn off the AI feature](#), while Community Edition users can avoid it by not installing the AI plugin. ★
- **Azure OpenAI privacy:** If you use Azure OpenAI, be aware that it operates under its own [privacy policy](#). It's recommended to review their terms before using. ★
- **Google Gemini privacy:** When utilizing Google Gemini, it is important to understand the specific data [privacy measures](#). ★

AI settings and customization

To utilize the AI-enhanced functionalities within DBeaver, certain configurations and setup processes are required. This section offers a comprehensive guide on initial setup, advanced configurations for PRO users, and customization options to tailor the AI integration according to specific preferences.

Basic configuration

By default, AI features are ready to use. To start, you need to specify the service credentials based on the AI service you choose: OpenAI, Azure AI, Ollama or Google Gemini.

Credentials for OpenAI

1. Sign up on the [OpenAI platform](#).

2. Navigate to the [API Keys section](#) and generate a new secret key.
3. Insert this key into DBeaver's **API token** setting.

Note: OpenAI services are available in specific countries. Consult the [supported countries](#) list to verify availability in your location.

Credentials for Azure AI

1. Sign up on the [Azure platform](#).
2. Navigate to the [Azure Portal](#) and create a new AI service under the AI + Machine Learning section.
3. Generate and copy the credentials for the newly created service.
4. Insert these credentials into DBeaver's **Engine Settings**.

Credentials for Google Gemini

1. Sign up on the [Google Cloud Platform](#).
2. Navigate to the [Google Cloud Console](#) and create a new project.
3. Enable the Gemini API for your project by searching for the Gemini API in the marketplace and clicking **Enable**.
4. Create credentials for your project by navigating to the **Credentials** page under **APIs & Services**. Choose **Create credentials** and select the appropriate type for your Gemini integration.
5. Copy the generated credentials.
6. Insert these credentials into DBeaver's **Engine Settings**.

Note: Google Gemini services are subject to regional availability. Check the list of [available regions](#) to ensure access in your area.

Credentials for Ollama

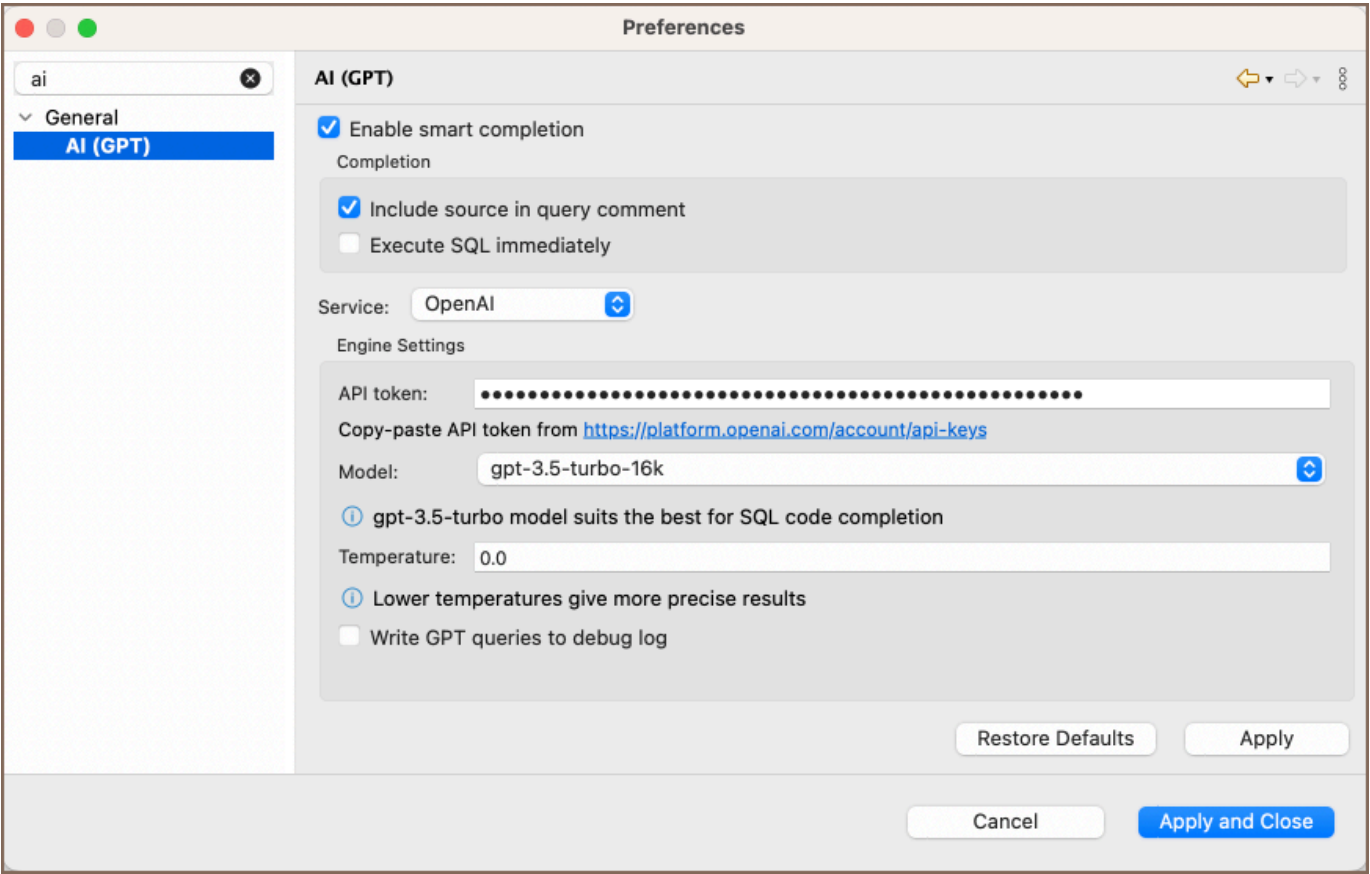
Ensure that Ollama is already installed and running on a server. You will need the host address where Ollama is installed to proceed.

1. Specify the host address of your Ollama server in the **Instance host** field, ensuring it follows the format `http://host:port`.
2. Click **Load Models**. If the host address is correct, DBeaver will display the available models from your Ollama server in the **Model** dropdown menu.
3. Select the model you need for your integration.

Preferences

For specific requirements or troubleshooting, you might want to adjust some of the following settings:

- Navigate to **Window -> Preferences -> General -> AI** to access these settings.



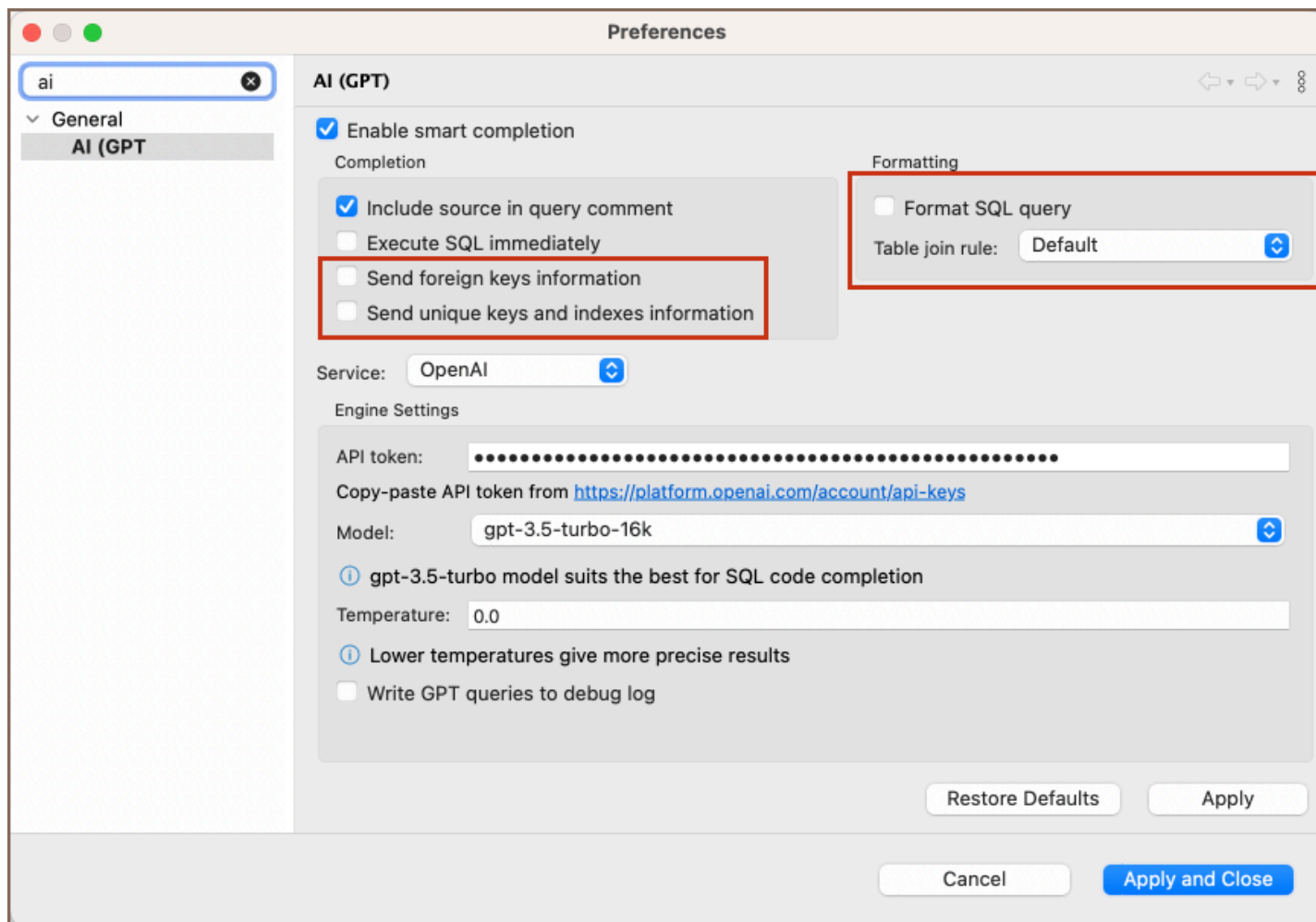
Setting	Description
Enable smart completion	Displays the AI features in the SQL Editor.
Include source in query	

comment	Shows your original request above the AI-generated query in the SQL Editor.
Execute SQL immediately	Runs the translated SQL query immediately after generation.
Send attribute type information	Send attribute type information to the AI vendor. It makes better completion, but consumes more tokens.
Send object description	Send object description to the AI vendor. Improves completion, but may consume significant amount of tokens.
API token	Input your secret key from the OpenAI platform.
Model	Choose the AI model (recommended: gpt-3.5-turbo for SQL).
Temperature	Control AI's creativity from 0.0 (more precise) to 0.9 (more diverse). Note that higher temperature can lead to less predictable results.
Write GPT/Ollama queries to debug log	Logs your AI requests.

PRO version exclusive settings

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

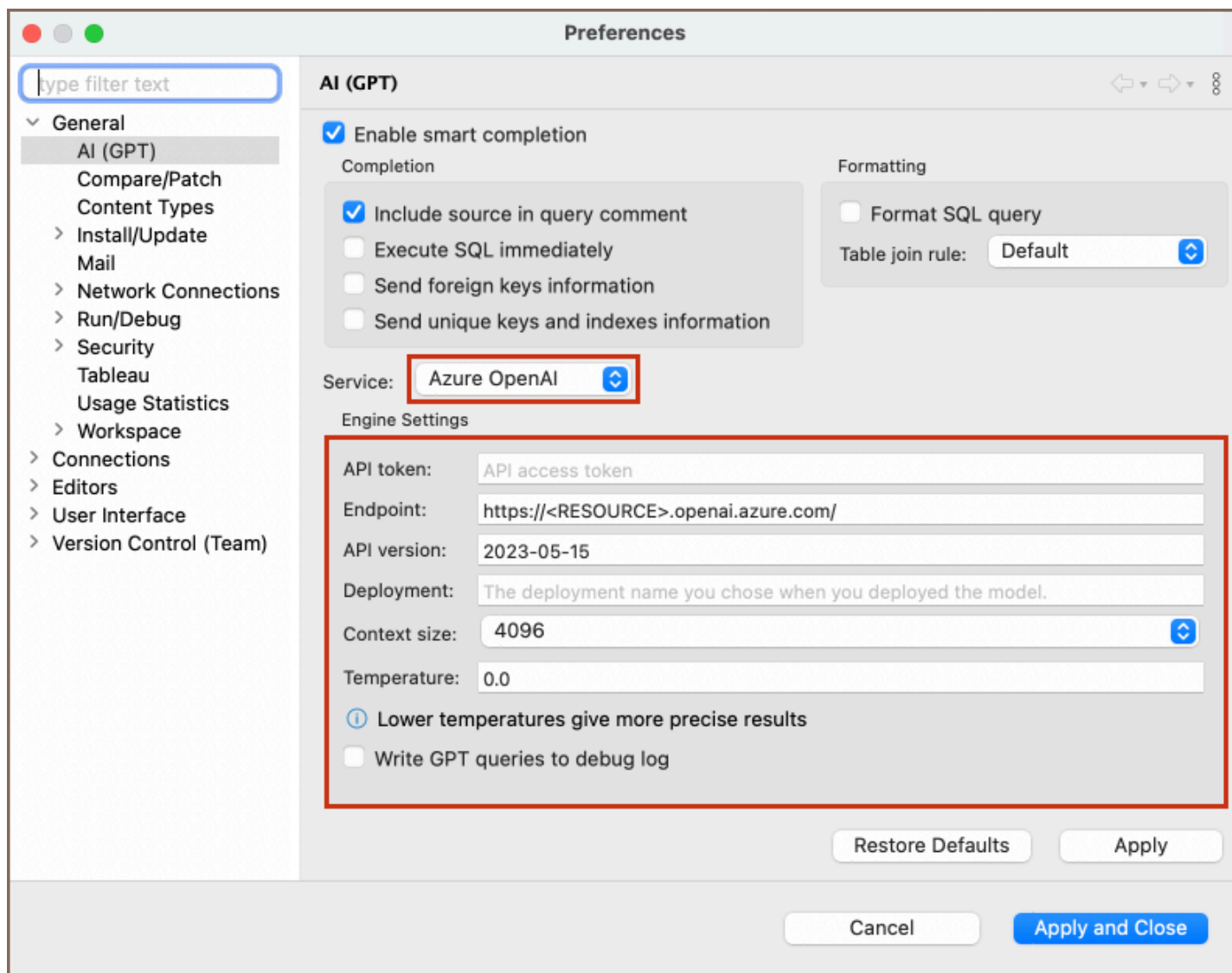
For users of the [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions, additional configurations are available:



Setting	Description
Send foreign keys information	Helps AI understand table relationships.
Send unique keys and indexes information	Assists AI in crafting complex queries.
Format SQL query	Adds formatting to the generated SQL.
Table join rule	Choose between explicit JOIN or JOIN with sub-queries.

There is also an option to switch the **Service** between **OpenAI**, **Azure OpenAI**, **Google Gemini** and **Ollama**.

Azure provides a set of distinct settings:



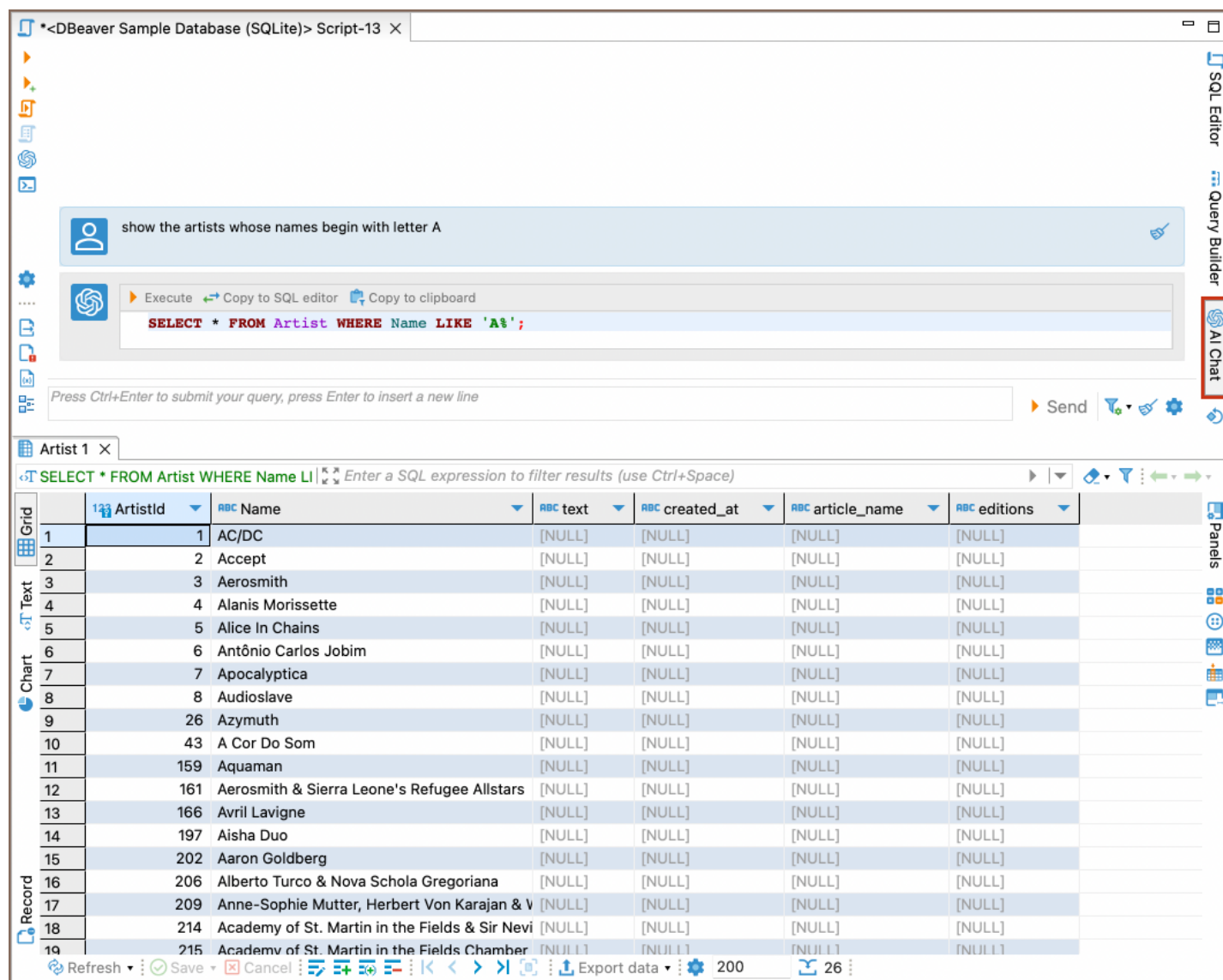
Setting	Description
Endpoint	Configure a custom endpoint URL for Azure OpenAPI interactions.
API version	Select the version of the API you wish to use.
Deployment	Specify the deployment name chosen during model deployment.
Context size	Choose the context size between 2048 and 32768 . A larger number allows the AI to use more data for better answers but may slow down response time. Choose based on your balance of accuracy and speed.

Utilizing AI capabilities in DBeaver

AI Chat usage


Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

To utilize the **AI Chat** feature in DBeaver:



The screenshot shows the DBeaver SQL Editor interface. The top toolbar includes buttons for 'Execute', 'Copy to SQL editor', and 'Copy to clipboard'. The AI Chat tab is highlighted on the right toolbar. The chat window shows a user prompt 'show the artists whose names begin with letter A' and the generated SQL query 'SELECT * FROM Artist WHERE Name LIKE 'A%';'. Below the chat window, the SQL query is executed, and the results are displayed in a table.

ArtistId	ABC Name	ABC text	ABC created_at	ABC article_name	ABC editions
1	AC/DC	[NULL]	[NULL]	[NULL]	[NULL]
2	Accept	[NULL]	[NULL]	[NULL]	[NULL]
3	Aerosmith	[NULL]	[NULL]	[NULL]	[NULL]
4	Alanis Morissette	[NULL]	[NULL]	[NULL]	[NULL]
5	Alice In Chains	[NULL]	[NULL]	[NULL]	[NULL]
6	Antônio Carlos Jobim	[NULL]	[NULL]	[NULL]	[NULL]
7	Apocalyptica	[NULL]	[NULL]	[NULL]	[NULL]
8	Audioslave	[NULL]	[NULL]	[NULL]	[NULL]
9	Azymuth	[NULL]	[NULL]	[NULL]	[NULL]
10	A Cor Do Som	[NULL]	[NULL]	[NULL]	[NULL]
11	Aquaman	[NULL]	[NULL]	[NULL]	[NULL]
12	Aerosmith & Sierra Leone's Refugee Allstars	[NULL]	[NULL]	[NULL]	[NULL]
13	Avril Lavigne	[NULL]	[NULL]	[NULL]	[NULL]
14	Aisha Duo	[NULL]	[NULL]	[NULL]	[NULL]
15	Aaron Goldberg	[NULL]	[NULL]	[NULL]	[NULL]
16	Alberto Turco & Nova Schola Gregoriana	[NULL]	[NULL]	[NULL]	[NULL]
17	Anne-Sophie Mutter, Herbert Von Karajan & V	[NULL]	[NULL]	[NULL]	[NULL]
18	Academy of St. Martin in the Fields & Sir Nevi	[NULL]	[NULL]	[NULL]	[NULL]
19	Academy of St. Martin in the Fields Chamber	[NULL]	[NULL]	[NULL]	[NULL]

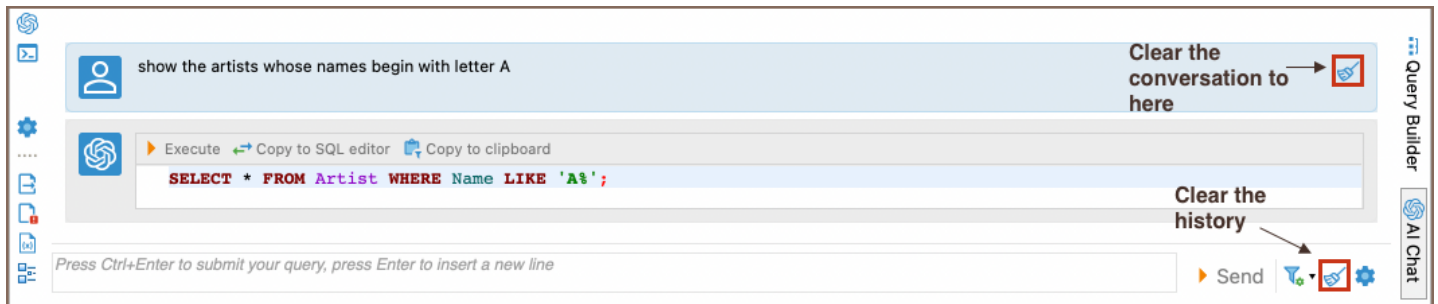
1. Launch the **SQL Editor**.
2. Click on the **AI Chat** tab , located on the right toolbar of the **SQL Editor**.
3. In the chat window that opens, input your request in natural language.
4. Press the **Send** button to get the SQL translation.
5. To execute the SQL query generated by AI, click the **Execute SQL query** button.

Note: The AI Chat logs your query history, enabling you to reference and expand on prior inputs. Each new or edited entry prompts the AI to generate a revised SQL query.

Resetting query context


To start a new conversation or to change the query context in the **AI Chat**, you can use one of the two reset options available:

- To clear the entire history, use the reset button located next to the **Save** button.
- To clear a portion of the conversation history up to a chosen point, use the reset option within the **AI Chat** itself, located at the end of each user's input prompt.



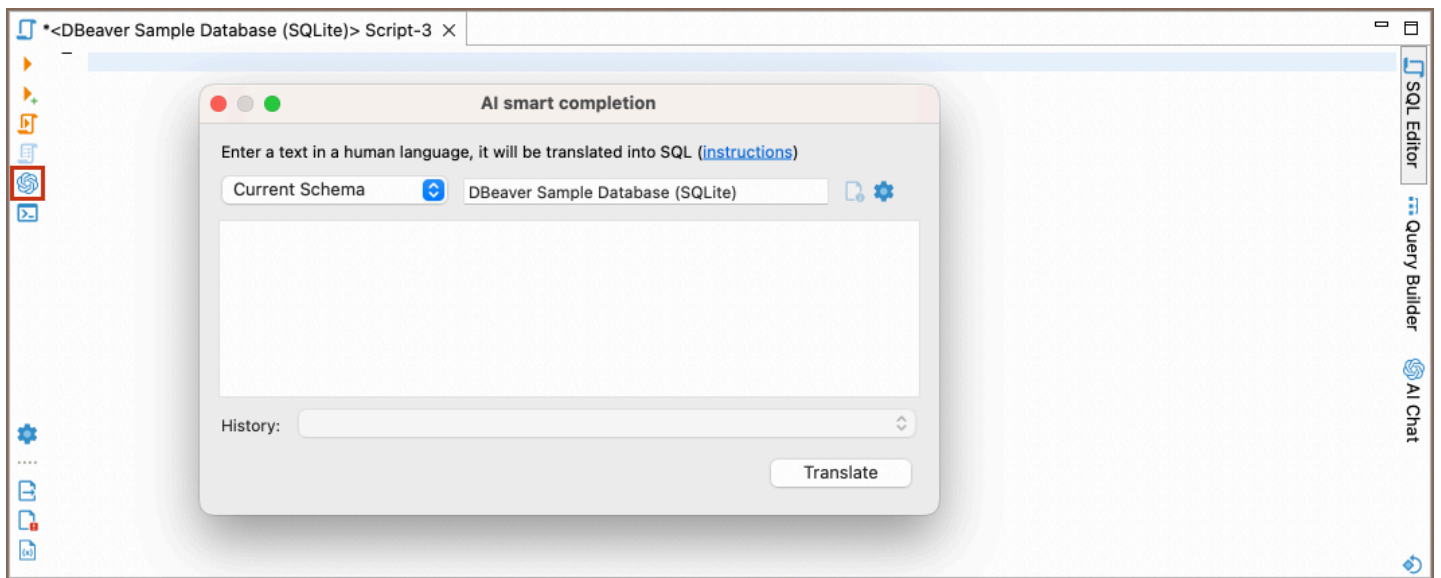
Defining the scope in AI Chat


For enhanced precision, especially in databases with extensive schemas, you can specify database objects in the **Scope** field to narrow down the context.

1. In the **AI Chat** interface, click on the arrow near the **Change scope** button  to open the context menu.
2. Specify the area of your database that the AI should concentrate on.
3. You can choose from:
 - **Current Schema**: Focuses the AI on the schema you are currently using.
 - **Current Database**: Limits the AI to the database currently selected.
 - **Connection**: Sets the AI to consider all schemas within the current database connection.
 - **Custom**: Allows you to define a more specific scope, such as a particular table or schema.

AI smart completion usage

To interact with databases using the **AI Smart completion** feature:



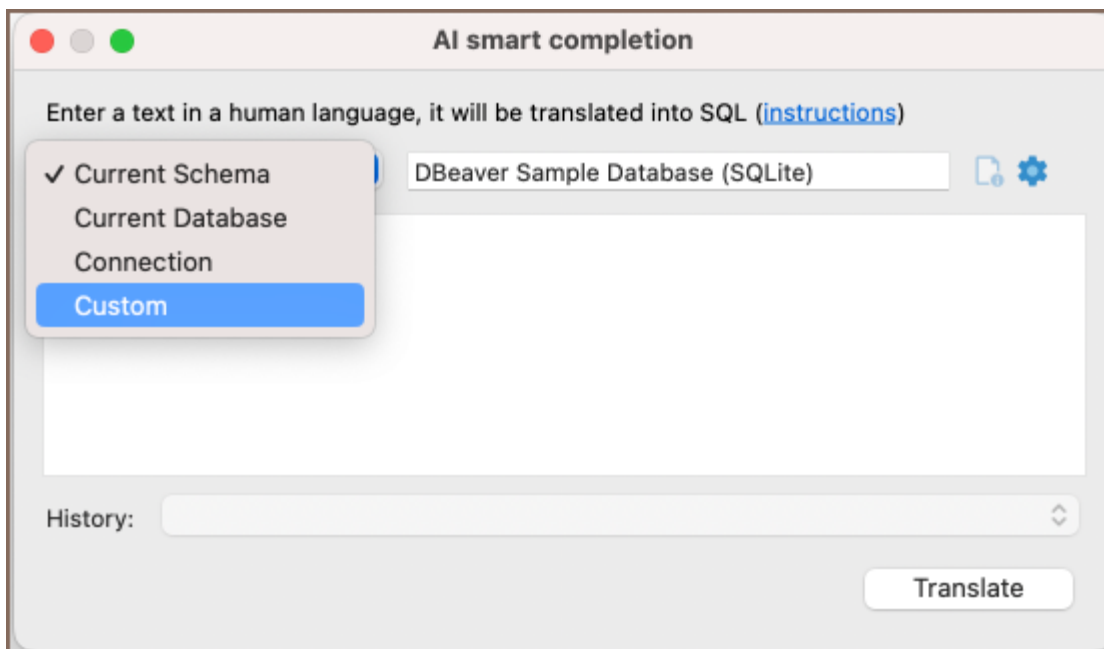
1. Launch the **SQL Editor**.
2. Click on the **GPT Chat smart completion** icon  located in the left toolbar of the **SQL Editor**. > **Note** The toolbar is customizable. For further information, refer to [Toolbar Customization](#) article.
3. Input your natural language request in the **AI smart completion** window.
4. Click **Translate** to obtain the SQL query.

For enhanced precision, especially in databases with extensive schemas, you can specify database objects in the **Scope** field to narrow down the context.

Defining the scope

To enhance the precision of AI-assisted SQL queries, especially in extensive database schemas, setting the scope is key. This action narrows down the AI's focus, ensuring it generates more relevant queries.

1. In the **AI Smart completion** interface, navigate to the **Scope** field.



2. You can choose from:

- **Current Schema:** Focuses the AI on the schema you are currently using.
- **Current Database:** Limits the AI to the database currently selected.
- **Connection:** Sets the AI to consider all schemas within the current database connection.
- **Custom:** Allows you to define a more specific scope, such as a particular table or schema.

Accessing query history

Query history allows you to review previous requests. In the PRO version, this includes history from past sessions, offering a track record of your SQL queries and AI interactions.

Disabling AI features

To hide the **GPT smart completion** and **AI Chat** icons in the SQL Editor:

- Navigate to **Window -> Preferences -> General -> AI**.
- Deselect **Enable smart completion**.

For users with PRO versions wanting to permanently disable this feature:

- Use the system variable by setting `DBEAVER_AI_DISABLED` to `true`.

- Or, add `Dai.disabled=true` to the `dbeaver.ini` file.

Detailed instructions on finding `dbeaver.ini` are available in [our article](#).

Note: Once this feature is permanently disabled in PRO versions, it cannot be re-enabled via Preferences.

Best practices for question formulation

When using AI to generate SQL queries, it's essential to provide clear and specific input. Here's how to optimize your questions:

- **Language:** While AI supports multiple languages, it's recommended to use English for best results.
- **Database knowledge:** Familiarity with your database structure enhances the accuracy of generated queries.
- **Iterative refinement:** Start with a general request and refine iteratively. As you provide more specific details, the AI's output will become more relevant.
- **Explicit details:** If you know certain tables or columns that should be part of the query, include them in your request for better accuracy.

For instance, if you're using the [DBeaver sample SQLite database](#), you might phrase your request as:

Example 1: "List all customers from Italy"

Resulting SQL:

```
SELECT *
FROM customers
WHERE country = 'Italy';
```

Example 2: "montre les clients de France"

```
SELECT *
FROM customer
WHERE country = 'France';
```

Example 3: "show customers who purchased blues tracks, use joins"

```
SELECT c.FirstName, c.LastName, t.Name, g.Name
FROM Customer c
      JOIN Invoice i ON c.CustomerId = i.CustomerId
      JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
      JOIN Track t ON il.TrackId = t.TrackId
      JOIN Genre g ON t.GenreId = g.GenreId
WHERE g.Name = 'Blues'
ORDER BY c.LastName, c.FirstName;
```

Example 4: "get names of customers who purchased blues tracks, use joins"

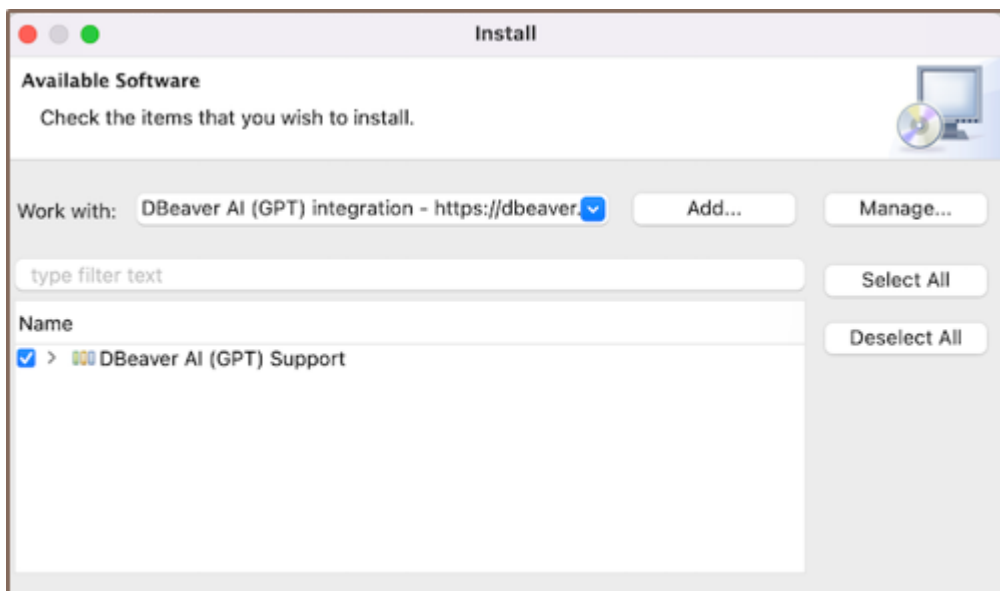
```
SELECT DISTINCT c.FirstName, c.LastName
FROM Customer c
      JOIN Invoice i ON c.CustomerId = i.CustomerId
      JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
      JOIN Track t ON il.TrackId = t.TrackId
      JOIN Genre g ON t.GenreId = g.GenreId
WHERE g.Name = 'Blues';
```

Installing the GPT extension


For users of the DBeaver Community version, the GPT extension requires manual installation.

Follow these steps to install the GPT extension in the Community version:

1. Navigate to **Help** from the main menu and select **Install New Software**.
2. Within the installation dialog, locate the **Work with** dropdown. Choose **DBeaver AI integration** from the list.



3. Check the box next to **AI Support**. Click **Next** and proceed through the installation prompts.
4. A **Trust Artifacts** window will appear. Ensure you select all the artifacts listed and click on **Trust Selected**.
5. Once the installation is complete, restart DBeaver.

After restarting, launch the SQL editor. On the left toolbar, you should now see the **GPT Chat** icon .

Client side scripting

You can use special commands in the SQL scripts.

These commands are executed on DBeaver's side, not on the server-side.

DBeaver supports the following commands:

Command	Database	Description
<code>@set var = value</code>	All	Sets a script variable. You can use expressions as a value. Variables can be used as SQL queries input parameters. For more information see Dynamic parameter bindings
<code>@unset var</code>	All	Unsets a script variable.
<code>@echo message</code>	All	Prints message to output log. You can use a macro in a message (for example <code>\${var}</code>).
<code>@include fileName</code>	All	<ul style="list-style-type: none">- Executes a specified file name,- Can be used in scripts,- Opens a new SQL console with the specified file and processes SQL queries as in a regular SQL editor.
<code>@export { ... }</code>	All	Opens the data transfer wizard with predefined settings. For more information see the main article .
<code>source fileName</code>	MySQL	The same as <code>@include</code> but in MySQL CLI syntax
<code>define var = value</code>	Exasol	The same as <code>@set</code> but in Exasol EXAPlus syntax.

Export Command

Table of contents

[Overview](#)

[Usage](#)

[Settings](#)

[Producer Settings](#)

[Consumer Settings](#)

[Processor Settings](#)

[CSV processor](#)

[DbUnit processor](#)

[HTML processor](#)

[JSON processor](#)

[Markdown processor](#)

[SQL processor](#)

[Source code processor](#)

[TXT processor](#)

[XML processor](#)

[XLSX processor](#)

Overview

The `@export` command allows you to open the data transfer wizard with prefilled settings.

It may be helpful in case you're editing several SQL queries and want to quickly perform the export of the produced results without creating any additional data transfer tasks.

Disclaimer: This article describes supported settings by the `@export` command, their purpose, and allowed values. Generally, this article contains every setting accessible in the data transfer wizard. Settings are written in the order they appear in the wizard, so you can always look at the wizard to quickly locate any of these settings.

Usage

The body of the command consists of JSON text, which looks like this:

```
{
  "type": <ID of the processor>,
  "producer": {
    <producer settings>
  },
  "consumer": {
    <consumer settings>
  },
  "processor": {
    <processor-specific settings>
  },
}
```

Due to certain limitations, it must be written on a single line, without line delimiters:

```
@export { "type": "csv", "producer": { ... }, "consumer": { ... }, "processor": { ... } }
```

The command itself doesn't do anything. It must be followed by any other query:

```
@export { "type": "csv", "producer": { ... }, "consumer": { ... }, "processor": { ... } }
SELECT * FROM Album;
```

You can either execute each line separately, or execute the entire script at once.

Settings

Here's the description of each attribute:

Attribute	Description
<code>type</code>	Type of the processor.
<code>producer</code>	Settings that affect how the data is extracted. See the full table of supported settings in the main section .
<code>consumer</code>	Settings that affect how the data is transformed before processing. See the full table of supported settings in the main section .
<code>processor</code>	Settings that affect how the data is processed. This includes formatting, transformations, etc. These settings are specific to the processor specified by the <code>type</code> attribute. See the full table of supported processors in the main section .

Producer Settings

Id	Name	Description	Type	Default Value	Allowed Values
<code>extractType</code>	Extract type	Data extraction mode. Denotes whether a single query or multiple segmented queries should be used to extract data.	String	<code>SINGLE_QUERY</code>	<code>SINGLE_QUERY</code> , <code>SEGMENTS</code>
<code>segmentSize</code>	Segment size	Specifies how many rows are read per segment during data extraction. See <code>extractType</code>	Integer	<code>100000</code>	Any
<code>fetchSize</code>	Fetch size	Number of rows to fetch per one server round trip. May greatly affect extraction performance.	Integer	<code>10000</code>	Any
<code>openNewConnections</code>	Open new connection (s)	Open new physical connection for data reading. Makes great sense if you are going to continue to work with your database during the export process.	Boolean	<code>true</code>	Any
<code>queryRowCount</code>	Select row count	Query row count before performing export. This will let you track export progress but may cause performance faults in some cases.	Boolean	<code>true</code>	Any

Consumer Settings

Id	Name	Description	Type	Default Value	Allowed Values
----	------	-------------	------	---------------	----------------

<code>formatterProfile</code>	Formatting Profile	Specifies the profile used for formatting data.	String		Any
<code>valueFormat</code>	Value Formatting	Specifies how the data is interpreted.	String	<code>UI</code>	<code>UI</code> , <code>EDIT</code> , <code>NATIVE</code>
<code>lobExtractType</code>	Binaries Policy	Specifies how binaries are processed.	String	<code>INLINE</code>	<code>SKIP</code> , <code>FILES</code> , <code>INLINE</code>
<code>lobEncoding</code>	Binaries Encoding	Specifies how binaries are encoded.	String	<code>BINARY</code>	<code>BASE64</code> , <code>HEX</code> , <code>BINARY</code> , <code>NATIVE</code>
<code>outputClipboard</code>	Copy to Clipboard	Specifies that the data should be copied to the clipboard rather written to files on a disk.	Boolean	<code>false</code>	Any
<code>outputFolder</code>	Output Directory	Output directory pattern. Specifies where the output files should be located.	String	N/A	Any
<code>outputFilePattern</code>	Output Filename	Output filename pattern.	String	<code>\${table}_\${timestamp}</code>	Any
<code>outputEncoding</code>	Output Encoding	Specifies the file encoding.	String	<code>UTF-8</code>	Any
<code>outputEncodingBOM</code>	Insert BOM	Specifies whether the byte order mark should be written to the output file. Common for encoding such as <code>UTF-16LE</code> , <code>UTF-16BE</code> , <code>UTF-32LE</code> , and <code>UTF-32LE</code> .	Boolean	<code>false</code>	Any

<code>outputTimestampPattern</code>	Timestamp Pattern	Pattern used for the <code>\${timestamp}</code> variable in <code>outputFolder</code> and <code>outputFilePattern</code> .	String	<code>yyyyMMddHHmm</code>	Any
<code>appendToFile</code>	Append to the end of the file	If file already exists, appends data at end of it. Only works against compatible processors.	Boolean	<code>false</code>	Any
<code>useSingleFile</code>	Write to the single file	Write all streams to the single file. Only works against compatible processors.	Boolean	<code>false</code>	Any
<code>compressResults</code>	Compress	Specifies whether the output file should be compressed using ZIP.	Boolean	<code>false</code>	Any
<code>splitOutFiles</code>	Split output file	Specifies whether the output file should be split using the <code>maxOutFileSize</code> threshold. If size exceeds this threshold, a separate file is created and so on.	Boolean	<code>false</code>	Any
<code>maxOutFileSize</code>	Maximum file size	Maximum size of a single file. See <code>splitOutFiles</code>	Integer	<code>10000000</code>	Any

Processor Settings

In this section, we will explore the variety of processor settings available in DBeaver. Navigate through the links provided in the **Options** column for detailed instructions on using each processor type.

Id	Name	Description	Options
----	------	-------------	---------

<code>csv</code>	CSV	Export to CSV file(s)	Navigate to CSV
<code>dbunit</code>	DbUnit	Export to DbUnit XML file(s)	Navigate to DbUnit
<code>html</code>	HTML	Export to HTML file(s)	Navigate to HTML
<code>json</code>	JSON	Export to JSON file(s)	Navigate to JSON
<code>markdown.table</code>	Markdown	Export to markdown file(s)	Navigate to Markdown
<code>sql</code>	SQL	Export to SQL INSERT statements	Navigate to SQL
<code>source.code</code>	Source code	Export to source code array	Navigate to Source Code
<code>txt</code>	TXT	Export to plain text format	Navigate to TXT
<code>xml</code>	XML	Export to XML file(s)	Navigate to XML
<code>xlsx</code>	XLSX	Export to XLSX (Excel spreadsheet) format	Navigate to XLSX

CSV processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>extension</code>	File extension		String	<code>csv</code>	<i>Any</i>
<code>delimiter</code>	Delimiter	Column delimiter. You can use special characters \ + t,n,r	String	<code>,</code>	<i>Any</i>
<code>rowDelimiter</code>	Row delimiter	Row delimiter. Default is system-specific line feed delimiter. You can use special characters \ + t,n,r	String	<code>default</code>	<code>default</code> , <code>\n</code> , <code>\r</code> , <code>\r\n</code> , <code>\n\r</code>
<code>header</code>	Header	CSV header settings	String	<code>top</code>	<code>none</code> , <code>top</code> , <code>bottom</code>
<code>headerFormat</code>	Header format	Header format	String	<code>label</code>	<code>label</code> , <code>description</code> , <code>both</code>
		Bad characters escaping model			

<code>escape</code>	Characters escape	(surrounded with quotes or escaped with '\' character)	String	<code>quotes</code>	<code>quotes</code> , <code>escape</code>
<code>quoteChar</code>	Quote character	Character which will be used to quote strings (space means no quote)	String	<code>"</code>	<i>Any</i>
<code>quoteAlways</code>	Quote always	Quote all cell values. Cannot be used with "quoteNever"	String	<code>disabled</code>	<code>disabled</code> , <code>all</code> , <code>strings</code> , <code>all but numbers</code> , <code>all but nulls</code>
<code>quoteNever</code>	Quote never	Do not quote cell values. Cannot be used with "quoteAlways"	Boolean	<code>false</code>	<i>Any</i>
<code>nullString</code>	NULL string	String which will be used instead of NULL values	String		<i>Any</i>
<code>formatNumbers</code>	Format numbers	Format numeric values using locale settings	Boolean	<code>false</code>	<i>Any</i>

DbUnit processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>upperCaseTableName</code>	Force upper case table name		Boolean	<code>true</code>	<i>Any</i>
<code>upperCaseColumnNames</code>	Force upper case column names		Boolean	<code>true</code>	<i>Any</i>
<code>extension</code>	File extension		String	<code>xml</code>	<i>Any</i>
<code>includeNullValues</code>	Include NULL values in export		Boolean	<code>true</code>	<i>Any</i>
<code>nullValueString</code>	Replace NULL values with		String	<code>[NULL]</code>	<i>Any</i>

HTML processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>extension</code>	File extension		String	<code>html</code>	<i>Any</i>

<code>tableHeader</code>	Output table header	Output query or table name as first row in generated table	Boolean	<code>true</code>	Any
<code>columnHeaders</code>	Output column headers	Output column names as extra row in generated table	Boolean	<code>true</code>	Any
<code>extractImages</code>	Images	Extract images to graphic files	Boolean	<code>true</code>	Any

JSON processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>printTableName</code>	Print table name		Boolean	<code>true</code>	Any
<code>formatDateISO</code>	Format dates in ISO 8601		Boolean	<code>true</code>	Any
<code>extension</code>	File extension		String	<code>json</code>	Any

Markdown processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>extension</code>	File extension		String	<code>md</code>	Any
<code>nullString</code>	NULL string	String which will be used instead of NULL values	String		Any
<code>formatNumbers</code>	Format numbers	Format numeric values using locale settings	Boolean	<code>false</code>	Any
<code>showHeaderSeparator</code>	Show header separator	Print header separator (---). Required for GitHub markdown.	Boolean	<code>true</code>	Any
<code>confluenceFormat</code>	Confluence format	Use Confluence format (special format of header and no separator line)	Boolean	<code>false</code>	Any

SQL processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>includeAutoGenerated</code>	Include generated columns	Include auto-generated columns (e.g. auto-increment) in SQL INSERT	Boolean	<code>false</code>	Any
<code>extension</code>	File extension		String	<code>sql</code>	Any
<code>userTableName</code>	Target table name	<p>You can specify the name of the target table. It will be used as is to generate an INSERT statement.</p> <p>If you want the table name also contains the scheme or/and catalog identifier, you can prescribe it in this field by yourself.</p> <p>By default, the name of the original table is used.</p>	String		Any
<code>nativeFormat</code>	Native date/time format	Use native date/time format in INSERT statements	Boolean	<code>true</code>	Any
<code>omitSchema</code>	Omit schema name	Omit schema/catalog name in INSERT statements	Boolean	<code>false</code>	Any
<code>rowsInStatement</code>	Data rows per statement	Number of data rows per single insert statement	Integer	<code>10</code>	Any
	Insert line	Insert line feed before			

<code>lineBeforeRows</code>	before rows	values (for multi-row inserts)	Boolean	<code>true</code>	Any
<code>keywordCase</code>	Keyword case	You can choose lower or upper keyword case	String	<code>upper</code>	<code>upper</code> , <code>lower</code>
<code>identifierCase</code>	Identifier case	You can choose lower or upper keyword case for table and column names	String	<code>as is</code>	<code>as is</code> , <code>upper</code> , <code>lower</code>
<code>upsertKeyword</code>	Upsert keyword	You can choose different upsert keywords	String	<code>INSERT</code>	<code>INSERT</code> , <code>INSERT ALL</code> , <code>UPDATE OR</code> , <code>UPSERT INTO</code> , <code>REPLACE INTO</code> , <code>ON DUPLICATE KEY UPDATE</code> , <code>ON CONFLICT</code>
<code>insertOnConflict</code>	On conflict expression	Expression for the end of the statement. Enter the required value in this field. This is database specific setting	String		Any

Source code processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>language</code>	Language	Programming languages	String	<code>PHP &lt; 5.4</code>	<code>PHP &lt; 5.4</code> , <code>PHP 5.4+</code>
<code>formatDateISOPHP</code>	Format dates in ISO 8601		Boolean	<code>true</code>	Any
<code>extension</code>	File extension		String	<code>php</code>	Any
<code>quoteChar</code>	Quote character	Character which will be used to quote strings	String	<code>"</code>	<code>"</code> , <code>'</code>
		Row delimiter. Default is			

<code>rowDelimiter</code>	Row delimiter	system-specific line feed delimiter. You can use special characters \ + t,n,r	String	<code>default</code>	<code>default</code> , <code>\n</code> , <code>\r</code> , <code>\r\n</code> , <code>\n\r</code>
---------------------------	------------------	--	--------	----------------------	--

TXT processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>extension</code>	File extension		String	<code>txt</code>	Any
<code>batchSize</code>	Batch size		String	<code>200</code>	Any
<code>minColumnLength</code>	Min column length	Minimum column length. If the cell value is smaller, then spaces will be added to the specified length.	String	<code>3</code>	Any
<code>maxColumnLength</code>	Max column length	Maximum column length. If the cell value is longer than the specified parameter, then it will be crop.	String	<code>0</code>	Any
<code>showNulls</code>	Show NULLs	Show NULLs if this setting is enabled or nothing if disabled	Boolean	<code>false</code>	Any
<code>delimHeader</code>	Show header delimiter	Will add hyphen characters in the first row without a header, or between the header and data.	Boolean	<code>true</code>	Any
<code>delimLeading</code>	Show leading delimiter	Will add a pipe character at the start of the row.	Boolean	<code>true</code>	Any
<code>delimTrailing</code>	Show trailing delimiter	Will add a pipe character at the end of the row.	Boolean	<code>true</code>	Any
<code>delimBetween</code>	In- between	Will add the chosen custom character (ex. pipe, space, or nothing) between	String	<code> </code>	Any

	delimiter	data values.			
<code>showHeader</code>	Print header	Print column names on the top. You can disable this setting for the multi-export along with the header delimiter setting	Boolean	<code>true</code>	Any

XML processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>extension</code>	File extension		String	<code>xml</code>	Any
<code>includeDoctype</code>	Include DOCTYPE declaration		Boolean	<code>false</code>	Any

XLSX processor

Id	Name	Description	Type	Default Value	Allowed Values
<code>extension</code>	File extension		String	<code>xlsx</code>	Any
<code>rownumber</code>	Row number (s)	Set row index as first column	Boolean	<code>false</code>	Any
<code>border</code>	Border style	Cell borders style	String	<code>THIN</code>	<code>NONE</code> , <code>THIN</code> , <code>THICK</code>
<code>nullString</code>	NULL string	String which will be used instead of NULL values	String		Any
<code>header</code>	Header format	Header format	String	<code>label</code>	<code>label</code> , <code>description</code> , <code>both</code> , <code>none</code>
<code>headerfont</code>	Header row font	First row font properties	String	<code>BOLD</code>	<code>NONE</code> , <code>BOLD</code> , <code>ITALIC</code> , <code>STRIKEOUT</code> , <code>UNDERLINE</code>

<code>trueString</code>	Boolean string TRUE	String which will be used instead of TRUE boolean values	String	<code>true</code>	<i>Any</i>
<code>falseString</code>	Boolean string FALSE	String which will be used instead of FALSE boolean values	String	<code>false</code>	<i>Any</i>
<code>trimString</code>	Trim strings	Trim all string values (remove extra leading and trailing spaces).	Boolean	<code>false</code>	<i>Any</i>
<code>exportSql</code>	Export SQL	Export SQL to a second sheet	Boolean	<code>false</code>	<i>Any</i>
<code>splitSqlText</code>	Split SQL Text	Split exported SQL on rows by CR	Boolean	<code>false</code>	<i>Any</i>
<code>splitByRowCount</code>	Max row on sheet	Split by row count	Integer	<code>1048575</code>	<i>Any</i>
<code>splitByColNum</code>	Column group	Column number for grouping rows on sheet by column value	Integer	<code>0</code>	<i>Any</i>
<code>dateFormat</code>	Excel date format	Excel date and time format (e.g. m/d/yy h: mm) it can be changed in Excel application	String	<code>m/d/yy</code>	<code>m/d/yy</code> , <code>d-mmm-yy</code> , <code>d-mmm</code> , <code>mmm-yy</code> , <code>h:mm AM/PM</code> , <code>h:mm:ss AM/PM</code> , <code>h:mm</code> , <code>h:mm:ss</code> , <code>m/d/yy h:mm</code>
<code>appendStrategy</code>	Append strategy	A strategy that will be used when appending data to an existing file	String	<code>create new sheets</code>	<code>create new sheets</code> , <code>use existing sheets</code>

ER Diagrams overview

Table of contents

[Selection of Elements in Diagrams](#)

[Structure Adjustment](#)

[View Adjustment](#)

[Refresh](#)

[Diagram Notations and Routing types](#)

[Notations](#)

[Routing types](#)

[Setting Notation and Routing type](#)

[Notes](#)

[Search in Diagram Entities](#)

[Bindings](#)

[Navigation and selection](#)

[Table Manipulation](#)

[Focus](#)

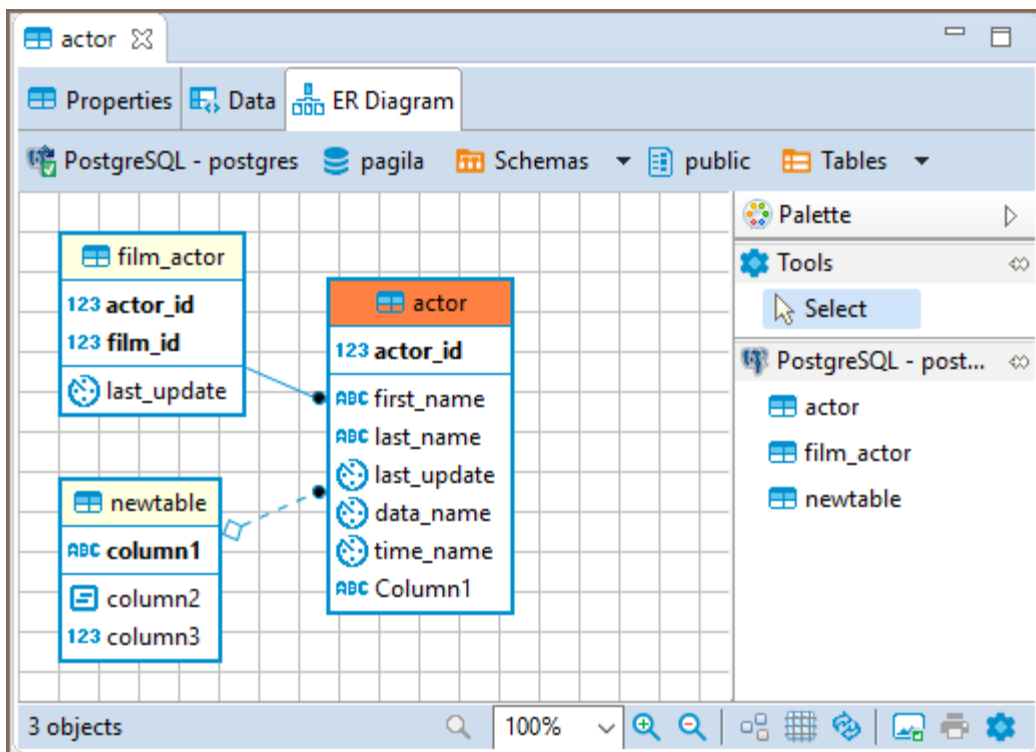
[Other Functions](#)

[Diagram Export](#)

[Diagram Printing](#)

[Settings](#)

ER diagrams appear on the rightmost tab of the [Database Object Editor](#):



Entity Relation Diagrams (ERD) are graphic presentations of database entities and the relations between them.

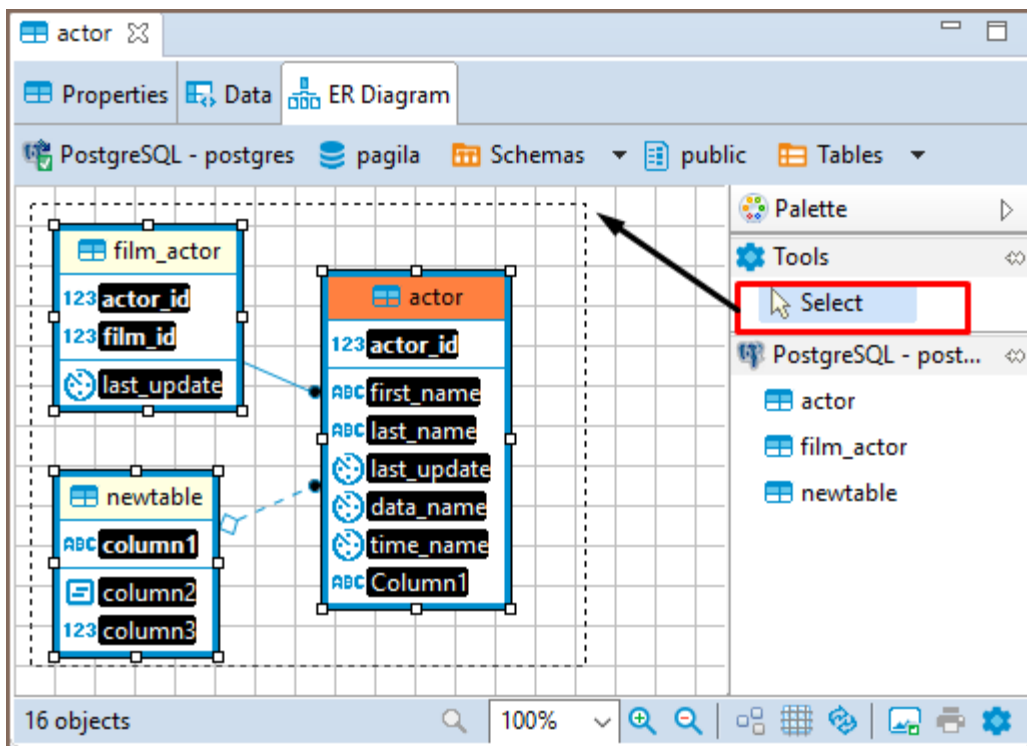
DBeaver allows you to view the diagrams of existing tables and whole database schemas, see [Database Structure Diagrams](#). DB also allows the creation of custom diagrams, see [Custom Diagrams](#).

Both types of diagrams provide the same tools to adjust their view and structure. They can be printed and exported to image file formats.

Selection of Elements in Diagrams

You can use one of the two tools to select elements in diagrams:



- **Select** – supports both, single and multi-select modes. To select a single element (table, connection, entity inside a table) in a diagram, just click that element. To select multiple elements, similar to using the Marquee tool, click outside the first element and draw until all elements you need are in focus:

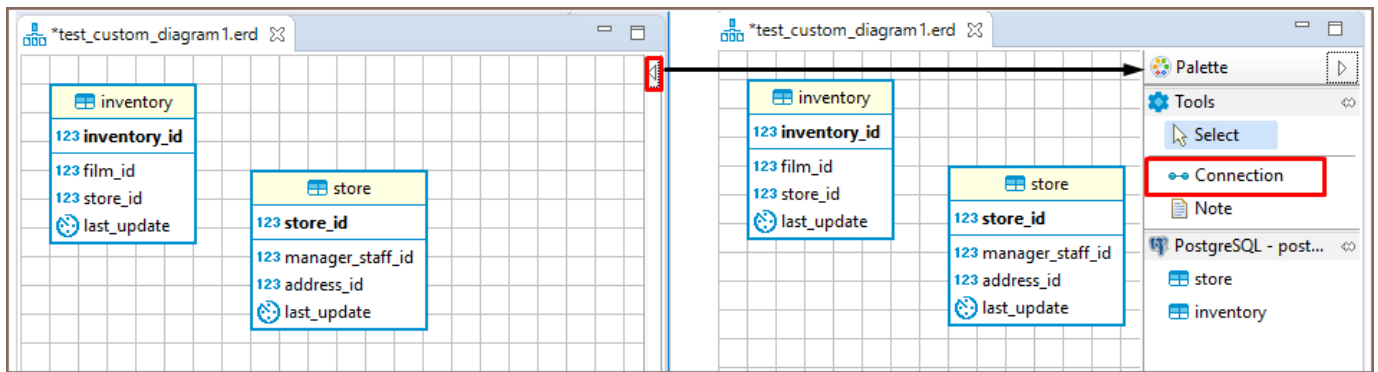


Structure Adjustment

NOTE: All changes to existing database schemas cannot be saved and are intended for exploration purposes only.

You can do the following structural changes in the diagrams.

- Add new tables to a diagram by drag-n-dropping them onto the diagram field from the [Database Navigator](#).
- Rearrange tables in the diagram by dragging them all over the space. You can select several tables and drag them to a new location.
- Auto-arrange tables into a compact view after manual rearrangements: click the **Arrange Diagram** () in the toolbar or on the context menu (right-click anywhere on the diagram tab).
- (Available for [Custom Diagrams](#) only) - connect tables with a connector: click the **Show Palette** button () in the upper-left corner of the diagram tab and then, in the Palette panel, click **Connection**:





Now click the tables that you want to connect with each other in turn, one by one. To stop the connection line, double-click the last table

- (Available for [Custom Diagrams](#) only) - removes tables and connections: right-click the table or connection and click **Delete** on the context menu or just click the table or connection and press **Delete**.

View Adjustment

You can adjust the view of any diagram in the following ways:

- Enable/disable the diagram grid: Click **Toggle Grid** () in the toolbar.
- Modify attributes visibility: Right-click the diagram and, on the context menu, click **Show Attributes** and then select one of the options:
 - **All** - all attributes
 - **Any keys** - primary and foreign keys
 - **Primary key** - only primary keys
 - **None** - no attributes
- Modify attributes presentation: Right-click the diagram and, on the context menu, click **View Styles** and then select one of the options:
 - **Show Icons**
 - **Show Data Types**
 - **Show Nullability**

- **Show Comments**
- **Show Fully qualified names**
- Change the color of the entities/notes: Right-click the header of the entity or comment and then click **Set color** on the context menu. Then you can select the color and click **OK**.
- For elements located in front of/behind others, bring an element to the front or send it to the back: Right-click the element and then click **Bring to front** / **Send to back** on the context menu.
- Zoom the diagram in/out: Click the **Zoom In/Zoom Out** buttons or choose the scaling value in the dropdown list in the toolbar: 

Refresh


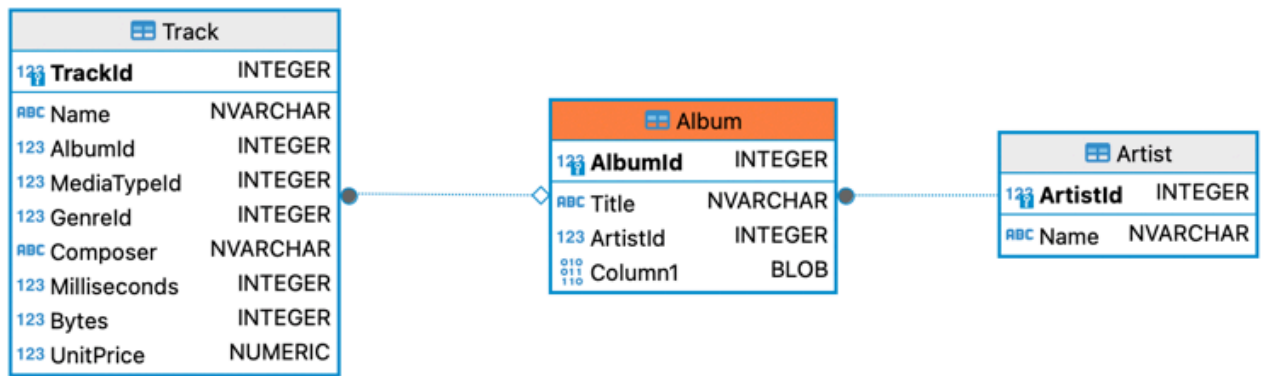
To see changes made by others to the database schema, you might need to refresh the diagram: click **Refresh Diagram** () in the toolbar.

Diagram Notations and Routing types

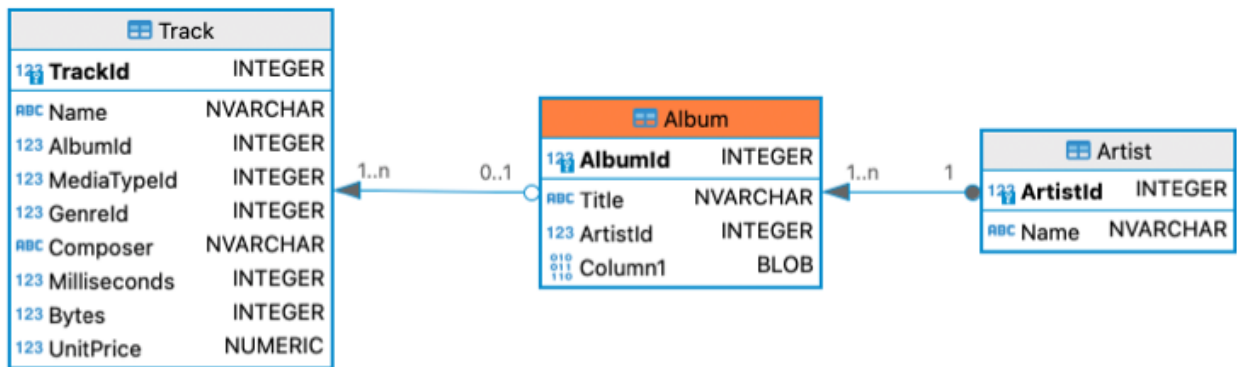
DBeaver offers a variety of notations and routing types for ER Diagrams, allowing you to customize database structure representations to suit your needs.

Notations

- **IDEF1X** (default): Highly recommended for designing relational databases. It places a strong emphasis on detailing entity relationships and constraints. For more information, see [IDEF1X Notation](#).

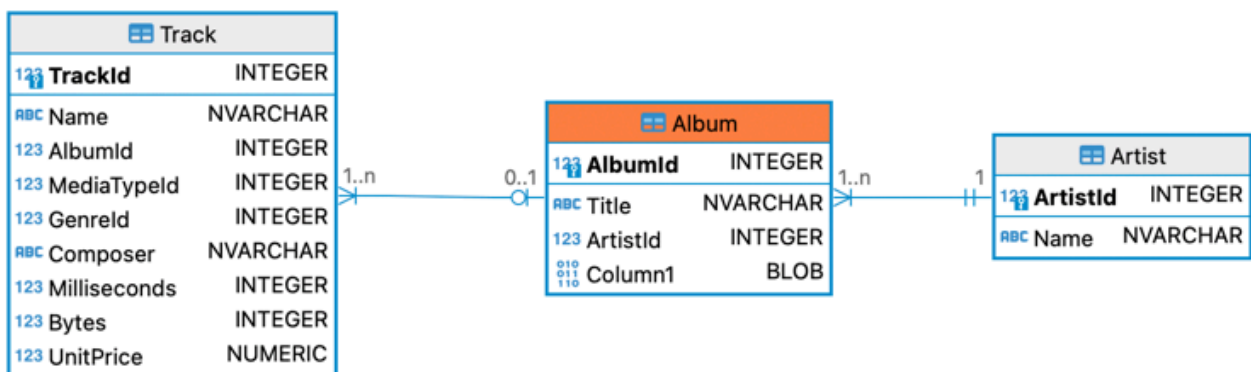


- **Bachman:** A notation that is particularly useful for data processing diagrams and reflects the data structure of the designed system from the data management perspective. For a more detailed understanding, refer to [Bachman Notation](#).



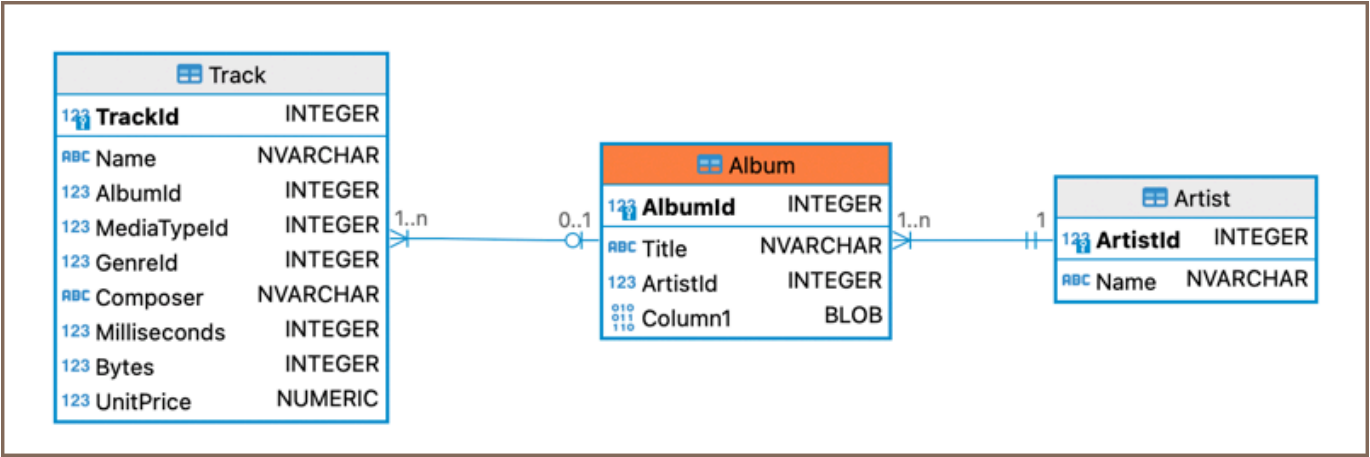
Note: Bachman notation is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

- **Crow's Foot:** This notation is widely used and is particularly intuitive for representing cardinality and relationships between entities. More information is available on [Crow's Foot Notation](#).



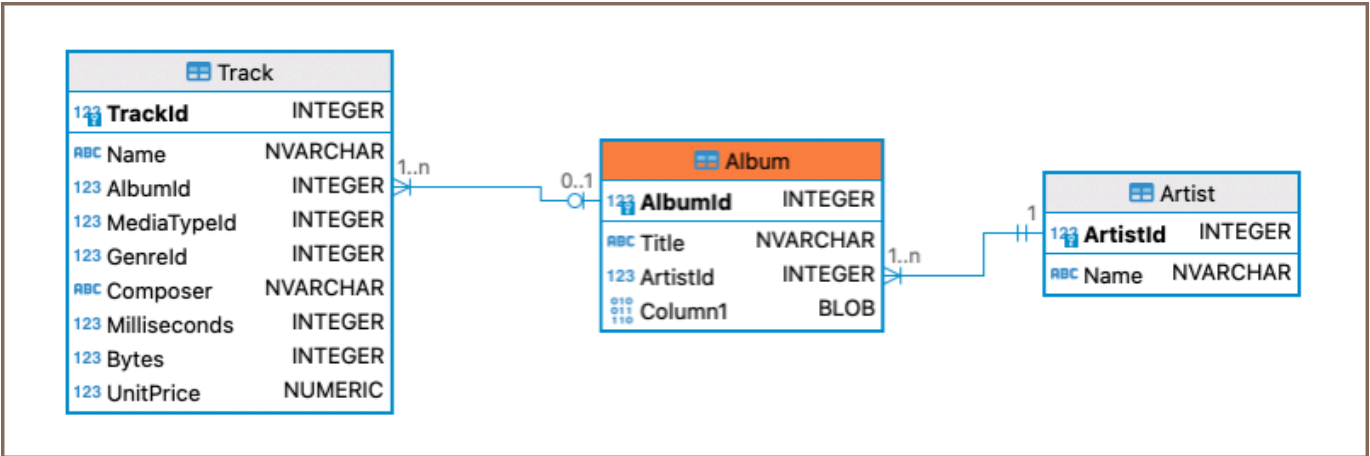
Routing types

- **Shortest paths** (default): Calculate and display the shortest possible lines connecting entities, ensuring a compact and efficient diagram representation.



Tip: Click on a connection to view detailed relationships.

- **Orthogonal paths**: Uses right-angled lines for clear, structured layouts; Showing direct relationships between tables and columns.




Note: Orthogonal paths are available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Setting Notation and Routing type


To set notation or routing type in an ER Diagram:

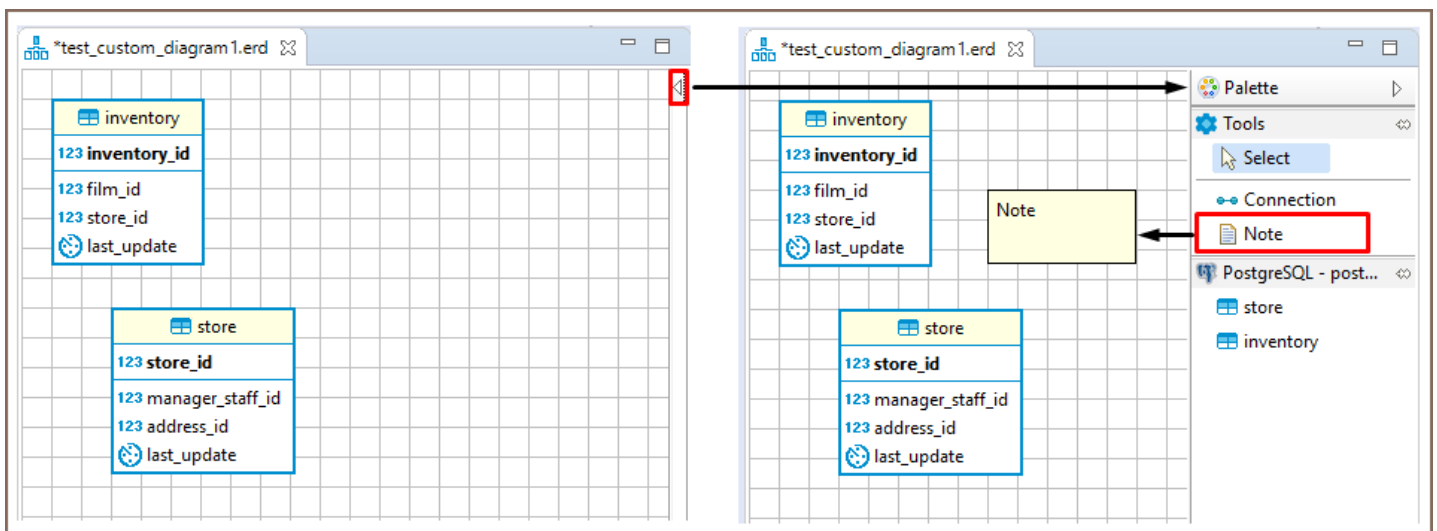
1. Right-click on an empty space within the ER Diagram area.
2. From the context menu, hover over **Notation** or **Routing** for the respective settings.
3. Select your preferred option from the submenu.

Alternatively, adjust these settings through the ER Diagram toolbar:


1. Click the **Settings** button .
2. In the **Advanced** section, find the **Notation type** or **Routing type** dropdown menu.
3. Choose your preferred setting.

Notes

You can create notes only in [Custom Diagrams](#). To create a note, click the **Show Palette** button () in the upper-left corner of the diagram tab. Then, in the Palette panel, click **Note** and click anywhere in the diagram tab. Now you can double-click the **Note** box to enter the note text:



Search in Diagram Entities

To search among entities of a diagram, click the **Search items** button () in the toolbar, then type in the search combination. The entities that contain the search combination are highlighted in the diagram. To remove the filter, click the cross icon next to the search field.

Bindings

Navigation and selection

- Use **ARROWS** to navigate between tables.



Press **SHIFT** | **#** + **ARROWS** to select multiple tables.

- Press **CTRL** | **#** + **ARROWS** to select additional tables using **SPACEBAR**.
- Press **SPACEBAR** to select the current table.

Table Manipulation

- Press **.** (period) to change the mode to move/resize tables, then use **ARROWS** and **ENTER** | **#** to move/resize tables.

Note: To use this feature on macOS, you may need to enable accessibility permissions for DBeaver. This can be done in **System Preferences -> Security & Privacy -> Accessibility**.

Focus

- Press **ENTER** | **#** to focus on attributes in the table.
- Press **BACKSPACE** | **#** to leave the focus.
- Use **|**, **,**, **?**, **** to focus on associations.
- Press **ALT/FN + 1** | **#/FN + 1** to focus on the diagram.
- Press **ALT/FN + 2** | **#/FN + 2** to focus on the palette.
- Press **ALT/FN + 3** | **#/FN + 3** to focus/open the outline.
- Press **ALT/FN + 4** | **#/FN + 4** to focus on the parameter view.

Other Functions

- Press **CTRL + SHIFT + ENTER** | **# + SHIFT + #** to open the selected table diagram.

Diagram Export



You can export (save) a diagram as an image (PNG, GIF, BMP formats) or as a file in GraphML format. To export a diagram, click **Save diagram in external format** () in the toolbar.

Diagram Printing

To print a diagram, press `CTRL+P` or click **Print Diagram** () on the toolbar.

Settings

To modify the diagram settings, click **Configuration** () on the toolbar.

Database Structure ER Diagrams

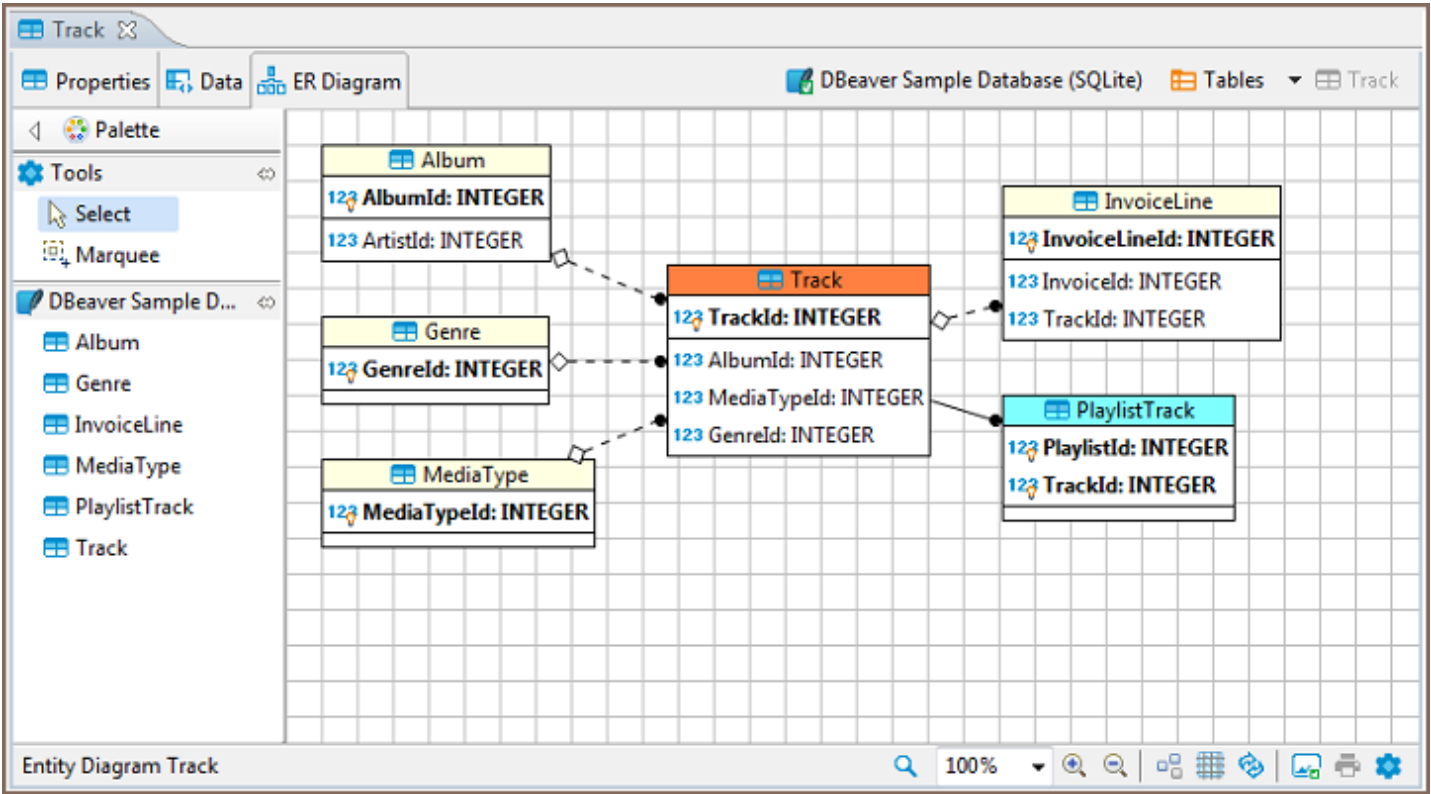
Table of contents

- [Overview](#)
- [Relationship Notation](#)

Overview

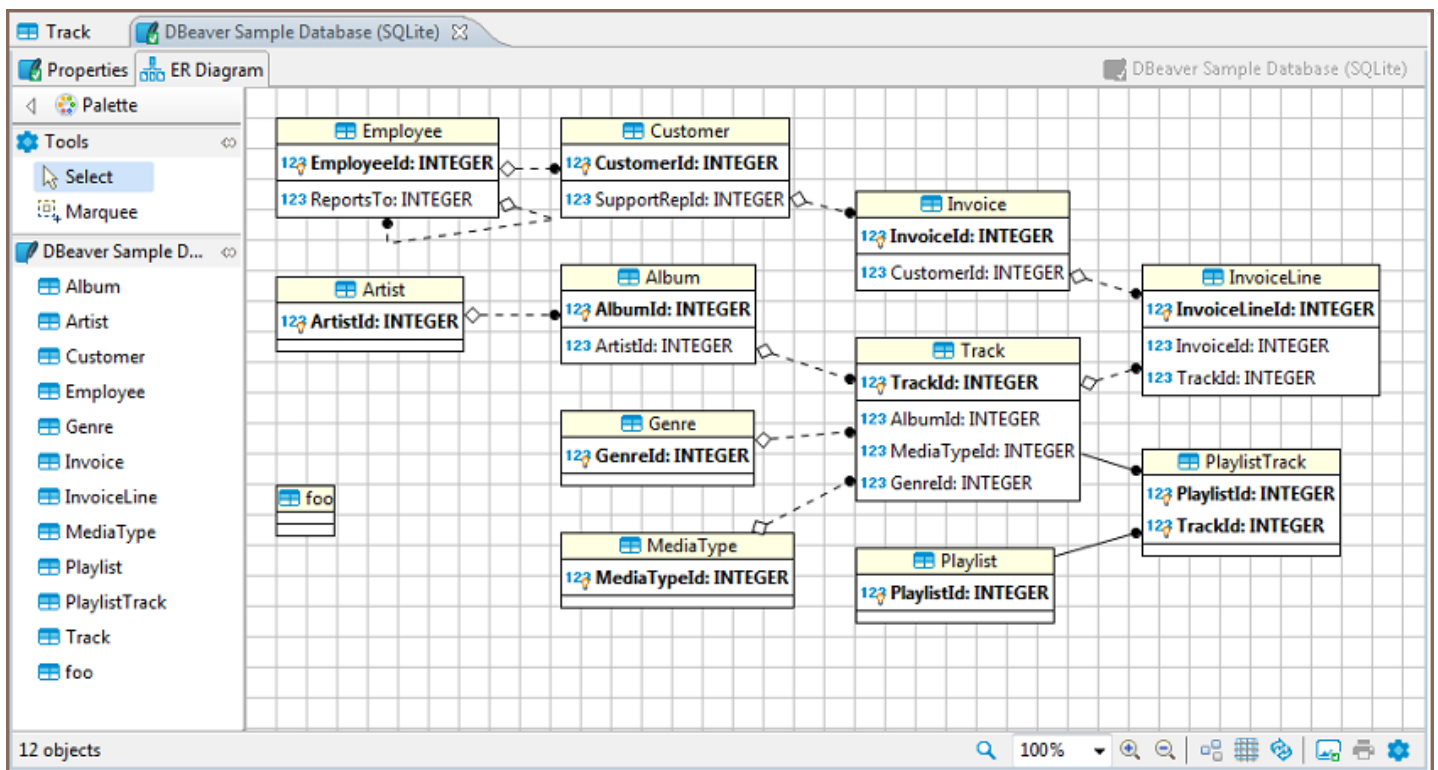
You can view a database structure in the standard ERD (Entity Relation Diagram) form. ER diagrams are available for all tables and schemas (databases).

The ER diagram for a table shows the table itself and its relations with other tables inside the schema. To view the ER diagram for a table or view, double-click the table or view in the [Database Navigator](#) and then, in the [Database Object Editor](#), switch to the **ER Diagram** tab:



To view the ER diagram for a full database schema, double-click the schema name in the Database Navigator or the previous node in the path (usually - **Tables**):

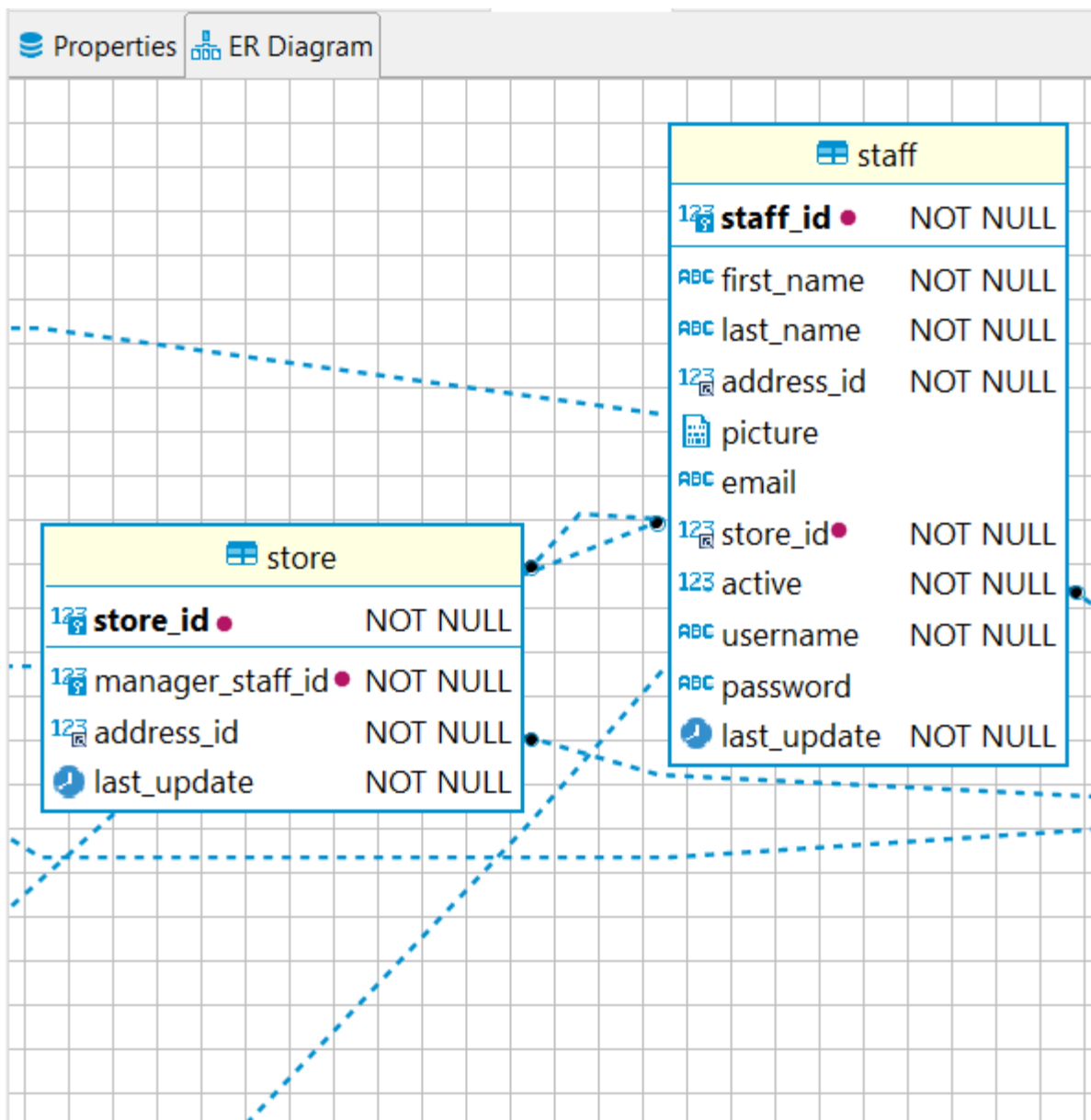




NOTE: Table and schema diagrams are read-only. You can rearrange the layout, drag-n-drop elements inside a diagram but you cannot save the changes state or delete/add anything. This is because the diagrams represent the actual state of databases.

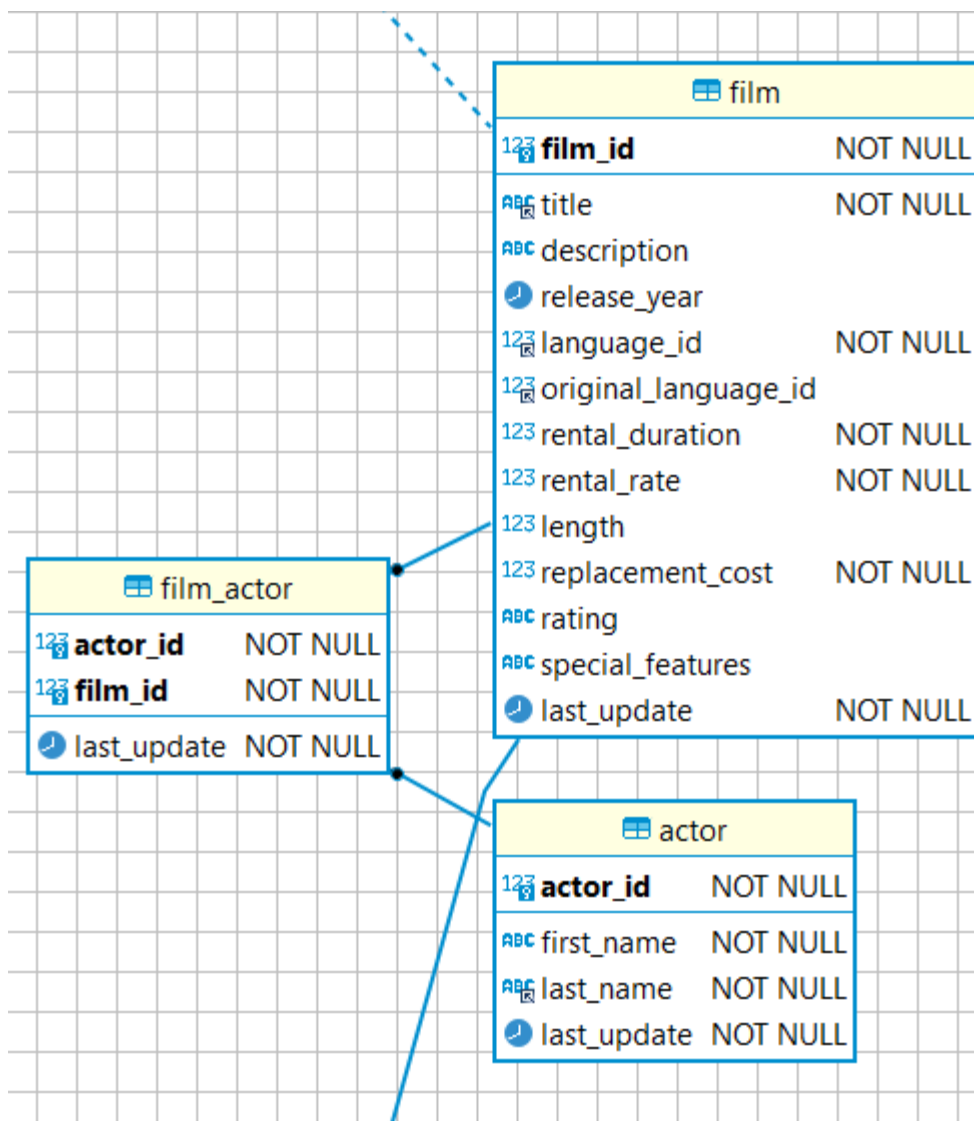
Relationship Notation

Lines representing the relationship between tables can look different depending on the nature of the relationship. Please note that any line can have only one beginning and one end. We may have links between two tables not only for one pair of attributes, which means that several associations will be displayed:

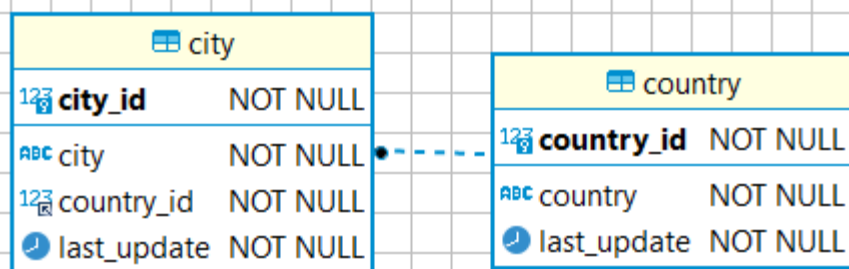


Notation	Description
	A solid line means that the foreign key column in one table is a primary key in another table
	A dashed line means that the foreign key column is not a primary key but a regular column in another table
	A black dot is used as the start of the line
	A diamond is used at the end of the link when the column in the source table is optional (there is no NOT NULL)

A one-to-one relationship is always a solid line due to the unique primary key - foreign key relationship:



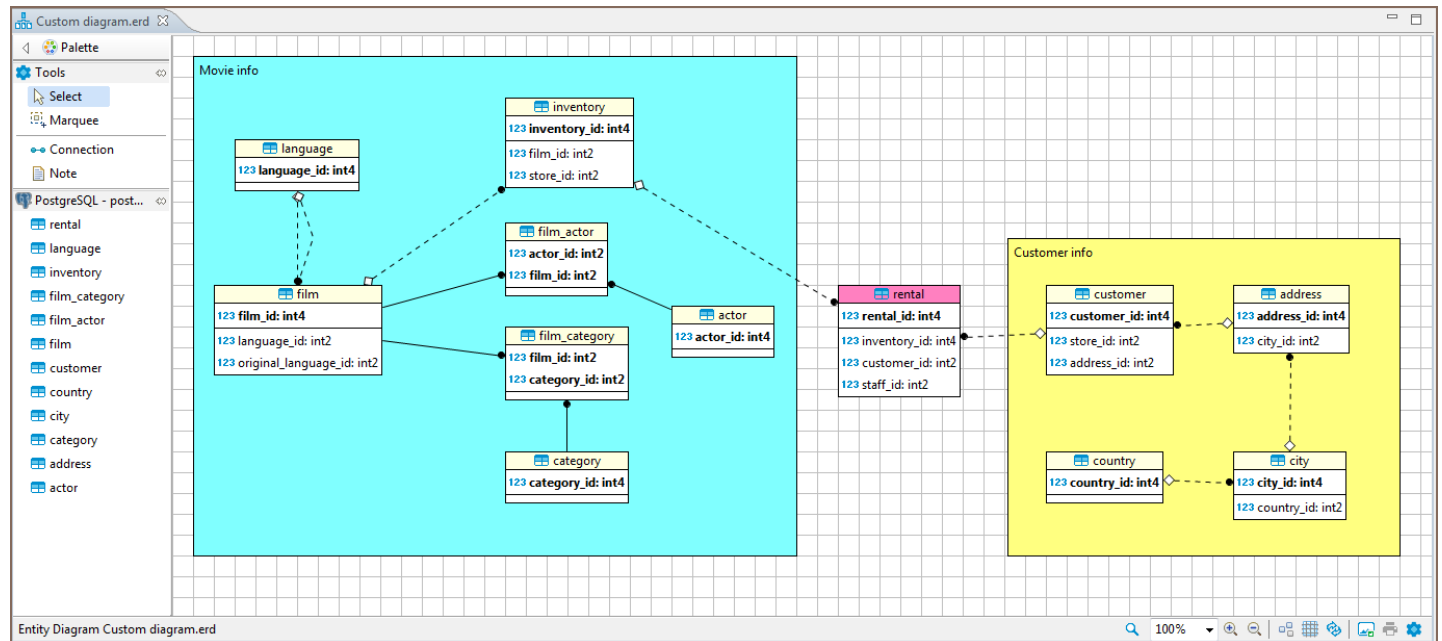
A dashed line is for the one-to-many relationship:



Custom ER diagrams

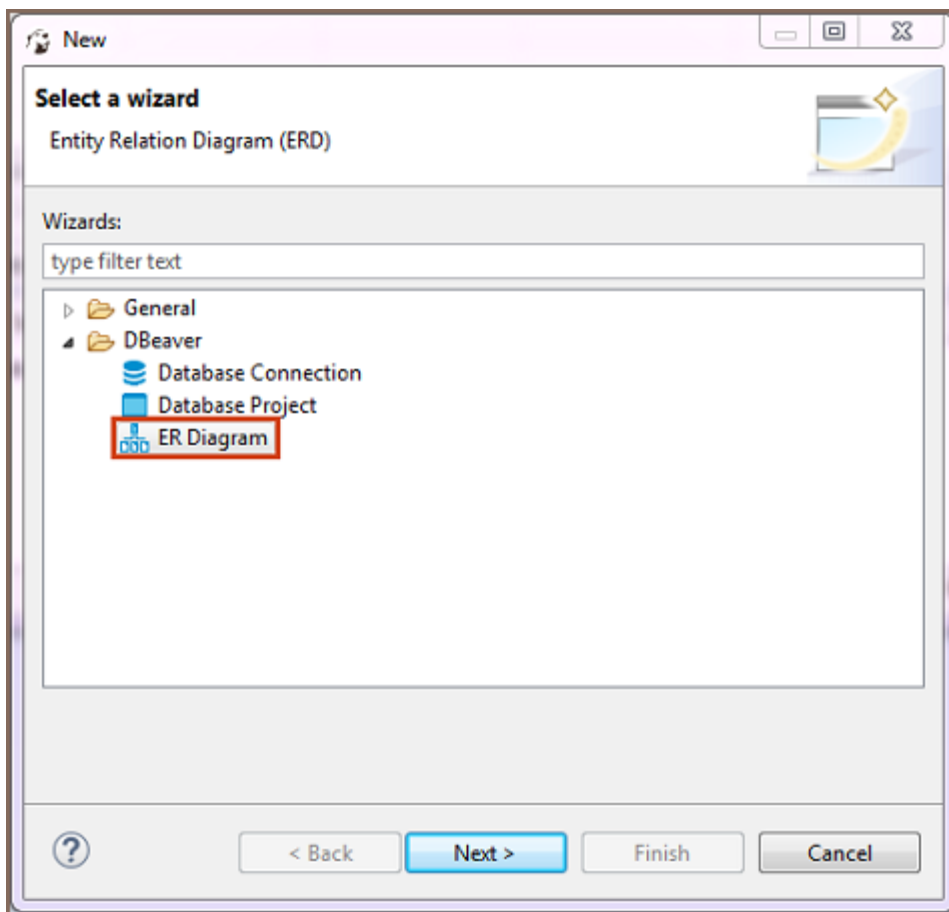
You can create custom ER diagrams that can contain any tables, relations and notes.

However, even custom diagrams may contain only real existing database entities (tables).

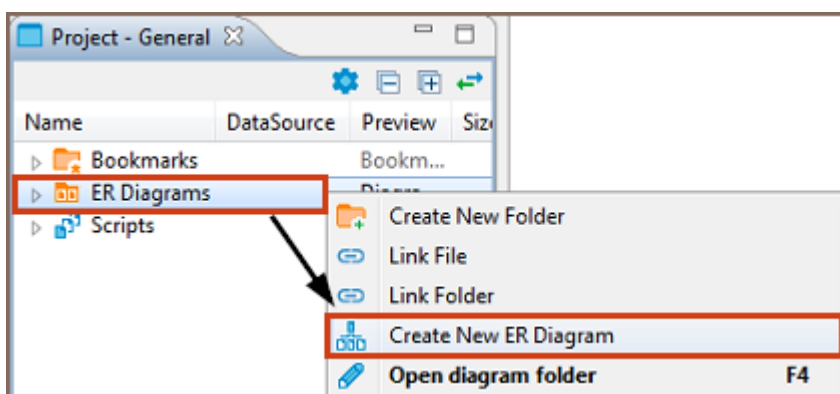


You can create a custom diagram in one of the ways:

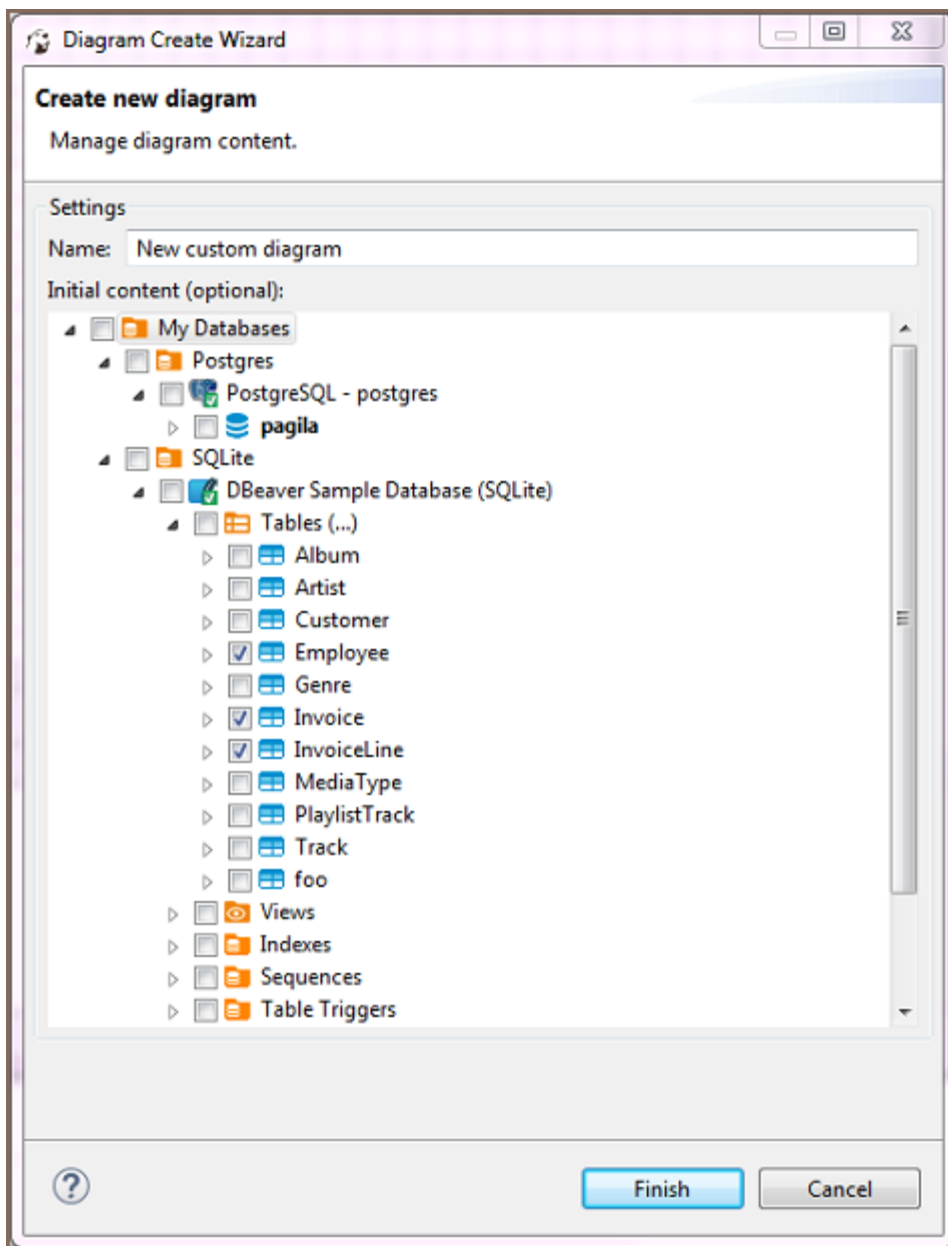
1. On the DBeaver main menu, click **File -> New**. Then in the new diagram wizard, click **DBeaver -> ER Diagram**, and then **Next**:



2. In the [Project Explorer](#) view, right-click the **ER Diagrams** node and then click **Create New ER Diagram** on the context menu.



In both cases, in the Diagram Create Wizard, specify the diagram name and (optional) choose the initial diagram contents (set of tables):



The new diagram appears in a separate editor. Now you can drag-n-drop any number of tables into it. You can add tables from different connections as well as from different database types (for example, combine Oracle and MySQL tables in one and the same diagram).

You can also add notes and custom relations (associations) using the ERD palette on the left side of diagram tab - see details in the [ER Diagrams](#) article. For example, to create a diagram similar to the one shown at the beginning of this article, you need to:

1. Add required tables and relationships between them and move them around to create a well-shaped structure (see *Structure Adjustment* section of the [ER Diagrams](#) article).
2. Add notes (see the *Notes* section of the [ER Diagrams](#) article).

3. Stretch the notes to cover the intended tables, then send the notes to the back, and then set a color to the tables and notes (see the *View Adjustment* section of the [ER Diagrams](#) article).

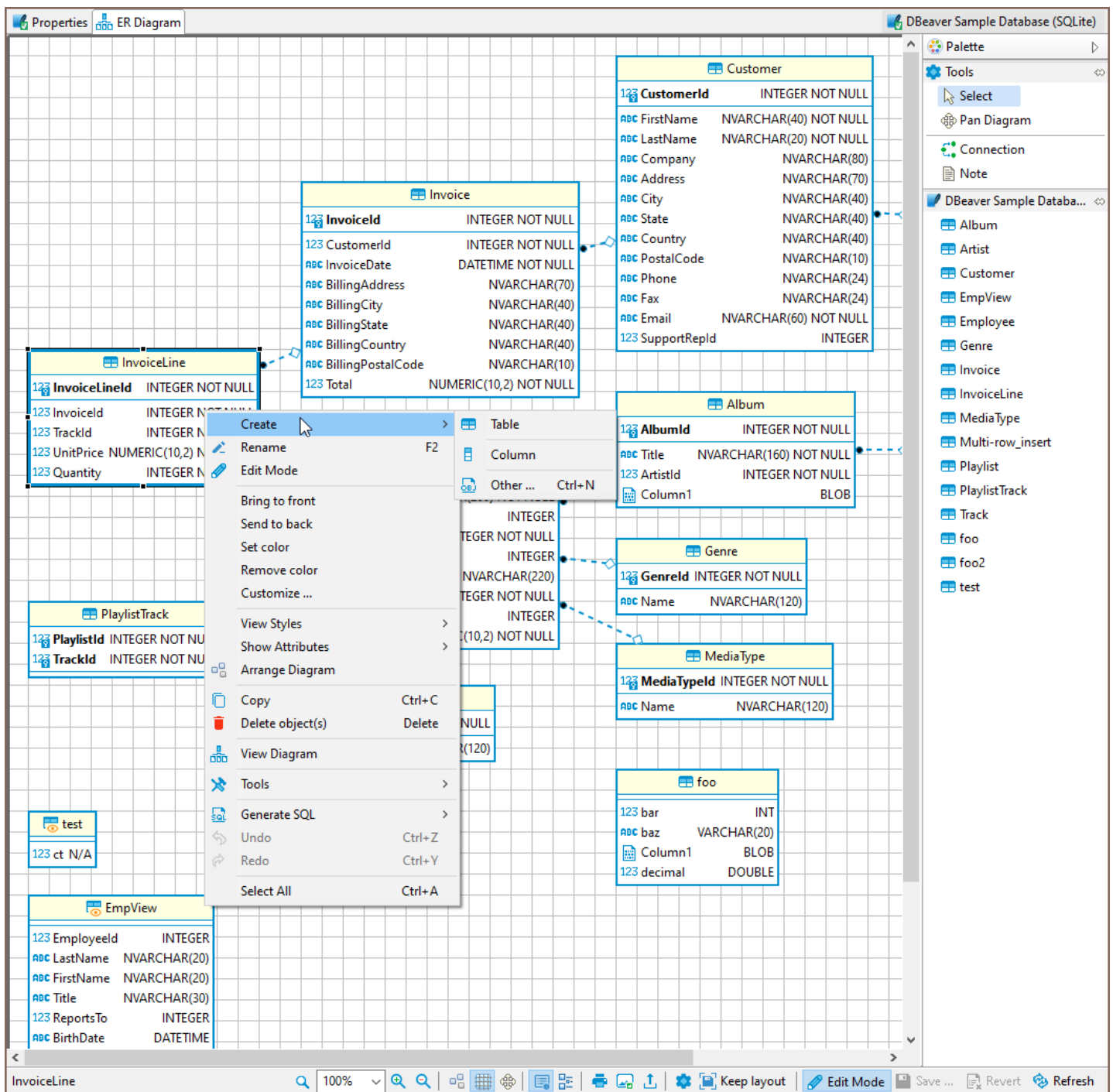
Undo/redo functions are fully supported in diagram editing.

Edit mode for ER Diagrams

Note: This feature is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Edit mode for ER Diagrams is a special feature that lets you create database objects while using the visual presentation of ER Diagrams. It can be accessed on the ERD tab of any database object that supports it. Changes made to a diagram in edit mode will generate an SQL script that can be executed to persist all the changes made. You can enable Edit mode by either choosing it from the context menu or by clicking a button on the bottom toolbar.

To create an object you need to right-click on the diagram and select an object that you want to create. Foreign keys are created similarly to virtual foreign keys in custom diagrams. Just drag a column from a table for which the foreign key is being added and drop it in the referenced table. This will open a window that lets you choose a unique key in the referenced column.



Saving changes to a database is performed by clicking a save button. This will show you a preview of a generated script that you then can execute or cancel. The revert button will cancel all the changes that you did to a diagram after the last save.

Properties Editor

Table of contents

- [Overview](#)
- [Key Operations](#)
- [Toolbar controls](#)
- [Context menu](#)

Overview

When you open the **Properties** tab of the [Database Object Editor](#), you'll encounter the following interface elements:

Database object's properties

Columns

Column Name	#	Data type	Identity	Collation	Not Null	Default	Comment
123 film_id	1	serial4			[v]	nextval('film_film_id_seq'::regclass)	some column
123 title	2	varchar(255)		default	[v]		
123 description	3	text		default	[]		
123 release_year	4	year			[]		
123 language_id	5	int2			[v]		
123 rental_duration	6	int2			[v]	3	
123 rental_rate	7	numeric(4, 2)			[v]	4.99	
123 length	8	int2			[]		
123 replacement_cost	9	numeric(5, 2)			[v]	19.99	
123 mpaa_rating	10	mpaa_rating			[]	'G'::mpaa_rating	
123 last_update	11	timestamp			[v]	now()	
123 special_features	12	text		default	[]		
123 fulltext	13	tsvector			[v]		

Sub-entities properties

Tabbed editors

Permissions

DDL

Virtual

Toolbar

- **Database object's properties:** displays the properties of the current database object.
- **Tabbed editors:** These vary depending on your database. Depending on the database you're working with, you'll see different tabs reflecting that particular database's unique characteristics and requirements.








- **Sub-entities properties:** These contain the properties of the object's sub-entities. To access these properties, click the tabs in the **Tabbed Editors**.
- The **Toolbar** provides tools for managing these sub-entities.



Key Operations

To interact with the **Properties Editor** page, you can use [Toolbar controls](#), execute commands from [context menu](#) associated with sub-entities, and utilize related [keyboard shortcuts](#).

Toolbar controls

The toolbar at the bottom of the editor provides the following tools for the majority of sub-entities, except for some specific ones like SQL-based views ([DDL and Source](#)):

Button	Name	Description
	Search items	Displays a search field next to the button: <ul style="list-style-type: none"> - Type in the search combination - the content updates dynamically - To remove the filter, click the cross icon next to the search field.
	Filter settings	Opens the Filtering window, which allows setting a custom filter, see Configure Filters .
	Configure columns	Opens the Configure columns dialog box to select the columns to display or hide in the current view.
	Refresh the selected items	Depending on the database type, refreshes either the current item or its parent or the whole database object – reloading data from the database.
	View	Opens an editor/viewer for the item currently in focus.
	Create new [items]	Creates a new item of the same type as currently displayed in the open view, for example, a column .
	Delete database object	Deletes the item currently in focus.

	<p>Save the current contents</p>	<ul style="list-style-type: none"> - Same as the Save button on the main application toolbar - Same as <code>Ctrl+S</code> (or <code>⌘S</code> for MacOS) <p>Opens the Persist Changes window that allows saving changes in the currently open sub-entity.</p>
	<p>Revert to the last saved state</p>	<p>Reverts all changes made to the database object to the previous saved state.</p>

Context menu

The **Properties** of the sub-entities have context menus that offer the same commands as the [Database Navigator](#).

To access these menus, right-click on the item.

SQL Code Editor

Table of contents

[Overview](#)

[Configuring SQL Code Editor settings](#)

[Miscellaneous settings](#)

[Query analysis settings](#)

[Auto close settings](#)

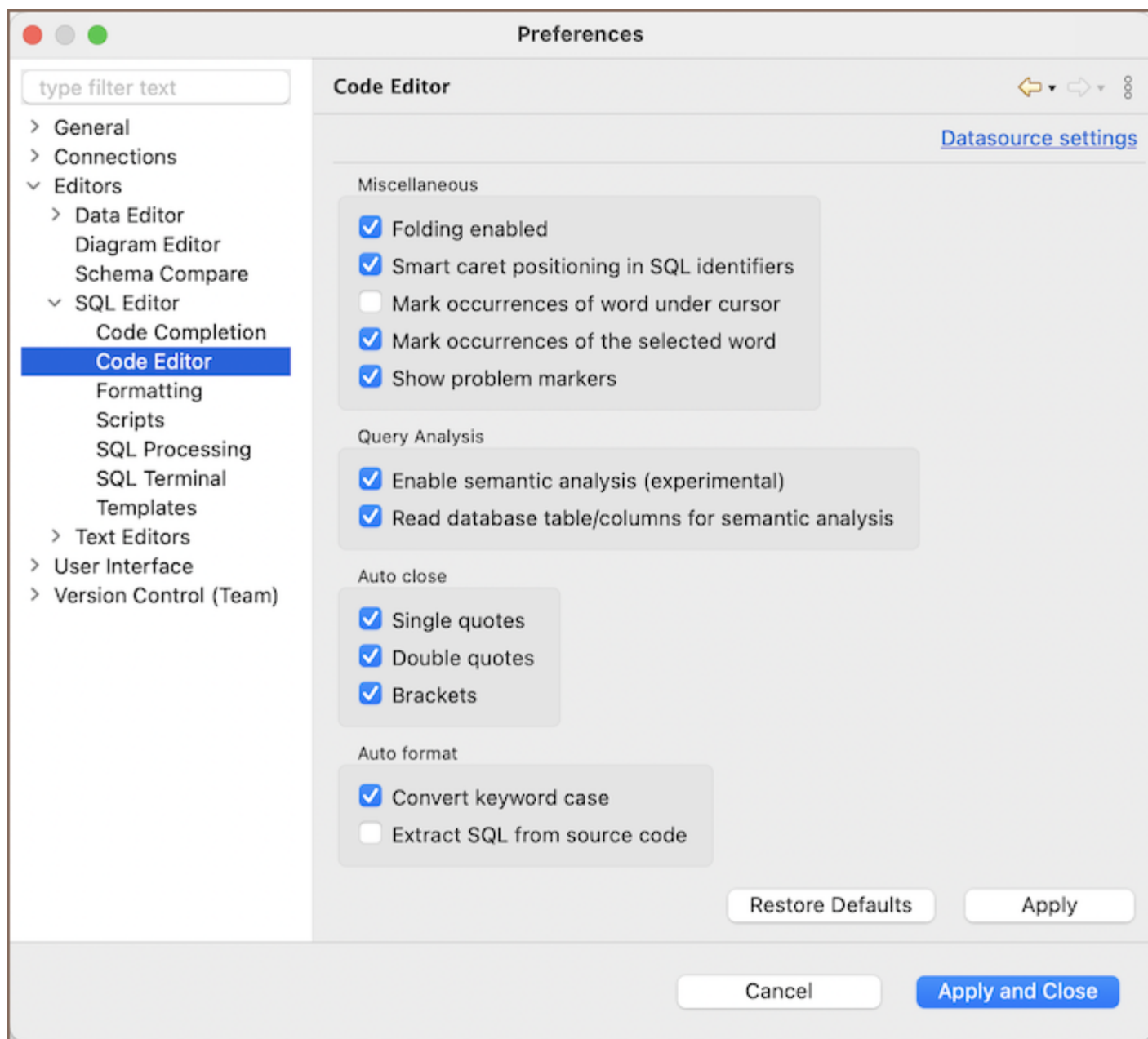
[Auto format settings](#)

Overview

The SQL Code Editor within DBeaver's SQL Editor provides you with advanced functionalities for SQL script management. This includes displaying problem markers, automatic closing of quotes and brackets, semantic analysis, and more. This article explores the settings of the SQL Code Editor, focusing on how they enhance the [Script panel's](#) behavior and ensure efficient management of your SQL code.

Configuring SQL Code Editor settings

To configure the SQL Script panel behavior, navigate to **Window -> Preferences -> Editors -> SQL Editor -> Code Editor**.



Further sections provide descriptions of the SQL Script panel settings.

Miscellaneous settings

Setting	Description
Folding enabled	Use folding in SQL scripts.
Smart caret positioning in SQL identifiers	Respects underscore and other identifier parts to navigate with cursor.
Mark occurrences of word under cursor	Mark occurrences of current word in the current SQL file.

Mark occurrences of the selected word	Mark occurrences of the selected word in the current SQL file.
Show problem markers	Shows individual problem markers near the failed queries and highlights the problematic segment of it in the editor. If disabled, selects the problematic segment in the editor.

Query analysis settings

Setting	Description
Enable semantic analysis	Highlight anything treated as table or column name and resolve all the aliases and table names according to FROM clauses.
Read database table/columns for semantic analysis	May be an expensive operation. Validate all the table and column names with respect to real database objects. This option is only available when Enable semantic analysis is activated.

- The **Enable semantic analysis** feature highlights column names, their aliases, table names, and schema names.

```
SELECT
    c.customer_id AS CustomerID,
    CONCAT(c.first_name, ' ', c.last_name) AS CustomerName,
    COUNT(r.rental_id) AS NumberOfRentals,
    SUM(p.amount) AS TotalRentalAmount
FROM
    customer c
JOIN
    rental r ON c.customer_id = r.customer_id
JOIN
    payment p ON r.rental_id = p.rental_id
GROUP BY
    c.customer_id
HAVING
    COUNT(r.rental_id) > 5;
```

In instances where a selection list item is absent from the FROM clause, it is indicated with a red highlight.

For example, the column `b.id` in the following SQL script will be highlighted, indicating it is missing from the FROM clause, thereby pinpointing a potential issue within the script.

```
SELECT a.Title t, b.id, FROM Album a WHERE a.AlbumId >3;
```

- The **Read database table/columns for semantic analysis** feature extends to verifying the existence of database objects. If an object referenced in the script does not exist or is not recognized in the database metadata, it is marked with a highlight.

For example, if a script attempts to query a nonexistent column `Biography` as shown below, it will be highlighted to indicate that the column does not exist in the current database context:

```
SELECT a.AlbumId, a.Biography FROM Album a WHERE a.ArtistId = 5;
```

Auto close settings

Setting	Description
Single quotes	Automatically closes a single quote mark with another single quote when the first one is typed.
Double quotes	Automatically closes a double quote mark with another double quote when the first one is typed.
Brackets	Automatically closes a bracket with the corresponding closing bracket when the first one is typed.

Auto format settings

Setting	Tooltip Description
Convert keyword case	Auto-convert keywords to upper/lower case on enter.
Extract SQL from source code	On source code paste will remove all source language elements like quotes, <code>+</code> , <code>\n</code> , etc.

Dashboards

Table of contents

[Managing Dashboards Panel](#)

[Opening Dashboard Panel](#)

[Adding Dashboards](#)

[Removing Dashboards](#)

[Resetting Dashboards](#)

[Changing Dashboard Representation](#)

[Adjusting Dashboard Configuration](#)

[Setting Connection Preferences](#)

[Detaching Dashboards](#)

[Changing Dashboard View](#)

[Copying Dashboards to Clipboard](#)

[Saving Dashboards](#)

[Printing Dashboards](#)

[Zooming](#)

[Managing Dashboards](#)

[Creating Dashboards](#)

[To create a dashboard from scratch](#)

[To create a dashboard from a template](#)

[Editing Dashboards](#)

[To edit dashboard's configuration](#)

[Deleting Dashboards](#)

[To delete a dashboard](#)

Dashboards tool allows DBAs and programmers to quickly identify performance, disk space issues, the number of connections and other important KPIs associated with a single database connection. To learn more about database connections, see [Database Connections](#).

By default, DBeaver is delivered with a number of predefined sets of dashboards for such databases as PostgreSQL, MySQL, Oracle and Exasol. Custom dashboards are also supported. To learn more about custom dashboards, see the Managing Dashboards section below.

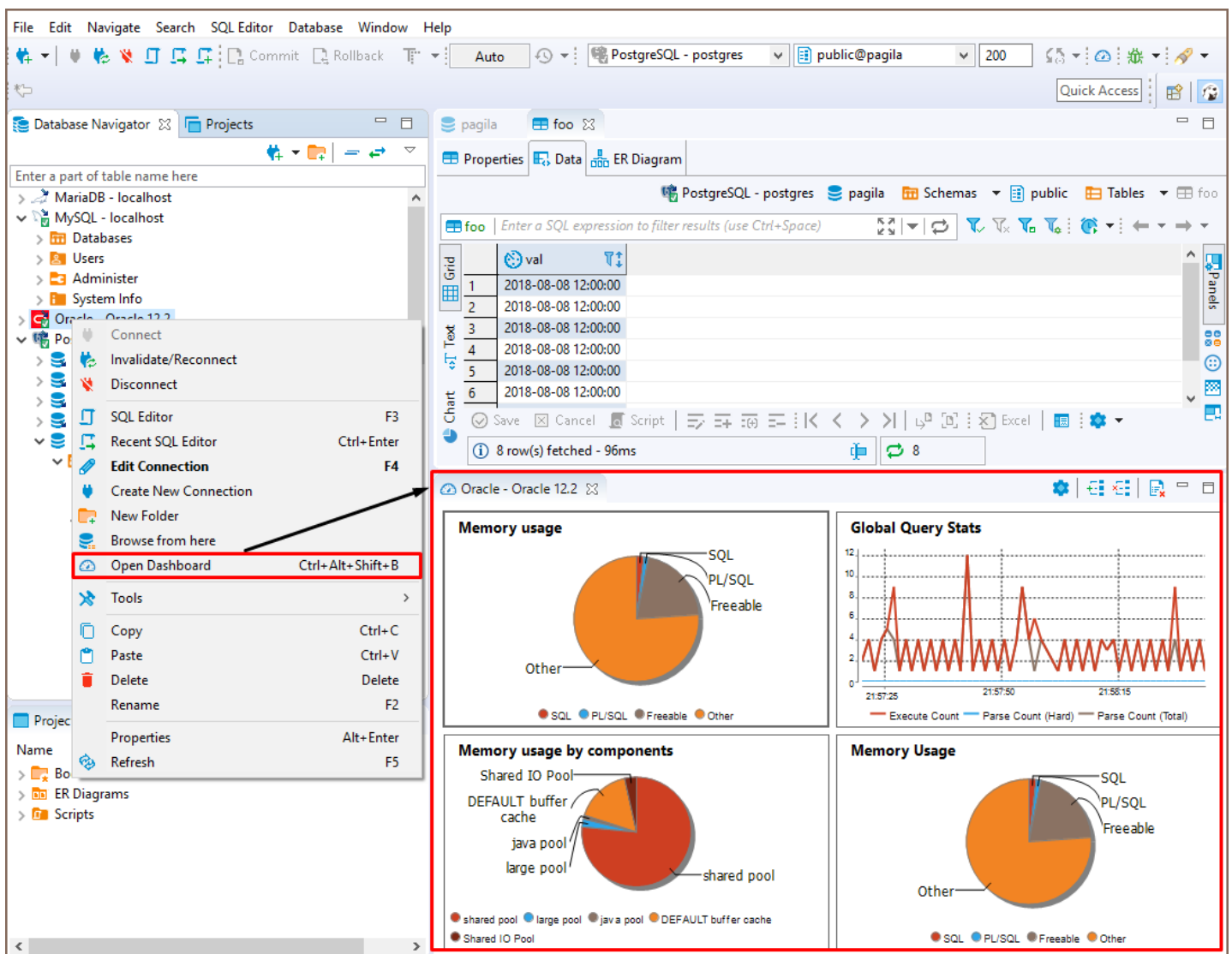
Managing Dashboards Panel

Dashboards panel is a collection of real-time dashboards, which are dashboards that are updated continuously. Dashboards displayed on the dashboards panel are actually a combination of continuously run SQL SELECT queries and charts continuously built on the data fetched.





Opening Dashboard Panel

To open the dashboards panel press the **Open Dashboard** button  on the main toolbar. The default configuration of the dashboards panel for the current database connection will appear. To learn more about database connections, see [Database Connections](#) article.


You can also right-click a connection name in the **Database Navigator** editor and select the **Open Dashboard** menu option or use the keyboard shortcut `Ctrl + Alt + Shift + B` and the dashboards panel will be opened.

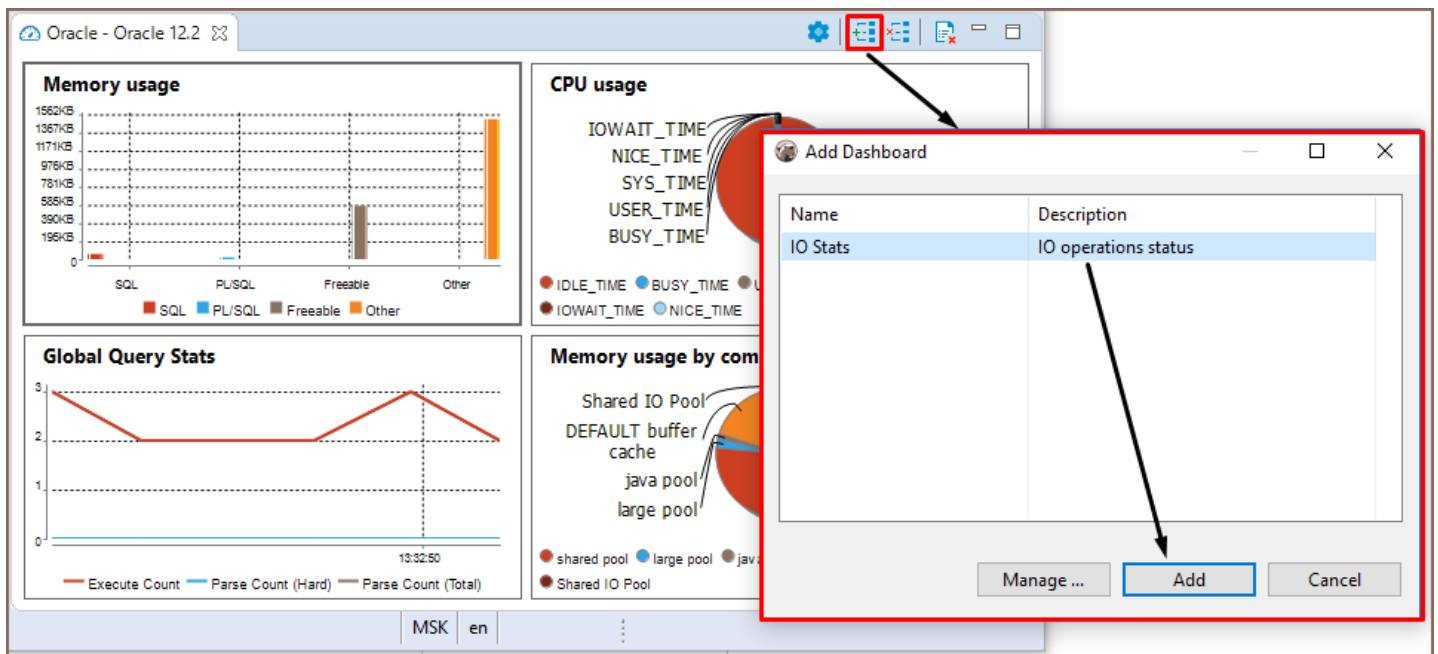


The following controls are available on the dashboards panel toolbar:

Icon	Name	Description
	Settings	Opens dashboard's configuration.
	Add dashboard	Adds dashboard to the dashboard panel.
	Remove dashboard	Removes dashboard from the dashboard panel.
	Reset dashboards	Restarts dashboard's calculation.

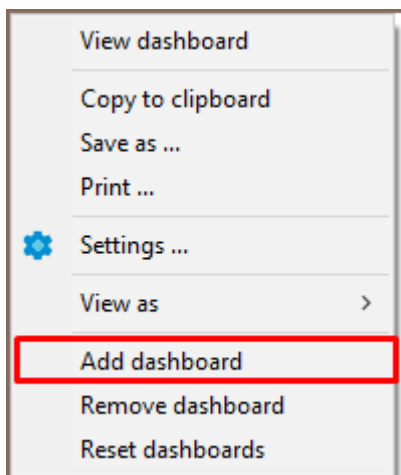
Adding Dashboards

To add a dashboard to the dashboards panel, press **Add dashboard** button  on the dashboards panel's toolbar, choose one of the dashboards from the list of available dashboards and press the **Add** button.




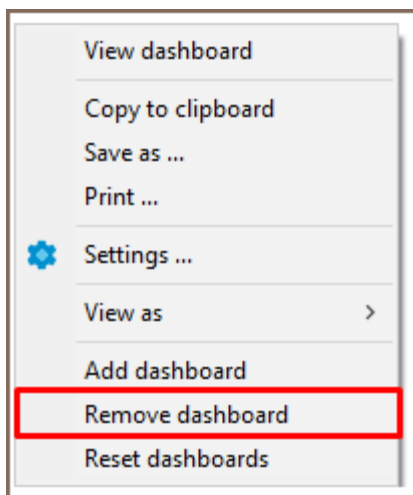
Note: Different databases have different sets of predefined dashboards. DBeaver is delivered with sets of predefined dashboards for such databases as PostgreSQL SQL, MySQL, Oracle, and Exasol. It is also possible to create new custom dashboards, for more details see Managing Dashboards.

You can also add a dashboard by right-click in any place of the dashboards panel and then select the **Add dashboard** menu option.




Removing Dashboards

To remove a dashboard from the dashboards panel, click on the dashboard you want to remove and press button **Remove dashboard**  in the dashboards panel toolbar or select **Remove dashboard** option in the dashboard's context menu.

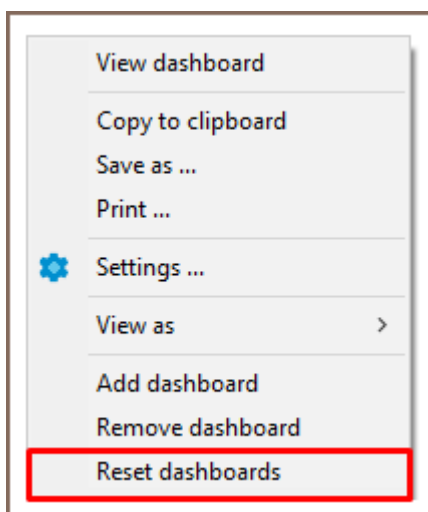


Resetting Dashboards

If you want to restart the dashboard's calculation you can reset it.

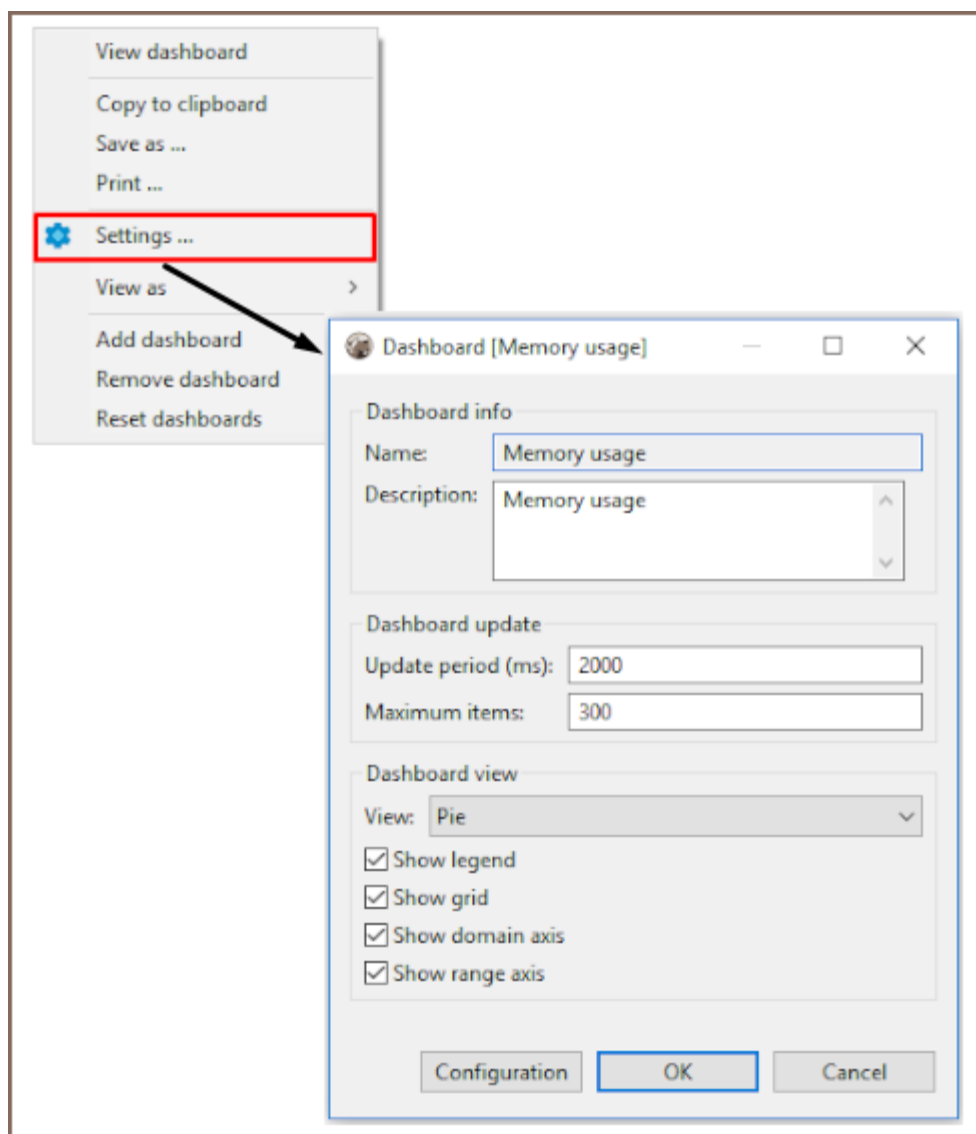
You can reset all the dashboards displayed on the dashboards panel by a single click on **Reset dashboards**  button on the dashboard panel's toolbar.

To reset a particular dashboard right-click on it and select **Reset dashboards** menu option or left click a dashboard and press **Reset dashboards** button on the dashboards panel's toolbar.



Changing Dashboard Representation

To adjust dashboard representation settings, right click on a dashboard and select the **Settings** menu option, then, in the opened dialog change the parameters you want.



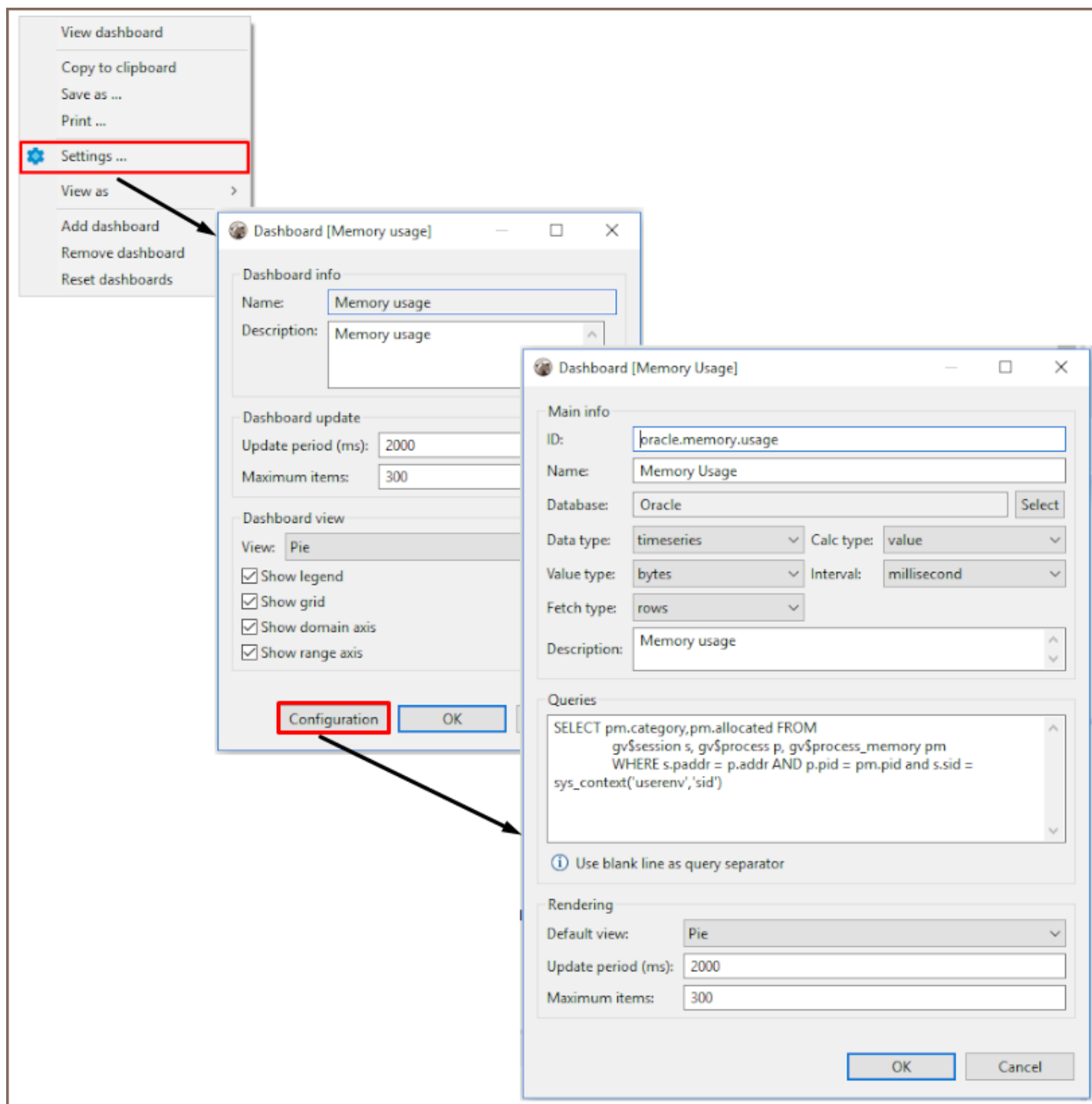
The following dashboard representation parameters can be adjusted:

Parameter	Description
Name	Defines a name of a dashboard.
Description	Defines dashboard's description. Use this field to make it easy to understand what kind of information the dashboard represents.
Update periods (ms)	Defines how often dashboard's rendering should be updated. The default value is 1000 ms.
Maximum items	Defines maximum number of fetched items. The default value is 300.
View	Defines visual representation of the dashboard. The following options are available: Bar, Pie, Time series.
Show legend	If this check-box is selected, the legend will be displayed on the dashboard.
Show grid	If this check-box is selected, the grid will be displayed on the dashboard.

Show domain axis	If this check-box is selected, the domain axis will be displayed on the dashboard.
Show range axis	If this check-box is selected, the range axis will be displayed on the dashboard.

Adjusting Dashboard Configuration

To adjust dashboard's configuration settings right-click on a dashboard, select the **Settings** menu option, then, in the opened dialog box press the **Configuration** menu option.



The following dashboard parameters can be configured:


Parameter	Description
ID	Defines dashboard's ID. Make sure that ID has numeric values in it.
Name	Defines dashboard name.
Database	Defines the database driver. To learn more about database drivers, see Database Drivers .
Data type	Defines the data type. The following options are available: timeseries (the default option) and statistics. Select timeseries type if you want to track the actual value returned by the server. Select the statistics type if your dashboard will show historical data.

Calc type	Defines how the data should be calculated. The following options are available: value (the default option) and delta. Select value if you are interested in the current value. Select delta if you want to track the difference between the current value and the previous one. This may be very useful when you work with statistics data, for example.
Value type	Defines the value to be shown on the range domain. The following options are available: decimal (the default option), integer, percent, bytes. Choose the value type in accordance with your data, for example, memory usage is convenient to be tracked in KBytes.
Interval	Defines time interval to be shown on the domain axis. The following time intervals are available: millisecond(the default option), second, minute, hour, day, week, month, year.
Fetch type	Defines whether the query should fetch data from rows or columns.
Description	Defines the description of a dashboard. Use this field to make it easy to understand what kind of information the dashboard represents.
Queries	Defines an SQL query whose fetched data will be used to build the chart displayed on the dashboard.
Default view	Defines the default visual representation of a dashboard on the dashboard panel. The following options are available: Bar, Pie, Time series(the default option).
Update period(ms)	Defines how often the dashboard's rendering should be updated.
Maximum items	Defines the maximum number of items to be fetched for the dashboard.

Note: Predefined dashboards are read-only and cannot be re-configured, but you can copy them and use them templates to create new dashboards with any query and other settings. To learn about creating new dashboards, see the Managing Dashboards section.

Setting Connection Preferences

By default, if there is no active connection to the database and you open its dashboards panel, all the dashboards on the panel will be empty.

You can force a database connection on the dashboard panel's activation by pressing the **Settings** button  on the dashboards panel's toolbar and then selecting the **Connect on activation** check-box.

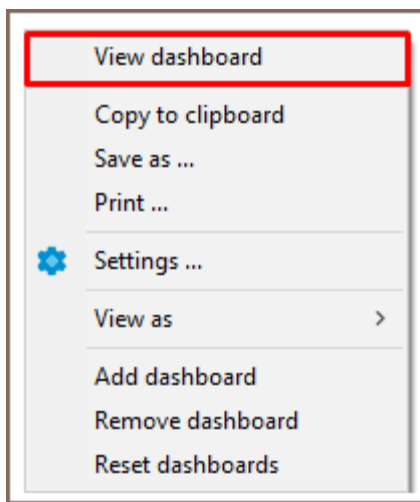
Detaching Dashboards

If you have several monitors and would like to place a dashboard into a separate screen, you can either detach the whole dashboards panel or a single dashboard, and drag-and-drop them to any place you want.

To detach the whole dashboard panel, right click on the dashboard's tab name and select the **Detach** menu option.

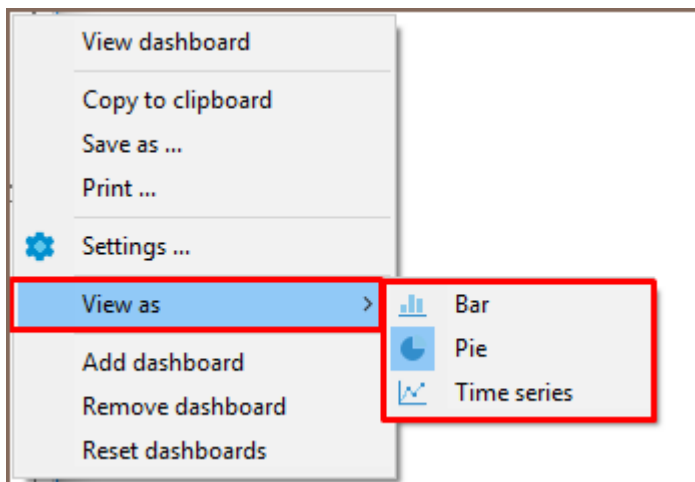


To detach a single dashboard, double left click over it. You can also right click the dashboard and then, select the **View Dashboard** menu option, the dashboard will be detached from the panel and you will be able to move it to any place on your screen.



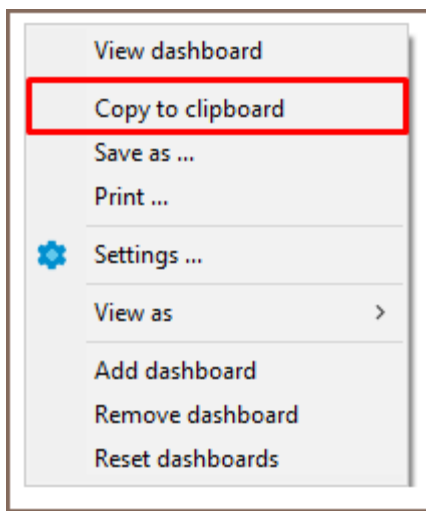
Changing Dashboard View

You can change the representation of a dashboard and view it as a Pie, Bar or Time series. To change the dashboard view, right click on it and select the **View as** menu option.



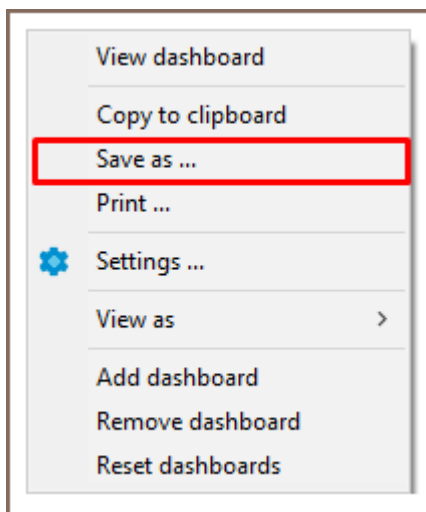
Copying Dashboards to Clipboard

To copy a dashboard onto the clipboard, right click on the dashboard and use the **Copy to Clipboard** menu option. The screenshot of the dashboard will be placed onto the clipboard.



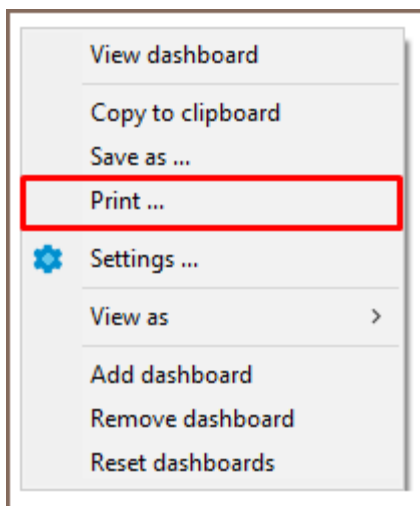
Saving Dashboards

If you want to save a screenshot of a dashboard locally in PNG format, right click on it and select the **Save as ...** option in the context menu displayed.



Printing Dashboards

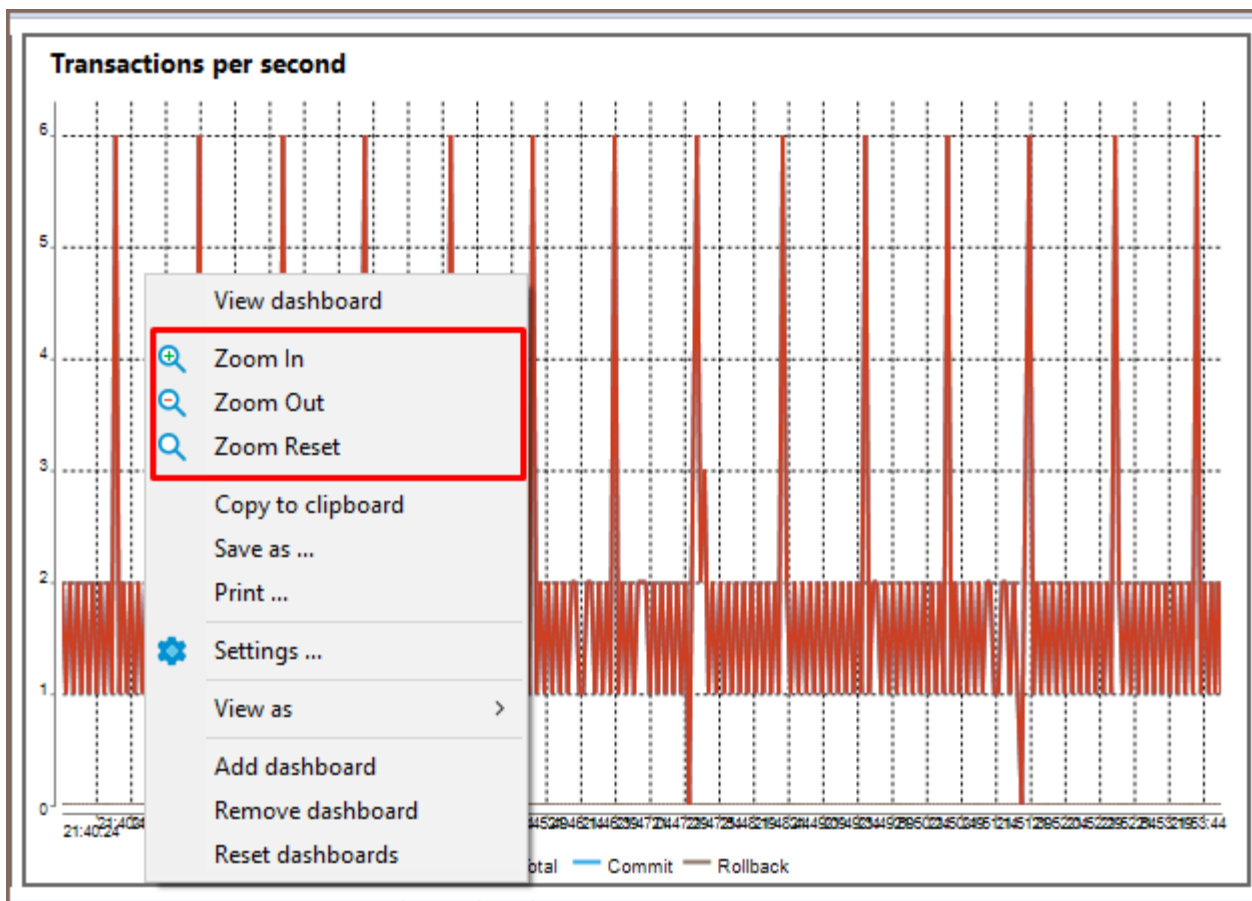
If you want to print out a screenshot of a dashboard, right-click the dashboard to be printed and select the **Print...** option.



Zooming

For Time series and Bar dashboard representations the following zooming options are available on the dashboard's context menu:

- Zoom In
- Zoom Out
- Zoom Reset




Managing Dashboards

You can extend the list of predefined default dashboards by creating your own custom dashboards. This section describes dashboards' list management.

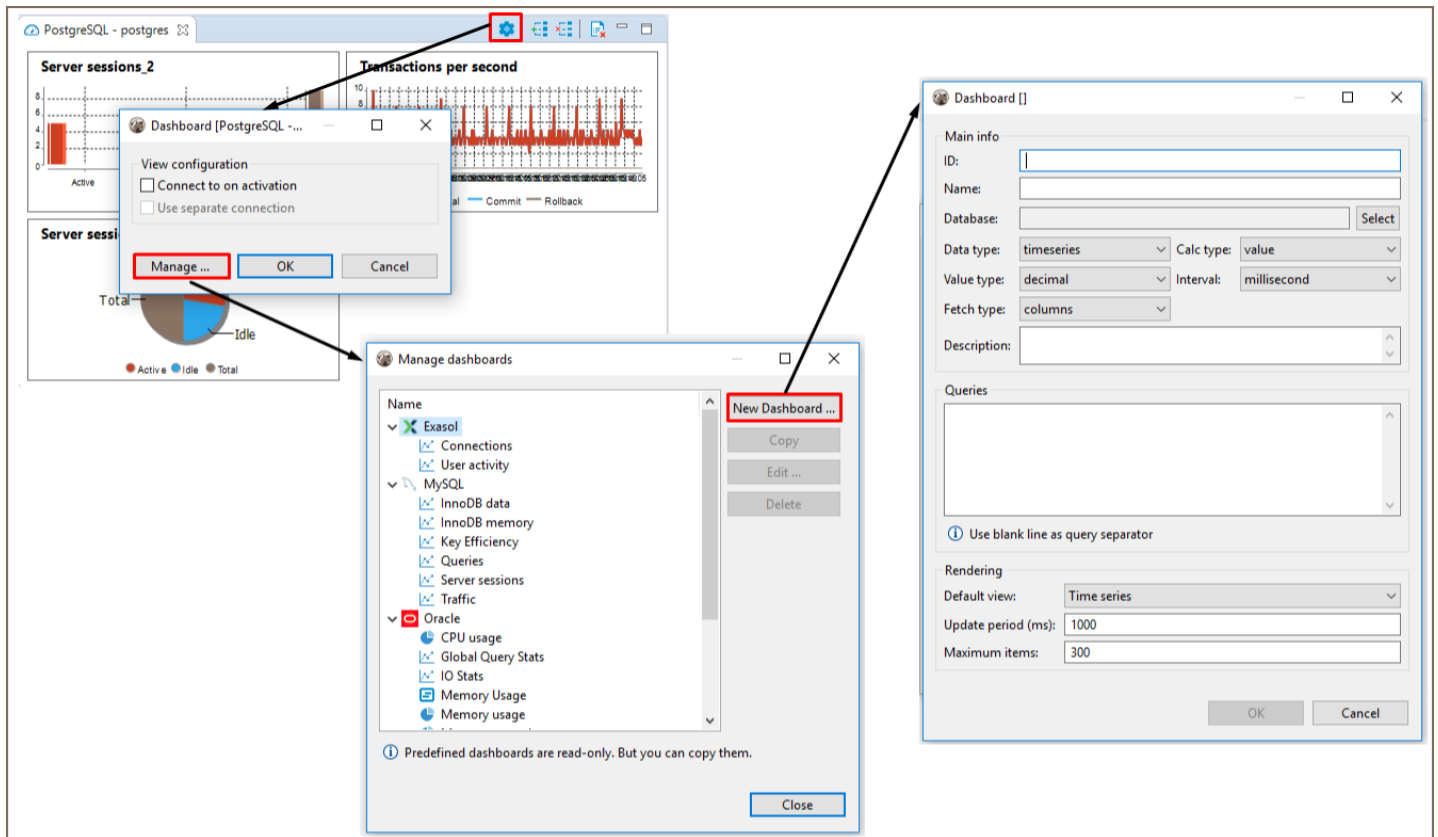
Creating Dashboards

You can create a new custom dashboard either from scratch or from any existing dashboards.


To create a dashboard from scratch

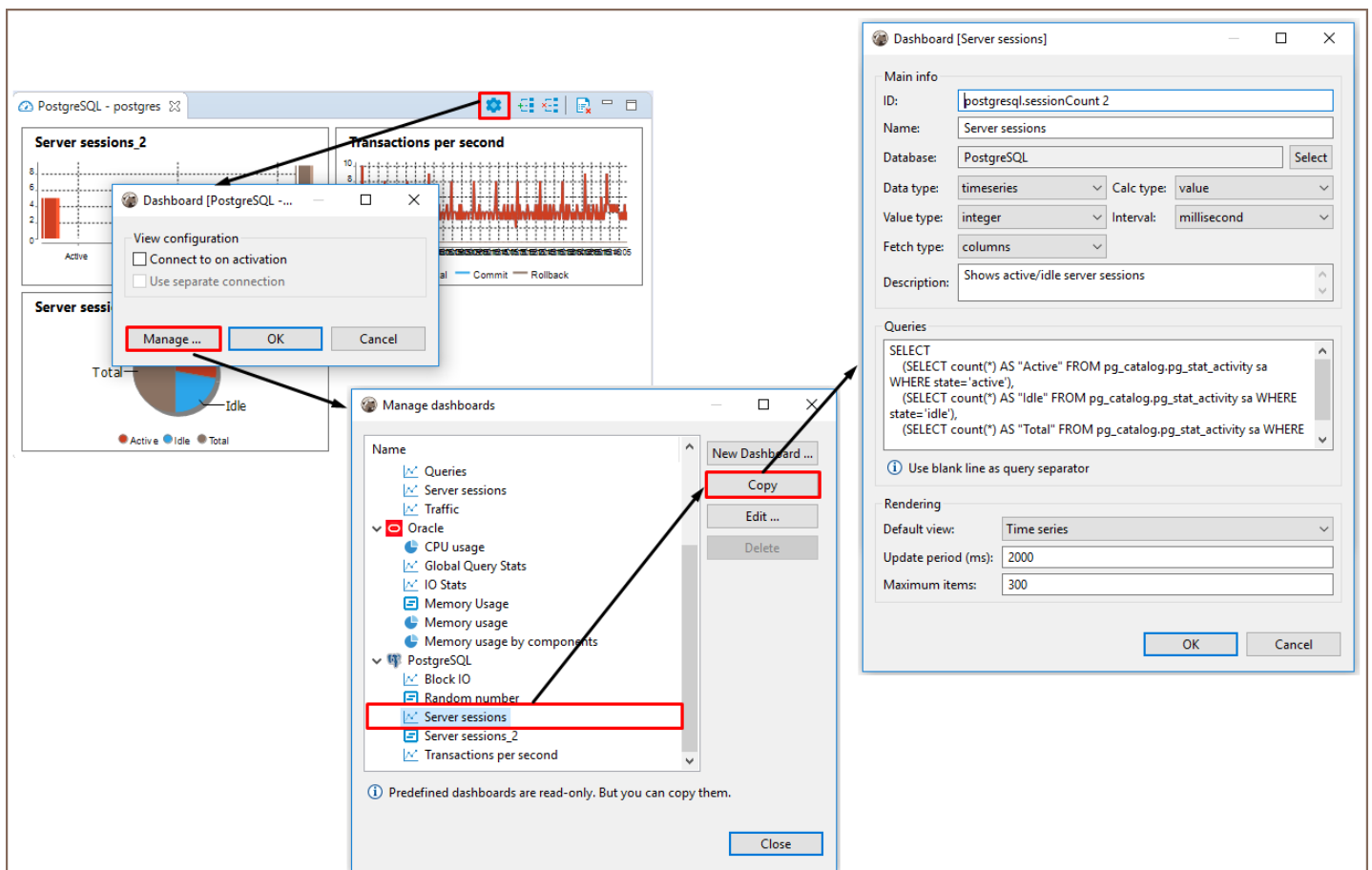
1. Press the **Settings** button  on the dashboards panel toolbar.
2. In the opened dialog box click the **Manage...** button.
3. In the **Manage dashboards** window click the **New dashboard...** button.

- Set up all configuration parameters as required and press **OK**. To learn more about the dashboard's configuration parameters, see [Adjusting Dashboard Configuration](#).



To create a dashboard from a template

- Press the **Settings** button  on the dashboards panel toolbar.
- In the opened dialog box click the **Manage...** button.
- In the **Manage dashboards** window select any of the existing dashboards from the list and click **Copy**.
- Adjust all configuration parameters as required and press **OK**. To learn more about the dashboard's configuration parameters, see [Adjusting Dashboard Configuration](#).




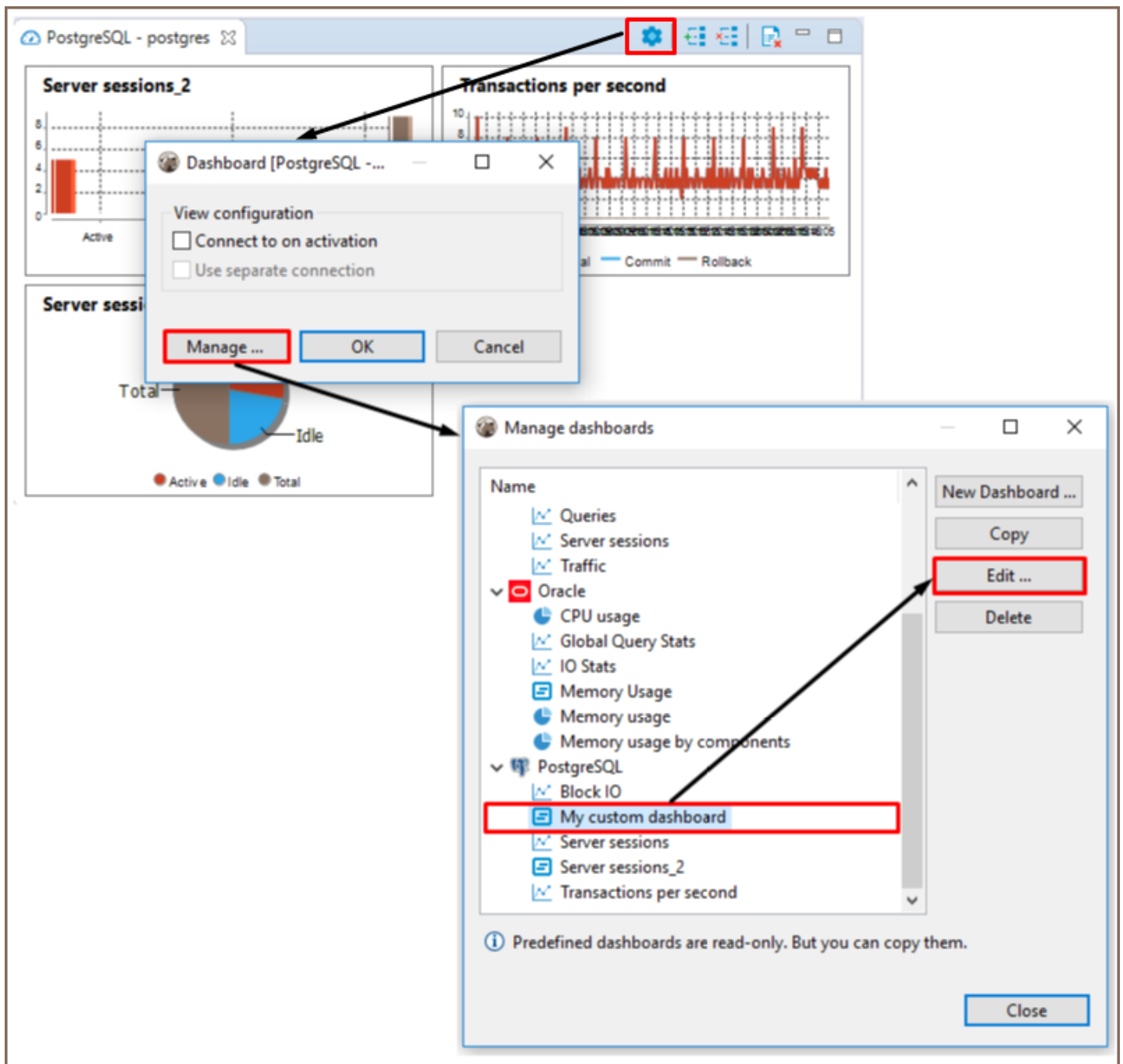
Editing Dashboards

If you need to change the dashboard's name, ID or any other configuration setting, you can edit a dashboard.

Note: Only custom dashboards can be edited, predefined dashboards are read-only, but you can use them as templates and create a custom dashboard whose parameters will be editable. To learn how to create dashboards from templates, see [Creating Dashboards](#).

To edit dashboard's configuration

1. Press the **Settings** button  on the dashboards panel toolbar.
2. In the opened dialog box click the **Manage...** button.
3. In the **Manage dashboards** window select any of the custom dashboards from the list and click **Edit...**
4. Adjust all configuration parameters as required and press **OK**. To learn more about the dashboard's configuration parameters, see [Adjusting Dashboard Configuration](#).



Deleting Dashboards

Note: Predefined dashboards cannot be deleted, but any custom dashboards can be deleted.

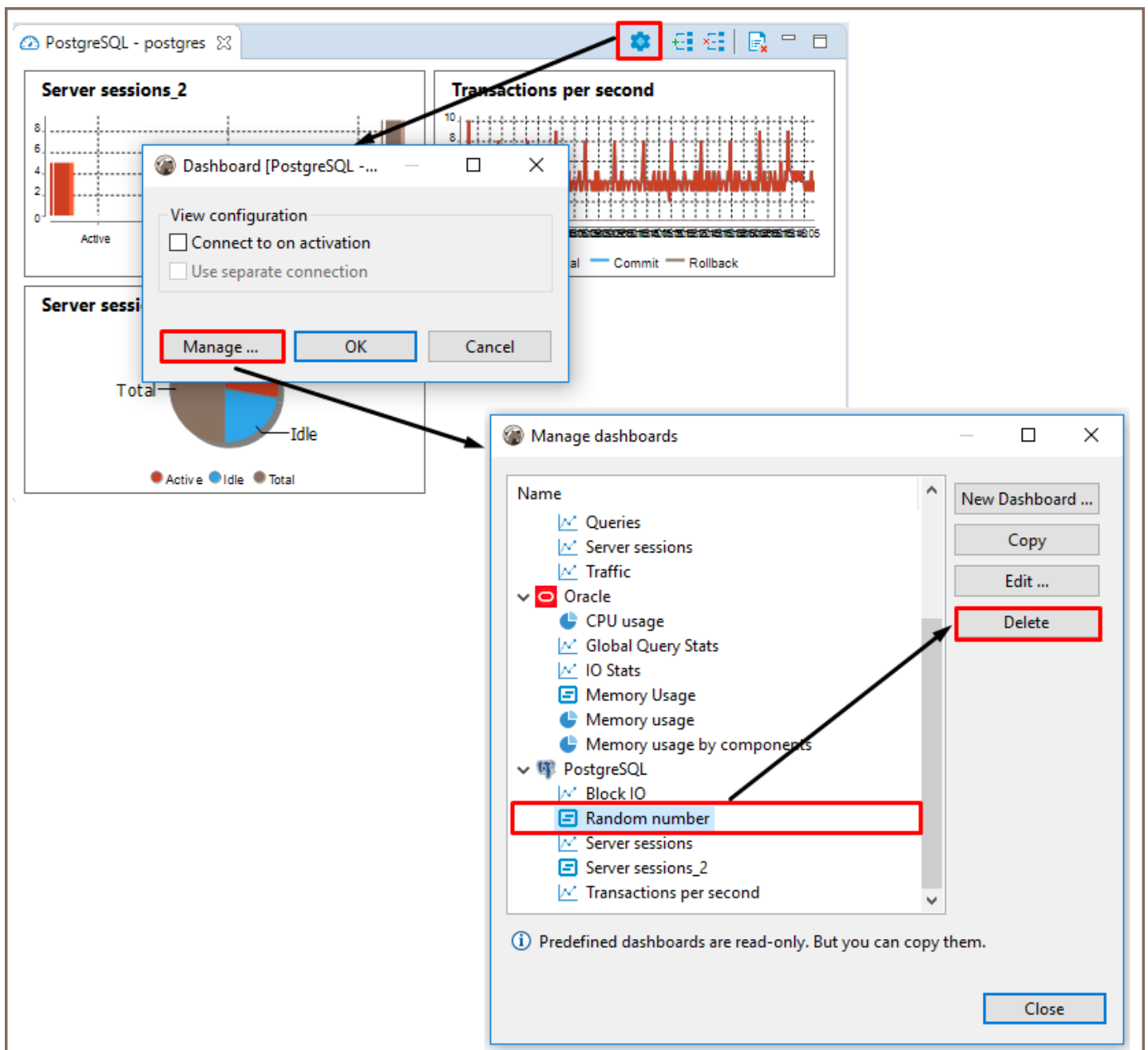
If you want to delete a dashboard, follow the steps described below.

To delete a dashboard

1. Press the **Settings** button  on the dashboards panel toolbar.

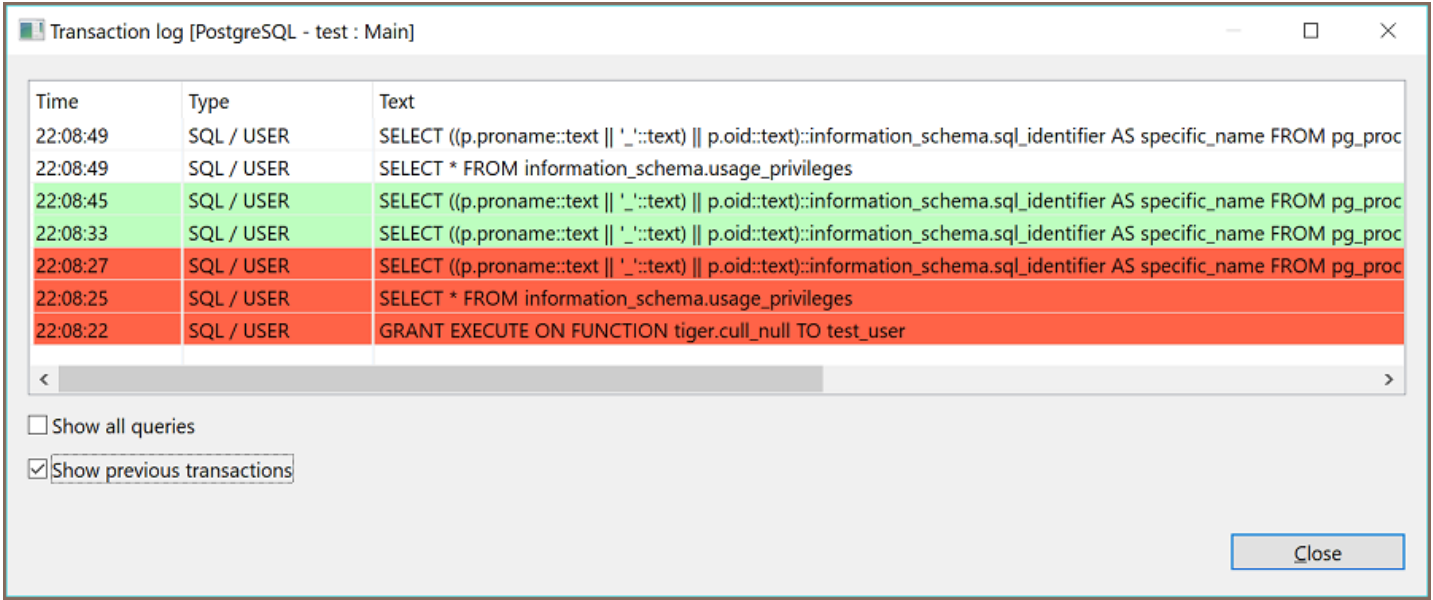
2. In the opened dialog box click **Manage...** dashboards.

3. In the **Manage dashboards** window select any of the custom dashboards from the list and click **Delete**.



Transaction log

The Transaction Log shows all transactions (queries of 'transaction' type such as INSERT/DELETE/UPDATE and others) made during the current DBeaver session. To open the Transaction Log, click the **Transaction log** button (🕒) in the toolbar or the statistics field to the left of it.



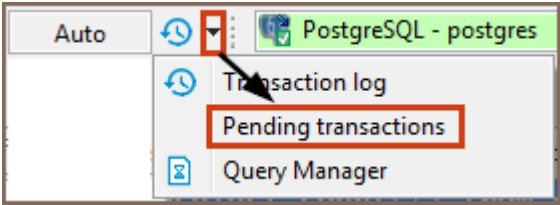
The Transaction Log window shows transactions that are:

- In progress or pending - shown without any special color
- Successfully committed – in green:
- Rolled back – in orange or red:

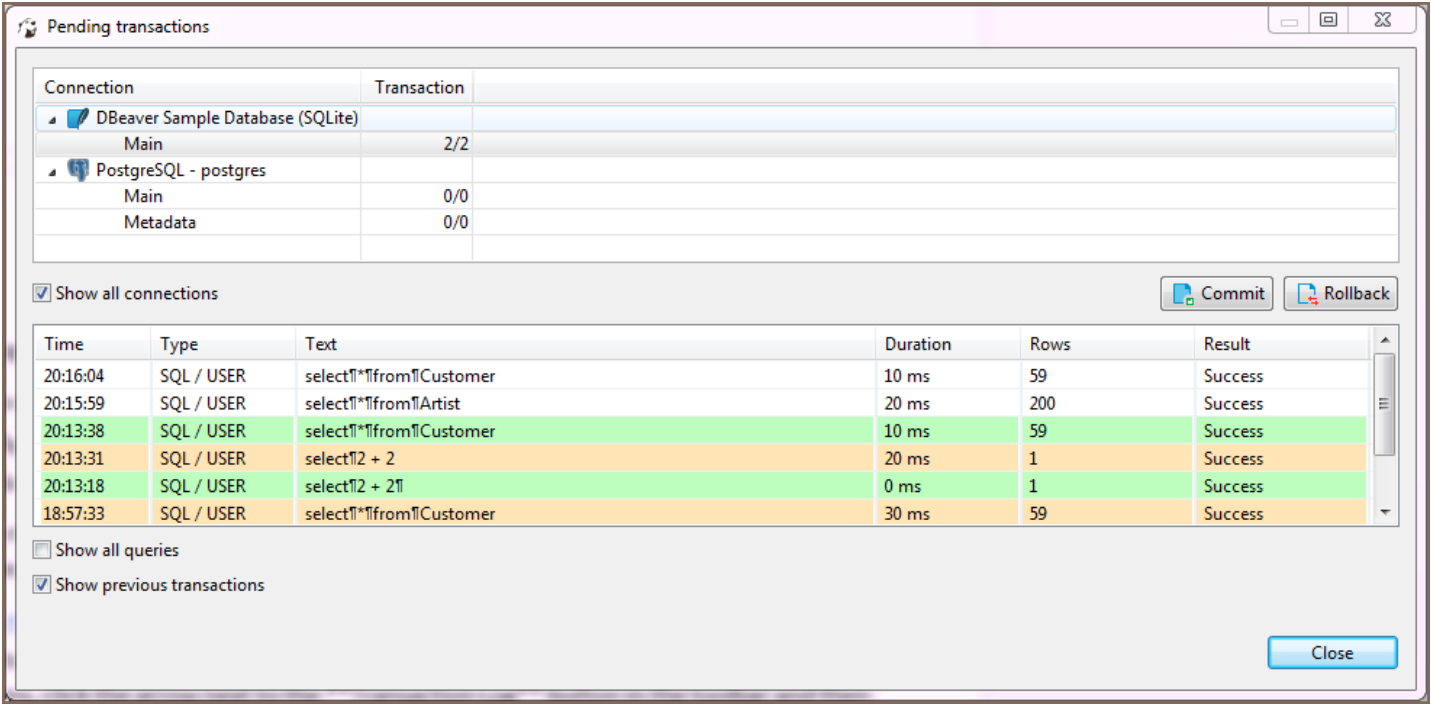
To see all previous transactions during the current session, select the **Show previous transactions** checkbox. To see all queries including non-transactional ones, select the **Show all queries** checkbox.

Pending Transactions

It might be useful to check your pending transactions because they might lock your database. To see your pending transactions, click the arrow next to the **Transaction Log** button in the toolbar and then click **Pending Transactions** on the dropdown menu:



The upper table of the Pending transactions window shows currently active connections and the number of their transactions. The bottom table shows query details of the connection that is currently in focus in the upper table:



When the Pending transactions window opens, the upper table shows only those connections that have pending transactions. If no connections have pending transactions, the table is empty. To see all connections that are currently active (connected), select the **Show all connections** checkbox.

You can commit or roll back transactions right from the Pending transactions window: in the upper table, click the row with the required uncommitted transactions and then click the **Commit** or **Rollback** button, depending on your purpose. If a transaction is committed/rolled back successfully, both buttons are disabled. If the operation is unsuccessful, the system displays an error message.

To see all previous transactions made during the current session, select the **Show previous transactions** checkbox.

To see all queries including non-transactional ones, select the **Show all queries** checkbox. The green rows are committed transactions, the orange (or red) ones are rolled back, rows without a special color are non-transactional or pending transactions.

Auto and manual commit modes

Table of contents

[Auto-Commit Mode](#)

[Manual Commit Mode](#)



[Smart Commit Mode](#)

[Transaction Isolation Level](#)

DBeaver supports two modes for committing changes to the database:



- **Auto-commit** transfers all changes that you make immediately to the database.
- **Manual commit** requires your confirmation before committing a change to the database or rolling it back.

Though available in many cases, the two modes are actionable only in [SQL Editor](#). See the next sections for details of using the modes.

To switch between the modes, use the mode selection button that appears in one of the two views:  or .

Auto-Commit Mode

Auto-commit mode is the default one for the Development and Test connection types, see [Connection Types](#).





Auto-commit mode is on if you can see the auto-commit view of the mode selection button () in the application toolbar. If you see the manual commit view () , then in order to switch to auto-commit mode, click the mode selection button – it changes to auto-commit. At the same time, this disables the two manual commit buttons in the toolbar: **Commit** and **Rollback** – these are available only in manual commit mode.

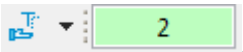
The statistics field next to the mode selection button always shows **Auto** in auto-commit mode: .

Clicking the statistics field opens the [Transaction Log](#).

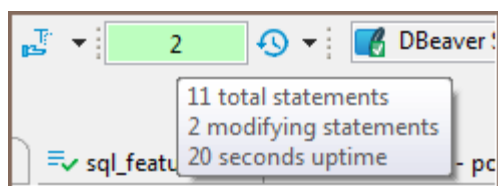
Manual Commit Mode

Manual commit is intended to protect your database from inadvertent changes and that is why it is the default mode for Production connection type, see [Connection Types](#).

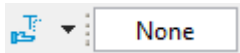
Manual commit mode is on if you can see the Manual commit view of the mode selection button () in the application toolbar. If you see the auto-commit view () , then in order to switch to manual commit mode, click the auto-commit button – it changes to manual commit. At the same time, this enables the two manual commit buttons in the toolbar: **Commit** () and **Rollback** () .

In manual commit mode, when you execute SQL statements (`Ctrl+Enter`), the number of database modifying statements that pend commitment to the database appears in the statistics field next to the mode selection button:  .

If you hover your mouse over the field, you can see the statistics of your SQL statements:



To commit statements to the database, click the **Commit** button in the toolbar. To discard them, click **Rollback**.

If no modifying statements have been made, the statistics field shows **None**:  .

Clicking the statistics field opens the [Transaction Log](#).

Smart Commit Mode

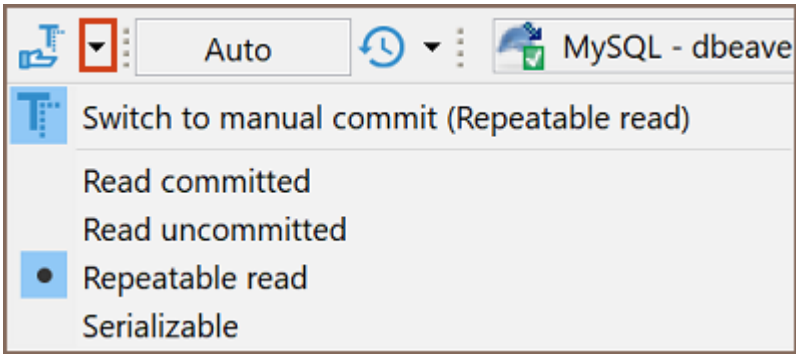
When smart commit is enabled and you are in auto-commit mode, which allows DBeaver to monitor your activity. When you try to execute any data modifying query (UPDATE, INSERT, DELETE, UPSERT, MERGE, etc) DBeaver will switch to manual commit mode before executing your query. Also if you edit any table data and save your changes - DBeaver will also switch to manual mode before the actual data modification.

If the option "Return to auto-commit on transaction end" is on, then DBeaver will switch back to auto-commit mode once you have executed Commit or Rollback command (using the main toolbar or the main menu actions).

Smart commit mode is very useful if you work mostly in the read-only mode. It does not lock tables when you perform SELECT queries. The transaction will only be started when you start to modify your data.

Transaction Isolation Level

For both, Auto and Manual commit modes, you can select the transaction isolation level. To do so, click the arrow next to the mode icon and then click the required option in the dropdown list:



Query Manager

Table of contents

[Overview](#)

[Query Manager Properties](#)

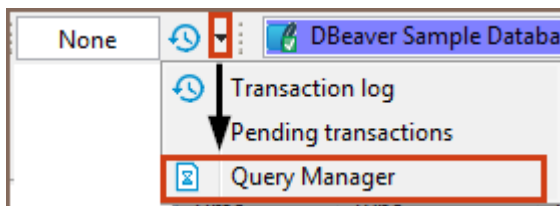
Overview

Query Manager is a view that shows the history of all SQL queries that DBeaver has executed during the current session.

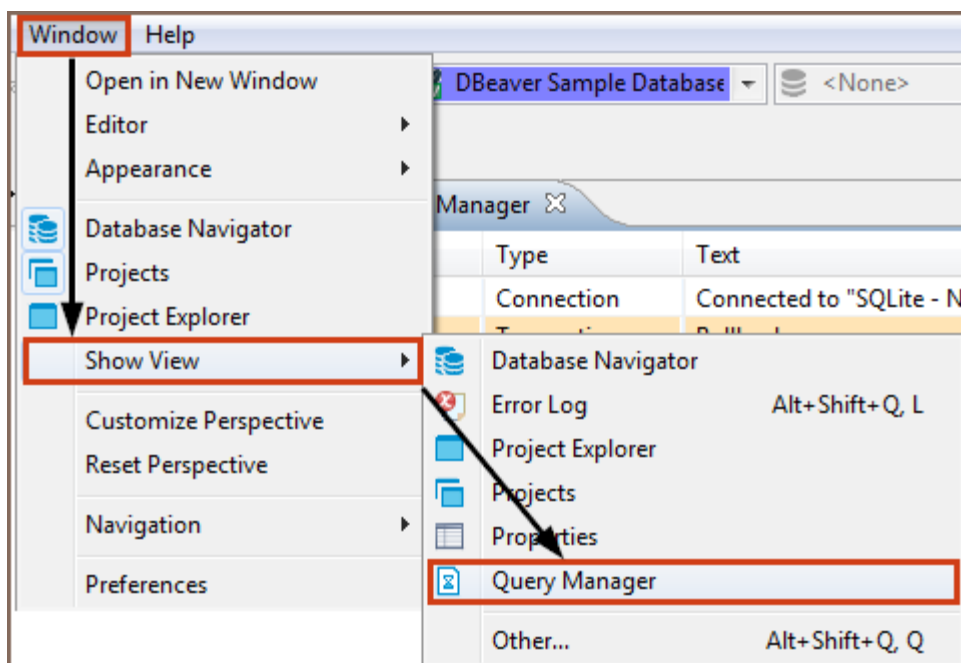
Note: DBeaver saves query history after restarting only in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions.**

To open the Query Manager, do one of the following:

- Click the arrow next to the **Transaction Log** button in the toolbar and then click **Query Manager** on the dropdown menu:




- On the **Window** menu, click **Show View -> Query Manager**:




The Query Manager logs all queries together with their execution statistics (execution time, duration, number of fetched/updated rows, errors, etc.):

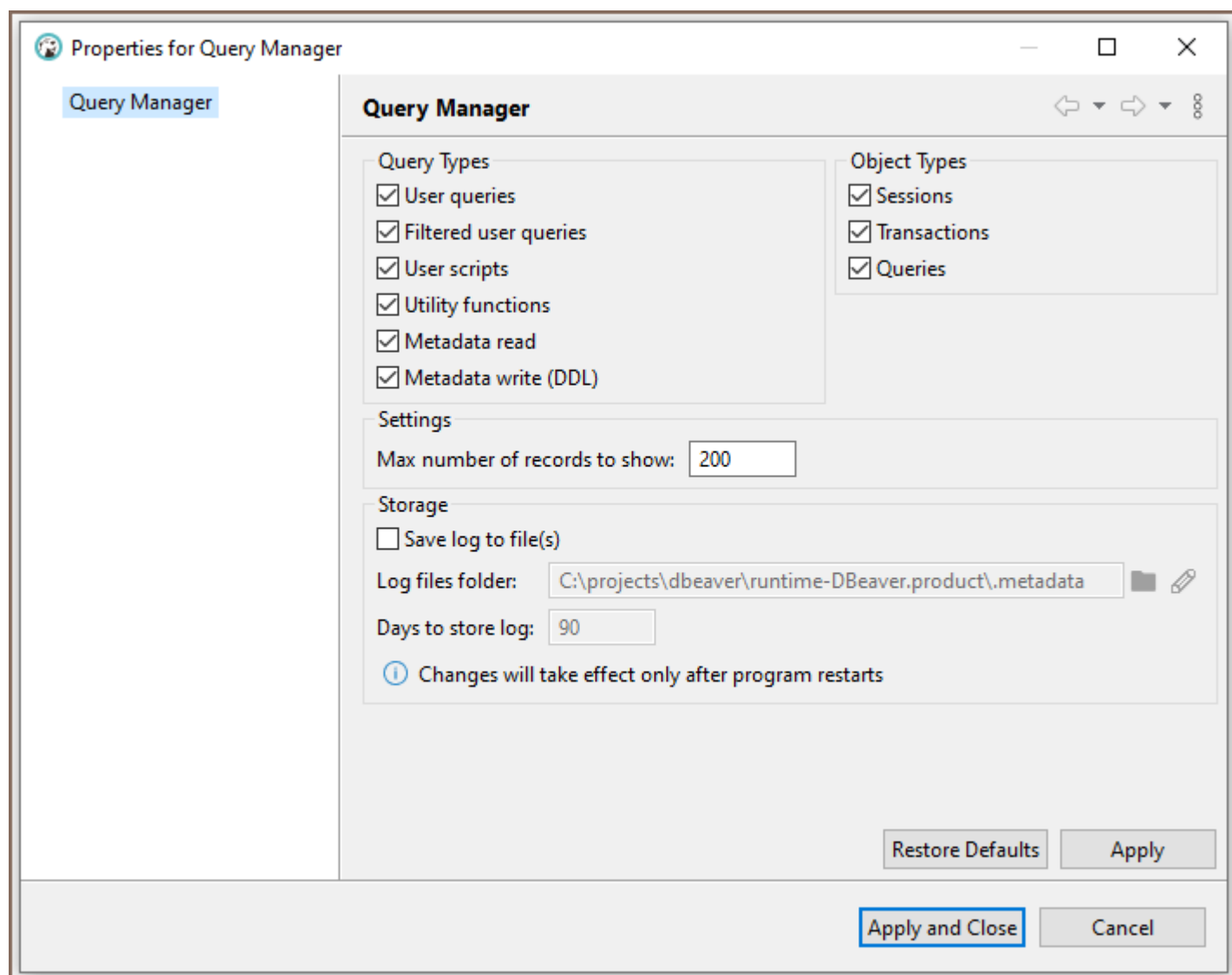
Time	Type	Text	Duration	Rows
18:36:34	Connection	Connected to "SQLite - New Connection"		
18:17:59	Transaction	Rollback	1 min 46 sec	
18:19:31	SQL / USER	select * from Customer	10 ms	1
18:19:29	SQL / USER	select * from Artist	20 ms	59
18:19:20	SQL / USER	select * from Artist	20 ms	200
18:16:45	Transaction	Commit	1 min 57 sec	
18:17:09	Transaction	Commit	0 min 49 sec	
18:17:27	SQL / USER	select * from Customer	4 ms	1
18:17:22	SQL / META	-- Load imported keys [null, null, Customer]	20 ms	1
18:17:22	SQL / META	-- Load primary keys [null, null, Employee]	1 ms	1
18:17:22	SQL / META	-- Load columns [null, null, Employee, %]	4 ms	15
18:17:22	SQL / META	-- Load tables [null, null, Employee, null]	0 ms	1
18:17:22	SQL / META	-- Load indexes [null, null, Customer]	1 ms	2
18:17:22	SQL / META	-- Load columns [null, null, Customer, %]	0 ms	13
18:17:22	SQL / META	-- Load tables [null, null, Customer, null]	0 ms	1
18:17:22	SQL / USER	select * from Customer	10 ms	59
18:17:17	SQL / META	-- Load imported keys [null, null, Artist]	0 ms	0
18:17:17	SQL / META	-- Load indexes [null, null, Artist]	14 ms	1
18:17:17	SQL / META	-- Load columns [null, null, Artist, %]	1 ms	2
18:17:17	SQL / META	-- Load tables [null, null, Artist, null]	2 ms	1
18:17:17	SQL / USER	select * from Artist	20 ms	200
18:17:09	SQL / META	-- Load schemas	0 ms	0
18:17:09	SQL / META	-- Load catalogs	0 ms	0
18:17:09	Transaction	Commit	0 min 0 sec	
18:17:09	Connection	Connected to "DBeaver Sample Database (SQLite)"		
18:17:06	SQL / USER	select count(*) from pg_catalog.table_to_xml	280 ms	

You can modify the look of the Query Manager by filtering queries and setting the number of entries displayed per page, as well as specifying some storage settings. See the 'Query Manager Properties' section below.

To erase all entries from the Query Manager, click the **Clear query manager log** button () in the view's toolbar. In **DBeaver PRO versions**, you can also open **Help** —> **Clear History** in the main menu and select **Query log history** option. The query history will disappear after the restart.

Query Manager Properties

To manage the look of the Query Manager, filter the entries, and modify the storage settings, click the **Set query manager filter** button () in the view's toolbar. The Properties for Query Manager window opens:



- Select or clear the checkboxes in the **Object Types** section to filter entries by object type. To filter entries by query type, select clear the checkboxes in the **Query Types** section.
- To change the max number of entries displayed, enter the new number in the **Max number of records to show** field.

- To set DBeaver to save the query log in a file, select the **Save log to file(s)** checkbox and then specify the file location in the **Log files folder** field.

After you make all necessary changes to the settings, click **Apply** to apply the changes and keep the window open or click **Apply and Close** to apply the changes and close the window. To discard all changes and return to the previous state, click **Restore Defaults**.

Data transfer

Table of contents

[Export data](#)

[Import data](#)

[Importing data from CSV file](#)

[Importing data from Excel file](#)

[Additional features](#)

[Importing data from XML file](#)

[Importing data from the database table](#)

Data transfer is a crucial feature that enables you to export and import data in various formats as well as moving data between tables in the same or to different databases.

Note: The data transfer operation runs in the background, allowing you to continue working with your database during the export or import process. However, try to avoid changing data in the tables that have been selected for export or import until the process is complete.

Export data

You can transfer data from one database to another or export it in different types and formats:

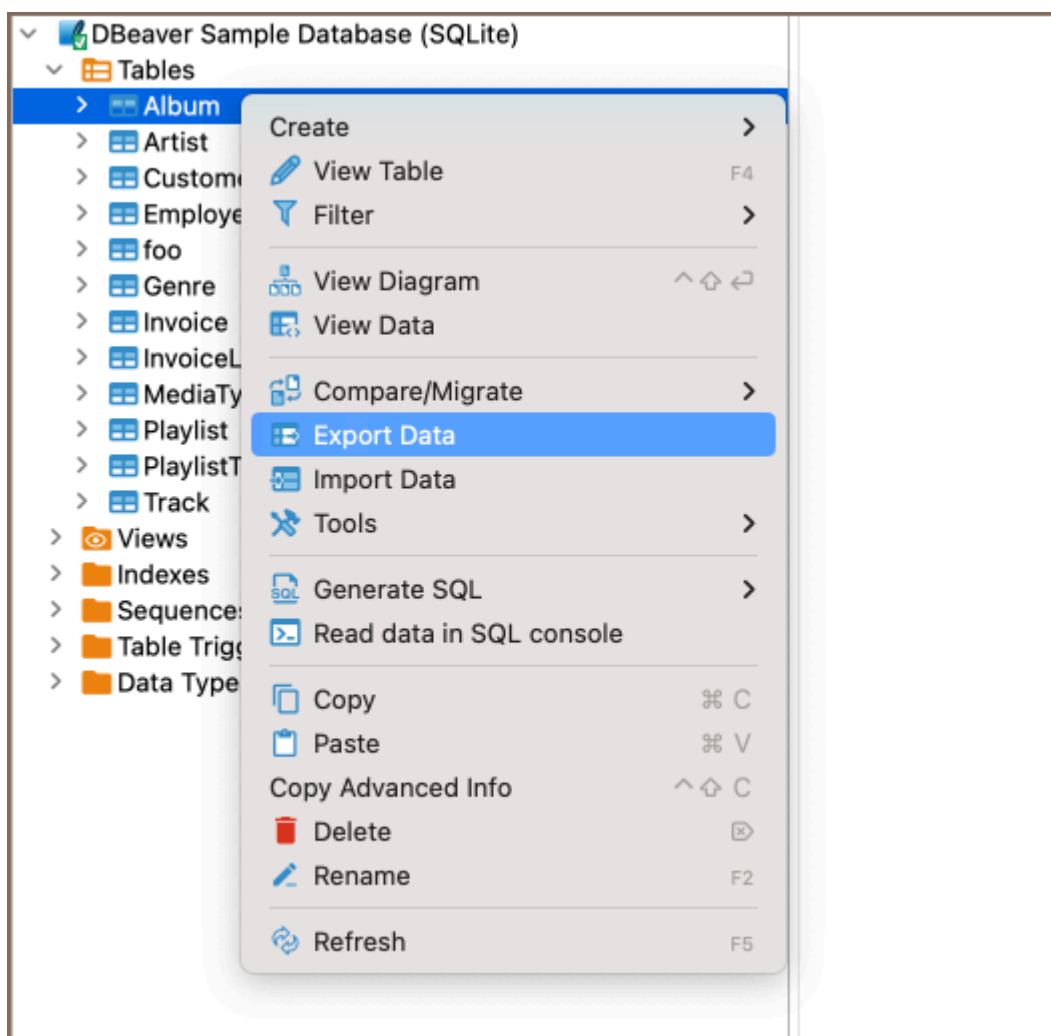
- CSV
- XLSX (Excel spreadsheet) ★
- HTML
- XML
- TXT
- JSON
- Markdown files
- SQL statements

- Source code arrays

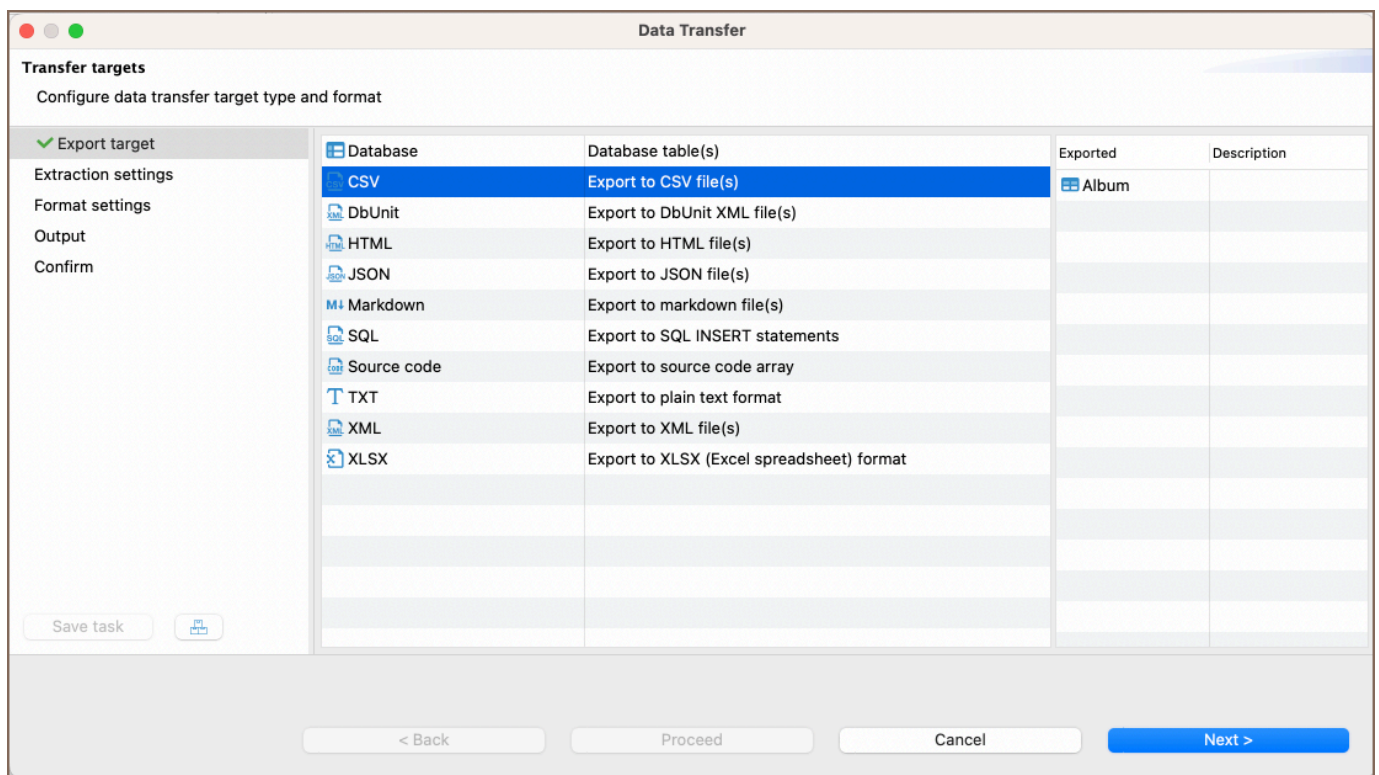
The process of exporting data follows a similar procedure for the supported formats. Therefore, in this guide, we will focus on the data exporting process using the **CSV** format.

1. Select the table or tables you want to export. In the context menu, choose **Export Data**.

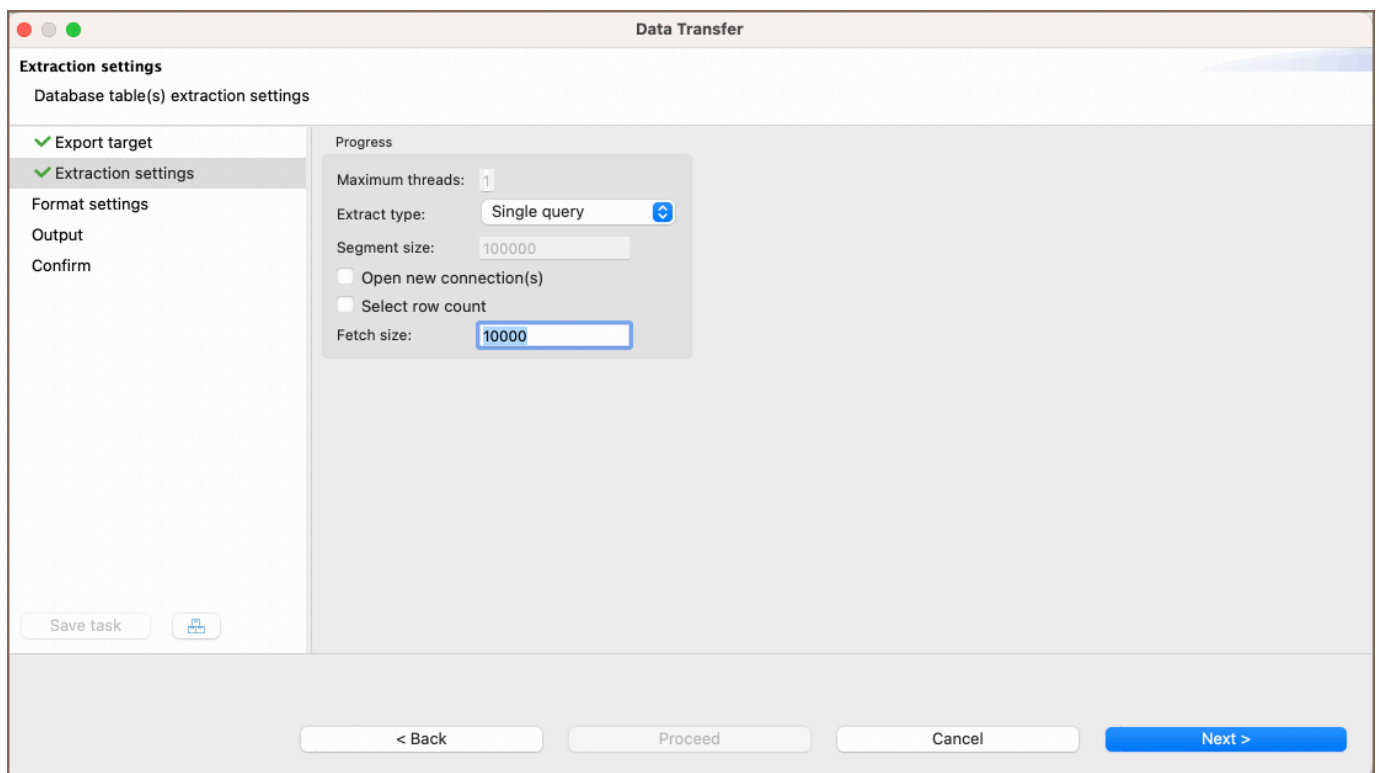
Note: you can also export data from custom SQL query results. To do that, choose **Export data** in the results context menu.



2. In the window that appears, choose **CSV** and click **Next**.



- Set your data extraction options (how the data will be read from the tables). This may affect the extraction's performance. And set the export format option. They are specific to the data format you chose in step 2:



Setting name	Description
Maximum threads	Number of threads used for data extraction.
Extract type	Defines the method of data extraction (Single query or Multiple queries).

Segment size	The size of data segments to be extracted at a time (active while Multiple queries are selected as an Extract type).
Open new connection(s)	Opens a new physical connection for data reading, recommended if you plan to continue working with the database during the export process.
Select row count	Queries row count before performing export to track progress, but may cause performance issues in some cases.
Fetch size	Number of rows the driver retrieves in one batch from the database.

Tip: The **Maximum threads** option is active when exporting data from multiple tables. It is recommended to set the number of threads to match the number of cores in your computer for optimal performance.

4. Configure the format settings.

At this stage you configure the format settings for the output file. These settings determine how the data will be presented and encoded in the exported file.

Data Transfer

Format settings
File format settings

- ✓ Export target
- ✓ Extraction settings
- ✓ **Format settings**
- Output
- Confirm

General

Formatting: <Connection's default> Edit ...

Binaries: Inline Encoding: Binary

Value display format: Display (default)

Configure Columns ...

Exporter settings

Name	Value
File extension	csv
Delimiter	,
Row delimiter	default
Header	top
Header format	label
Characters escape	quotes
Quote character	"
Quote always	disabled
Quote never	[]
NULL string	
Format numbers	[]

Save task

< Back Proceed Cancel Next >

Setting name	Description
Formatting	Selects the formatting settings for the exported data. Read more in the related article
Binaries	Specifies how binary data is represented in the export.

Encoding	Sets the character encoding for the exported file.
Value display format	Chooses the format for displaying values in the export.
Configure columns	Allows you to set which columns to export. Configure the columns to export by clicking Configure . In the Action column, use the down arrow on a row to toggle between skip or export for that column/table.

Note: The options within the Exporter settings section will vary and provide specific controls tailored to the format selected for export.

5. Set options for output files or clipboards.

The screenshot shows the 'Data Transfer' application window. The 'Output' tab is selected, showing configuration options for export output parameters. The left sidebar lists 'Export target', 'Extraction settings', 'Format settings', 'Output', and 'Confirm'. The 'Output' section is expanded, showing a 'General' tab with various settings:


- ☐ Copy to clipboard
- Directory: [Global Settings](#)
- File name pattern:
- Encoding: ☒ Timestamp pattern: ☐ Insert BOM
- ☐ Write to the single file
- ☐ Compress
- ☐ Split output file Maximum file size:
- On object data file name conflict: Ask; On blob value file name conflict: Ask
- [You can use variables in output parameters.](#)

The 'Results' section shows:

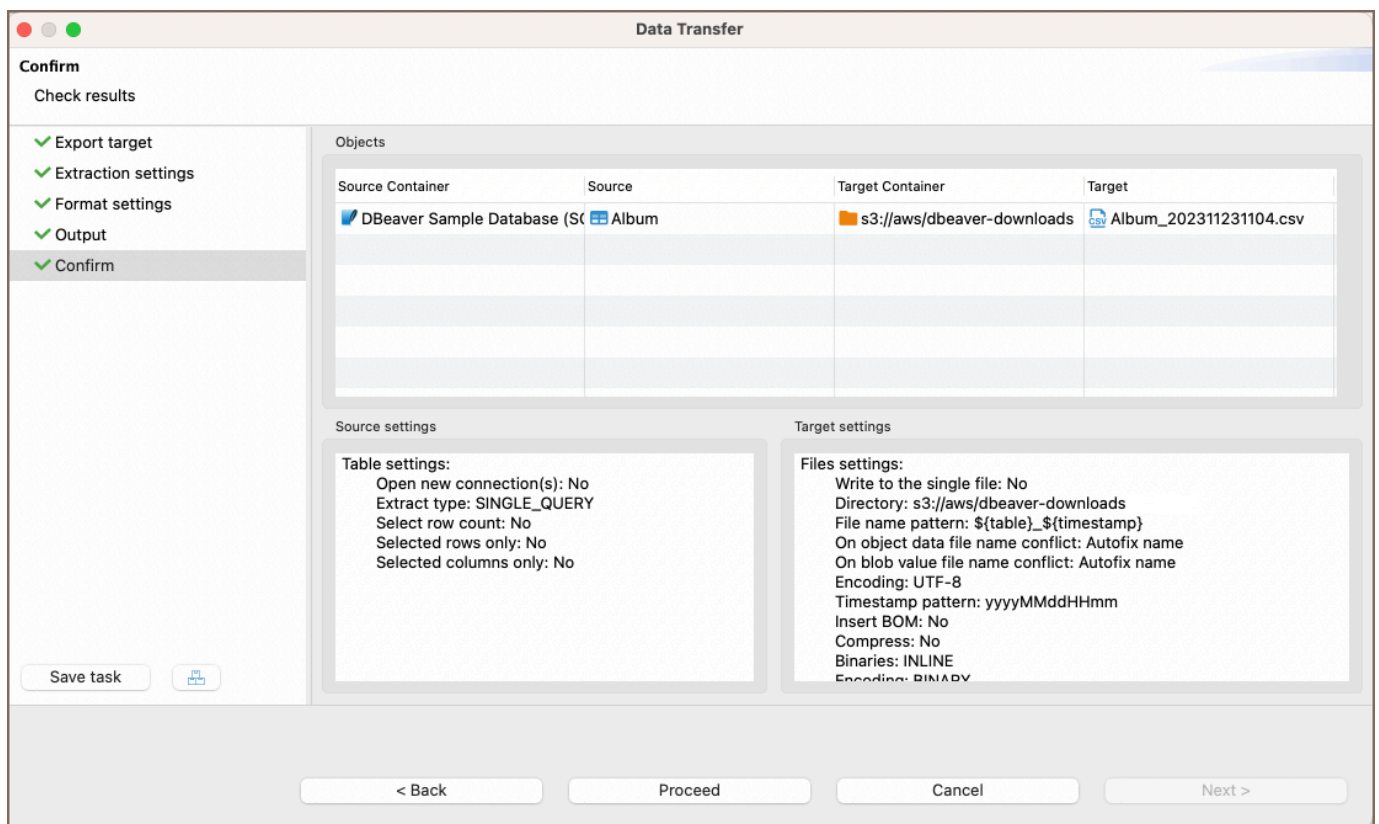
- ☒ Show finish message
- ☐ Show exported file in system explorer
- ☐ Execute process [Configure](#)
- ☐ Send results by E-Mail [Configure](#)

At the bottom, there is a link: [See how you can export files to external storage on our wiki](#)

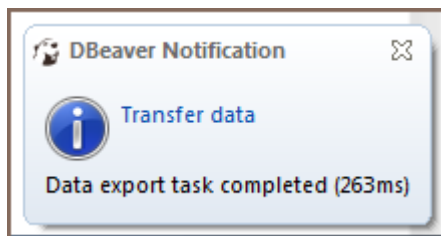
Navigation buttons at the bottom: < Back, Proceed, Cancel, Next >

Tip: You can export files to a remote file system via [Cloud Storage](#) using the **Browser remote file system** button . This feature is exclusively available to users of the Ultimate Edition, Team Edition, and CloudBeaver versions.

6. Review what you want to format and which format you will export. You can also save all your settings as a task in this step or change the task variables:



7. Press finish. See extraction progress. You can keep working with your database during the export process as the extraction will be performed in the background.

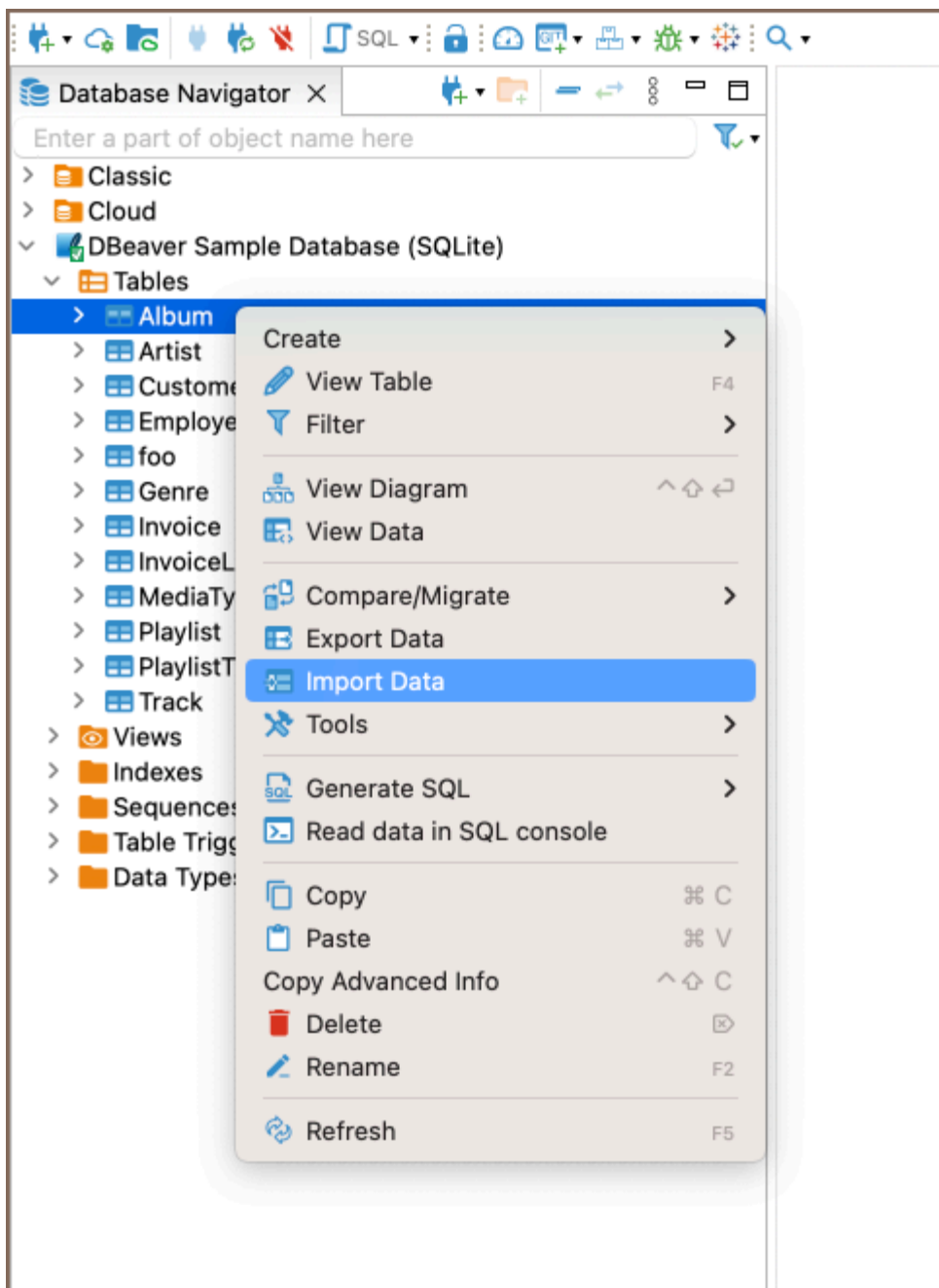


Import data

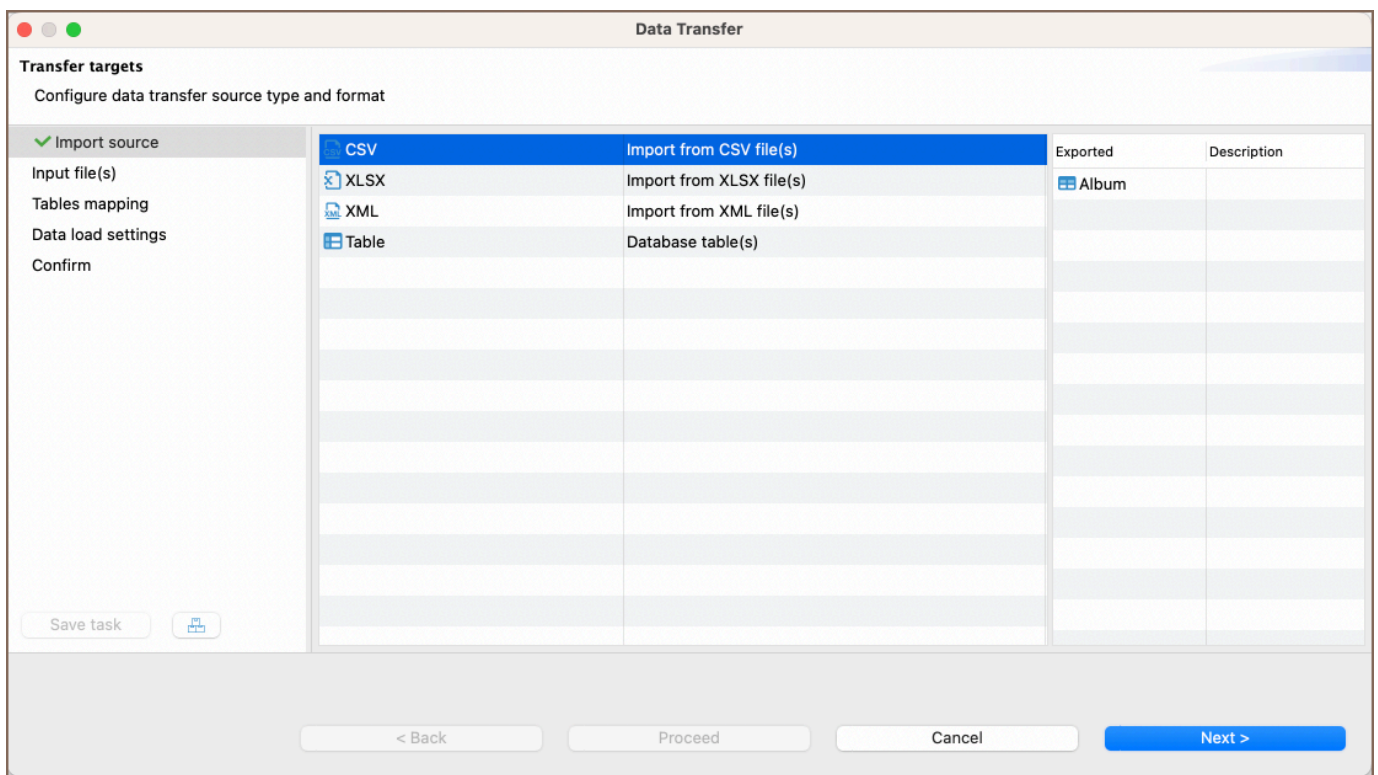
You can import data to your database from **CSV** , **XLSX** , and **XML** files.

Importing data from CSV file

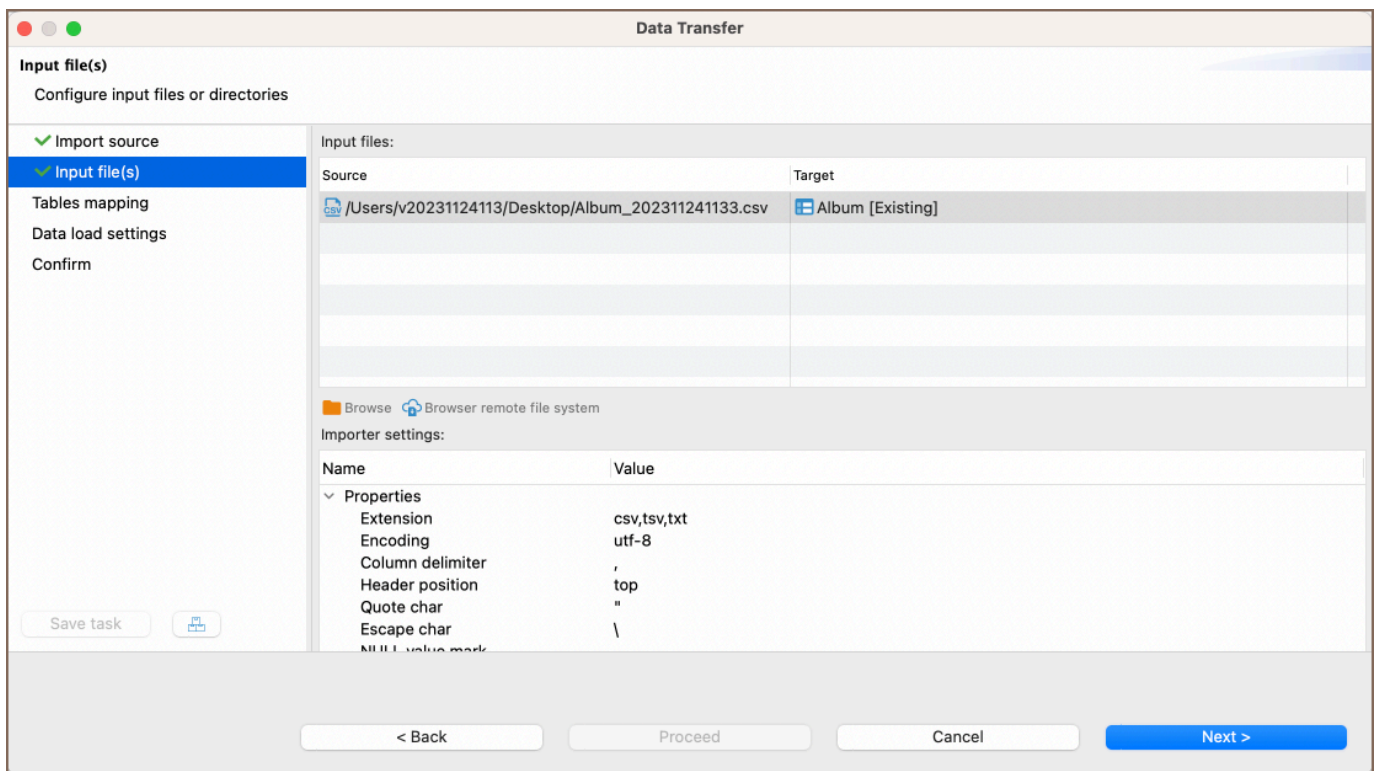
1. Select one or more tables to which you want to import data. In the context menu, choose **Import Data**:




2. In the window that appears, choose **CSV** and click **Next**.

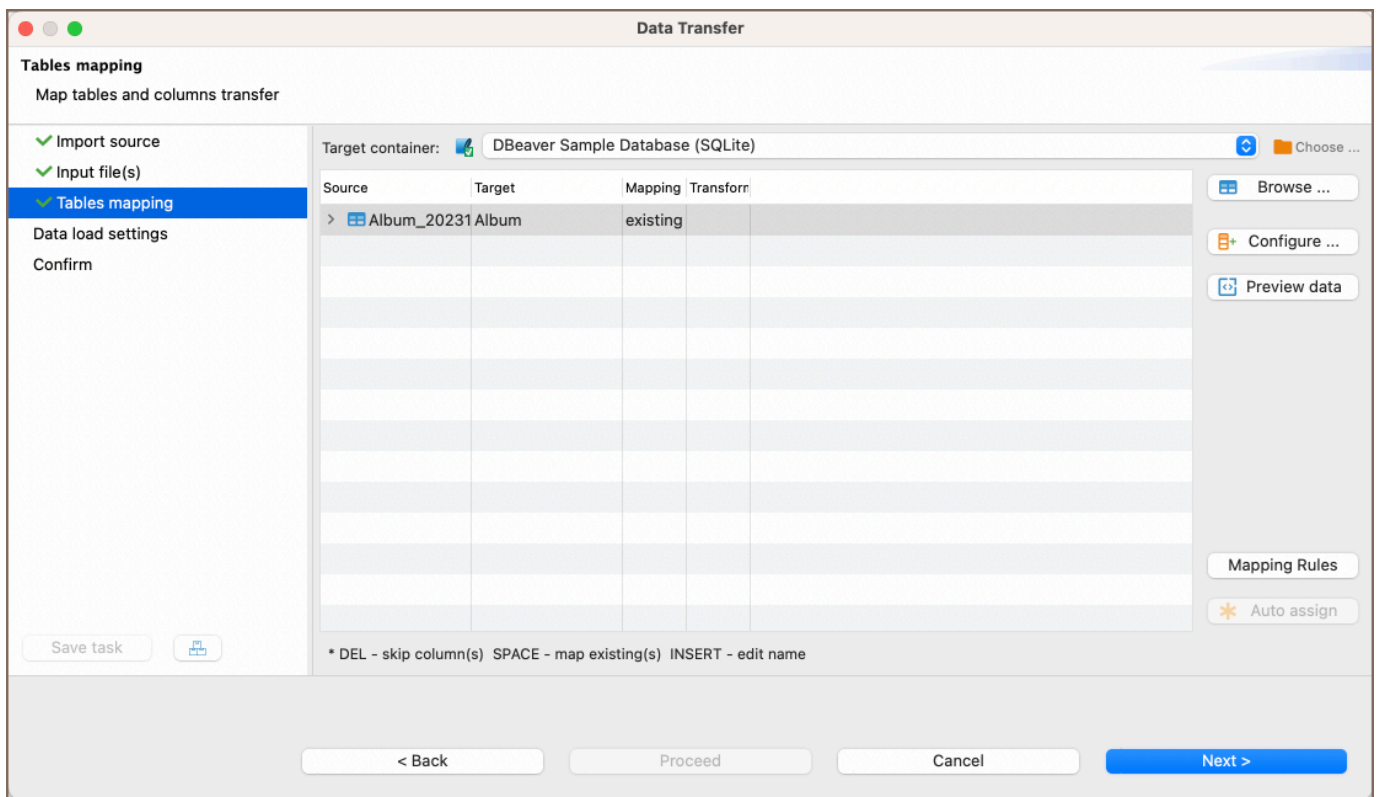


3. Select the input **CSV** file for each table you want to import. You can change the Importer settings (format specific) during this step:

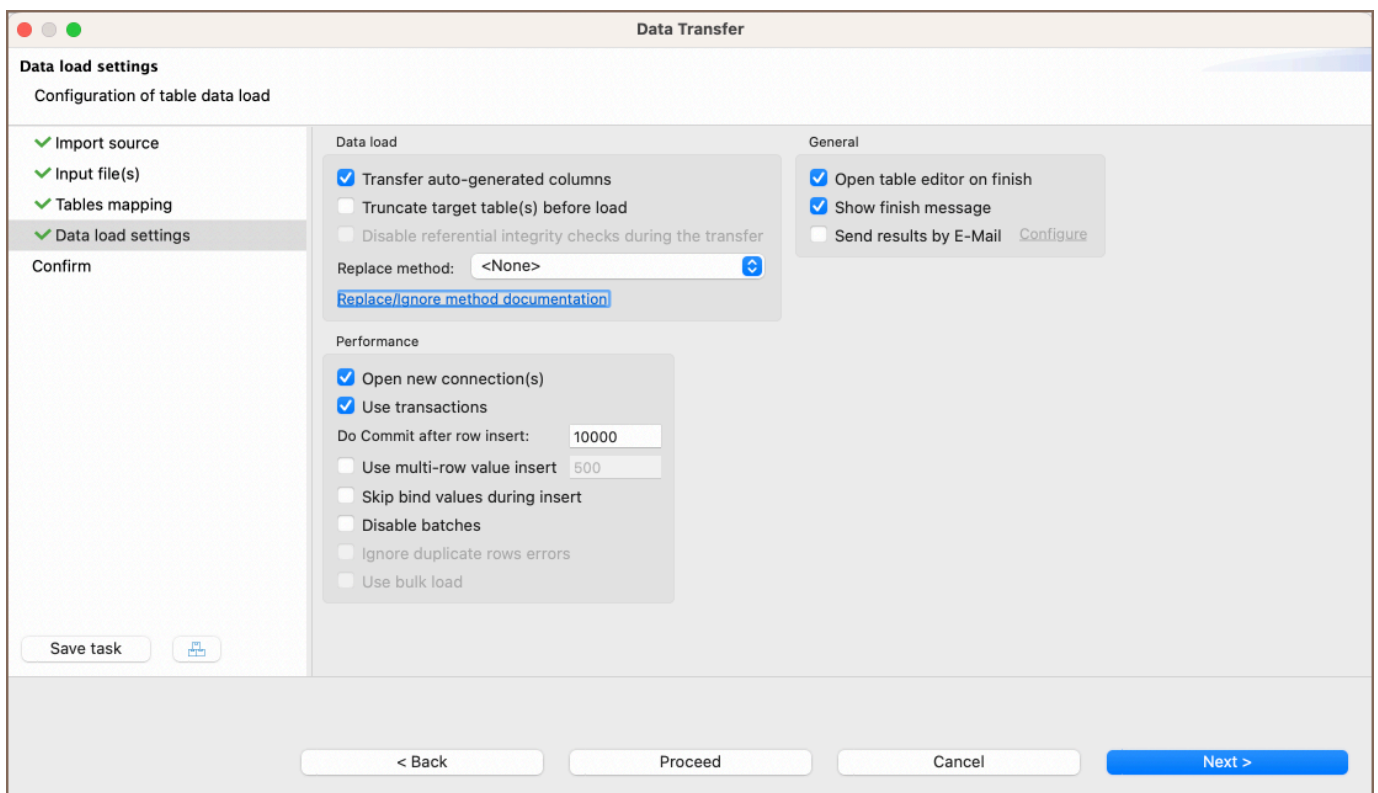


Tip: You can import files from a remote file system using the [Cloud Storage](#) by clicking the **Browser remote file system** button . This feature is exclusively available to users of the Ultimate Edition, Team Edition, and CloudBeaver versions.

4. Set mappings. Please refer to our [mapping process](#) guide for more detailed information.

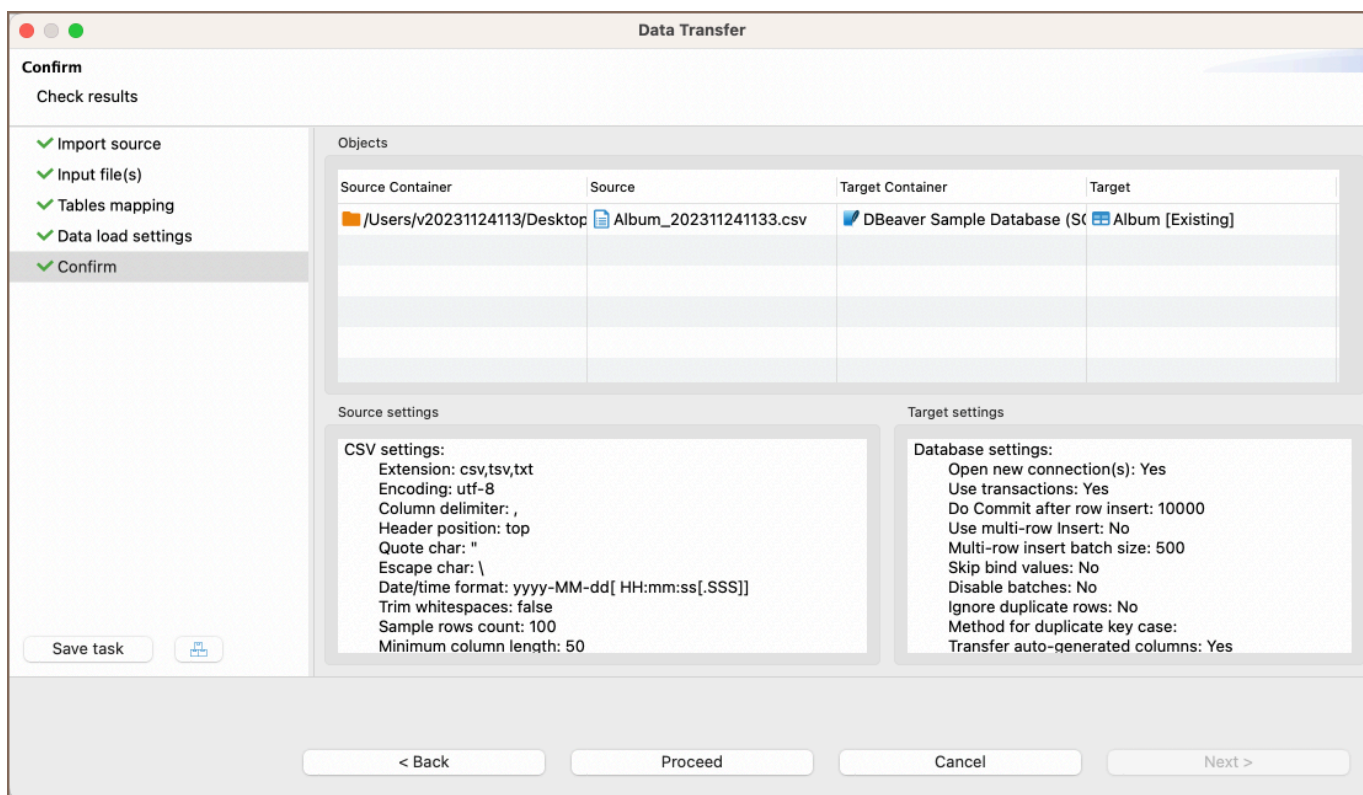


5. Set options for loading data in the database. These options may affect the loading's performance:

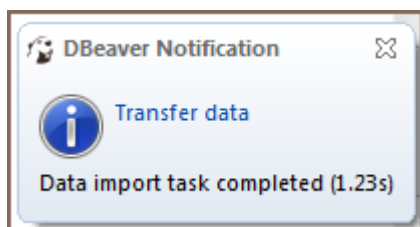


Read our guide on [Data Import and Replace](#) to learn more about the replacing method option.

6. Review which file(s) and to which table(s) you will import. You can also save all your settings as a task in this step:



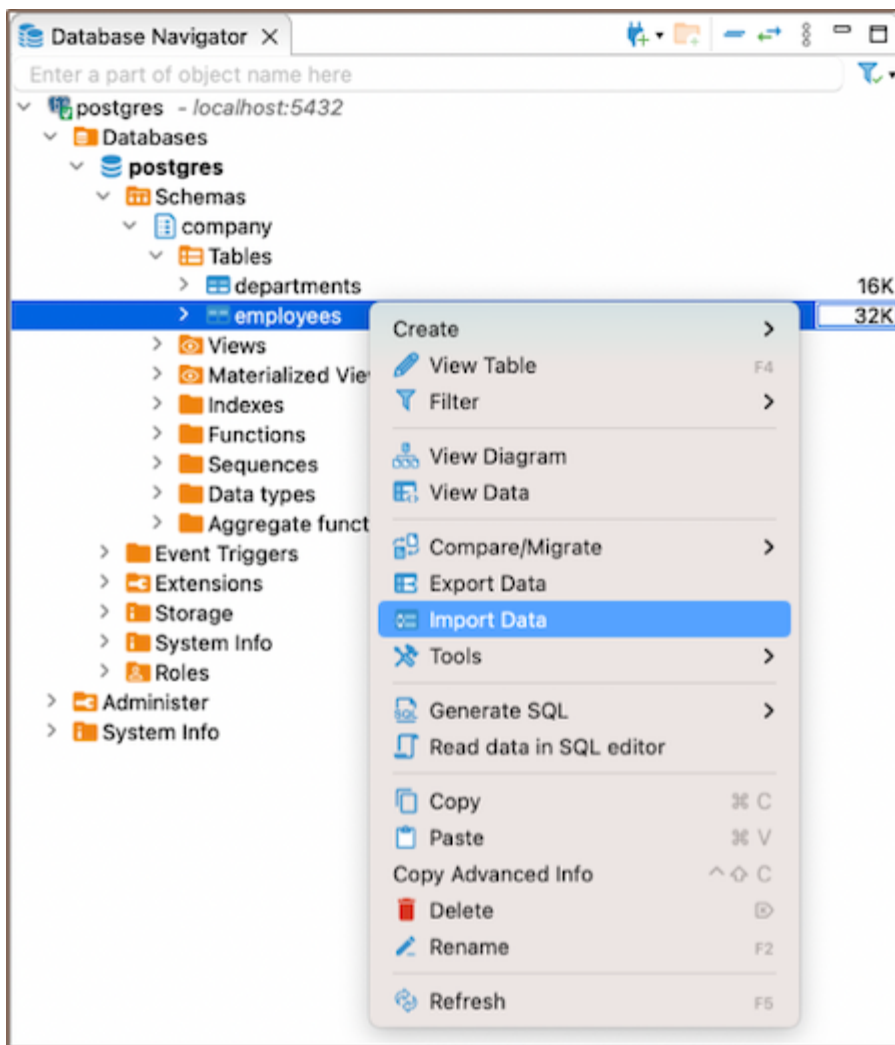
7. Press finish. See extraction progress. You can keep working with your database during the export process as the data loading will be performed in the background. In the end, you will see the following status message:



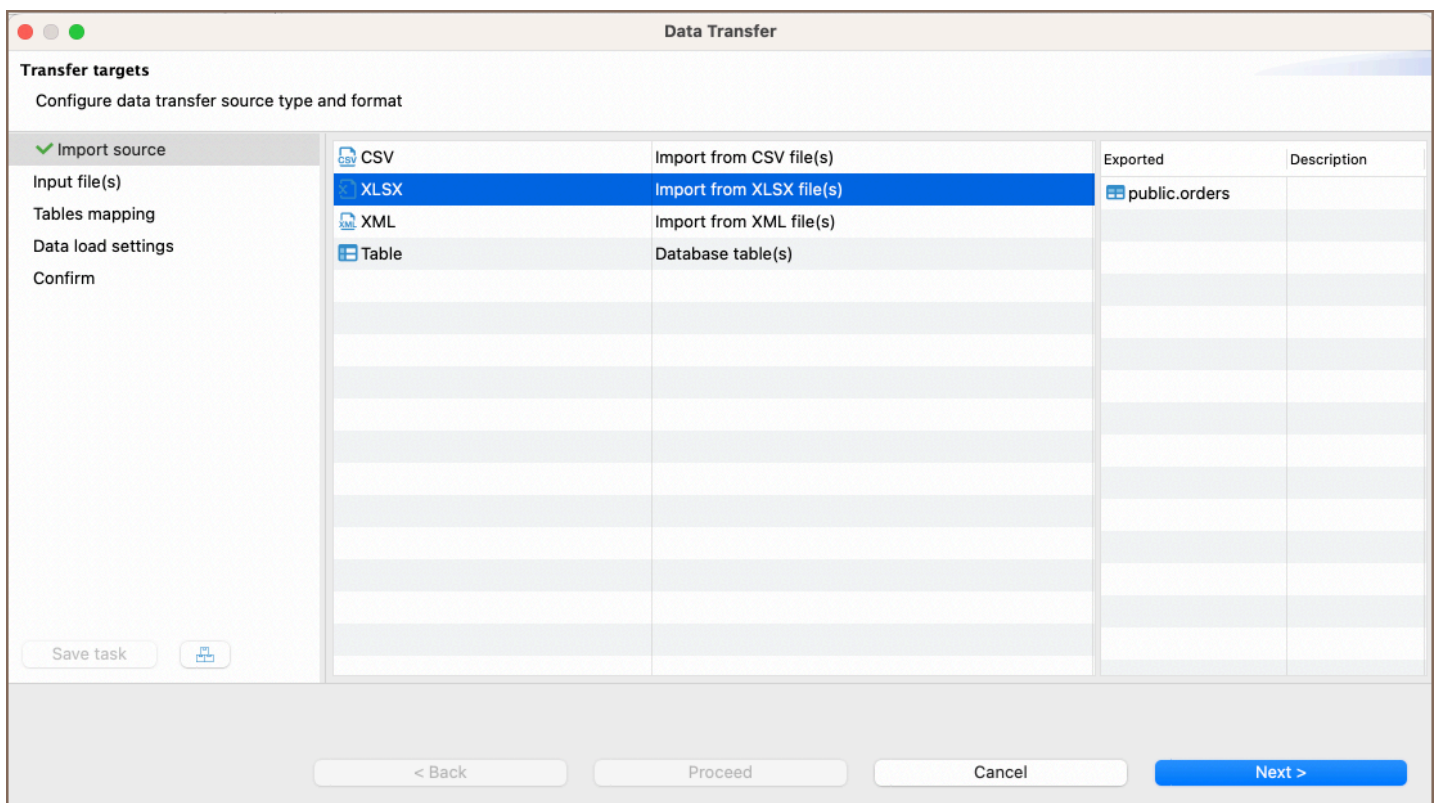
Importing data from Excel file

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.


- 1) Choose the database table (or tables) you want to import data into. Do this by right-clicking on the table name in the **Database Navigator** section and then clicking on **Import Data**.

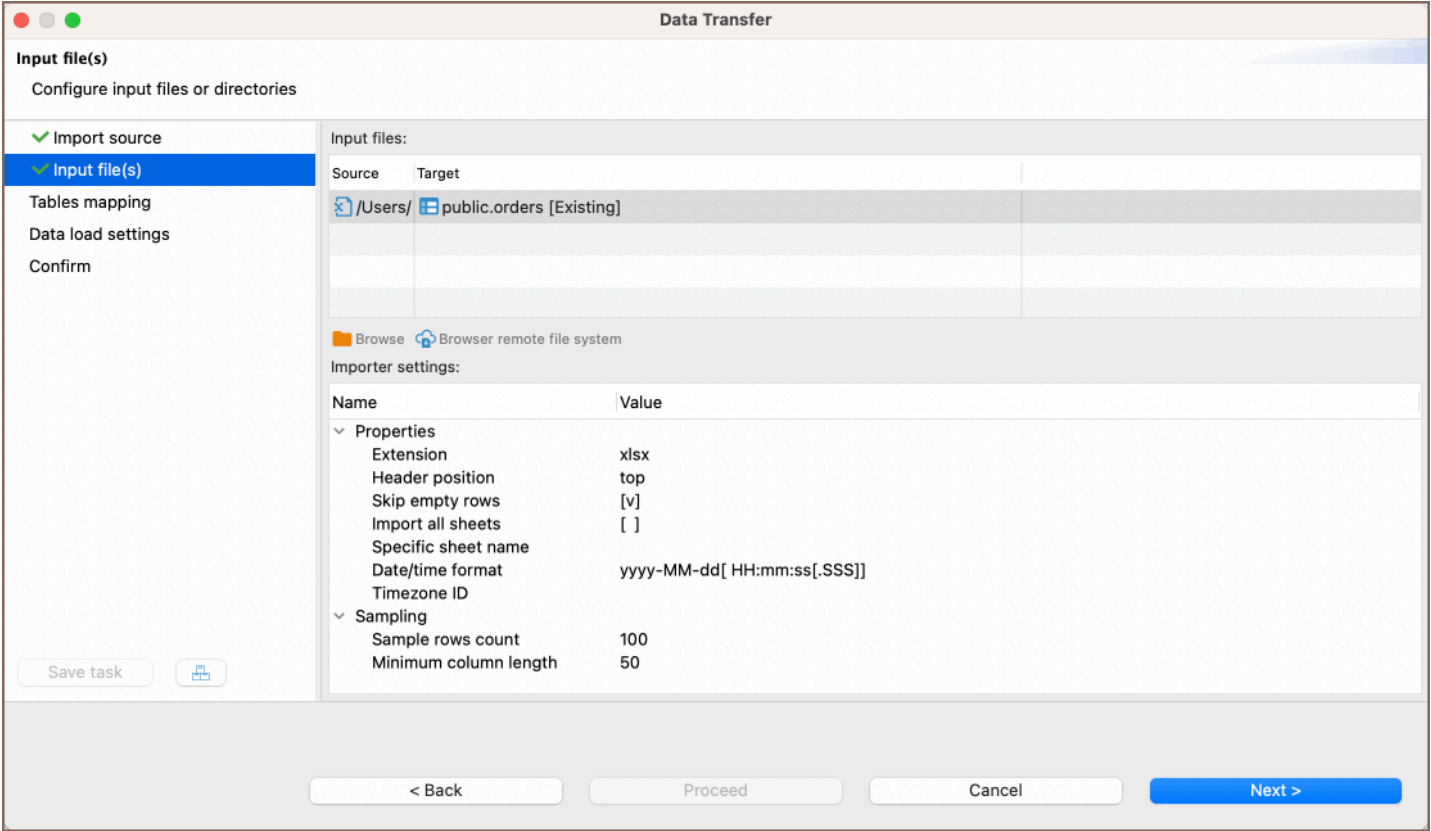


2) In the window that appears, choose **XLSX** and click **Next**.



3) In the following window, choose the file that contains the data you wish to import into the table. Select the appropriate settings described below, then click **Next**.

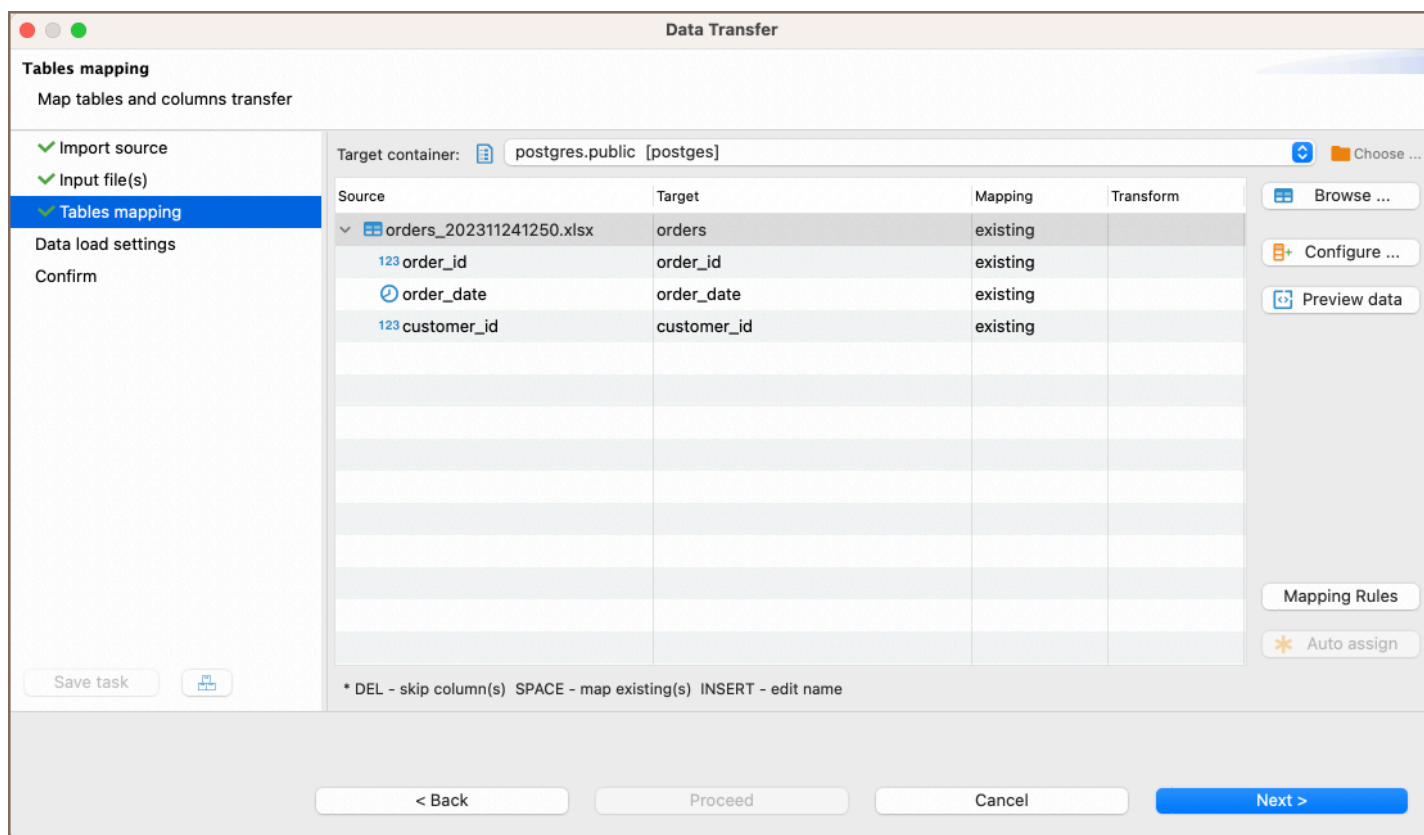
Tip: You can import files from a remote file system using the [Cloud Storage](#) by clicking the **Browser remote file system** button . This feature is exclusively available to users of the Ultimate Edition, Team Edition, and CloudBeaver versions.



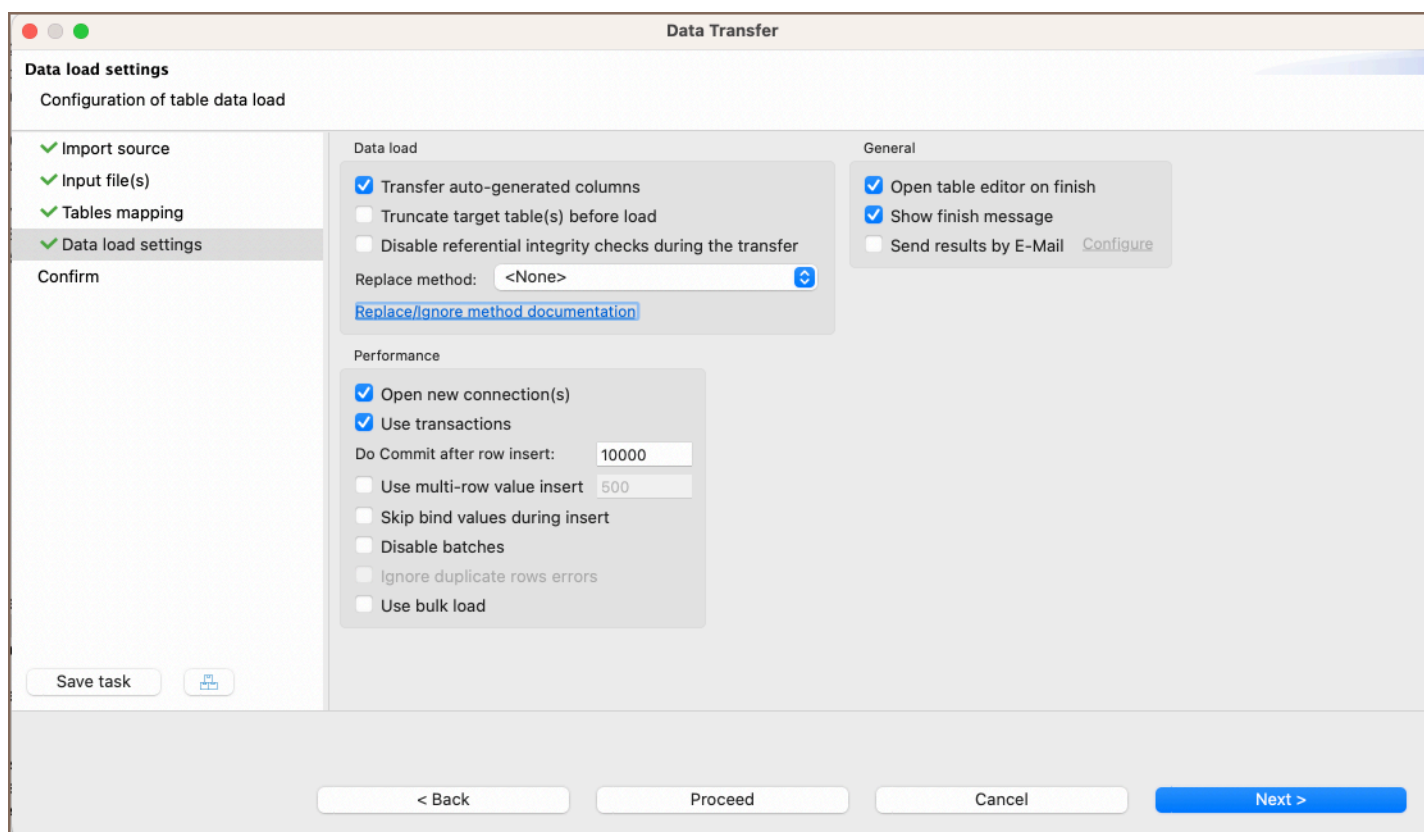
Setting name	Description
Header position	Determines the location of the column names in the Excel table, either at the top or none. This setting specifies whether the column names are located in the first row of the Excel table or if there are no column names present.
Skip empty rows	If this setting is enabled, any open string values encountered during the data processing will be ignored and not inserted into the corresponding cells in the row. If the setting is disabled, all cells in the row will be filled with a NULL value if an empty string is encountered.
Import all sheets	Specifies that all sheets in the file should be imported during the data import process.
Specific sheet name	Enables you to choose a particular sheet from the Excel file for importing during the data transfer process.

Date /time format	Use this setting to specify the date format used in the XLSX file. This is used to clarify the date format during the import process and does not affect the output data. You can refer to the java DateTimeFormatter documentation for details on the format pattern syntax.
Timezone ID	<p>The local machine timezone is used by default. There are three ways to specify the timezone:</p> <ol style="list-style-type: none"> 1) Local zone offset: Specify the offset from UTC in the format of either a positive or negative number (e.g., +3, -04:30). 2) Specific zone offset: Specify the offset from GMT or UTC in the format of GMT+/-X or UTC+/-X (e.g., GMT+2, UTC+01:00). 3) Region-based: Specify the timezone using a region-based identifier such as UTC, ECT, PST, etc.
Sample rows count	Determines the number of rows that will be used as a sample to estimate the length and data types of the imported data.
Minimum column length	This value is used when creating a new column and, if necessary, specifying its type. It indicates the minimum number of characters or digits expected in the column. This information helps determine the appropriate data type and size for the column during the creation process.
Save task	Opens the Save Task window to assist in creating a task during the data transfer process. This window provides options and settings for creating and configuring a task related to the data transfer operation.

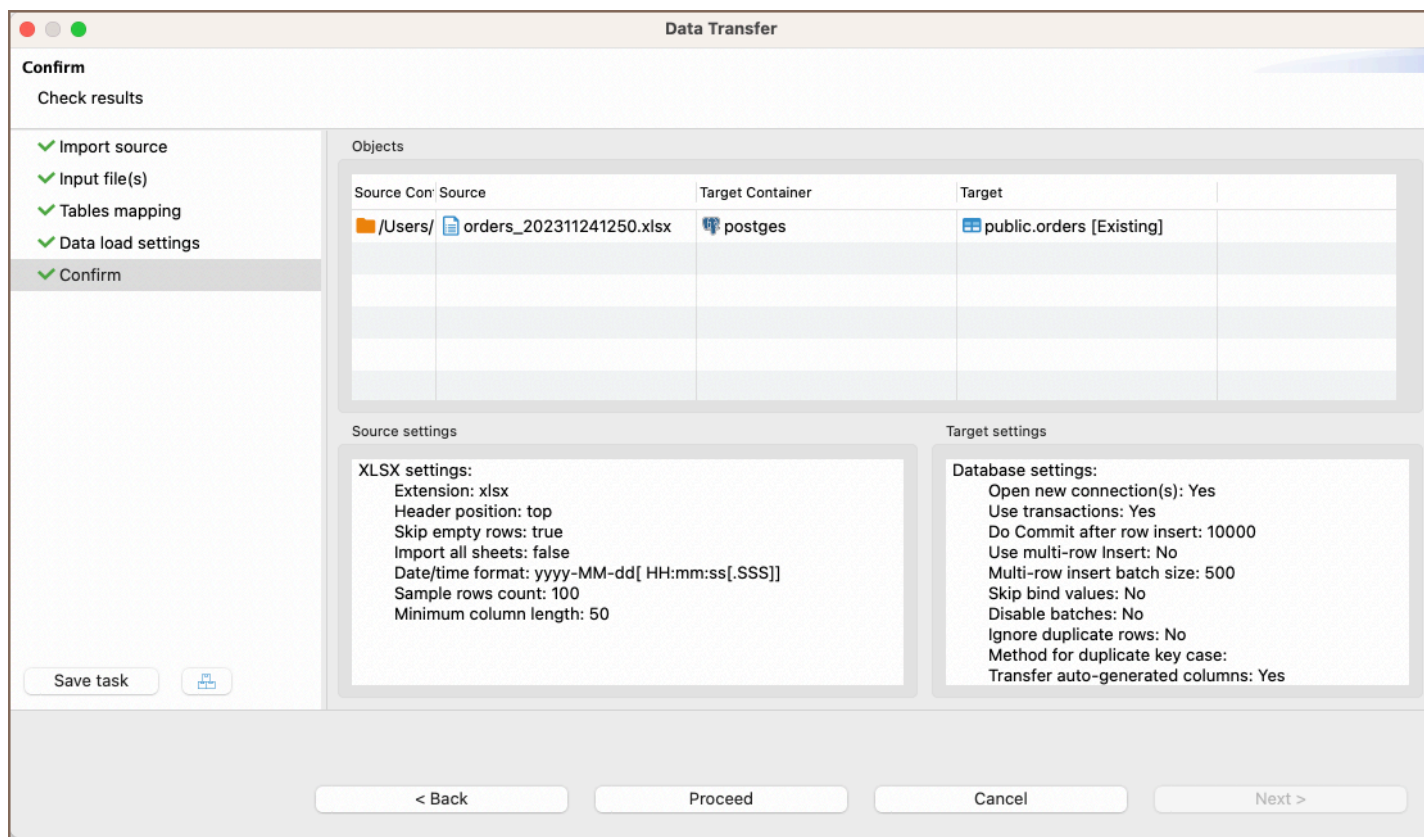
5) In the next window, set XLSX-to-table mappings. Please refer to our [mapping process](#) guide for more detailed information.



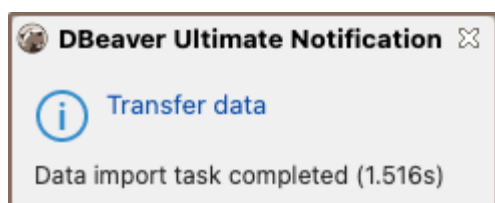
6) Select your data load settings in the subsequent window, and then click **Next**. For more information, please refer to our article's section [Data load settings](#) article.



7) In the final window, you can review all the settings you selected earlier. If you missed something, you could go **Back** and fix it. When you are ready, finish the import by clicking **Proceed**.



8) If the **XLSX** file is valid and there are no errors, you will see a notification window with information about the completion of the task. You can keep working with your database during export, as the data loading will be performed in the background.



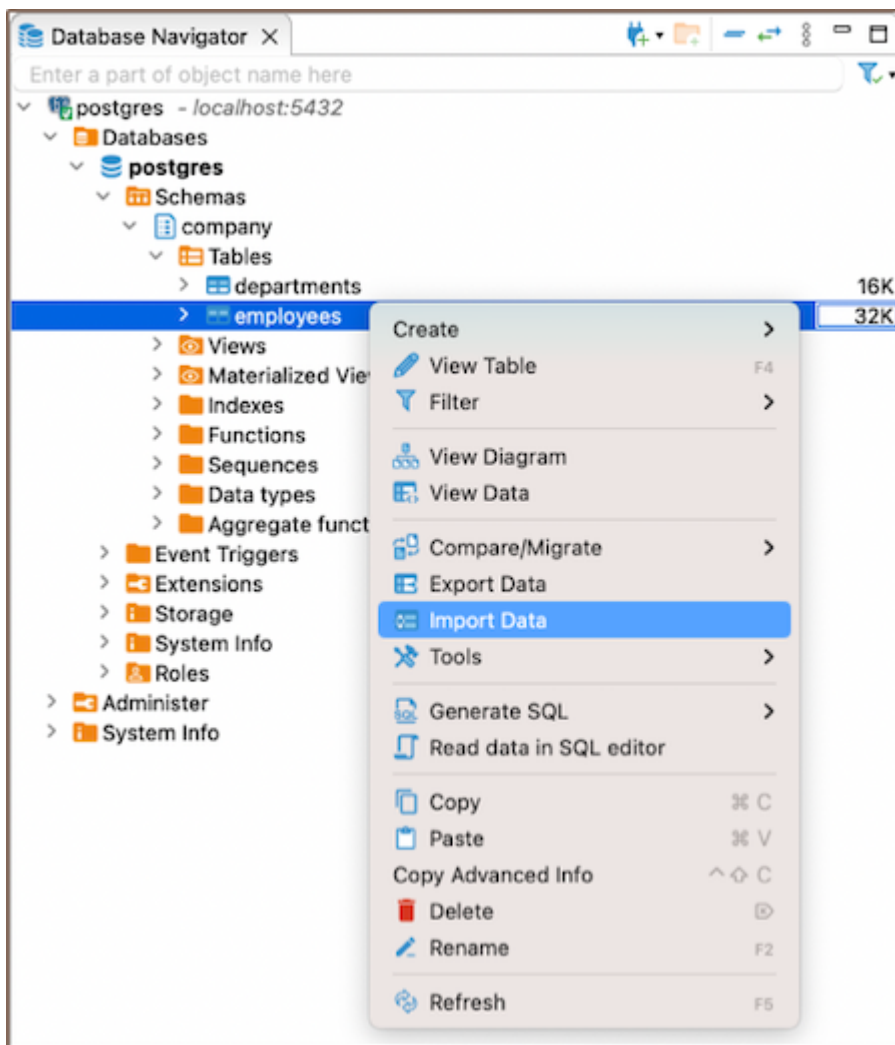
Additional features

When importing from Excel files, DBeaver provides additional features to enhance your data transfer experience:

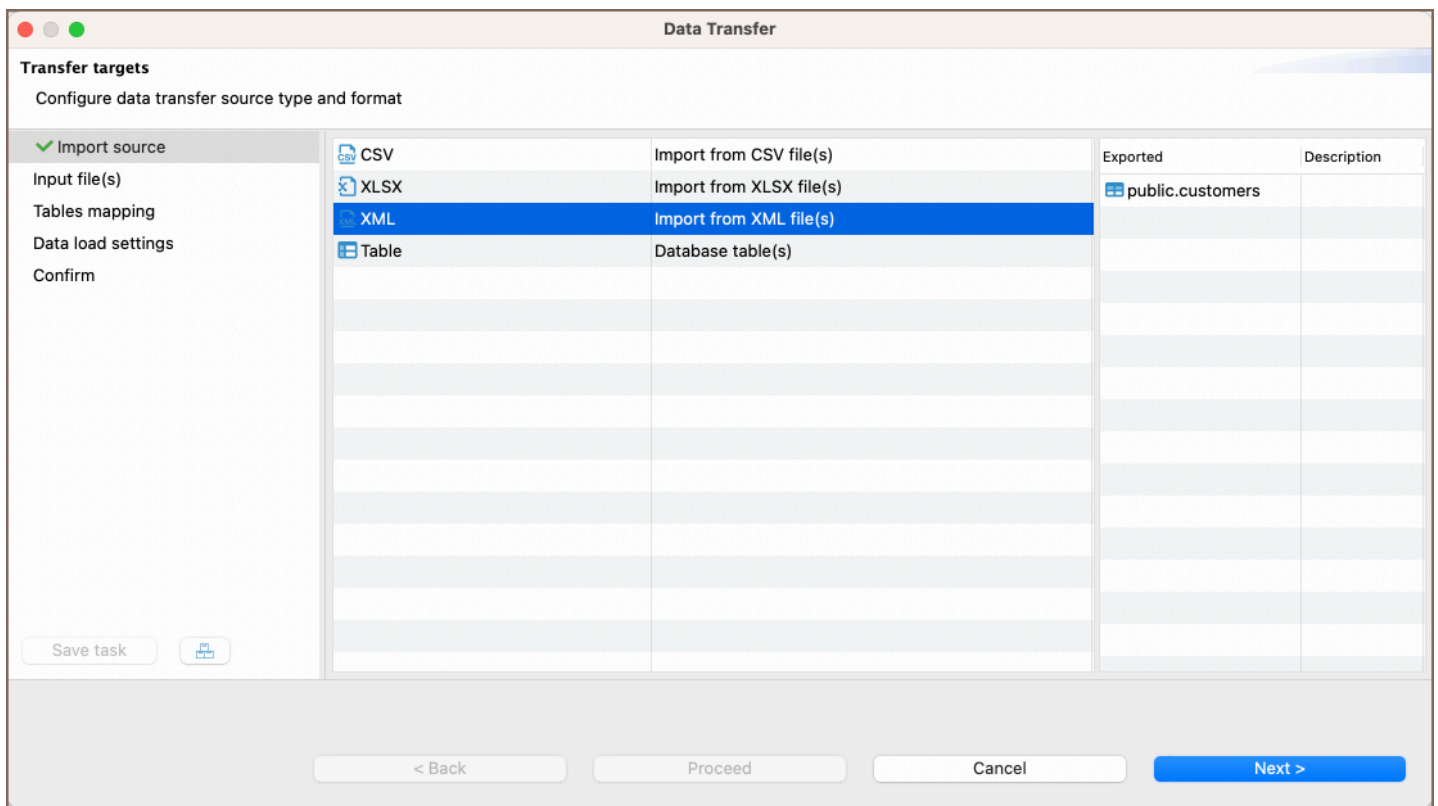
- **Allow importing multiple sheets into different tables:** DBeaver allows you to import multiple sheets from a single **XLSX** file into different tables in your database.

Importing data from XML file

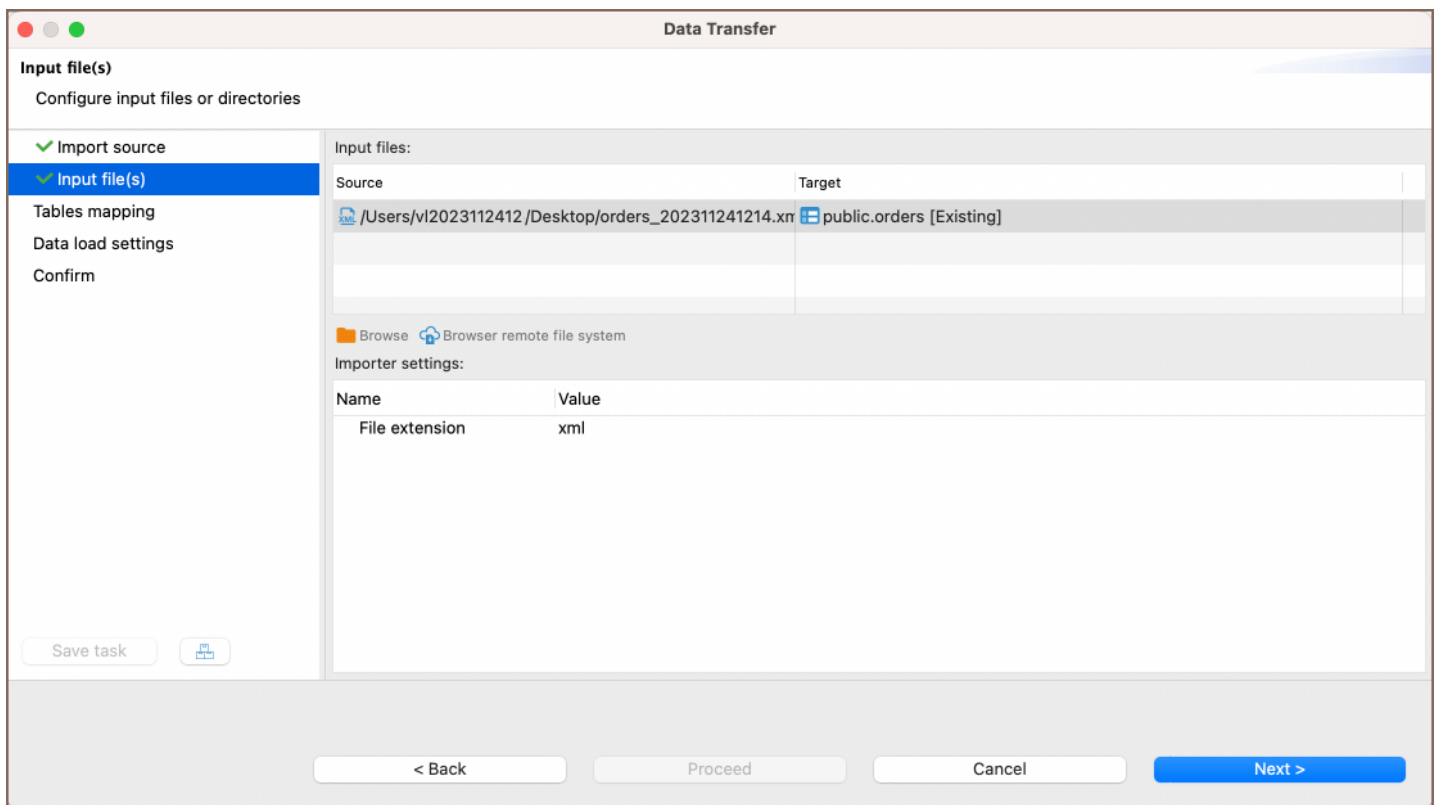
1) Select the database table (or tables) where you want to import data. Do this by right-clicking on the table name in the **Database Navigator** section, then clicking on **Import Data**.




2) In the window that appears, select **XML** and then click **Next**.



3) In the following window, select the **XML** file that contains the data you want to import, then click **Next**.



Tip: You can import files from a remote file system using the [Cloud Storage](#) by clicking the **Browser remote file system** button . This feature is exclusively available to users of the Ultimate Edition, Team Edition, and CloudBeaver versions.

4) In the next window set XLSX-to-table mappings. Please refer to our guide for detailed information on the [mapping process](#).

Tables mapping

Map tables and columns transfer

✓ Import source

✓ Input file(s)

✓ Tables mapping

Data load settings

Confirm

Save task

Target container: postgres.public [postgres]

Source	Target	Mapping	Transform
▼ orders_202311241214.xml	orders	existing	
123 order_id	order_id	existing	
order_date	order_date	existing	
123 customer_id	customer_id	existing	

* DEL - skip column(s) SPACE - map existing(s) INSERT - edit name

Browse ...

Configure ...

Preview data

Mapping Rules

Auto assign

< Back

Proceed

Cancel

Next >

5) select your data load settings in the subsequent window, and then click **Next**. For more information, please refer to our article's section [Data load settings](#).

⏏

⏏

⏏

Data Transfer

Data load settings

Configuration of table data load

✓ Import source


✓ Input file(s)

✓ Tables mapping

✓ Data load settings

Confirm

Save task




Data load

☒ Transfer auto-generated columns

☐ Truncate target table(s) before load

☐ Disable referential integrity checks during the transfer

Replace method: <None> 

[Replace/ignore method documentation](#)

Performance

☒ Open new connection(s)

☒ Use transactions

Do Commit after row insert: 10000

☐ Use multi-row value insert 500

☐ Skip bind values during insert

☐ Disable batches

☐ Ignore duplicate rows errors

☐ Use bulk load

General

☒ Open table editor on finish

☒ Show finish message

☐ Send results by E-Mail [Configure](#)

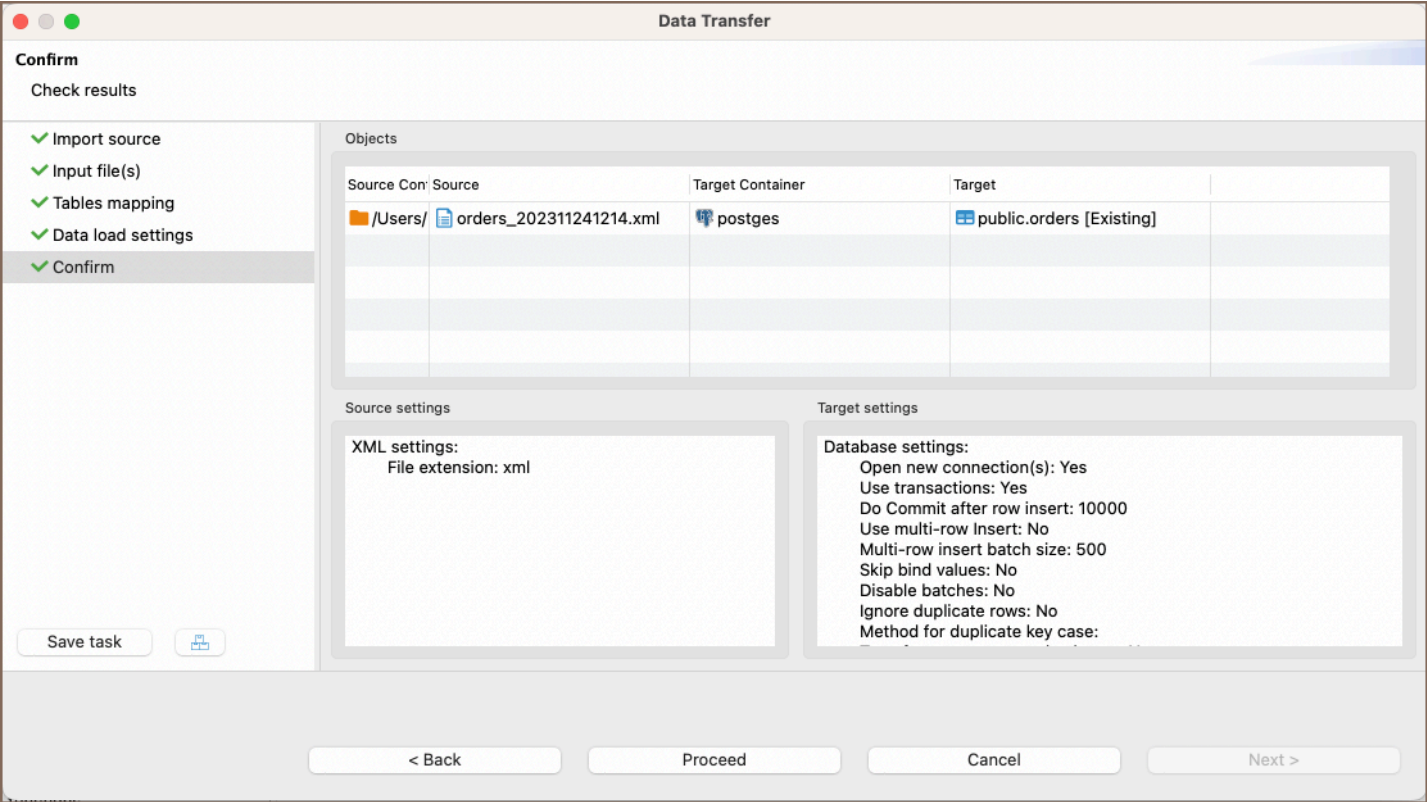
< Back

Proceed

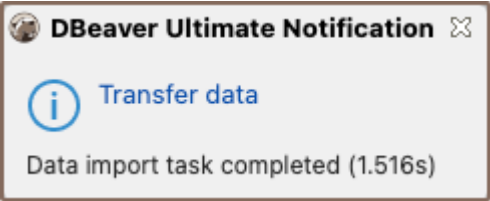
Cancel

Next >

6) In the final window, you can review all the settings you selected earlier. If you missed something, you can go back and adjust it. Once you are happy with everything, finish the import by clicking **Proceed**.



7) If the file is valid and there are no errors, you will see a notification window with information about the completion of the task. You can keep working with your database during export, as the data loading will be performed in the background.



Importing data from the database table

DBeaver offers seamless data migration capabilities, allowing you to transfer data from one database table to another. For more detailed instructions and insights on data migration, you can refer to our [Data Migration](#) guide.

Data import and replace

Table of contents

[Overview](#)

[Which databases support replace and insert methods?](#)

[MySQL](#)

[SQLite](#)

[PostgreSQL](#)

[FireBird](#)

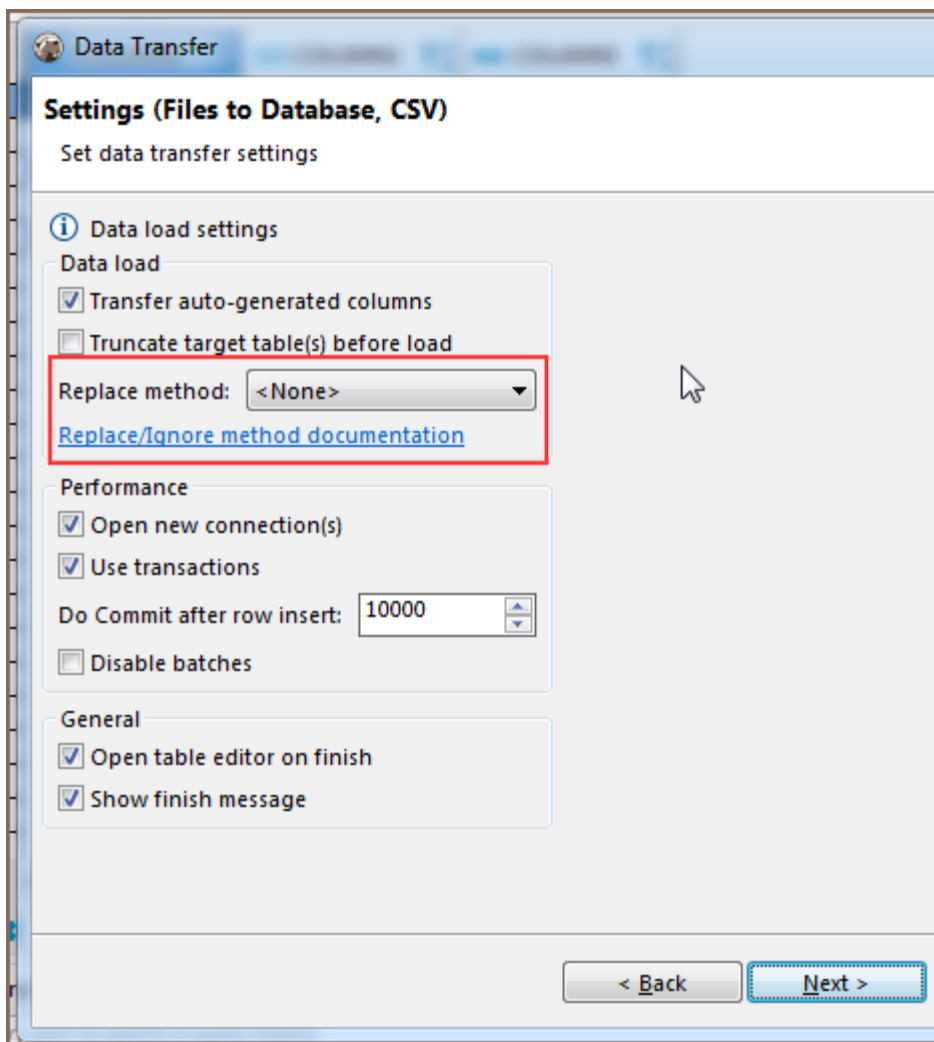
[Oracle](#)

Overview

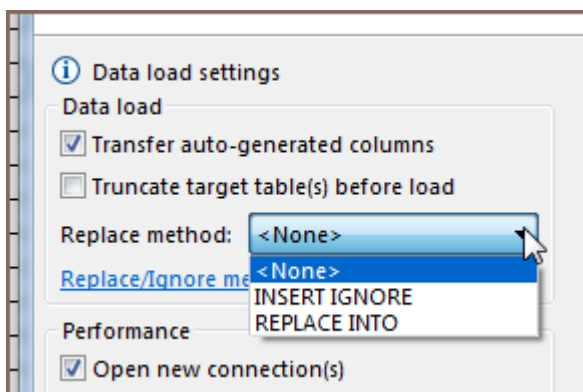
Sometimes there are situations when you want to ignore the current primary key/unique value when importing into a table.

Some databases have syntax constructs in addition to the INSERT INTO that may help.

The choice of the replacement method is in the import settings - in "Data load settings".



By default, the selection is **<None>**, you can select other options from the drop-down list. The options available depend on the target database you are importing to.



The database can only support the replace method or the ignore method. In this case, the list of methods will consist of only one item except **<None>**. If the base does not support the replacement methods, or if we have not added an implementation yet, then the combo with the list will be disabled.

Further, you will find a list of databases supporting these methods and examples of syntax.

Let's take a look at an example of how this works. We use a small, simple, slightly-modified Sakila (MySQL) table - sakila.language

```
CREATE TABLE language_insert (  
  language_id tinyint unsigned NOT NULL,  
  name char(20) NOT NULL,  
  last_update timestamp NOT NULL,  
  PRIMARY KEY (language_id)  
);
```

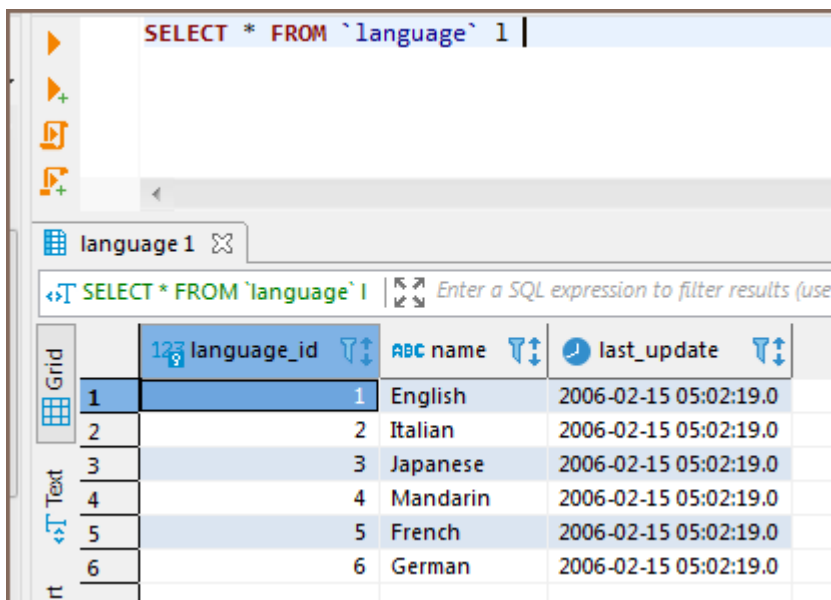
```
INSERT INTO sakila.language_insert (language_id,name,last_update) VALUES  
(1,'English','2006-02-15 05:02:19.0'),  
(2,'Italian','2006-02-15 05:02:19.0'),  
(3,'Japanese','2006-02-15 05:02:19.0');
```

If we try to execute this request twice, we will get the following error: SQL Error [1062] [23000]: Duplicate entry '1' for key 'language_insert.PRIMARY' (This message may look different in other databases).

Let's take a new .csv file with the following content and try to use the replace methods.

```
"language_id","name","last_update"  
1,Spanish,"2020-04-20 05:02:19.0"  
2,Russian,"2020-05-20 05:02:19.0"  
3,Belgian,"2020-06-20 05:02:19.0"  
4,Mandarin,"2006-02-15 05:02:19.0"  
5,French,"2006-02-15 05:02:19.0"  
6,German,"2006-02-15 05:02:19.0"
```

If we set the "INSERT IGNORE" method in the settings, the result of the insert will look like this:

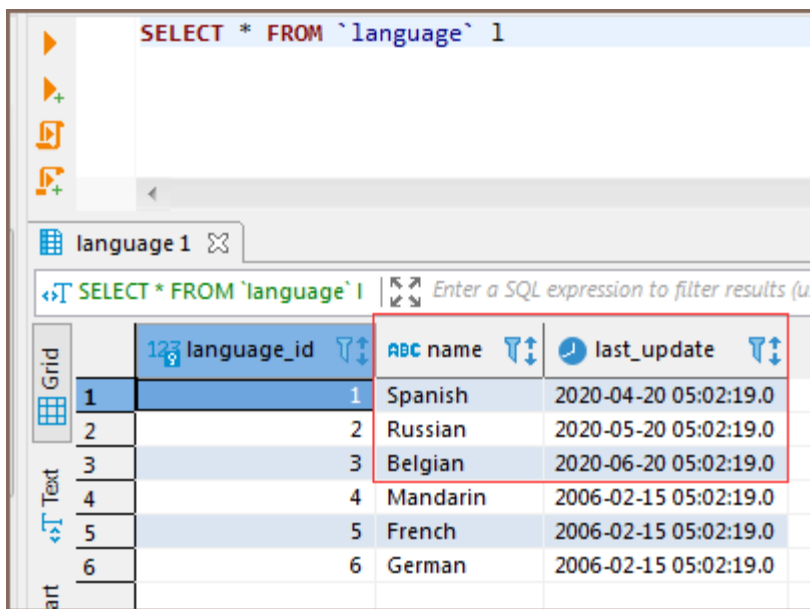


The screenshot shows the DBeaver SQL editor interface. The SQL query editor at the top contains the query: `SELECT * FROM `language` 1`. Below the query editor, the table structure is displayed with columns: `language_id`, `name`, and `last_update`. The table data is shown in a grid view with 6 rows. The first three rows (English, Italian, Japanese) are highlighted in blue, and the last three rows (Mandarin, French, German) are highlighted in light blue.

	language_id	name	last_update
1	1	English	2006-02-15 05:02:19.0
2	2	Italian	2006-02-15 05:02:19.0
3	3	Japanese	2006-02-15 05:02:19.0
4	4	Mandarin	2006-02-15 05:02:19.0
5	5	French	2006-02-15 05:02:19.0
6	6	German	2006-02-15 05:02:19.0

There will be no insertion errors, the first three lines will not change, and the fourth to sixth lines will be added to the table.

If we set the "REPLACE INTO" method in the settings, the result of the insert will look like this:



The screenshot shows the DBeaver SQL editor interface. At the top, the SQL editor contains the query: `SELECT * FROM `language` 1`. Below the editor, the results pane shows a table named 'language 1'. The table has three columns: 'language_id', 'name', and 'last_update'. The results are as follows:

	language_id	name	last_update
1	1	Spanish	2020-04-20 05:02:19.0
2	2	Russian	2020-05-20 05:02:19.0
3	3	Belgian	2020-06-20 05:02:19.0
4	4	Mandarin	2006-02-15 05:02:19.0
5	5	French	2006-02-15 05:02:19.0
6	6	German	2006-02-15 05:02:19.0

There will be no insertion errors, the first three lines will be replaced and the fourth to sixth lines will be added to the table.

Which databases support replace and insert methods?

MySQL

[INSERT IGNORE](#) and [REPLACE INTO](#)

Insert examples:

"INSERT IGNORE"

```
INSERT IGNORE INTO language_insert(language_id, name, last_update)
VALUES(1, 'English', '2006-02-15 05:02:19.0');
```

"REPLACE INTO"

```
REPLACE INTO language_insert(language_id, name, last_update)
VALUES(1, 'English', '2006-02-15 05:02:19.0');
```

SQLite

[Documentation](#)

"INSERT OR IGNORE" and "INSERT OR REPLACE"

Insert examples:

"INSERT OR IGNORE"

```
INSERT OR IGNORE INTO language_insert(language_id, name, last_update)
VALUES(1, 'English', '2006-02-15 05:02:19.0');
```

"INSERT OR REPLACE"

```
INSERT OR REPLACE INTO language_insert(language_id, name, last_update)
VALUES(1, 'English', '2006-02-15 05:02:19.0');
```

PostgreSQL

Available for [PostgreSQL version 9.5](#).

"ON CONFLICT DO NOTHING" and "ON CONFLICT DO UPDATE SET"

Insert examples:

"ON CONFLICT DO NOTHING"

```
INSERT INTO language_insert(language_id, name, last_update)
VALUES(1, 'English', '2006-02-15 05:02:19.0') ON CONFLICT DO NOTHING;
```

"ON CONFLICT DO UPDATE SET"

```
INSERT INTO language_insert(language_id, name, last_update)
VALUES(1, 'English', '2006-02-15 05:02:19.0')
ON CONFLICT (language_id)
DO UPDATE SET (language_id, name, last_update) = (EXCLUDED.language_id, EXCLUDED.name, EXCLUDED.last_update);
```

FireBird

Available for [FireBird version 2.1](#).

"UPDATE OR INSERT INTO"

Insert examples:

"UPDATE OR INSERT INTO"

```
UPDATE OR INSERT INTO language_insert(language_id, name, last_update)
VALUES(1, 'English', '2006-02-15 05:02:19.0');
```

Oracle

Available for [Oracle version 11.2](#).

"INSERT IGNORE ROW INDEX"

Insert examples:

"INSERT IGNORE ROW INDEX"

```
INSERT /*+ IGNORE_ROW_ON_DUPKEY_INDEX(LANGUAGE_INSERT, LANGUAGE_INSERT_PK) */  
INTO LANGUAGE_INSERT(LANGUAGE_ID, NAME, LAST_UPDATE) VALUES(1, 'English', TIMESTAMP '2006-02-15 05:02:19.6
```

Data migration

Table of contents

[Overview](#)

[Data migration process](#)

[Step 1 Define the data source](#)

[Step 2 Define data transfer target type](#)

[Step 3 Tables mapping](#)

[Available options](#)

[Target container and target table](#)

[Specify target container](#)

[Define a target table](#)

[Mapping](#)

[Transform column values](#)

[Configure](#)

[Column mapping](#)

[Table properties](#)

[Target DDL](#)

[Mapping Rules](#)

[Keyboard Shortcuts](#)

[Step 4 Extraction settings](#)

[Step 5 Data load settings](#)

[Step 6 Confirm](#)

[Step 7 Export completion notification](#)

Overview

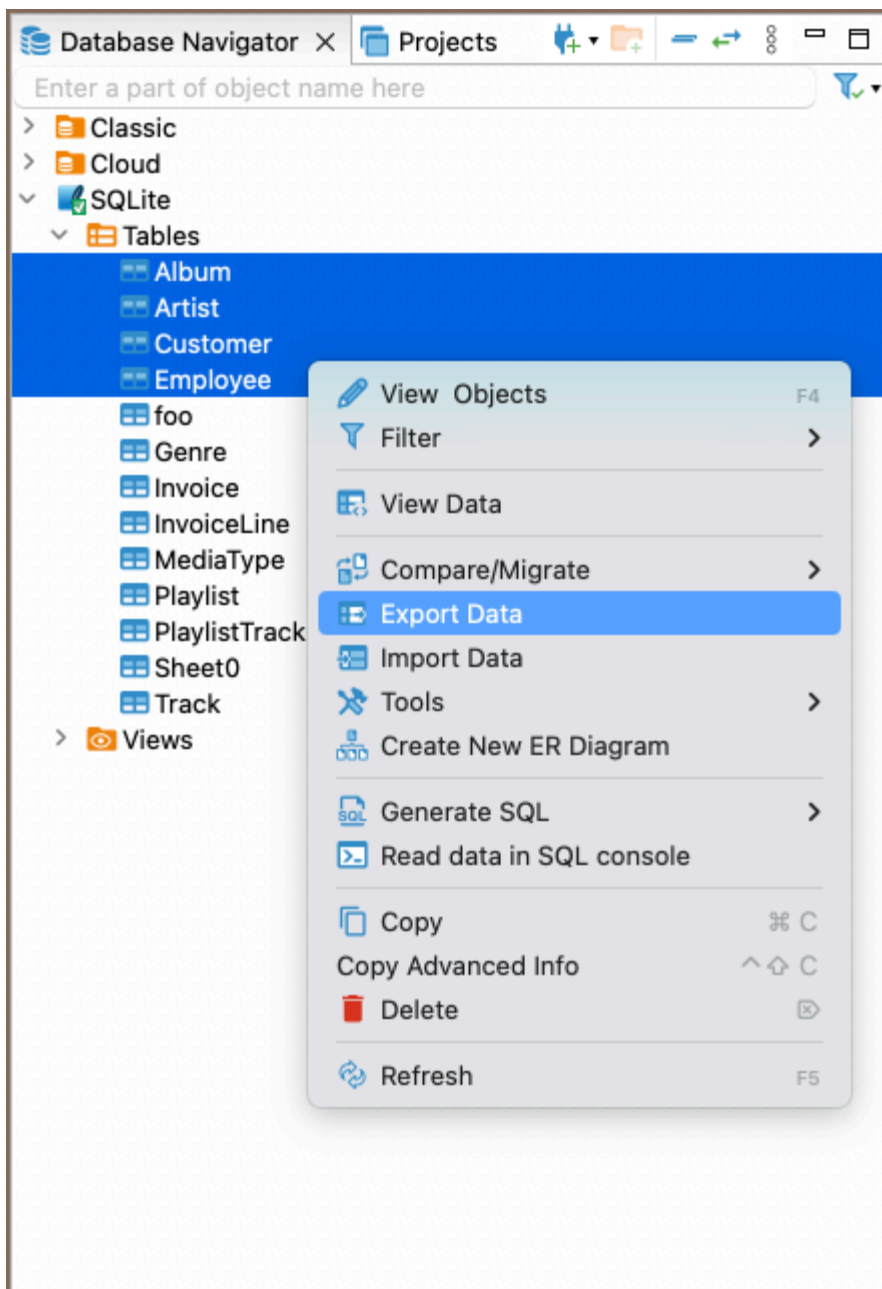
Data Migration in DBeaver provides the functionality to transfer data between different databases or between tables within the same database. This guide walks you through the steps to execute a data migration using DBeaver.

Data migration process

Step 1 Define the data source

To initiate the data migration, you need to select your data source. Follow the steps below:

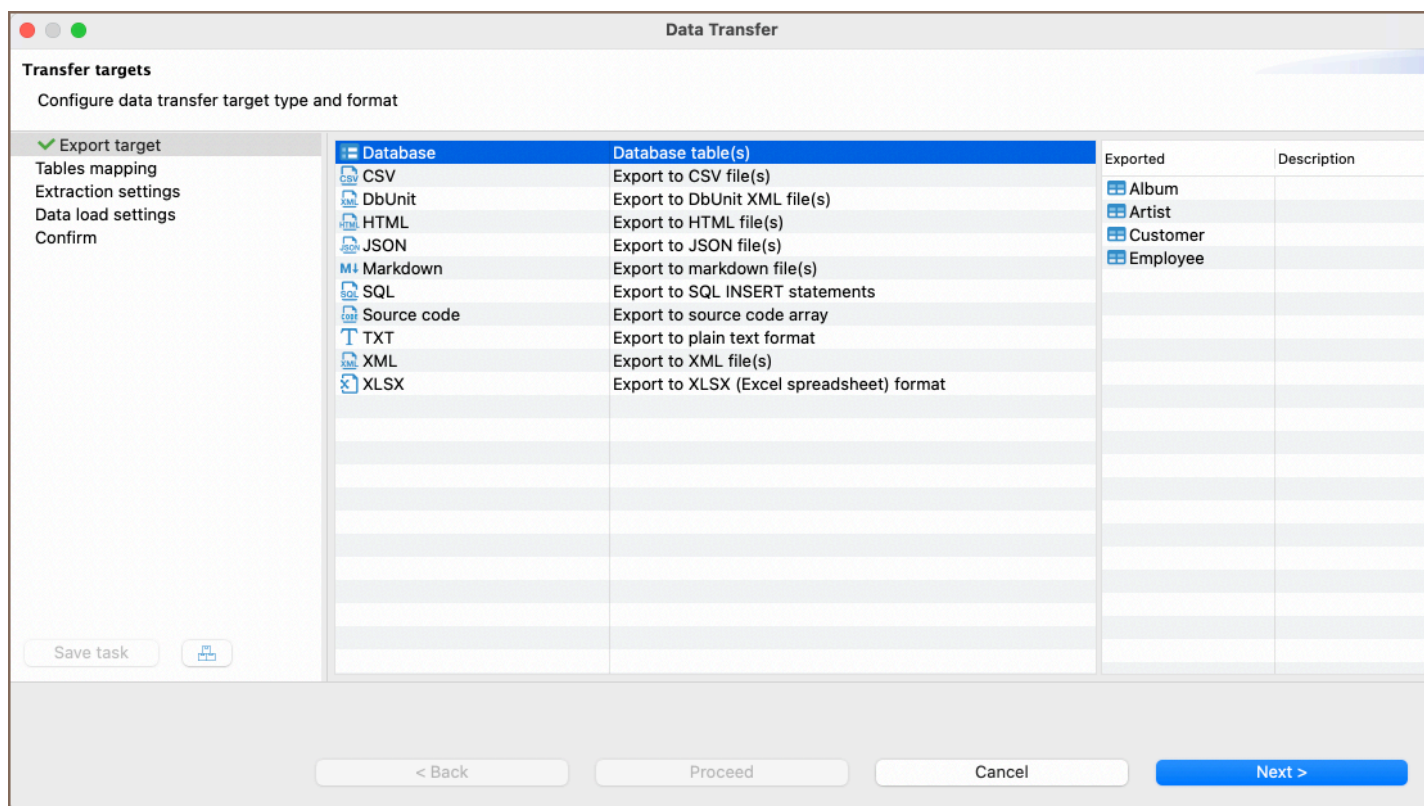
1. Navigate to the [Database Navigator](#).
2. Select one or multiple tables that you want to export.
3. Right-click to open the context menu.
4. Choose **Export Data** from the options.



Tip: You can export data from a custom SQL query. Execute the query and then choose **Export Data** from the results context menu.

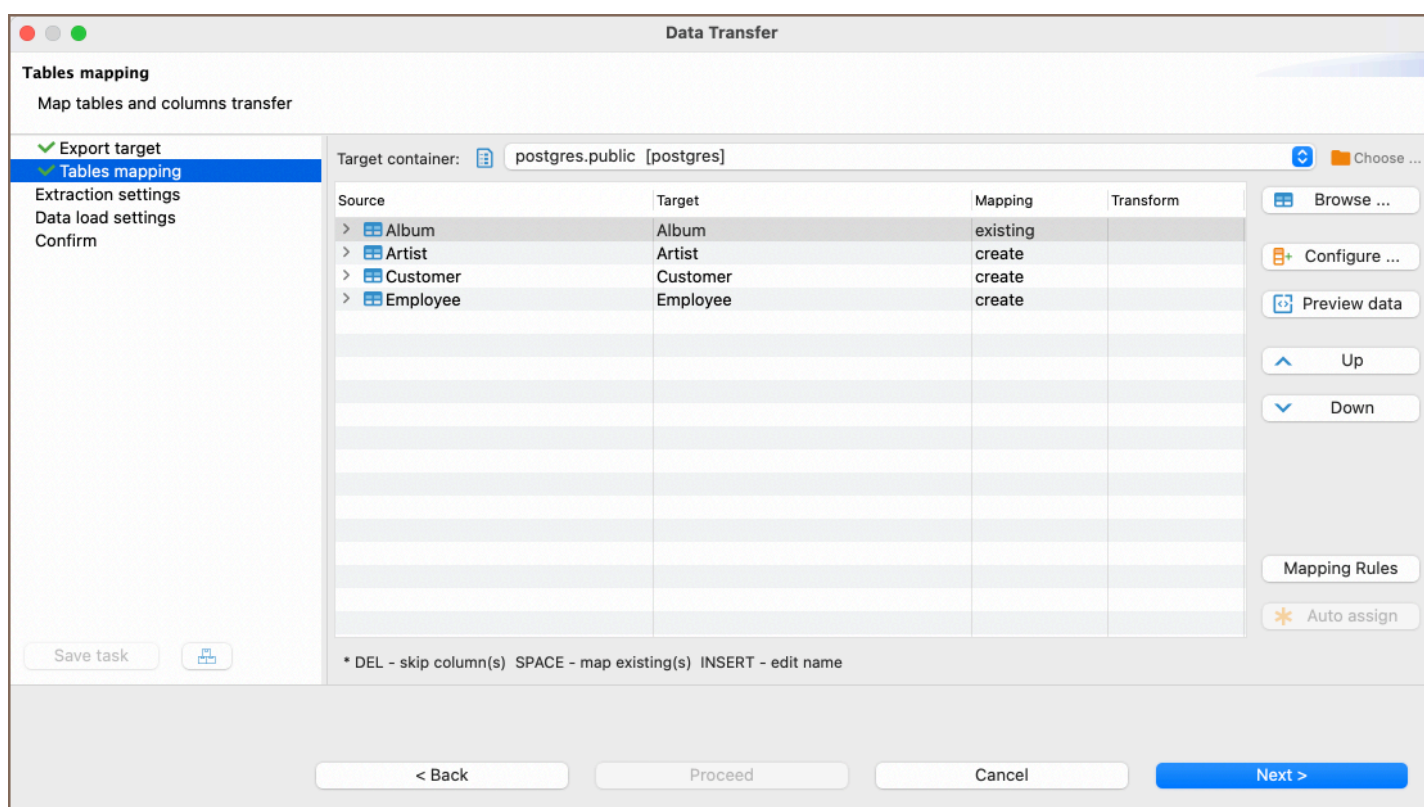
Step 2 Define data transfer target type

After selecting the data source, the next step is to specify the type of destination for the data transfer. Choose **Database** type as the data transfer target and press **Next**.











Step 3 Tables mapping

Once you have chosen **Database** as the transfer target type, the next step involves mapping data. This process includes specifying options, selecting the target container, and setting other configurations. Configure your data mapping settings and press **Next** to proceed to the next step.



Available options

Buttons:

Icon	Option	Description
	Browse	Allows selection of target container.
	Configure	Opens additional settings. For details, see the Configure section.
	Preview data	Shows a preview of the data to be transferred.
	Up	Moves the selected mapping closer to the head of the queue.
	Down	Moves the selected mapping closer to the tail of the queue.
	Mapping Rules	Choose how new tables and column names are transformed when transferring data. For details, see Mapping Rules section.
	Auto assign	Automatically populates the Target column based on the source.
	Choose...	Specify the target container where the data will be transferred in.

Fields:

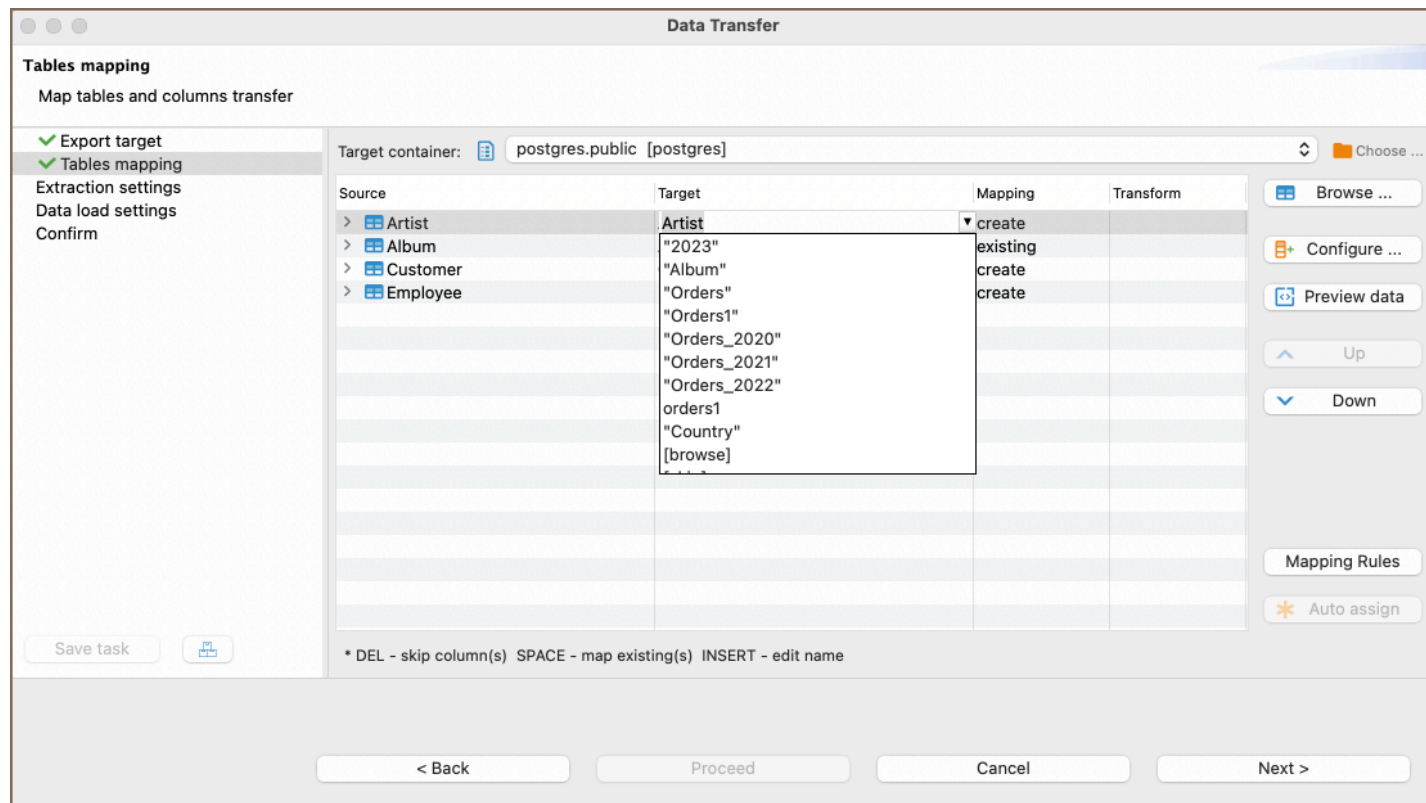
Option	Description
Target container	Defines the database or schema for the data transfer. For more details, see the Target Container section.
Source	Displays the names of selected tables and their columns.
Target	Shows the names of destination tables.
Mapping	Lists the available actions for data transfer. For more details, see the Mapping section.
Transform	Allows for the transformation of column values during the data transfer. For more details, see the Transform section.

Target container and target table

Specify target container

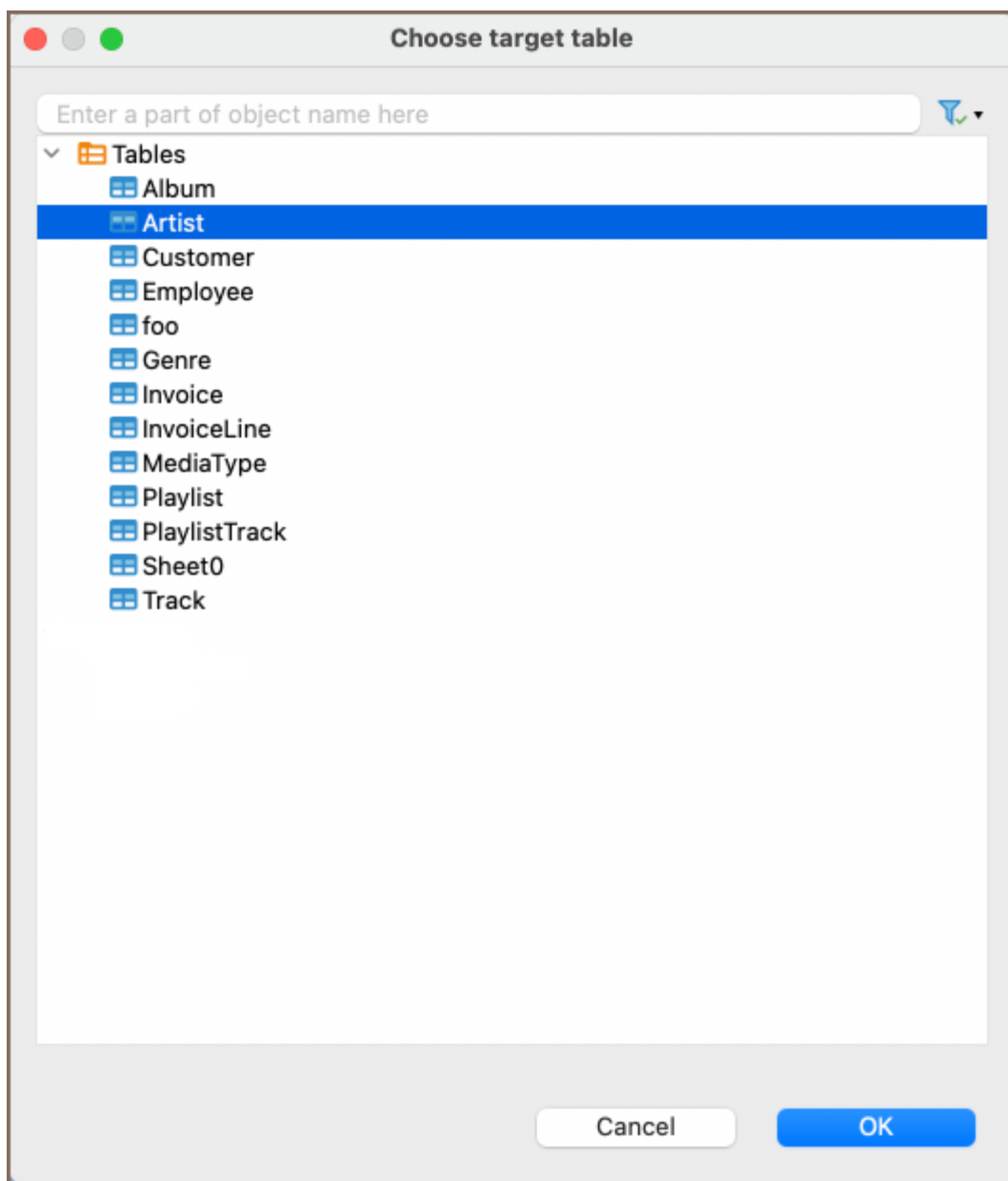
Press the **Choose...** button to select the container where the data will be transferred.

Define a target table



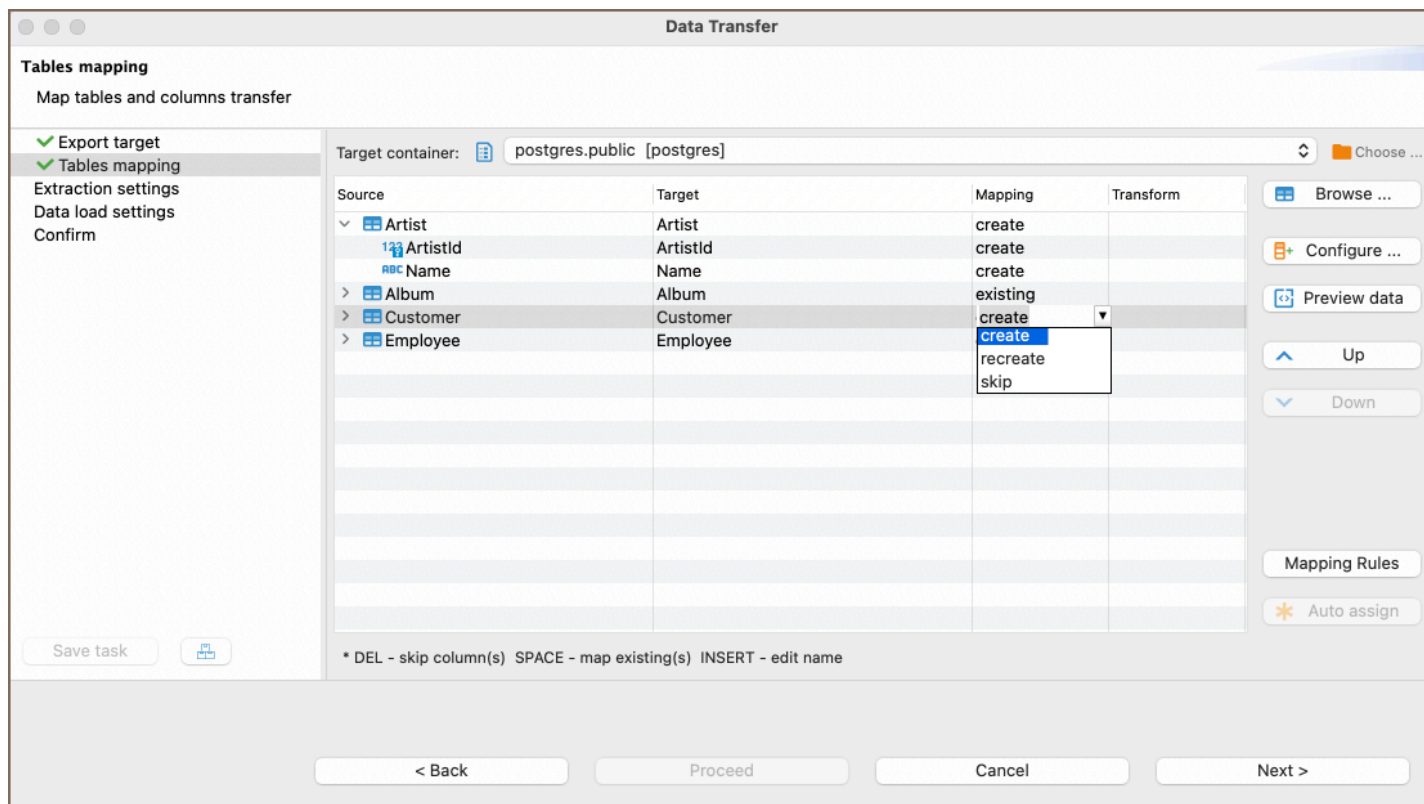
You can specify the target table where the data will be transferred in multiple ways:

- **Manual Entry:** Click on a cell in the **Target** column and manually enter the name of the table where you wish to transfer the data.
- **Drop-down list:** Use the drop-down list next to the **Target** column to choose among the following options:
 - Pre-existing table names.
 - **Skip:** Skips the data transfer for this table.
 - **Browse:** Opens the **Choose Target Table** window
- **Browse button:** Alternatively, press the **Browse** button to select a table from the existing tables in the target container.



Mapping

To change the mapping type, click a cell in the **Mapping** column of **Table mapping dialog box** and select the required mapping type.

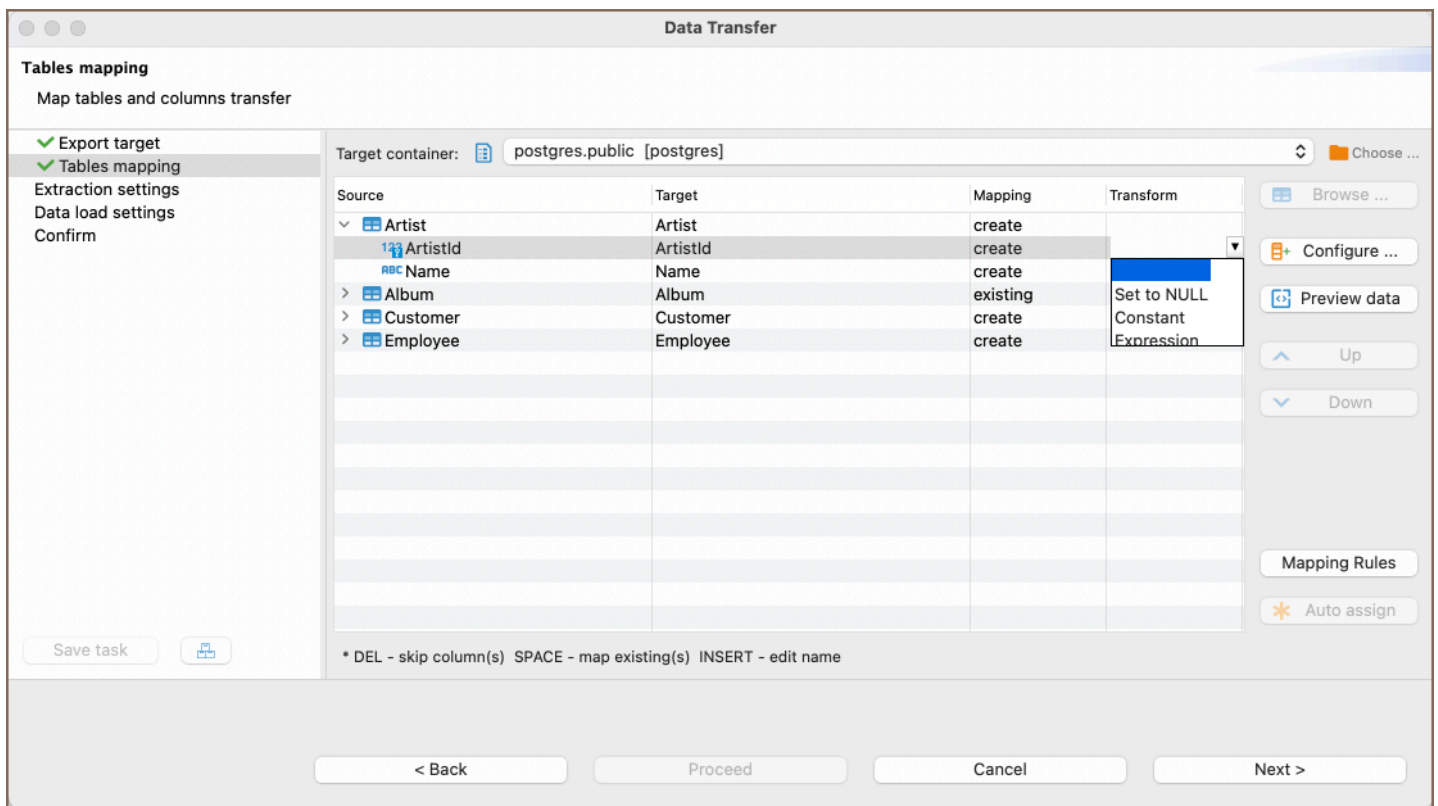


Action	Description
Create	Transfers source data into a newly created table or column in the target container.
Skip	Does not transfer the source data.
Existing	Transfers source data to an existing table in the target container.
Recreate	Recreate the table, which means that the available data, keys, indexes, and other possible entities of the existing table will be lost.

Tip: If the cells are marked with ?, it means that in the target table, there are no columns with matching names, otherwise the names will be filled in automatically.

Transform column values

You may also want to transform the values of some columns during the transfer. To do that, define column transformers by clicking on corresponding cells in the **Transform** column. You can choose one of three options:



Option	Description
Set to NULL	All values in the corresponding column are set to <code>null</code> .
Constant	Sets column value to a constant value.
Expression	Uses <code>JEXL</code> expressions to calculate the column's value.

Configure

Click the **Configure** button to open the **Configure metadata structure** window. This window allows you to delve into additional settings are distributed across the following tabs:

Column mapping

By navigating to the **Column mapping**, you can explore detailed mapping between the source and target columns.

[illegible]

Element	Description
Source Column	Contains names of columns existing in the selected source table.
Source Type	Lists the data types assigned to the columns in the selected source table.
Target Column	Contains names of columns in the target table where the data from the source column will be transferred.
Target Type	Lists the data types that will be assigned to the columns in the target table.
Mapping	Contains the list of actions to be applied to the data on data transfer.
Transform	Displays transformations for the data in a column during the transfer.

Important: Data types that are supported in the source database may not be supported in the target, and vice versa. To set a data type for a target column, click the cell in the **Target Type** column and choose from the dropdown list.

Table properties

By selecting the **Table properties** tab, you can modify properties of the target table, such as:

Property	Description
Tablespace	Specifies the tablespace for the target table.
Partition By	Sets the partitioning for the target table.
Comment	Allows you to add comments to the target table.

Note: The availability of these settings may vary depending on the database you are using.

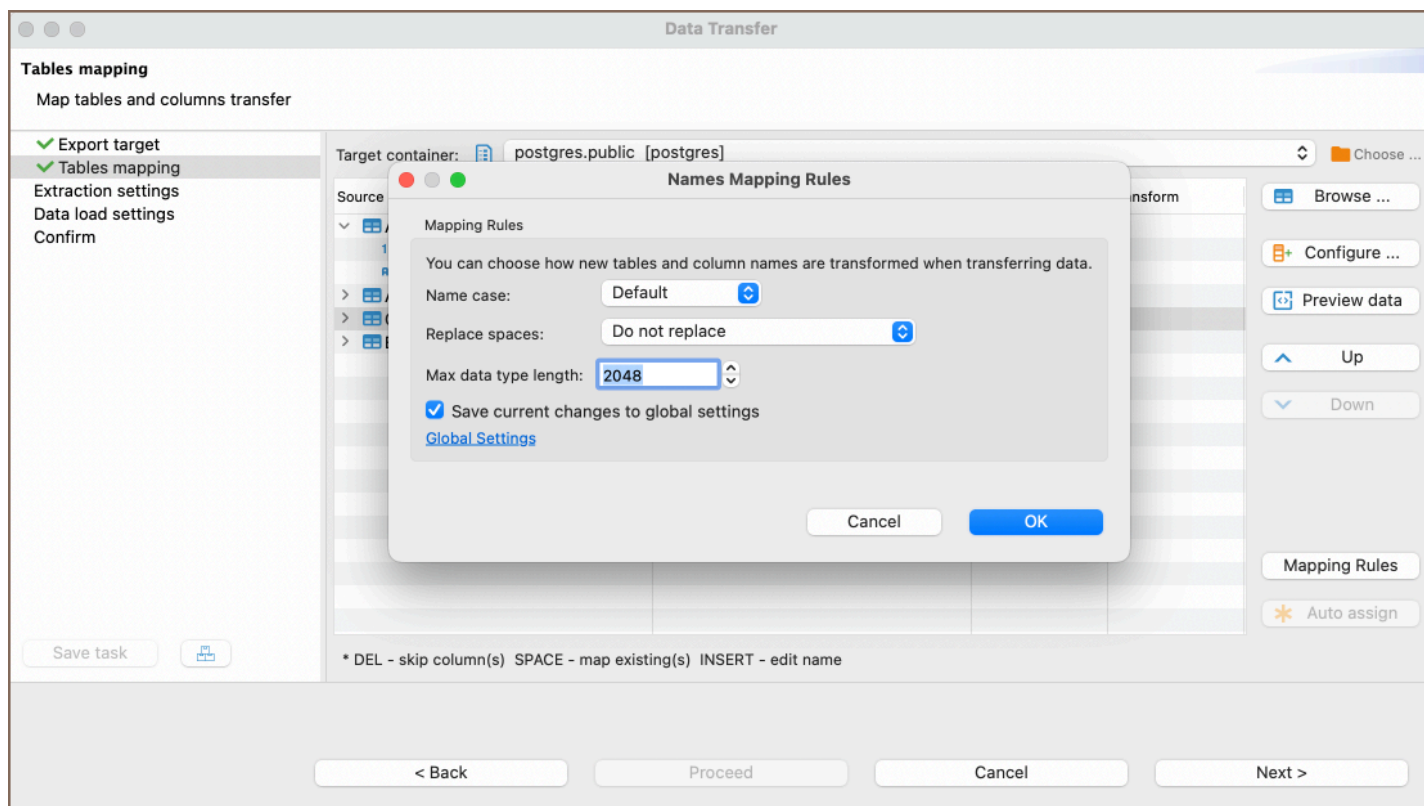
Target DDL

By selecting the **Target DDL** tab, you can view the SQL script that will be executed during the data transfer.

If you're exporting data to a new table or recreating an existing one, the tab will display the necessary SQL statements. If not, the DDL tab will remain empty.

Mapping Rules

Clicking the **Mapping Rules** button opens a window that provides options for customizing how new tables and column names are transformed during the data transfer.



In the window, the following settings are available:

Option	Description
Name case	Sets the letter casing for table and column names. Choices include Default , Upper case , and Lower case .
Replace spaces	Determines how spaces in table and column names are handled. Options are: Do not replace , Replace with underscore , and Remove, convert to CamelCase .
Max data type length	Sets the maximum length for data types.
Save current changes to global settings	If checked, saves the current mapping rules to global settings. Otherwise, they will be saved at the data source settings level.
Global settings	Opens the global settings window for Names Mapping Rules . Alternatively, to open global settings, go to Window -> Preferences -> Connections -> Data Transfer

In addition to using the Global settings button, you can also access these settings by navigating to **Window -> Preferences -> Connections -> Data Transfer**.

Important: After modifying the **Mapping Rules** settings, you will be prompted to confirm your changes. Modifying the **Mapping Rules** may result in the loss of names that were already changed.

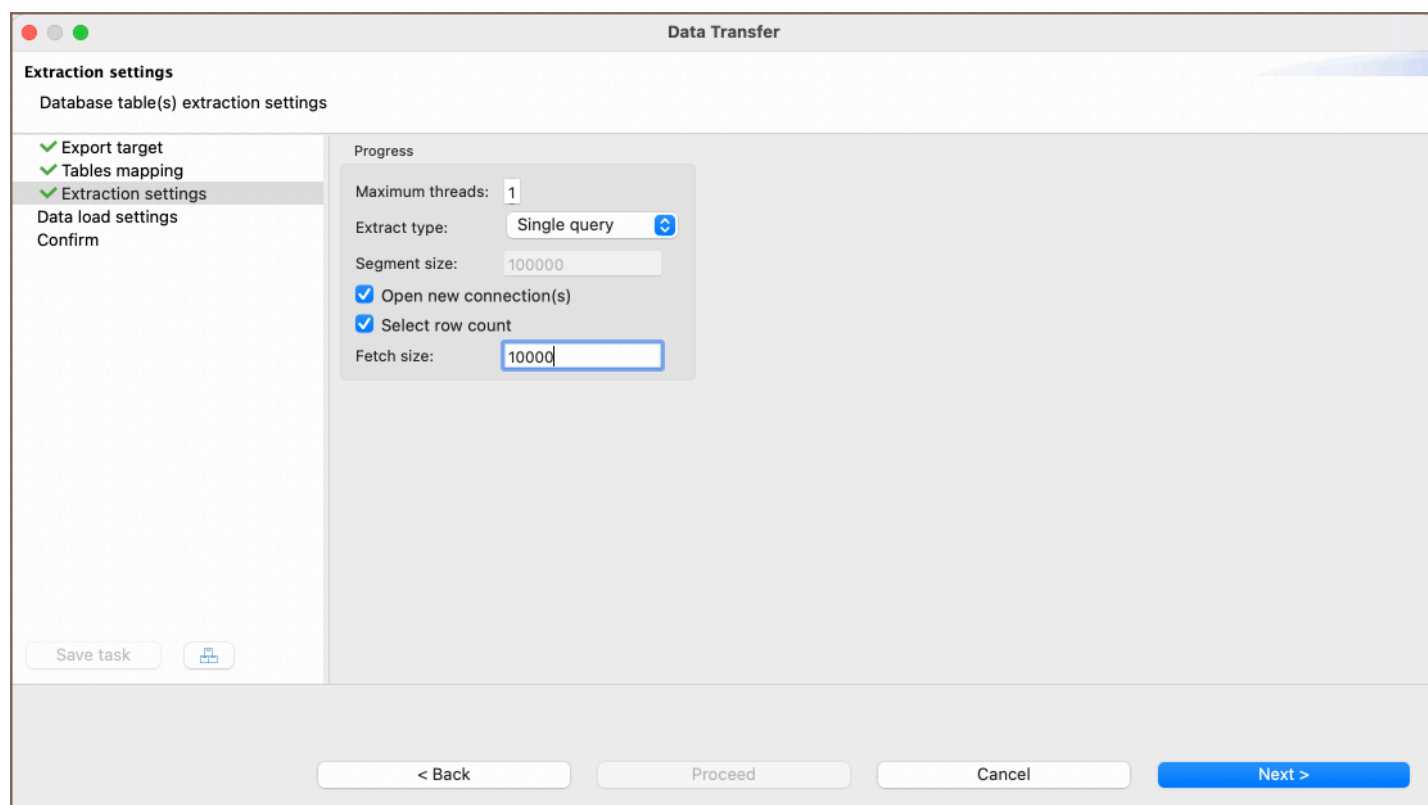
Keyboard Shortcuts

The following keyboard shortcuts for easy navigation within the mapping table area of the **Tables mapping** tab are supported:

Shortcut	Action
Up	Move one row up.
Down	Move one row down.
Right	Expand list of source table columns.
Left	Collapse list of source table columns.
Space	Auto-assigns the target.
Del	Sets mapping type to skip.

Step 4 Extraction settings

After setting up your table mappings, the next step is to define how the data will be extracted from the source. The **Extraction settings** tab offers various options to optimize this process. Configure these settings to suit your specific data transfer requirements and press **Next**.



The screenshot shows the 'Data Transfer' window with the 'Extraction settings' tab selected. The window has a title bar with standard macOS window controls (red, yellow, green buttons). The main content area is divided into a left sidebar and a main panel. The sidebar contains a list of steps: 'Export target', 'Tables mapping', 'Extraction settings' (highlighted with a green checkmark), 'Data load settings', and 'Confirm'. The main panel is titled 'Database table(s) extraction settings' and contains a 'Progress' section with the following settings: 'Maximum threads' set to 1, 'Extract type' set to 'Single query' (with a dropdown arrow), 'Segment size' set to 100000, 'Open new connection(s)' checked, 'Select row count' checked, and 'Fetch size' set to 10000. At the bottom of the window, there are four buttons: '< Back', 'Proceed', 'Cancel', and 'Next >' (highlighted in blue). A 'Save task' button is also visible in the bottom left corner of the main panel.

Option	Description
Maximum threads	Defines the number of threads to be used for data transfer.
Extract type	Use Multiple queries if database doesn't support resultset scrolling. You can also set the Segment Size value when this option is selected.
Open new connection(s)	If checked, a new connection is established, ensuring data transfer does not affect other database operations.
Select row count	Enables a progress bar to display data migration status.
Fetch size	Indicates the number of rows fetched per server round trip, affecting extraction performance.

Step 5 Data load settings

Data Transfer

Data load settings
Configuration of table data load

- ✓ Export target
- ✓ Tables mapping
- ✓ Extraction settings
- ✓ Data load settings
- Confirm

Data load

- ☒ Transfer auto-generated columns
- ☐ Truncate target table(s) before load
- ☐ Disable referential integrity checks during the transfer
- Replace method: [Replace/ignore method documentation](#)

Performance

- ☒ Open new connection(s)
- ☒ Use transactions
- Do Commit after row insert:
- ☐ Use multi-row value insert
- ☐ Skip bind values during insert
- ☐ Disable batches
- ☐ Ignore duplicate rows errors
- ☐ Use bulk load

General

- ☒ Open table editor on finish
- ☒ Show finish message
- ☐ Send results by E-Mail [Configure](#)

Save task

< Back Proceed Cancel Next >

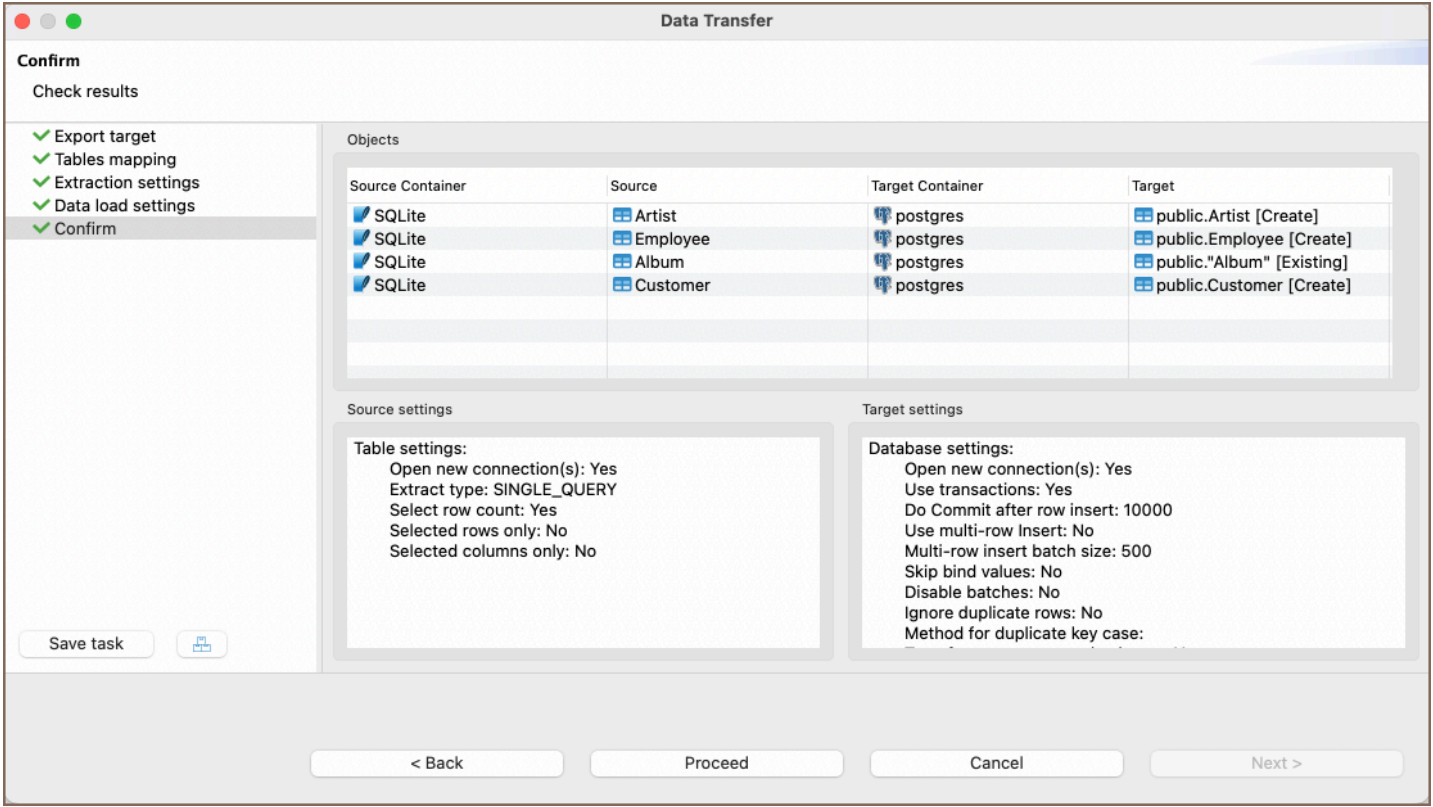
After configuring the extraction settings, you'll need to specify how the data will be loaded into the target database. The **Data load settings** tab provides a range of options to control this part of the process. Adjust the settings according to your needs and then press **Next**.

Data load settings tab defines how the extracted data will be pushed to the target. The following options are available.

Option	Description
Transfer auto-generated columns	Fill in or skip columns marked with the "autogenerated" status. Some databases accept values in such columns, while others will throw a syntax error.
Truncate target table(s) before load	Select this checkbox only if you want all the data to be cleared from the target table. Be very careful with this option!
Disable referential integrity checks during the transfer	Disabling constraints in the target table. This setting prevents database errors by temporarily disabling the constraints. However, please note that not all databases support this functionality.
Replace method	Read our guide on Data Import and Replace to learn more about the replacing method option.
Open new connection(s)	Use this option to speed up data transfer. If selected, a new connection will be opened and the data transfer will not interfere with other calls to the database where data is being transferred to.
Use transactions	This option allows you to speed up the data transfer and to define the number of rows for each transaction by setting the Commit after insert of parameter.
Do Commit after row insert	Performing a commit after a certain number of inserted rows. This setting specifies that a commit operation should be executed after a specified number of rows have been inserted into the table.
Use multi-row value insert	Use multi-row insert with extended values number for higher performance. Database-specific setting.
Skip bind values during insert	This option can drastically increase performance for some drivers like Redshift by skipping a process of binding values and setting them directly, but it opens up a vulnerability to SQL injections. Not recommended if you are not sure of imported file contents.
Disable batches	Select this checkbox if you want to disable the use of batch imports. The import will be made row by row. Enabling this function will show all import errors, but make the import process slower.
Ignore duplicate rows errors	In the import process, if a database encounters a duplicate key from the import row in the target table, such errors are ignored, and the import operation continues without failure.
Use bulk load	Bypasses transaction settings and loads the entire dataset using the native tool provided by the database.
Open table editor on finish	If selected, the table editor is to be opened when data transfer is finished.
Show finish message	If selected, a notification message will be shown when the transfer is finished.

Send results by E-Mail	Sends data transfer results by E-Mail on finish. Read our guide to using email to transfer data to learn more.
------------------------	--

Step 6 Confirm



The final step before initiating the data transfer is the **Confirm** tab. This tab provides a summary of all configurations set up in the previous steps. It serves as a last review to ensure all settings are correct.

Here is a table summarizing the configurations:

Option	Description
Source Container	Displays the source container from which data will be pulled.
Source	Shows the specific source within the container.
Target Container	Displays the target container to which data will be pushed.
Target	Shows the specific target within the container.
Source Settings	Summarizes settings related to data extraction from the source.
Target Settings	Summarizes settings related to data loading into the target.

These settings are final and cannot be changed at this stage. If all configurations are correct, click **Finish** to start the data transfer.

Tip: You can save these configurations for future use and create a task related to the data transfer operation by selecting **Save [task](#)**. This option consolidates your settings into a reusable task.

Upon clicking **Proceed**, the data transfer process will initiate.

Step 7 Export completion notification

If there are no errors, you will see a notification window indicating the successful completion of the export task.

You can continue working with your database during the export process, as it will be performed in the background.

Data transfer via email

Note: This feature is available in [Enterprise](#) and [Ultimate](#) editions only.

Table of contents

[SMTP profile configuration](#)

[Setting up data transfer](#)

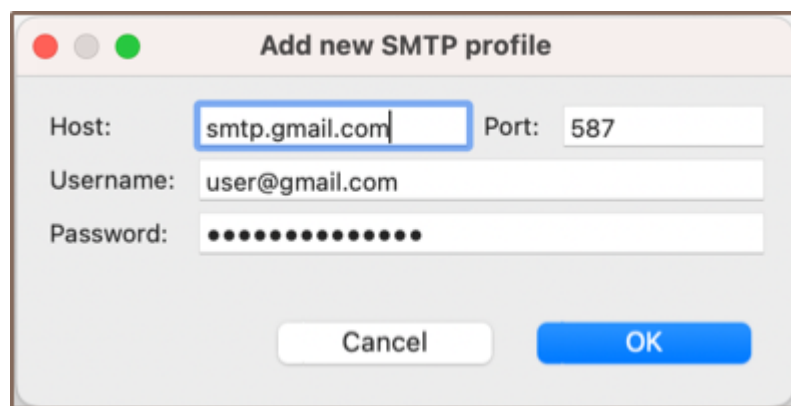
[Authorization troubleshooting](#)

DBeaver offers a way to send data exported via [Data Transfer](#) by email.

SMTP profile configuration

First, you'll need to add an SMTP profile to send the email. Go to `Window ⇒ Preferences ⇒ General ⇒ Mail` and create a new profile.

Parameters `Host` and `Port` may depend on the mail service you use. Use the latter if the service offers SSL and TLS ports. Gmail, for example, uses host `smtp.gmail.com` and port `587`. An example of a configured profile:



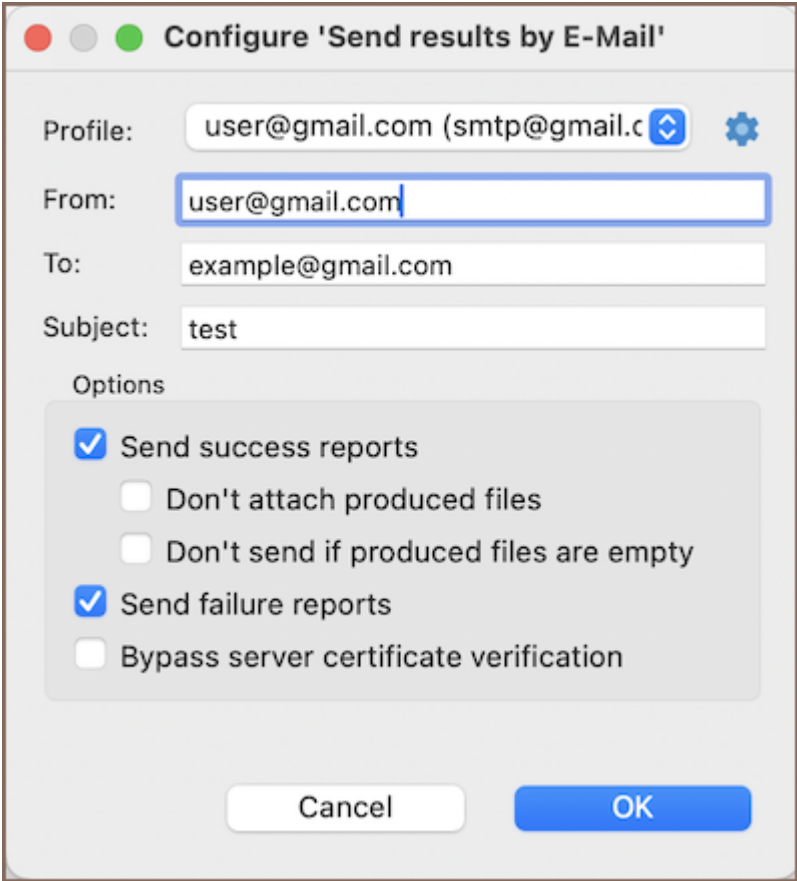
Then you can use the `Test connection` button to verify that the host and credentials are valid.

Please take a look at the [troubleshooting](#) section for more information on resolving common authorization problems.

Setting up data transfer

After configuring at least one [SMTP profile](#), you are able to set up the email notification for data transfer. To do this, follow these steps:

- 1. Create an [export task](#).
- 2. Navigate to the **Output** page.
- 3. Ensure the **Send results by E-mail** option is checked.
- 4. Click the **Configure** button to set up email details.



Within the email configuration, define the following settings:

Setting	Description
Send success reports	Sends an email upon successful completion of a task.
Don't attach produced files	Excludes the output files from the email notification.
Don't send if produced files are empty	Prevents sending an email when the output files have no data.
Send failure reports	Notifies by email if a task encounters a failure.
Bypass server certificate verification	Disables server certificate validation, useful for self-signed certificates.

Once the task runs, the recipients you configure will receive an email. The email will contain the exported file in the selected format if the appropriate options are enabled.

Authorization troubleshooting

You may face various problems while setting up a new SMTP profile.

Several common errors when using Gmail and solutions for them are described below:

- **535-5.7.8 Username and Password not accepted** . Check that the username and password are correct. If you are confident that you have entered valid credentials, then try creating an app password.

Read more at <https://support.google.com/accounts/answer/185833>

- **534-5.7.9 Application-specific password required** . You have two-factor authorization enabled. You will need to generate a special password for DBeaver to use.

Read more at <https://support.google.com/accounts/answer/185833>

There were also several cases when the antivirus would block DBeaver from sending an email.

Data transfer external storage

Note: This feature is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Exporting files to Google Drive](#)

[Exporting automatically](#)

Using [Data Transfer](#), you can export files to external storage, such as Google Drive.

Exporting files to Google Drive

1. Download and install Google Drive for desktop [here](#)
2. Configure Drive for streaming mode according to [this article](#)

After that, in the data transfer wizard, change the output directory to the Drive directory.

Exporting automatically

It can be achieved by using [Task Scheduler](#).

Data compare

Note: This feature is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Steps for comparing data](#)

[Choose tables](#)

[Keys configuration](#)

[Column mapping](#)

[Compare settings](#)

[Execution log](#)

[Viewing the results](#)

[Additional tools](#)

Overview

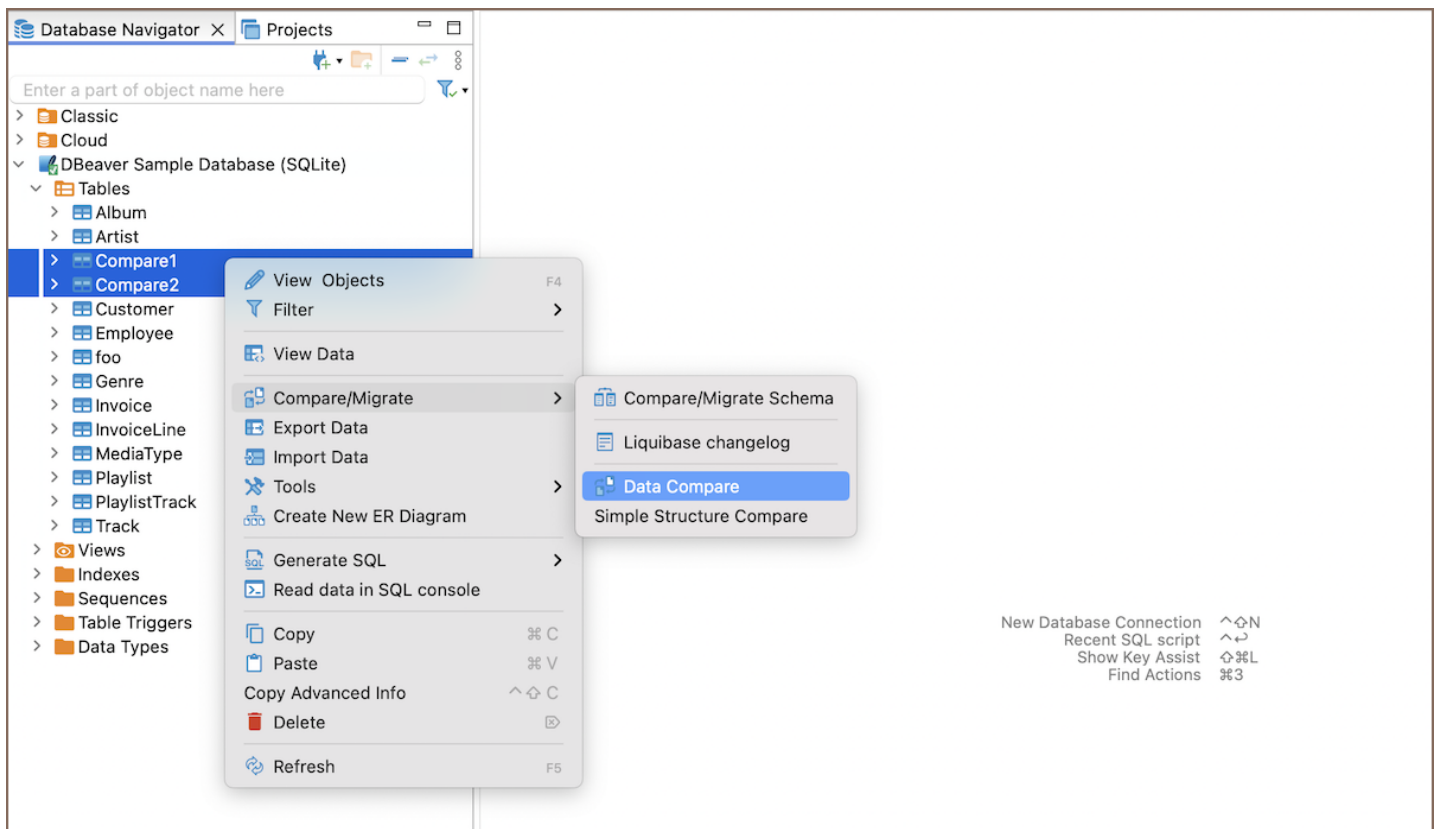
When you work with data, it is common to encounter scenarios where two tables contain almost identical datasets with minor differences. Comparing data across these tables can be crucial for several tasks:

- Visualizing and navigating differences quickly.
- Exporting the differing data.

Steps for comparing data

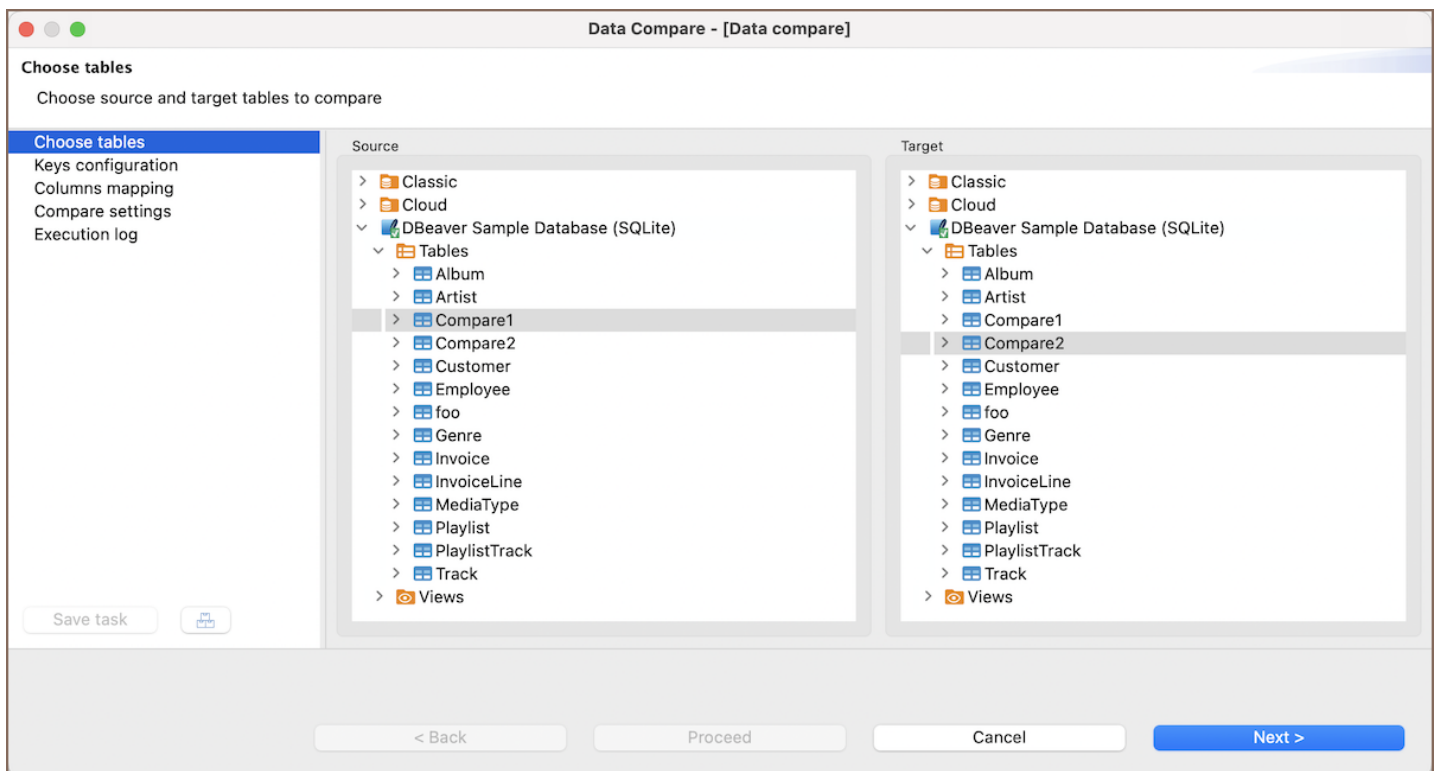
Choose tables

1. First, select the tables you wish to compare from the [Database Navigator](#).



Tip: You can compare tables from the same or different databases, or even from different RDBMS, such as PostgreSQL and MySQL.

2. Navigate to **Database -> Compare/Migrate -> Data Compare** in the menu. This will open the **Data Compare** wizard. At this stage, confirm your choice of source and target tables for the comparison.



Keys configuration

Select the columns that will serve as the **unique key** for the comparison. Ensure that the number of columns selected is equal. Choosing the wrong **keys** may result in invalid comparison results.

Keys configuration
Choose unique columns to use as a key during comparison

✓ Choose tables
Keys configuration
Columns mapping
Compare settings
Execution log

Source

Key: COMPARE1_PK

Columns:

Column	#	Type
<input checked="" type="checkbox"/> 123 InvoiceId	1	INTEGER
<input type="checkbox"/> 123 CustomerId		INTEGER
<input type="checkbox"/> ABC InvoiceDate		DATETIME
<input type="checkbox"/> ABC BillingAddress		NVARCHAR
<input type="checkbox"/> ABC BillingCity		NVARCHAR
<input type="checkbox"/> ABC BillingState		NVARCHAR
<input type="checkbox"/> ABC BillingCountry		NVARCHAR
<input type="checkbox"/> ABC BillingPostalCode		NVARCHAR
<input type="checkbox"/> 123 Total		NUMERIC

Clear All

Target

Key: COMPARE2_PK

Columns:

Column	#	Type
<input checked="" type="checkbox"/> 123 InvoiceId	1	INTEGER
<input type="checkbox"/> 123 CustomerId		INTEGER
<input type="checkbox"/> ABC InvoiceDate		DATETIME
<input type="checkbox"/> ABC BillingAddress		NVARCHAR
<input type="checkbox"/> ABC BillingCity		NVARCHAR
<input type="checkbox"/> ABC BillingState		NVARCHAR
<input type="checkbox"/> ABC BillingCountry		NVARCHAR
<input type="checkbox"/> ABC BillingPostalCode		NVARCHAR
<input type="checkbox"/> 123 Total		NUMERIC

Clear All

Rows will be sorted and compared according to the selected key columns. Keys must be unique.

< Back Proceed Cancel Next >

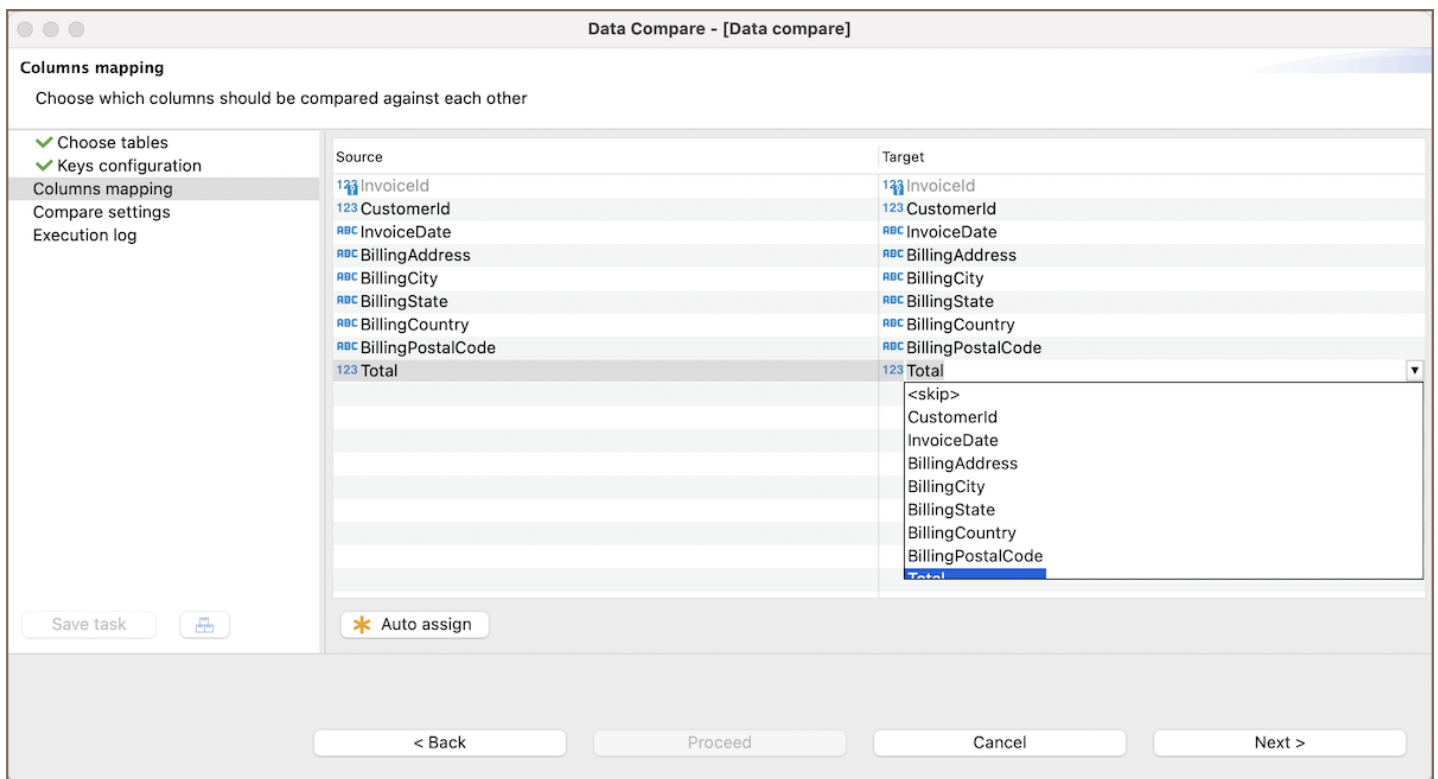
Note: By default, if the table has a unique key, it will be selected automatically during the initial setup.

Without unique keys, rows cannot be properly compared as there is no way to distinguish between individual rows.

Column mapping

If the tables you are comparing have different ordering, naming, or quantities of columns, you can adjust their mappings or skip columns that are not of interest.

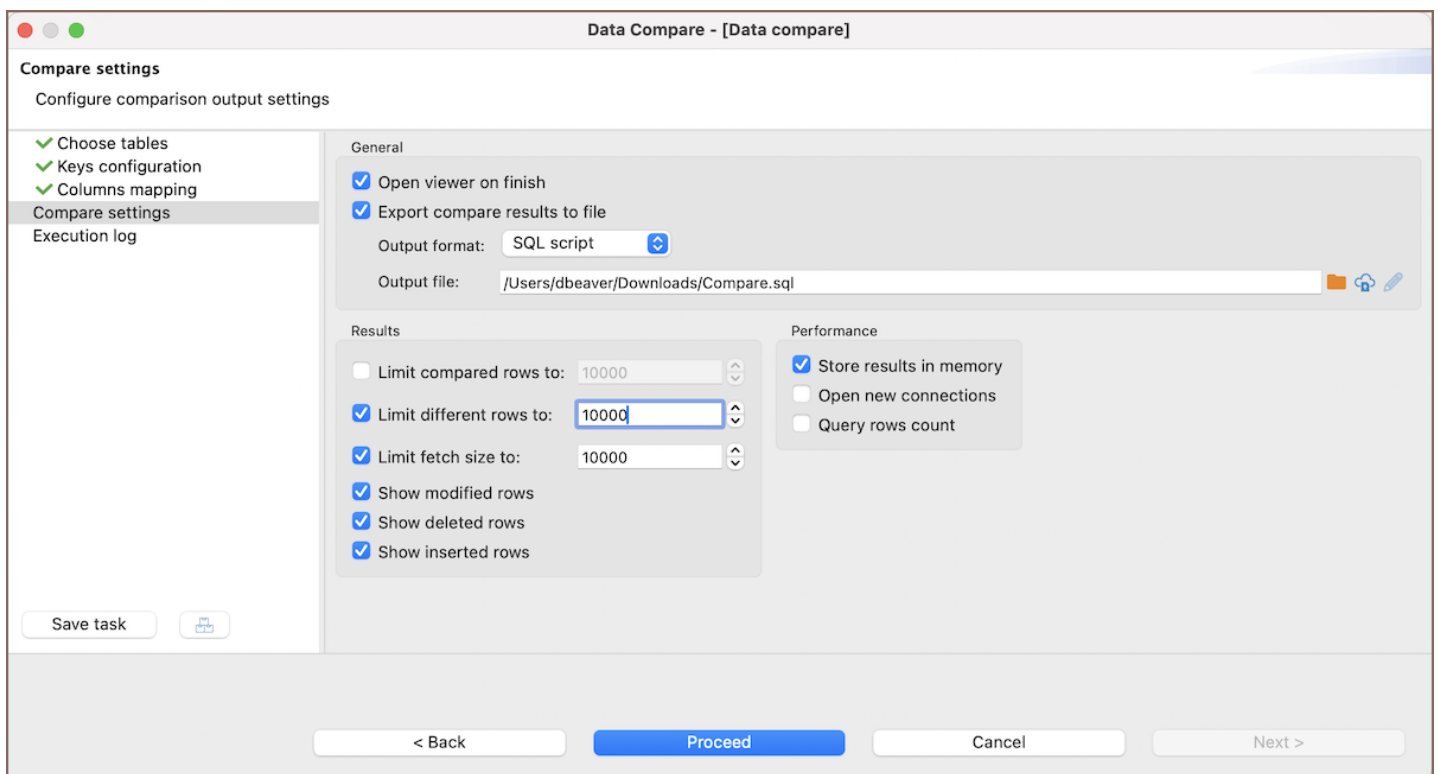
You can use the **Auto assign** button to automatically link columns with the same names.




Note: This page will only feature columns that are not used as the primary key configured on the previous page.

Compare settings

On the final page of the **Data Compare** wizard, configure settings to tailor the data comparison to your needs:



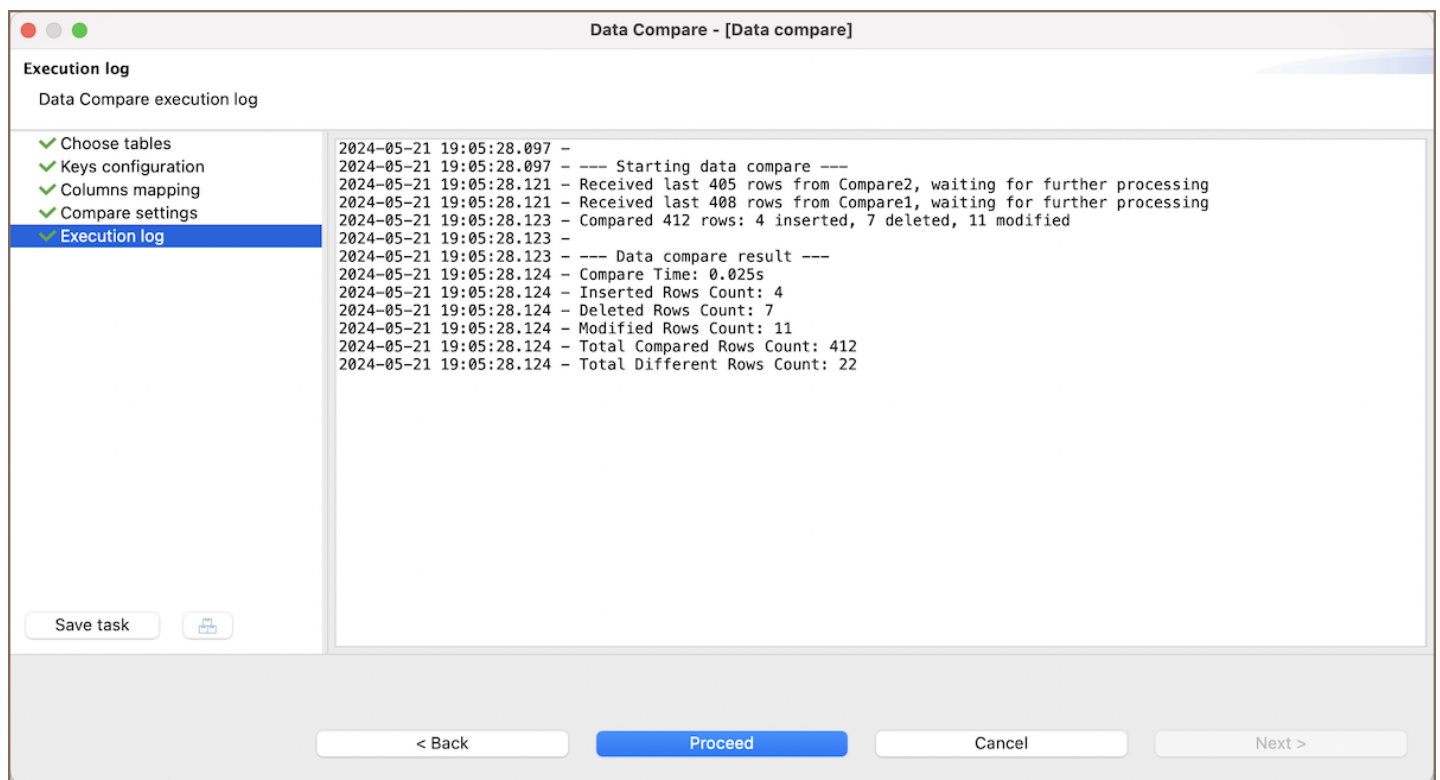
Setting	Description
Open viewer on finish	Opens the compare viewer for a visual overview of the differences once the comparison is complete.
Export compare results to file	Exports the SQL script diff to a file. You can choose the output format and specify the output file path.
Limit compared rows to	Limits how many rows must be compared against each other before ending the comparison.
Limit different rows to	Limits how many rows must be differ between each other before ending the comparison.
Limit fetch size to	Determines how many rows should be fetched at once, which helps manage memory and performance during comparison.
Show modified rows	Determines whether modified rows should be included in the diff or not, allowing you to track changes in the data.
Show deleted rows	Determines whether deleted rows should be included in the diff or not, providing insight into data that has been removed from the database.
Show inserted rows	Determines whether newly inserted rows should be included in the diff or not, allowing you to see additions to the database.
Store results in memory	Stores results in memory for further processing. Note that DBeaver might run out of memory when comparing large tables if this option is enabled. When disabled, results will be stored in a temporary database.
Open new connections	Opens new physical connections for data reading, which can help manage resources and ensure the stability of data comparison operations.
Query rows count	Queries row count before performing compare. This lets you track the progress of the comparison but may cause performance faults in some cases.
Save task	Opens the Save Task window to assist in creating a task during the compare process. This window provides options and settings for creating and configuring a task related to the compare operation.

Tip: You can export file to a remote file system via [Cloud Storage](#) using the **Browser remote file system** button .

Execution log

Once you have configured all settings and clicked **Proceed**, the Data Compare wizard will perform the comparison operation:

- An **Execution Log** window will display the progress and outcome of the comparison process.
- Simultaneously, a separate window will open showing the comparison results. This window provides a detailed view of differences between the datasets based on the specified settings.



Viewing the results




After the comparison process is complete, you can view the results in the viewer. It highlights the differences between the datasets in detail, with each row that shows a discrepancy being clearly marked.


Data Compare ('Compare1' - 'Compare2') ×											
Red rows are absent from the target table; green rows are absent from the source table; values marked with yellow have different values in both tables						Export Changes... Swap Tables Show Summary					
Source - Compare1 (DBever Sample Database (SQLite))						Target - Compare2 (DBever Sample Database (SQLite))					
123 InvoiceId	123 CustomerId	123 InvoiceDate	ABC BillingAddress	ABC BillingCity		123 InvoiceId	123 CustomerId	123 InvoiceDate	ABC BillingAddress	ABC BillingCity	
1	39	27	2007-06-10 00:00:00	1033 N Park Ave	Tucson	1	39	27	2007-06-10 00:00:00	1033 N Park Ave	Tucson
2	40	36	2007-06-15 00:00:00	Tauentzienstraße 8	Berlin	2	49	30	2007-08-06 00:00:00	230 Elgin Street	Ottawa
3	96	45	2008-02-18 00:00:00	Erzsébet krt. 58.	Budapest	3	96	45	2008-02-18 00:00:00	Erzsébet krt. 58.	Budapest
4	99	3	2008-03-10 00:00:00	1498 rue Bélanger	Montréal	4	106	41	2008-04-10 00:00:00	11, Place Bellecour	Lyon
5	106	41	2008-04-10 00:00:00	11, Place Bellecour	Lyon	5	110	3	2008-04-20 00:00:00	1498 rue Bélanger	Montréal
6	110	3	2008-04-20 00:00:00	1498 rue Bélanger	Montréal	6	146	30	2008-09-30 00:00:00	230 Elgin Street	Ottawa
7	153	9	2008-10-31 00:00:00	Sander Boulevard 51	Copenhagen	7	153	9	2008-10-31 00:00:00	Sander Boulevard 51	Copenhagen
8	158	24	2008-11-18 00:00:00	162 E Superior Street	Chicago	8	173	50	2009-01-24 00:00:00	C/ San Bernardo 85	Madrid
9	173	50	2009-01-24 00:00:00	C/ San Bernardo 85	Madrid	9	180	29	2009-02-24 00:00:00	796 Dundas Street West	Toronto
10	180	29	2009-02-24 00:00:00	796 Dundas Street West	Toronto	10	210	19	2009-07-19 00:00:00	1 Infinite Loop	Cupertino
11	210	19	2009-07-19 00:00:00	1 Infinite Loop	Cupertino	11	240	57	2009-11-21 00:00:00	Calle Lira, 198	Santiago
12	217	57	2009-09-19 00:00:00	Calle Lira, 198	Santiago	12	243	17	2009-11-30 00:00:00	1 Microsoft Way	Redmond
13	240	57	2009-11-21 00:00:00	Calle Lira, 198	Santiago	13	250	55	2009-12-31 00:00:00	421 Bourke Street	Sidney
14	250	55	2009-12-31 00:00:00	421 Bourke Street	Sidney	14	267	30	2010-03-24 00:00:00	230 Elgin Street	Ottawa
15	267	30	2010-03-24 00:00:00	230 Elgin Street	Ottawa	15	307	19	2010-09-13 00:00:00	1 Infinite Loop	Cupertino
16	276	15	2010-04-26 00:00:00	700 W Pender Street	Vancouver	16	320	22	2010-11-06 00:00:00	120 S Orange Ave	Orlando
17	307	19	2010-09-13 00:00:00	1 Infinite Loop	Cupertino	17	324	41	2010-11-28 00:00:00	11, Place Bellecour	Lyon
18	320	22	2010-11-06 00:00:00	120 S Orange Ave	Orlando	18	340	9	2011-02-02 00:00:00	Sander Boulevard 51	Copenhagen
19	324	41	2010-11-28 00:00:00	11, Place Bellecour	Lyon	19	374	16	2011-07-04 00:00:00	1600 Amphitheatre Parkv	Mountain View
20	335	53	2011-01-15 00:00:00	113 Lupus St	London	20	381	54	2011-08-04 00:00:00	110 Raeburn Pl	Edinburgh
21	374	16	2011-07-04 00:00:00	1600 Amphitheatre Parkv	Mountain View	21	390	48	2011-09-12 00:00:00	Lijnbaansgracht 120bg	Amsterdam
22	381	54	2011-08-04 00:00:00	110 Raeburn Pl	Edinburgh						
23	390	48	2011-09-12 00:00:00	Lijnbaansgracht 120bg	Amsterdam						
24	394	8	2011-10-04 00:00:00	Grétrystraat 63	Brussels						

Note: A complete visual preview of the results is available only when **Data Compare** is launched through the wizard. If launched as part of a task, the visual editor will not open. Instead, only statistics will be recorded in the [execution log](#).

Additional tools

The viewer not only displays the differences but also provides tools to manage and interpret these differences effectively. Here's what you can do within the compare viewer:

Feature	Button image	Description
Export changes		<p>The results of the Data Compare can be exported as an SQL script, which includes DELETE , INSERT , and UPDATE statements to make the data in the second table match the first.</p> <p>Alternatively, you can export the results in an HTML format as a table for easy viewing and sharing.</p>
Swap tables		The Swap tables button allows you to switch the positions of the source and target data panels within the viewer.
Show summary		By clicking the Show Summary button, you can access a statistical summary and analysis of the differences found between the datasets. This summary provides a quick overview of the total changes, including the number of modified, deleted, and inserted rows.

Tip: You can also export the comparison results table through [Export data](#) button  Export data ▼ in the bottom toolbar or [copy](#) rows or individual values that differ.

Backup and restore

Note: This feature is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Database specific settings](#)

[MySQL](#)

[Backup MySQL database](#)

[Restore MySQL database](#)

[PostgreSQL](#)

[Backup PostgreSQL database](#)

[Global PostgreSQL database backup](#)

[Restore PostgreSQL database](#)

[Additional features](#)

[Execute Script Command](#)

[Extra command arguments](#)

[Scheduling dump and restore operations as tasks](#)

Overview

This article provides an in-depth guide to the Backup and Restore features available in DBeaver. DBeaver supports a wide range of databases for backup and restore operations:

- **Classic databases:**

- PostgreSQL

- MySQL

- MariaDB

- TiDB

- SingleStore
- EnterpriseDB
- Greenplum
- YugabyteDB
- CrateDB
- Materialize
- TimescaleDB
- Yellowbrick
- **Cloud databases:**
 - AlloyDB
 - Google Cloud SQL for PostgreSQL
 - Google Cloud SQL for MySQL

The native backup and restore differs from the standard DBeaver [Data Transfer](#) feature by focusing on creating and applying database backups using specialized utilities. These tools are tailored for high-performance interactions with databases, potentially accelerating the backup and restore processes compared to general data transfer methods.

These functions can be accessed from the Context Menu's **Tools** or the Main Menu's **Database -> Tools**.

Database specific settings

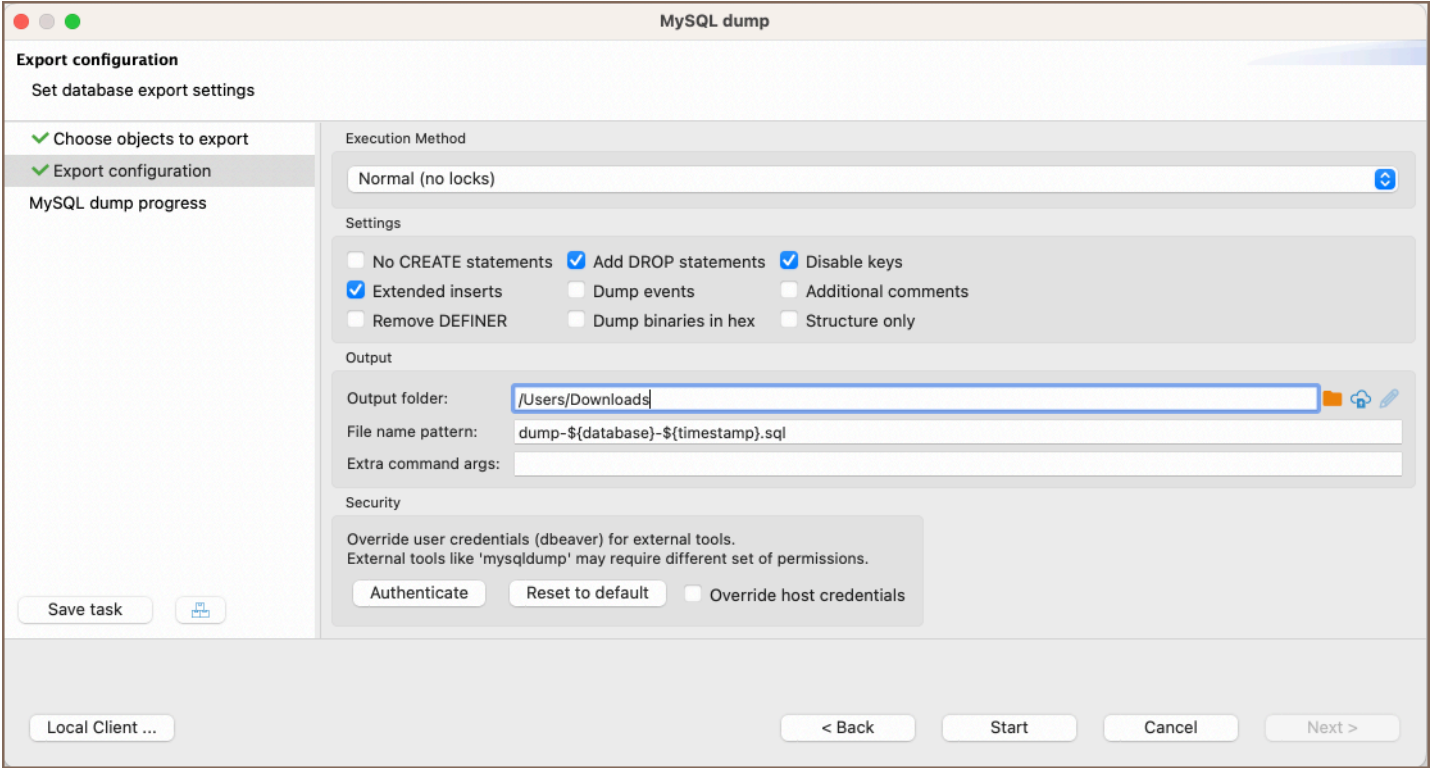
Different database systems possess distinct settings for backup and restore processes. This section presents the settings specific to MySQL and PostgreSQL.

MySQL


Backup MySQL database

To initiate a backup:

- 1. Select the desired database.
- 2. Right-click on the database and choose **Tools -> Dump database**.
- 3. The **Dump** window will appear. Select the necessary objects and click **Next**.
- 4. The **Export configuration** tab will be displayed. Here you can find the following checkboxes and fields:



Option /Field	Description
Execution Method	Method for execution. The default is Normal (no locks) .
Execution Method options:	Normal (no locks) : This is the default method. It performs the dump without acquiring any locks on the tables. Suitable for databases where minimal disruption is required during the dump process.
	Online backup in single transaction : This method ensures the consistency of the database by performing the dump within a single transaction. It's ideal for databases that support transactional storage engines like InnoDB. The backup occurs without interrupting other database operations but requires a higher isolation level.
	Lock all tables : As the name suggests, this method locks all tables in the database during the dump process. It

	ensures complete consistency of the dump but may interrupt other database operations. Suitable for databases or situations where you want a consistent snapshot and are okay with potential disruptions.
No CREATE statements	Do not include CREATE statements in the dump.
Add DROP statements	Include DROP statements before CREATE statements.
Disable keys	Disable keys during the dump process.
Extended inserts	Use extended INSERT statements in the dump.
Dump events	Include events in the dump.
Additional comments	Include comments in the dump.
Remove DEFINER	Remove DEFINER clause from the dump.
Dump binaries in hex	Display binary data as HEX in the dump.
Structure only	Only dump table structures, not data.
Output folder	Specify the directory where the dump file will be saved. For Ultimate Edition, Team Edition, and CloudBeaver versions, users have the option to save to a remote file system using the Browser remote file system button  .
Extra command args	Provide additional command-line arguments. See details in the Extra command arguments section.
Authenticate	Click to open the Authentication window where you can fill in the Username/Password for override.
Reset to default	Reset credentials to their default values.
Override host	Use different host credentials, if necessary.

credentials	
Local Client	Open a window to specify the path to the Local client. More about setting it up.

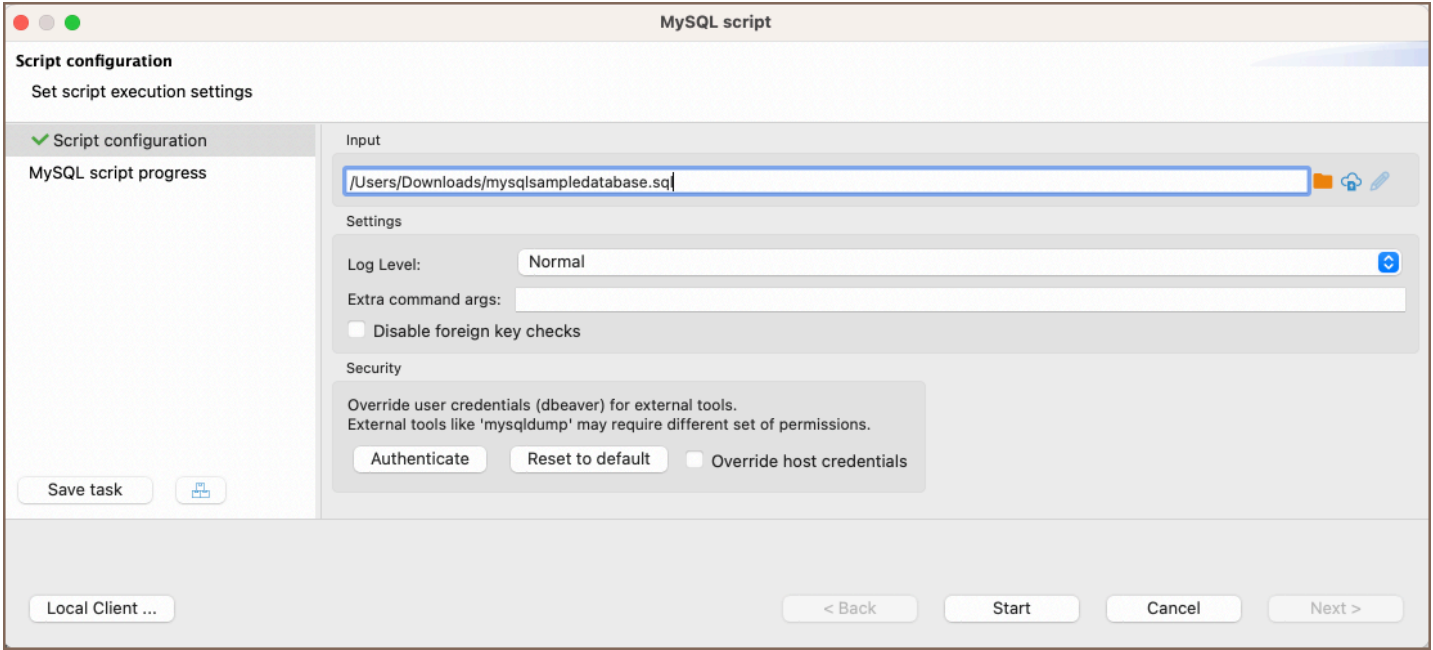
Note: The particular set of configuration options depends on the database type.

- After configuring the settings, click **Start**.
- Upon successful completion, a notification will appear with information about the process.
- The backup file can be found in the folder specified during the **Export configuration** step.

Restore MySQL database

To initiate a restore:


- Select the desired database.
- Right-click on the database and choose **Tools -> Restore database**.
- The **Restore** window will appear. In this window you have to provide the path to the SQL file you wish to restore from in the designated field. Example path: `/path/to/file.sql`



Below are the settings for the restore process:

Setting Name	Description
--------------	-------------

Log Level	Choose the level of logging for the restore process.
Log Level options:	Normal: Provides standard logging information. It captures the essential details of the restore process, ensuring that the logs remain concise and focused on primary events.
	Verbose: Gives more detailed logging information than the Normal level. It includes additional context and information, making it suitable for situations where you want to understand the process more in-depth without getting into debugging details.
	Debug: Offers the most detailed logging information. It captures all events, including low-level operations and potential issues. This level is ideal for troubleshooting problems or understanding the inner workings of the restore process.
Extra command args	Provide additional command-line arguments. See details in the Extra command arguments section.
Disable foreign key checks	Disable foreign key checks during the restore process.
Authenticate	Click to open the Authentication window where you can fill in the Username/Password for override.
Reset to default	Reset credentials to their default values.
Override host credentials	Use different host credentials, if necessary.
Local Client	Specify the path to the Local client.

Tip: In the input field for specifying the path to the SQL file, you can choose a file from a local directory or use the **Browser remote file system** button  to select a file from a remote file system. This functionality is accessible > to users of the Ultimate Edition, Team Edition, and CloudBeaver.

- After configuring the settings, click **Start**.
- Upon successful completion, a notification will appear with information about the process.

PostgreSQL


Backup PostgreSQL database

To initiate a backup:

1. Select the desired database.
2. Right-click on the database and choose **Tools** -> **Backup**.
3. The **Dump** window will appear. Select the necessary objects and click **Next**.
4. The **Backup settings** tab will be displayed. Here you can find the following checkboxes and fields:

The screenshot shows the 'PostgreSQL dump' window. On the left, there's a sidebar with 'Backup settings' selected. The main area is divided into 'Settings' and 'Output' sections. The 'Settings' section includes fields for 'Format' (Directory), 'Compression' (empty), and 'Encoding' (Big5). Below these are several unchecked checkboxes: 'Use SQL INSERT instead of COPY for rows', 'Do not backup privileges (GRANT/REVOKE)', 'Discard objects owner', 'Add drop database statement', and 'Add create database statement'. The 'Output' section has fields for 'Output folder' (/Users/Downloads), 'File name pattern' (dump-\${database}-\${timestamp}.sql), and 'Extra command args' (empty). At the bottom, there's a 'Security' section with a text box for overriding user credentials and two buttons: 'Authentication' and 'Reset to default'. At the very bottom of the window, there are navigation buttons: '< Back', 'Start', 'Cancel', and 'Next >'.

Setting	Description
Format	Choose the format for the dump. Options include Directory, Tar, Custom, and Plain.
Format options:	Directory: Splits the output into one file per table. Suitable for larger databases as it facilitates parallel restoration.
	Tar: Produces archive in tar format. Useful for backups since it can be read by standard tools.
	Custom: A flexible format that allows selective restore and other operations.
	Plain: Produces plain-text SQL script file. It can be used with psql for restoring.

Compression	Set the compression level for the dump, ranging from 0 (no compression) to 9 (maximum compression).
Encoding	Select the character encoding for the dump. The available encodings depend on database configuration and locale.
Use SQL INSERT instead of COPY for rows	Use the <code>INSERT</code> command instead of the <code>COPY</code> command for row data.
Do not backup privileges (GRANT /REVOKE)	Exclude privilege commands (<code>GRANT</code> / <code>REVOKE</code>) from the dump.
Discard objects owner	Exclude the ownership information from the dump.
Add drop database statement	Include a statement in the dump to drop the database before restoring.
Add create database statement	Include a statement in the dump to create the database when restoring.
Output folder	Specify the directory where the dump file will be saved. For Ultimate Edition, Team Edition, and CloudBeaver versions, users have the option to save to a remote file system using the Browser remote file system button  .
File name pattern	Define the naming pattern for the dump file, with variables like <code>\${database}</code> and <code>\${timestamp}</code> to customize the filename.
Extra command args	Provide additional command-line arguments. See details in the Extra command arguments section.
Authentication	Click to open the Authentication window where you can fill in the Username/Password for override.
Reset to default	Reset credentials to their default values.
Override host credentials	Use different host credentials, if necessary.
Local Client	Specify the path to the Local client.

Note: The particular set of configuration options depends on the database type.

5. After configuring the settings, click **Start**.
6. Upon successful completion, a notification will appear with information about the process.
7. The backup file can be found in the folder specified during the **Backup settings** step.

Global PostgreSQL database backup

When performing a Global PostgreSQL database Backup, the entire database is dumped, including roles and tablespaces. This differs from standard backup procedures where only specific schemas and their contents can be selected. Additionally, multiple databases can be chosen for backup at once in the global method.

To initiate a global backup:

1. Select the desired database.
2. Right-click on the database and choose **Tools -> Global Backup**.
3. The **Global Dump** window will appear. Select the necessary objects and click **Next**.
4. The **Backup settings** tab will be displayed. Here you can find the following checkboxes and fields:

PostgreSQL global dump

Global backup settings

Global backup settings

- ✓ Choose objects to export
- ✓ Global backup settings
- PostgreSQL global dump progress

Settings

Encoding:

- ☐ Dump only the object definitions, not data
- ☐ Dump only global objects, no databases
- ☐ Dump only roles
- ☐ Dump only tablespaces
- ☐ Do not backup privileges (GRANT/REVOKE)
- ☐ Discard objects owner
- ☐ Dump passwords for roles

Output

Output folder:


File name pattern:

Extra command args:

Security

Override user credentials ("postgres") for objects "".

External tools like 'psql' and 'pg_dump' may require different set of permissions.

Setting	Description			
Encoding	Set the character encoding for the backup.			
Dump only the object definitions, not data	Includes only the structure of the database objects, excluding the data.			
Dump only global objects, no databases	Includes only global objects and excludes individual databases.			
Dump only roles	Includes only user roles.			
Dump only tablespaces	Captures only tablespaces without the databases.			
Do not backup privileges (GRANT /REVOKE)	Excludes privilege statements like <code>GRANT</code> and <code>REVOKE</code> .			
Discard objects owner	Excludes the ownership information of the database objects.			
Dump passwords for roles	Includes passwords associated with the user roles.			
Output folder	Specify the directory where the dump file will be saved. For Ultimate Edition, Team Edition, and CloudBeaver versions, users have the option to save to a remote file system using the Browser remote file system button  .		File name pattern	Define the naming pattern for the dump file, with variables like <code>\${database}</code> and <code>\${timestamp}</code> to customize the filename.
Extra	Provide additional command-line arguments. See details in the			

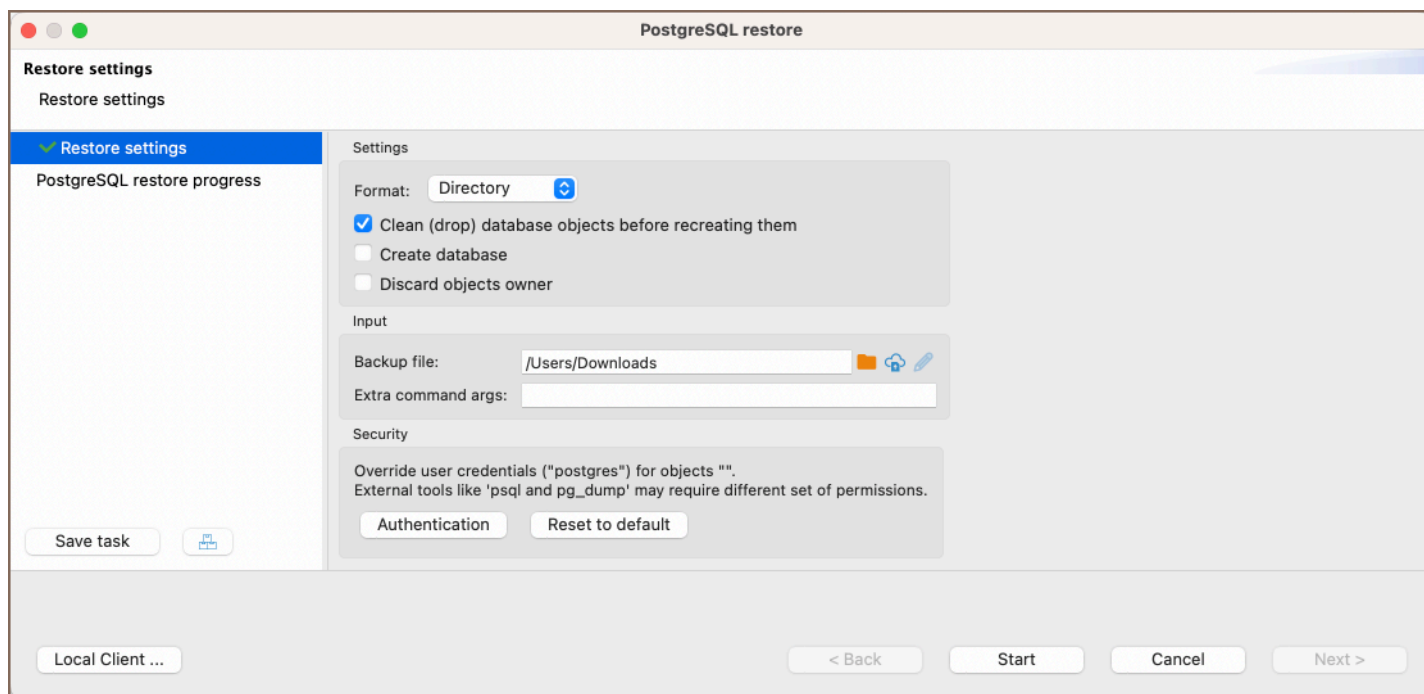
command args	Extra command arguments section.
Authenticate	Click to open the Authentication window where you can fill in the Username/Password for override.
Reset to default	Reset credentials to their default values.
Override host credentials	Use different host credentials, if necessary.
Local Client	Specify the path to the Local client.


5. After configuring the settings, click **Start**.
6. Upon successful completion, a notification will appear with information about the process.
7. The backup file can be found in the folder specified during the **Global backup settings** step.

Restore PostgreSQL database

To initiate a restore:

1. Select the desired database.
2. Right-click on the database and choose **Tools -> Restore**.



Setting	Description
Format	Choose the format for the restore operation. Options include Directory , Tar , Custom , and Plain .
Format options:	<p>Directory: Assumes the input consists of one file per table. Suitable for larger databases as it supports parallel restore.</p> <p>Tar: Assumes the input is an archive in tar format. This is useful when restoring backups made in tar format.</p> <p>Custom: A flexible input format that allows selective restore and other operations.</p> <p>Plain: Assumes the input is a plain-text SQL script file. Suitable for scripts that can be executed with psql.</p>
Clean (drop) database objects before recreating them	Drop database objects before recreating them from the backup.
Create database	Create a new database from the backup.
Discard objects owner	Exclude ownership information during the restore.
Backup file	Specify the path to the backup file that you wish to restore. For Ultimate Edition, Team Edition, and CloudBeaver versions, use the Browser remote file system button  to select a backup file from a remote file system via Cloud Storage .

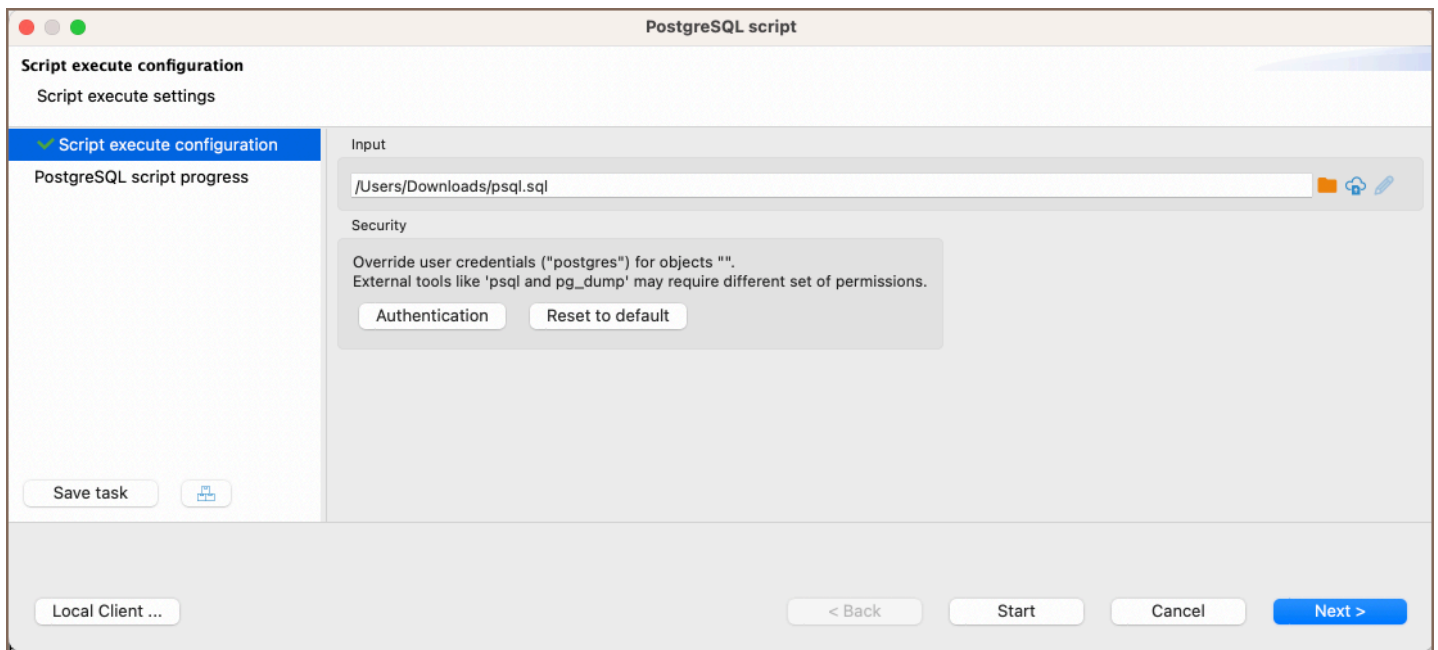
Extra command args	Provide additional command-line arguments. See details in the Extra command arguments section.
Authentication	Click to open the Authentication window where you can fill in the Username/Password for override.
Reset to default	Reset credentials to their default values.
Local Client	Specify the path to the Local client.

3. After configuring the settings, click **Start**.
4. Upon successful completion, a notification will appear with information about the process.

Additional features


Execute Script Command

In DBeaver, besides the standard backup and restore operations, there's an additional capability to run scripts directly through the **Execute script** command. This functionality uses native database clients.



Steps to use the **Execute script** command:

1. Right-click on the desired database in the navigator tree.
2. Navigate to **Tools -> Execute script**.
3. A window will appear prompting you to provide the path to your script.

Tip: In the input field for specifying the path to the SQL file, you can choose a file from a local directory or use the **Browser remote file system** button  to select a file from a remote file system. This functionality is accessible > to users of the Ultimate Edition, Team Edition, and CloudBeaver.

The **Execute script** command provides an alternative way to perform database operations that may not be covered by the traditional restore functionality. This includes running complex scripts or batch operations that require direct execution in the database's native environment.

Extra command arguments

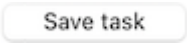
The **Extra command args** field allows you to input additional command-line arguments to further customize the dump process. This offers more specific control over the operation.

For example, when configuring backup/restore for PostgreSQL, you may want to exclude specific tables. To exclude a table named `employee_data`, use the `--exclude-table` option. Enter `--exclude-table=employee_data` in the **Extra command args** field during the setup process.

By doing this, the resulting process won't contain any data or structure from the `employee_data` table.

Note: Always consult the database's official documentation for a comprehensive list and explanation of available command-line arguments.

Scheduling dump and restore operations as tasks

You can set up both restore and dump operations as tasks. To save an operation as a task, click the **Save task** button  during the dump/restore process. For detailed instructions and features related to task management in DBeaver, refer to [Task management article](#).

Schema compare overview

Note: This feature is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Databases supporting schema comparison](#)

[How it works](#)

[What is possible to compare](#)

[How to use schema compare](#)

[Step1. Select two entities to compare](#)

[Step 2. Check the selected entities](#)

[Step 3. Specify compare settings](#)

[Step 4. Look at the comparison results and save the report](#)

[Step 4. Comparison results](#)

[Quote objects names in the report](#)

[Compare logs](#)

[Liquibase Changelog generation](#)

[Save operation as a task](#)

[Using schema compare with Liquibase PRO key](#)

[Object types being compared by LiquibasePRO](#)

Databases supporting schema comparison

-----	-----
Cockroach	Oracle
Databricks	PostgreSQL
DB2	Redshift
Derby	SAP HANA
EnterpriseDB	Snowflake

Firebird	SQLite
Greenplum	SQLServer
Informix	Teradata
MySQL/MariaDB	Vertica
Netezza	YugabyteDB
DB2i	

How it works

You can compare two schema/database structures and generate a report in the following formats:

- DDL script (series of create/alter/drop statements)
- Diff diagram (sort of ER diagram)
- Liquibase changelog
- Liquibase change report (JSON, YAML, or plaintext)

What is possible to compare

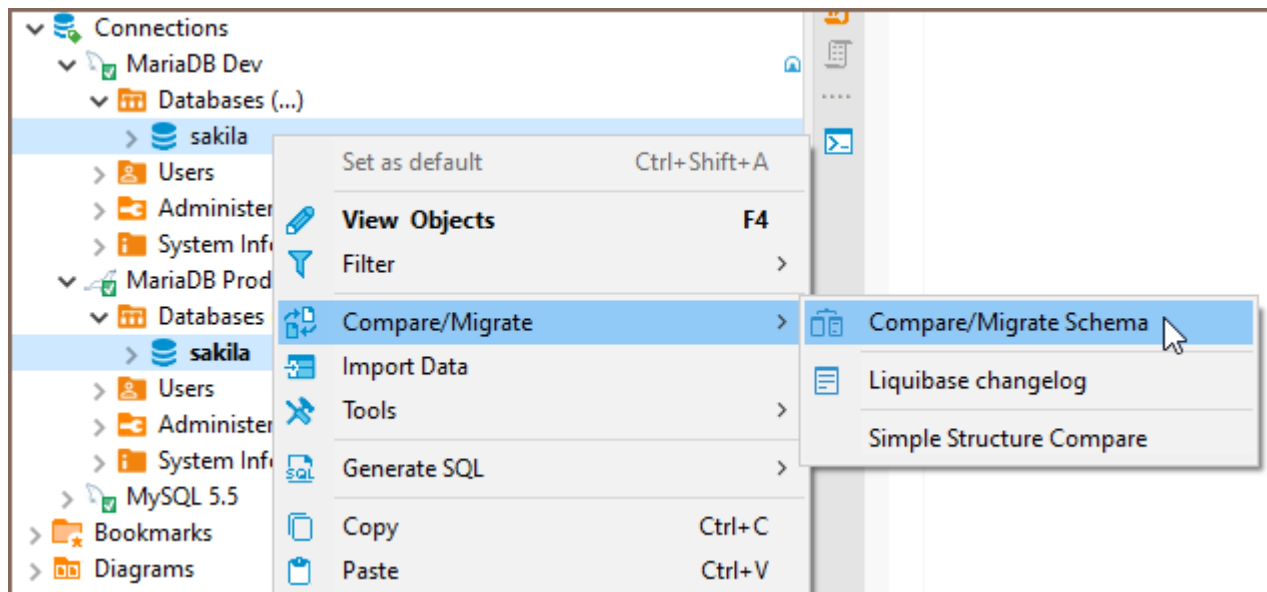
In most cases, you can compare **schemas, databases, or tables**. However, some databases (such as SQLite and Firebird) do not have catalogs and schemes that can be compared. In this case (and only for these databases), it is possible to compare the entire **datasources**.

Usually, you can compare the following database objects: **columns, primary keys, foreign keys, indexes**, and so on (it depends on your database). If you want to compare [more objects](#), such as **check constraints, procedures, functions, triggers**, you need to enable [Liquibase PRO](#).

How to use schema compare

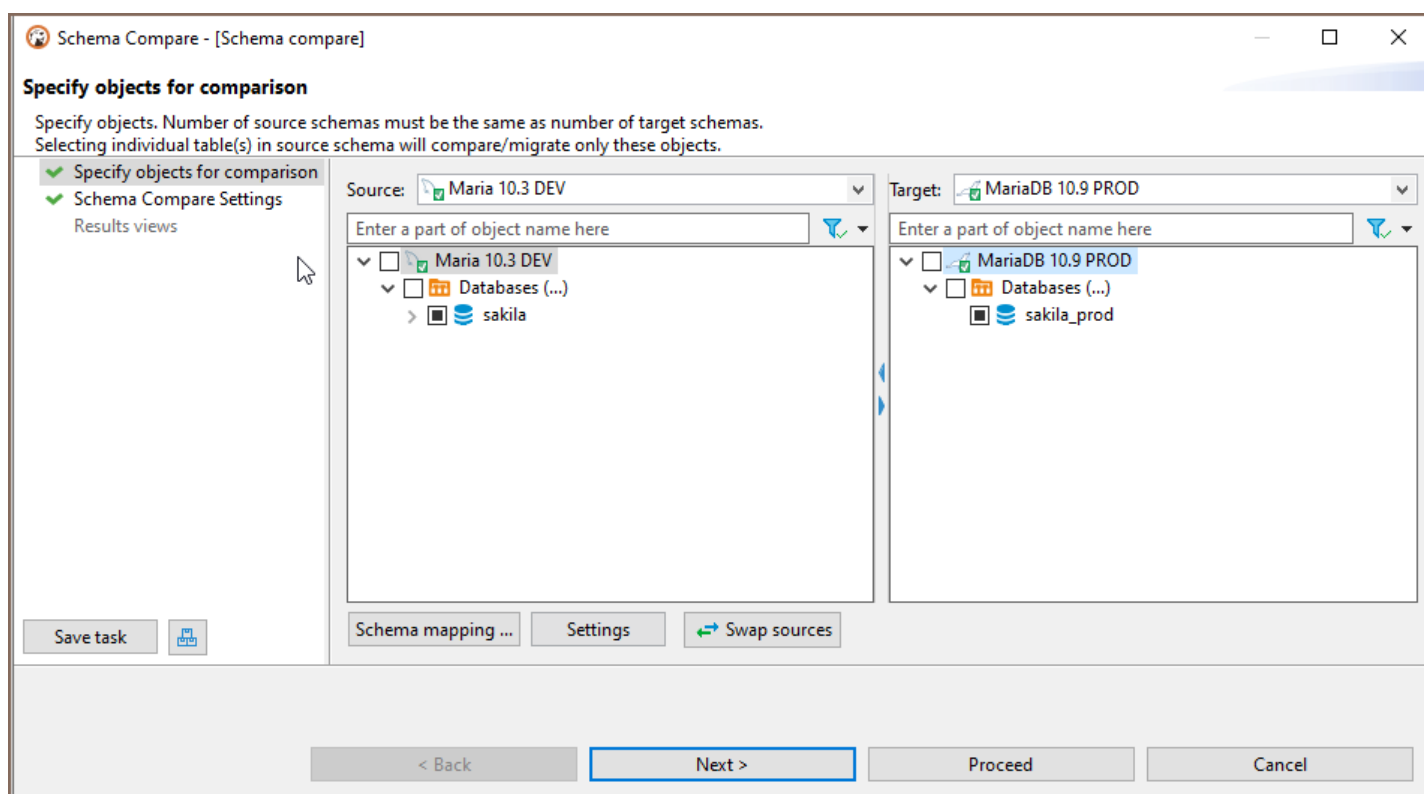
Step1. Select two entities to compare

1. Select the two objects (schemas, databases, or tables) you want to compare in the **Database Navigator**.
2. Open the context menu.
3. Open the sub-menu **Compare/Migrate** and click the **Compare/Migrate Schema**. You'll see the comparison window.



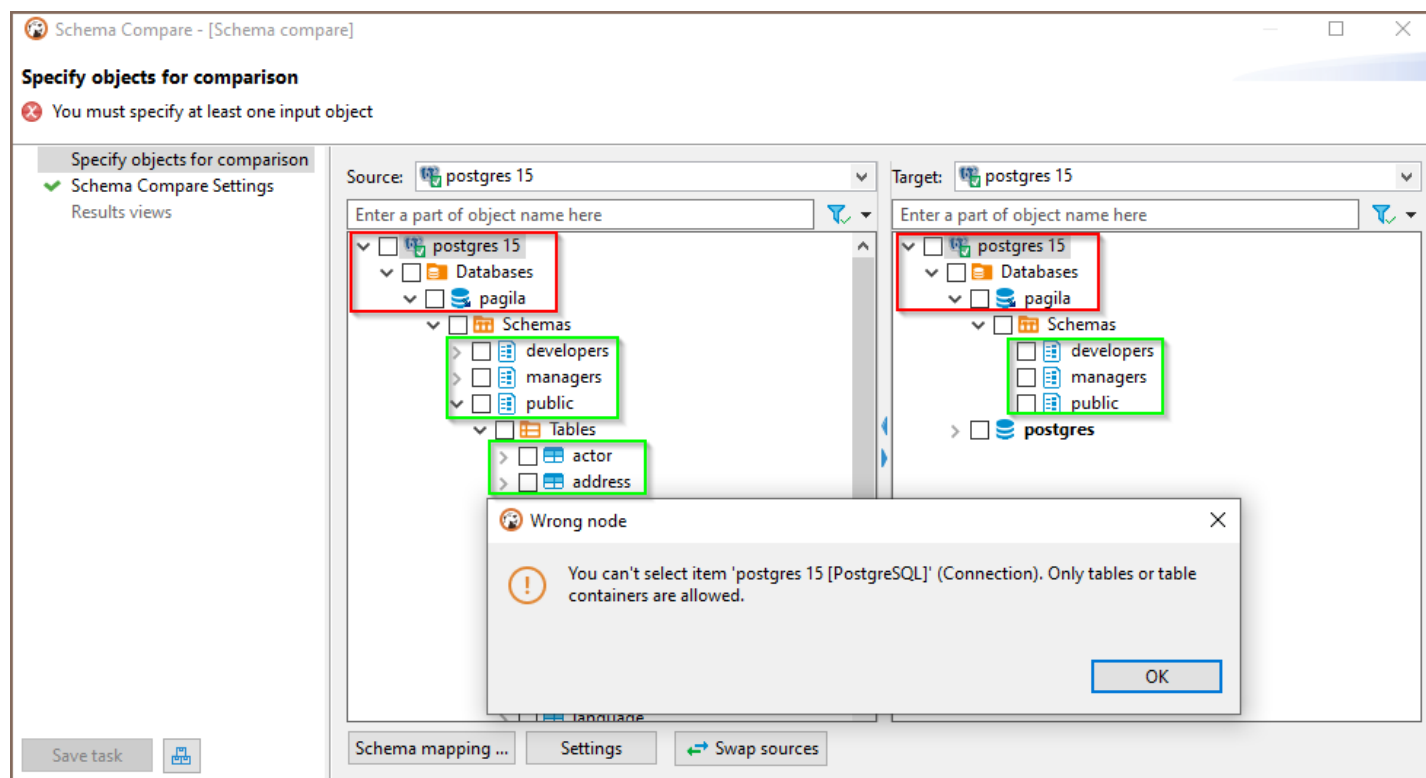
Step 2. Check the selected entities

Re-validate that you have chosen the correct objects to compare. You can change target and source containers by clicking the **Swap sources**.

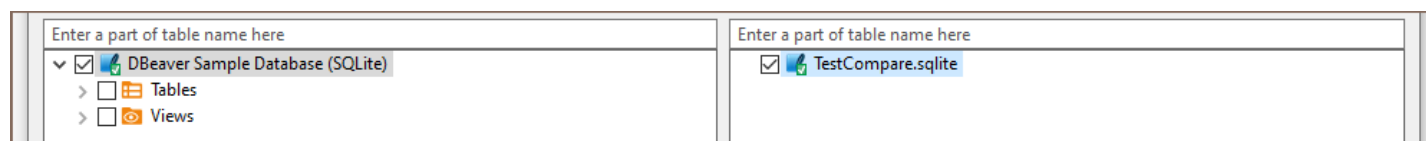


Note: You must select only those schemas/directories/containers that contain tables:

- Schemas – if the database supports schemas.
- Databases – if the database supports catalogs and does not support schemas.



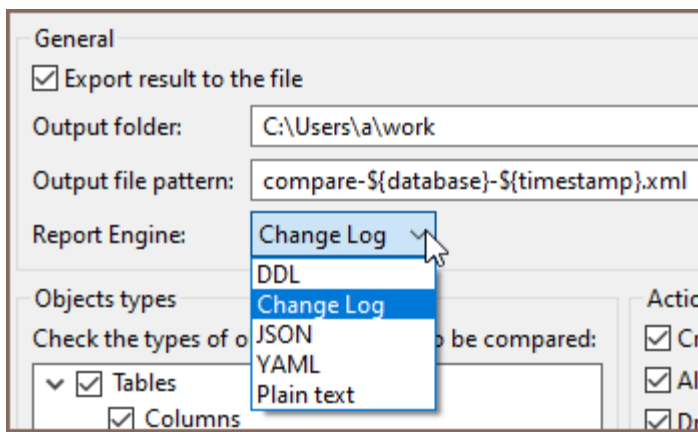
- Datasources – if there is no support for schemas or catalogs.



If everything is correct, click **Next**. You'll see the comparison settings.

Step 3. Specify compare settings

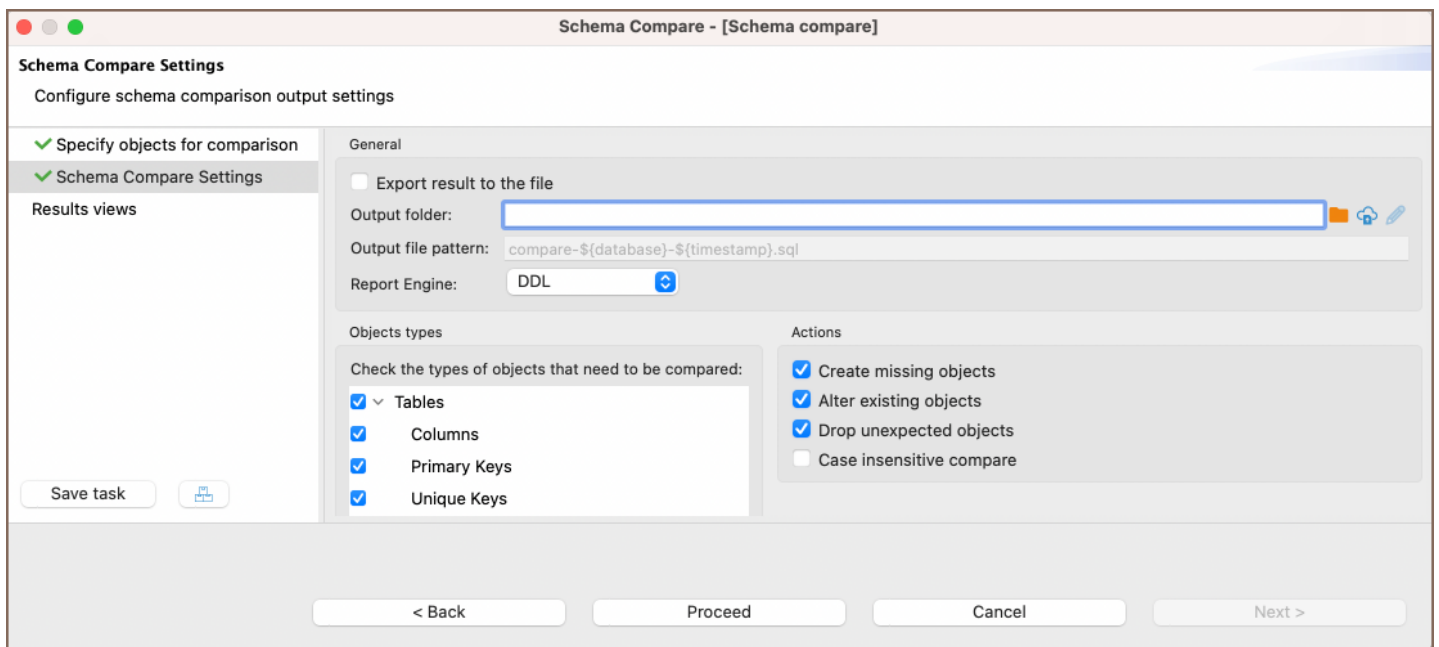
1. If you want to export the comparison result in a file, select **Export result to the file** option, then specify the folder and file name. You can use variables in the file name. Click on the field with a file name to see a list of available variables.
2. Select the report format in the **Report Engine** field.




3. You can simply exclude specific types of objects from comparison.

For example, it is possible not to show sequences, views, or external keys in the final comparison result.

4. You can also specify which changes should be processed: create, drop, or alter. By default, all kinds of changes are enabled. If you do not want to compare objects with equal names but in different cases (like "test" and "Test"), enable the **Case insensitive compare**. (Note: This settings section is unavailable for the generation changelog process.)



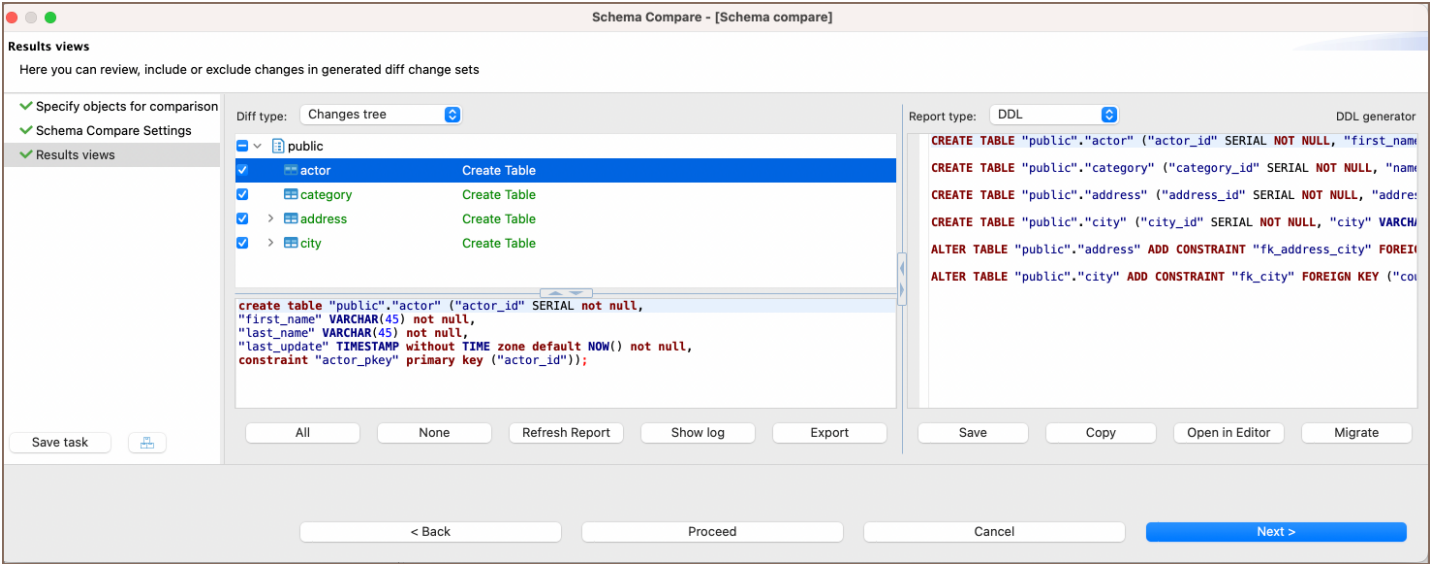
Tip: You can export the results of a schema compare operation to a remote file system via [Cloud Storage](#) using the **Browser remote file system** button . This feature is exclusively available to users of the Ultimate Edition, Team Edition, and CloudBeaver versions.

Step 4. Look at the comparison results and save the report

Click **Proceed** to generate a diff report.

Step 4. Comparison results

By default, DDL diff is generated. It contains a series of creating, altering, and/or dropping statements that will modify the schema on the right side. Thus, it will make it identical to the schema on the left side.

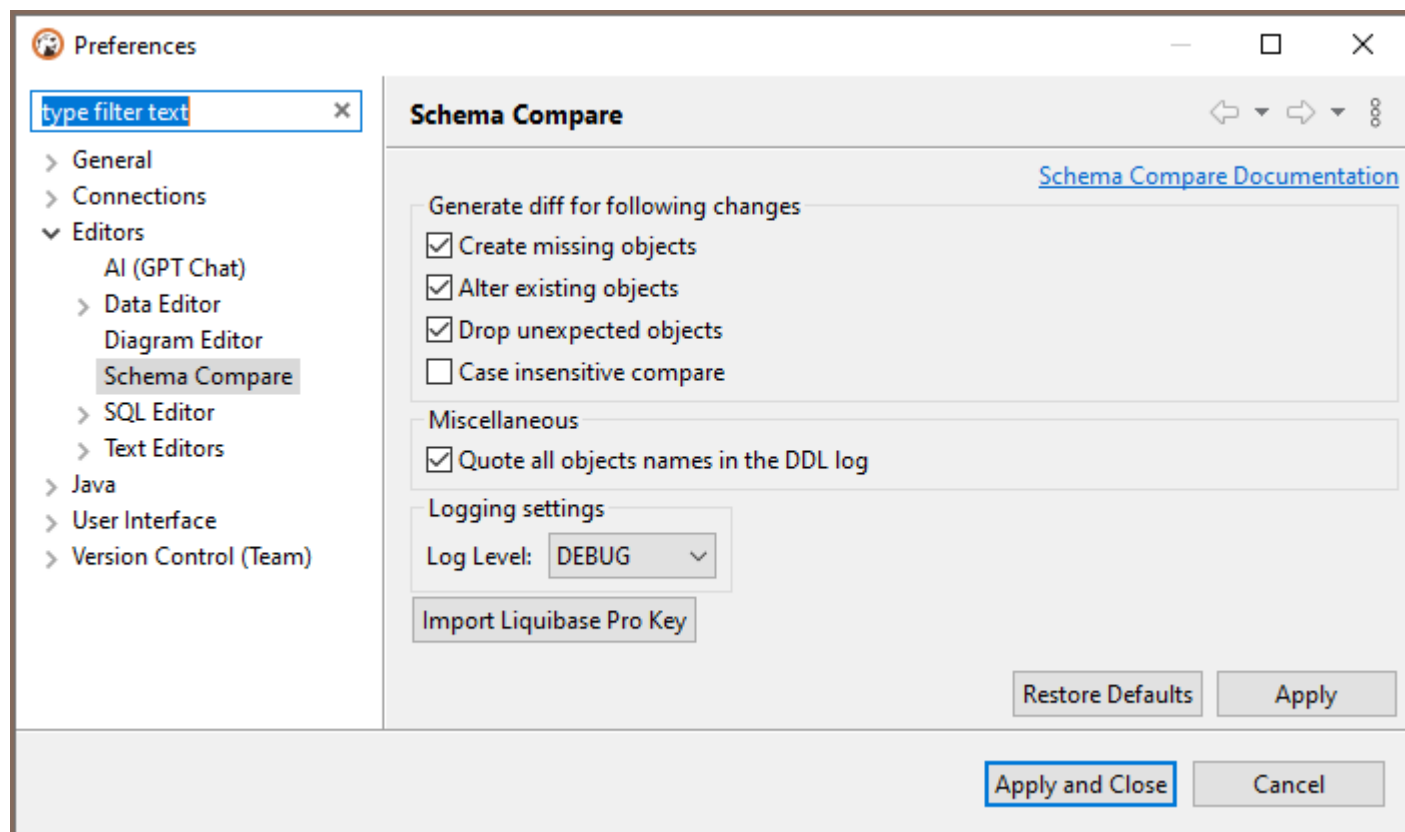


This comparison interface provides a suite of options to manage your results effectively:

Button /Option	Description
All	Enables all changes in the changes tree on the left side of the diff page.
None	Disables all changes in the changes tree on the left side of the diff page.
Refresh Report	Updates the comparison report to reflect any changes made to the selection or schema.
Show log	For detailed information about logs, refer to the section Compare logs .
Export	Exports the current diff report to a <code>.txt</code> file.
Save	Saves the current state of the report.
Copy	Copies the generated SQL or report text to the clipboard for easy pasting elsewhere.
Open in Editor	Opens the current diff report in the SQL Editor for further editing or review.
Migrate	Executes the generated SQL statements on the target database, applying all selected changes.

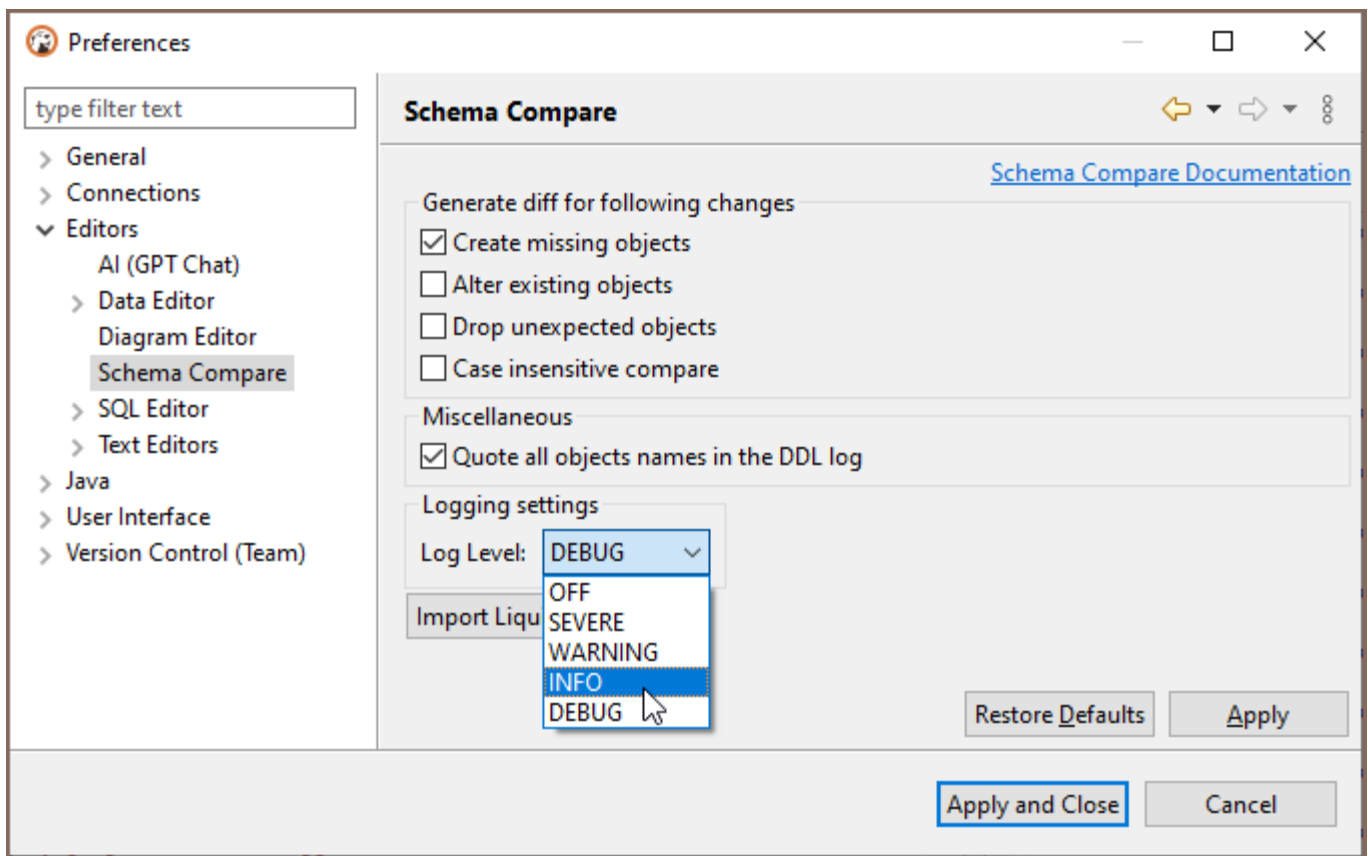
Quote objects names in the report

For the objects in the report to be dressed in quotation marks, select the **Quote all objects names** option in Preferences.

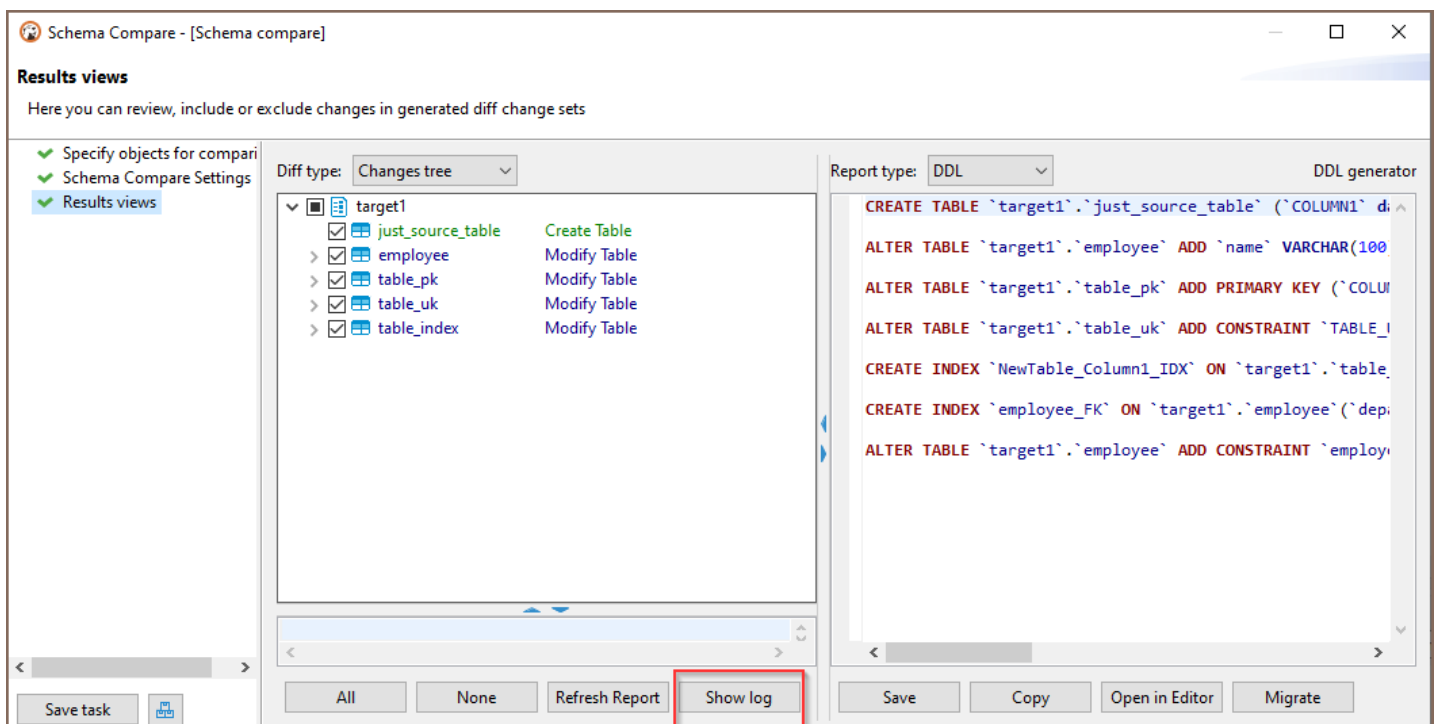


Compare logs

To get acquainted with the comparison logs, you first specify the logging level on the **Preferences-> Editors-> Schema Compare** preference page. Specify one of the logging levels and click on **Apply**. By default, the logging level is the OFF level. To get complete information, you can choose the DEBUG level.

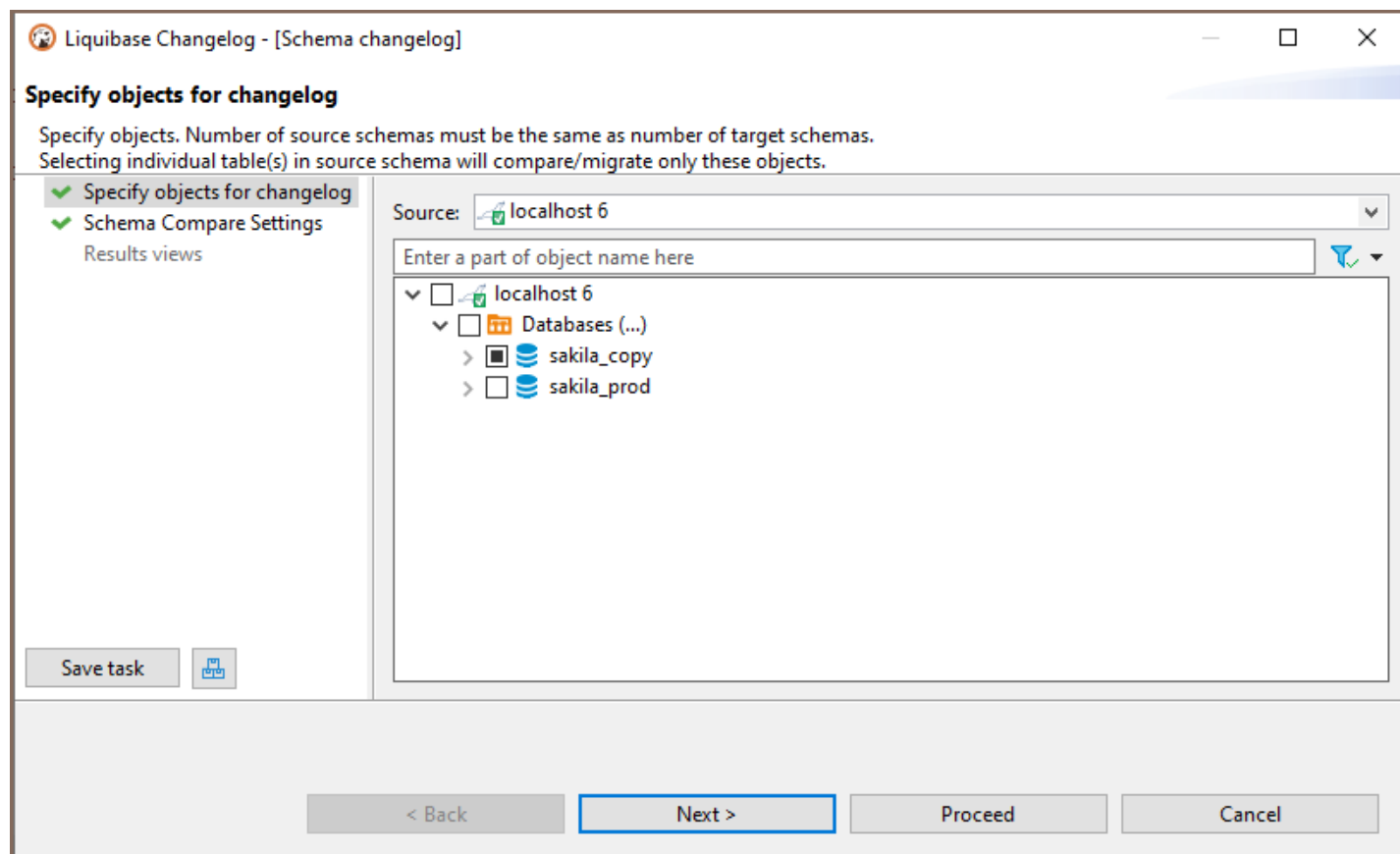


After comparing operations, click the **Show log** button. A log will be open in the Editor, and the content of this log will depend on the logging level you choose in the settings. Log level changes from preferences will not be applied to the comparison wizard if it is already open in another window. Close and open the schema compare wizard in this case.

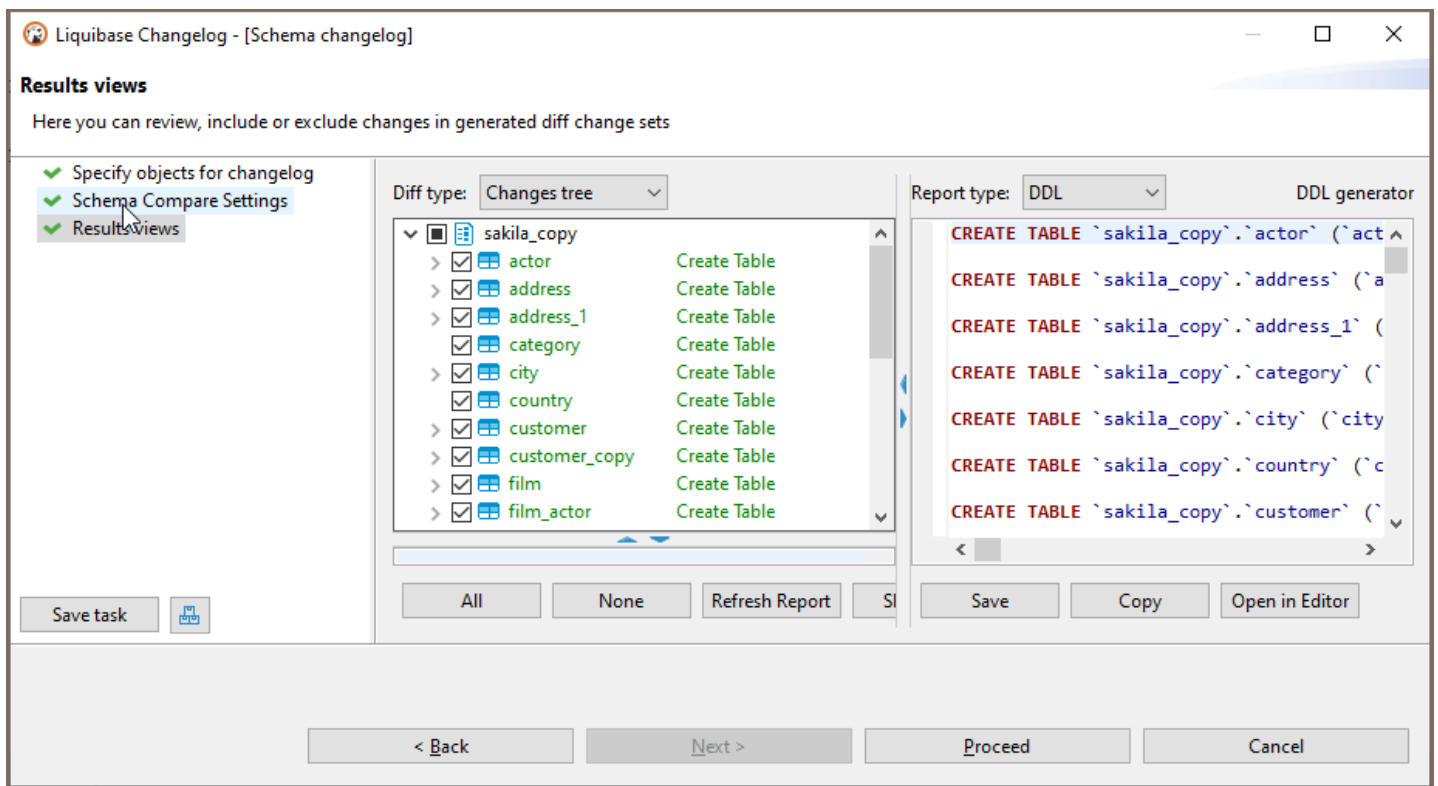


Liquibase Changelog generation

Suppose you want to create a report about the objects in your table container (similar to the metadata dump operation). In that case, you can select in the navigator tree on your container **Compare/Migrate -> Liquibase changelog** command.

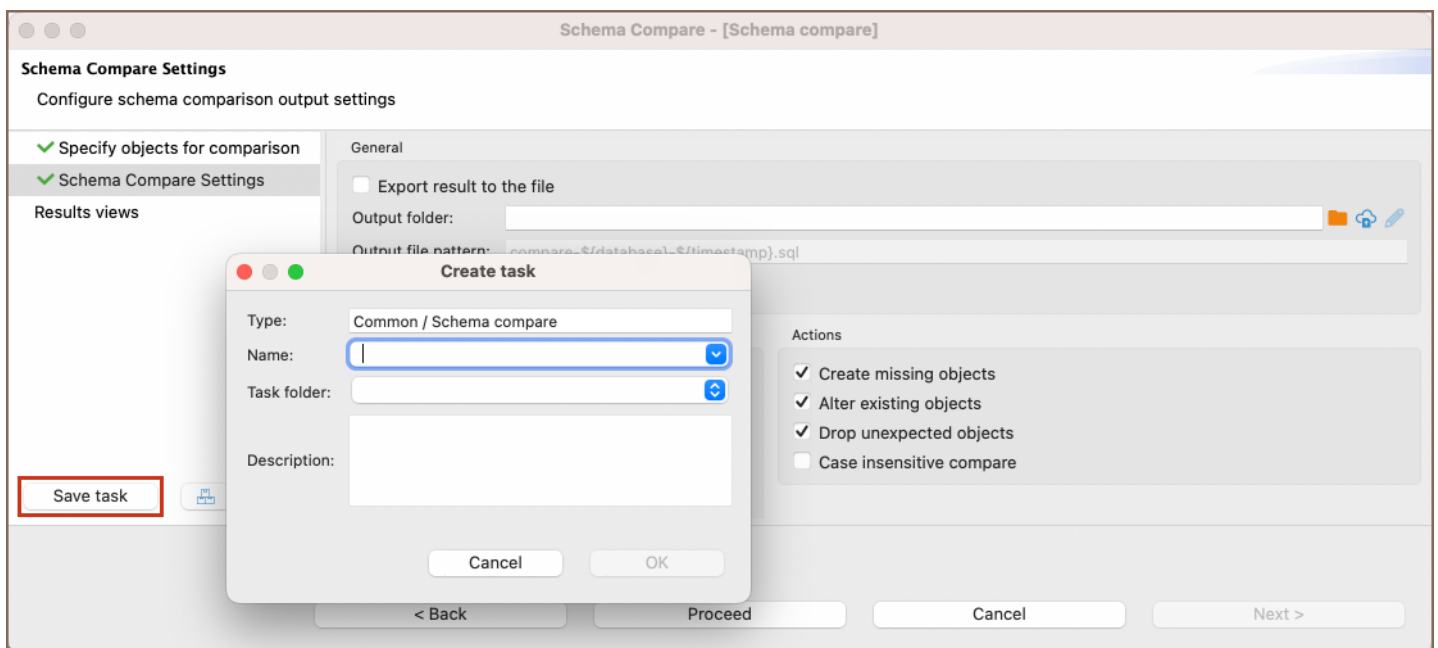


One or several table containers can be chosen. The report will contain creation statements of tables/views/keys /sequences - metadata from the table containers. But without data from tables/views. You can use this report in the future to restore the structure of your database.

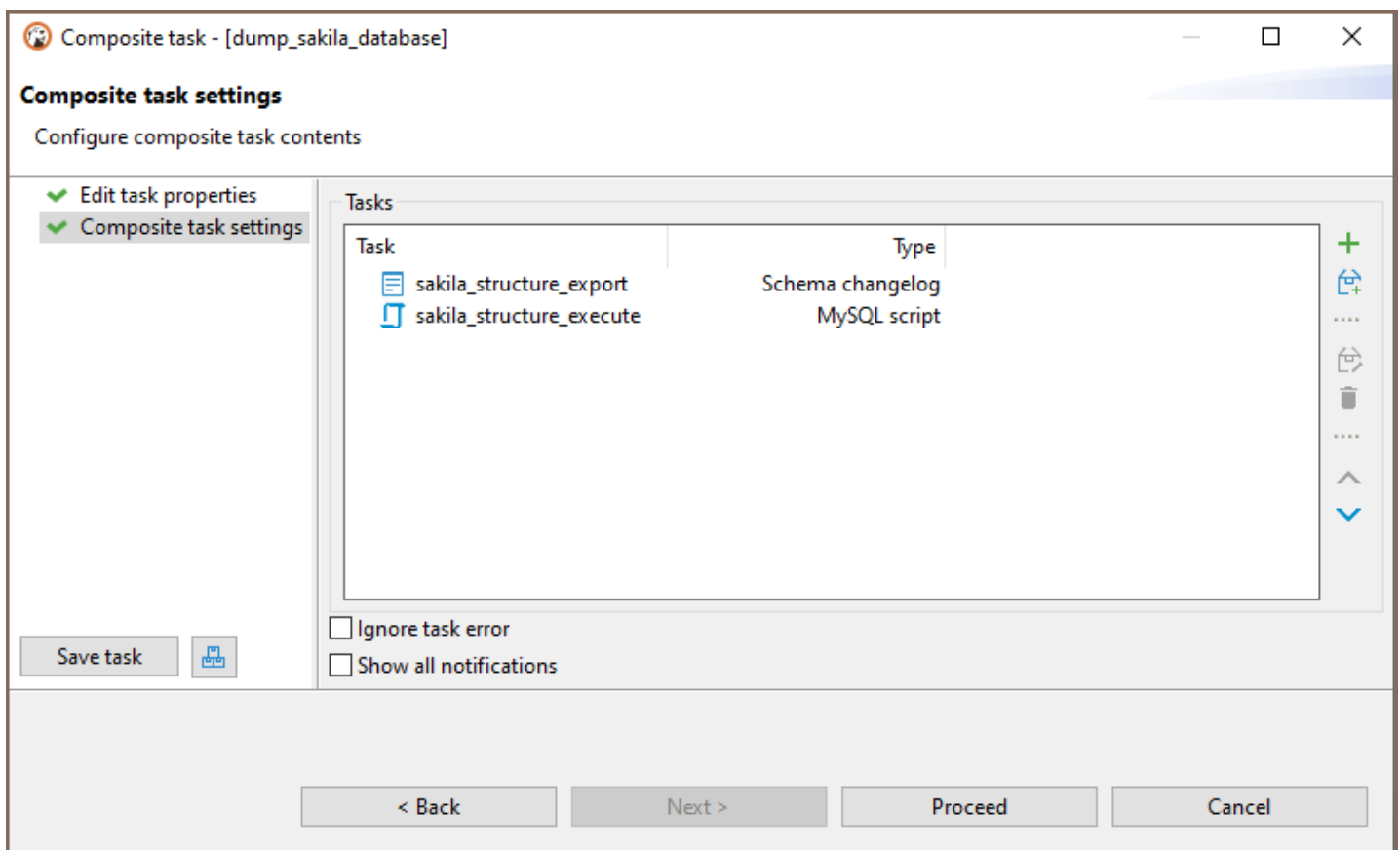


Save operation as a task

If you plan to constantly use the comparison or generation changelog/schema comparison, exporting the result as a file, you can save this as a task. The task can be completed anytime and put on the [schedule](#).



Also, generating a changelog or schema comparison can be part of the [composite task](#). For example, if you keep the log as a script with a certain name, then the implementation of this script can be set by the second operation of the composite task.



Using schema compare with Liquibase PRO key

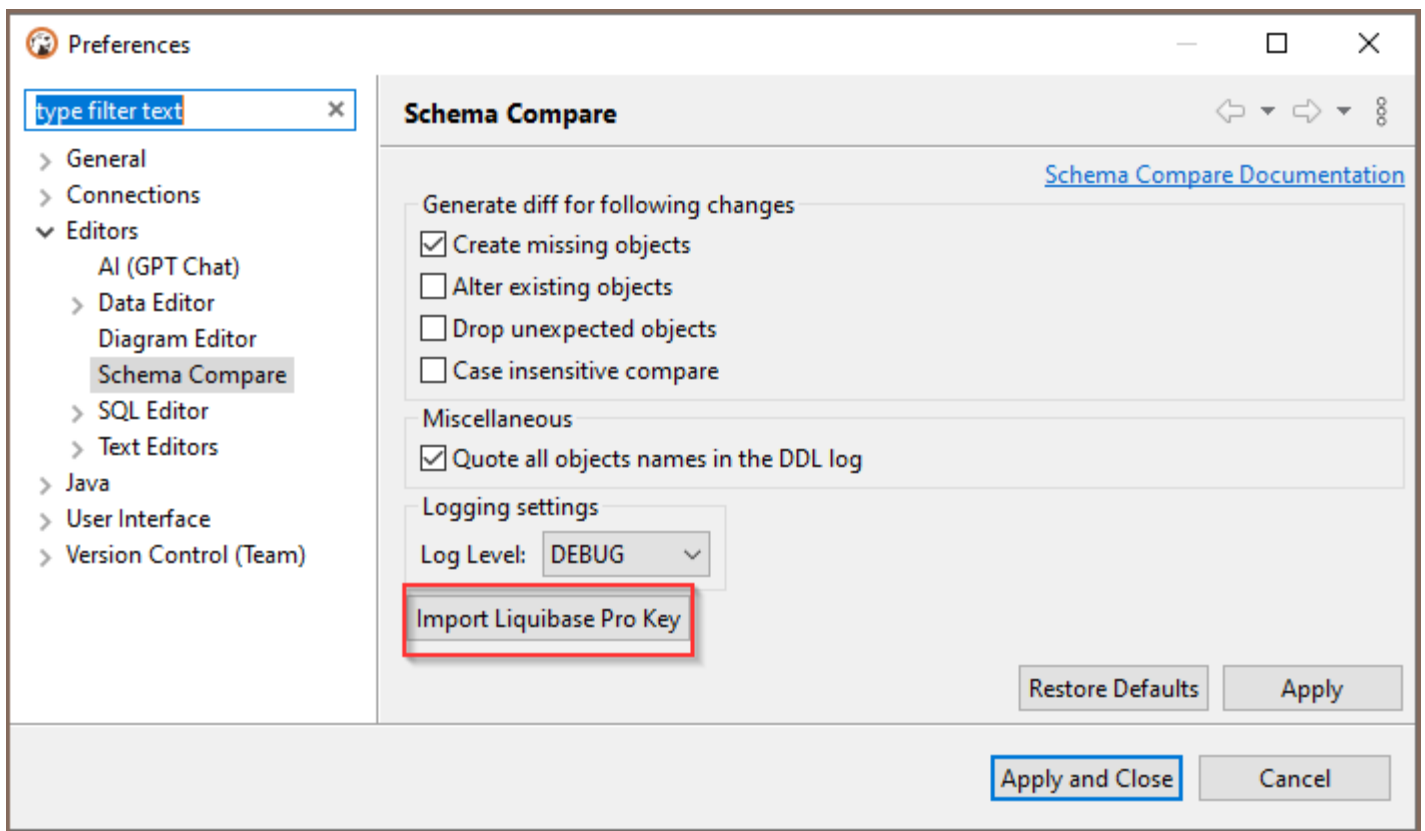
If you have a Liquibase PRO key, then you can use it with DBeaver. Steps you need:

- Find and open your dbeaver.ini file.

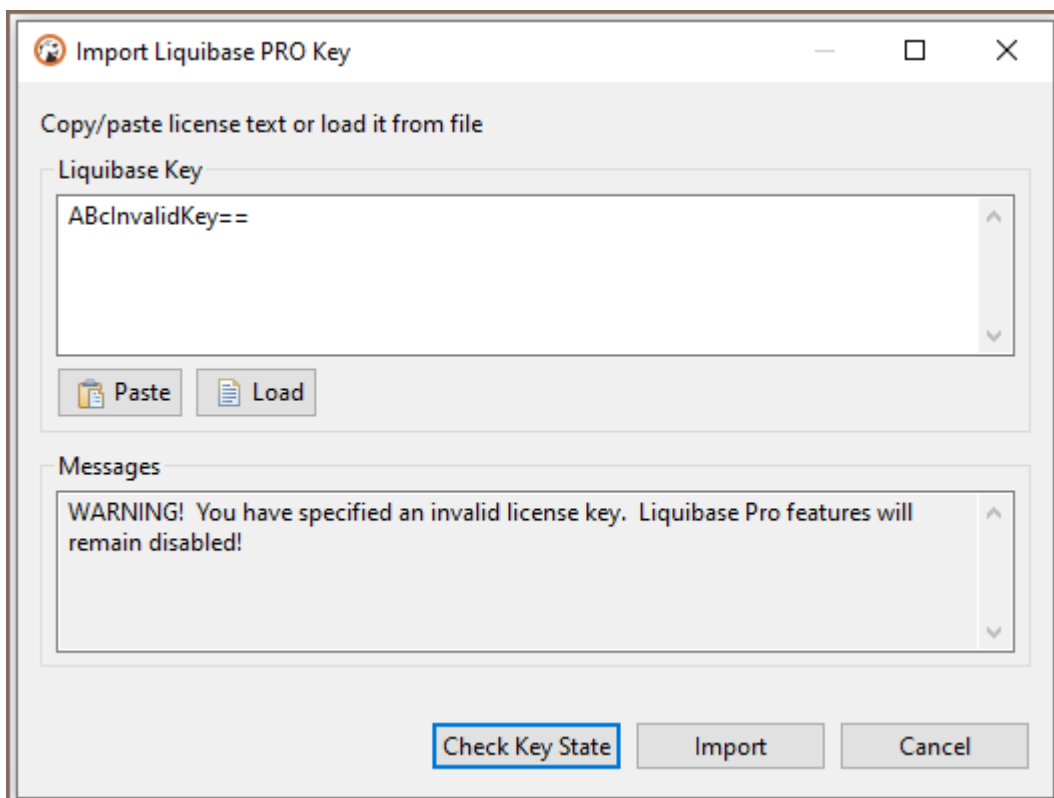
Tip: Detailed instructions on finding `dbeaver.ini` are available in [our article](#).

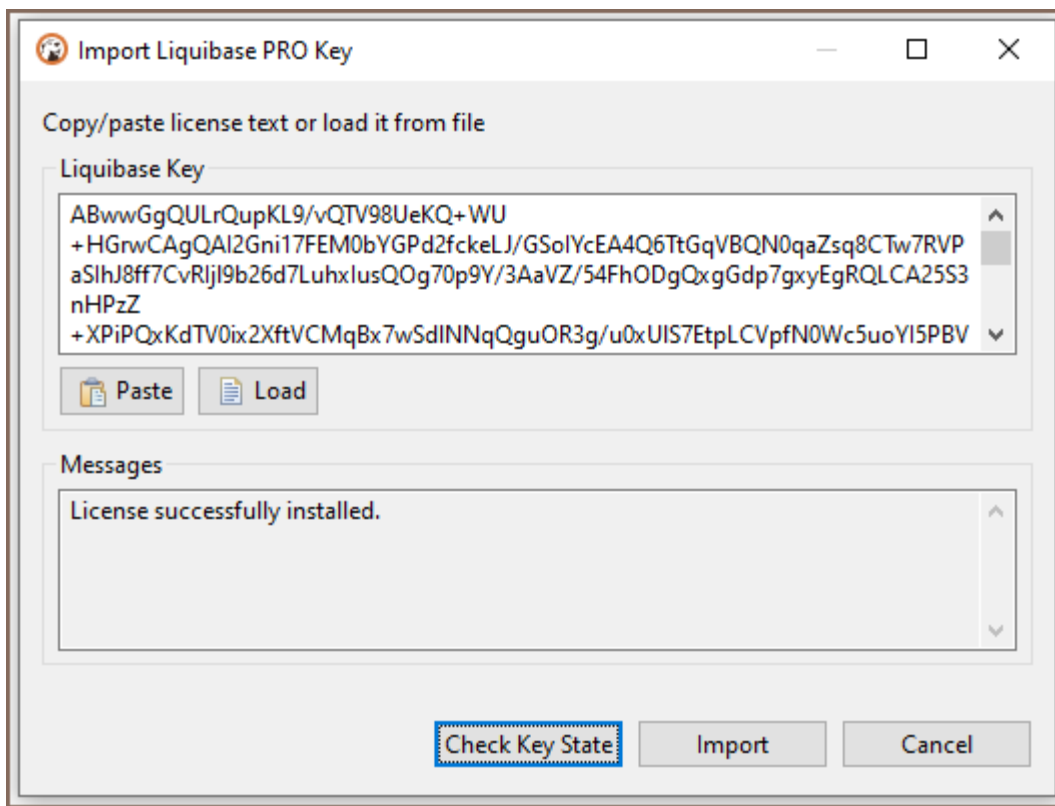
- Find -vmargs command
- Add a new line after this command: -Dliquibase.license.key=yourKey (example: -Dliquibase.license.key=ABwwGgQU...)
- Open DBeaver and the "Schema compare" window. The key will be checked at this step.

You can also add the Liquibase Pro key via UI in Preferences->Editors->Schema Compare preference page. Use the **Import Liquibase Pro Key** button to open the Import key dialog.



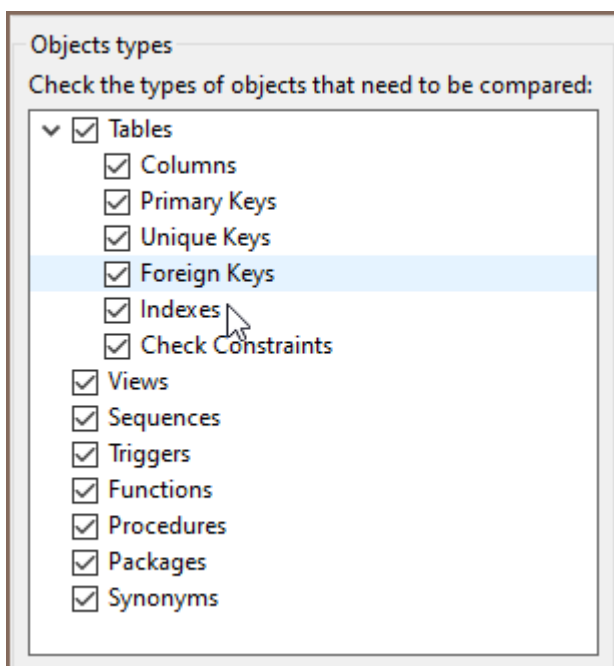
You can manually add your key in the Liquibase Key text field, throw the **Paste** button, or use the **Load** button to download a file. You can check the license state with the **Check Key State** button. After pressing the button, you can see the result of the checking in the **Messages** field.





We suggest you restart the program after adding a key for the correct program work. Settings changes will not be applied to the comparison wizard if it is already open in another window. The key will be saved in the DBeaver settings. If you specify the key in the .ini file and install another key through the Import Key dialog, the key from the .ini file will be in priority.

If the license key is valid, the **Object types** dialog will be extended to PRO objects. (If PRO objects didn't appear in the schema compare changelog - check your logs. Maybe the license expired, or the key is invalid)



Object types being compared by LiquibasePRO

- Check Constraints
- Procedures
- Functions
- Triggers
- Synonyms (Oracle)
- Package with the body (Oracle)

Using Liquibase in DBeaver

Note: This feature is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Supported Databases](#)

[Using Liquibase in DBeaver](#)

[Additional Resources](#)

DBeaver integrates with Liquibase, an open-source tool that helps you track, manage, and apply database changes.

Supported Databases

Liquibase works with a variety of databases. The specific databases that DBeaver supports are listed in the [Schema Compare](#) article.

Using Liquibase in DBeaver

With DBeaver's Liquibase support, you can efficiently manage your database changes. You can create and manage Liquibase [changelogs](#) directly in DBeaver, making tracking and applying database changes easier. With the [Liquibase PRO key](#), you can also use DBeaver's [Schema Compare](#) feature for an enhanced comparison of database schemas.

Additional Resources

For more information on Liquibase, refer to these resources:

- [How Liquibase Works](#)
- [Fundamentals of Liquibase](#)

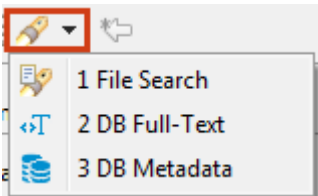
- [Core Liquibase Usage Concepts](#)
- [Liquibase Quickstart](#)
- [Evolving Your Database Schema Safely with Liquibase](#)

Search overview

Table of contents

- [Search View](#)
- [Search History](#)

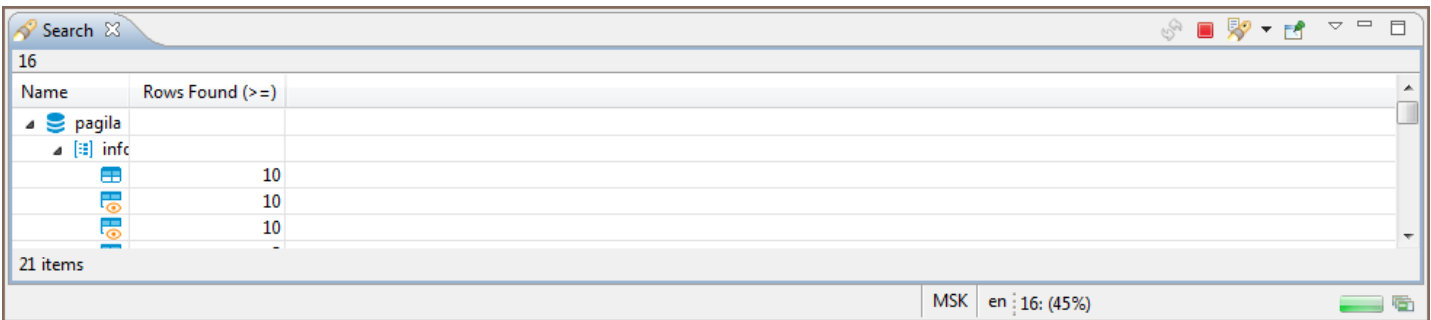
To use search, click the Search button on the main toolbar:




Please see the dedicated articles for information about searching for different types. This article describes common features of the three search types.




Search View

Search results for any of the search types appear in a separate Search view. The following image shows the Search view for the database full text search:






The view contains a toolbar that provides common tools for all types of search as well as specific tools for the File Search type. The following are common tools:

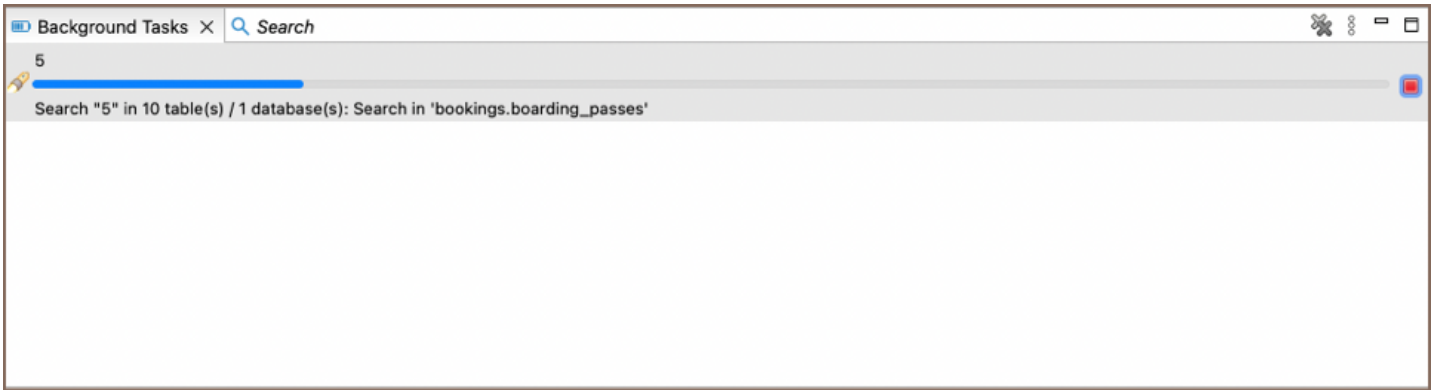
Button	Name	Description
	Run the Current Search	Refreshes the search results

	Again	
	Cancel Current Search	<ul style="list-style-type: none">- Active state (red) indicates that the search is still in progress and appears if the search takes some time to complete. Clicking the button in this state stops the current search.- Inactive state (grey) indicates that the search is complete. The button in this state is non-actionable.
	Show Previous Searches	<ul style="list-style-type: none">- Clicking the button itself opens the Previous Searches window.- Clicking the arrow next to the button opens a dropdown menu. See the <i>Search History</i> section further in this article.
	Pin the Search View	Ties the current search results to the Search view. If you click this button, the current results stay in the view while the results of the next search appear in a new Search view. Otherwise, every new search replaces the previous results with new results.

For information about specific tools of File search, see the [File Search](#) article.

If the search is short, the results appear almost instantly. But if it takes some time, the Search view indicates the progress in the following ways:

- The **Cancel Current Search** button in the toolbar has the Active state ()
- The progress bar appears in the bottom-right corner of the view indicating the process: 
- The button to show the search progress in a separate view () appears in the bottom-right corner of the view next to the search progress bar. Clicking the button opens the Background Tasks view:



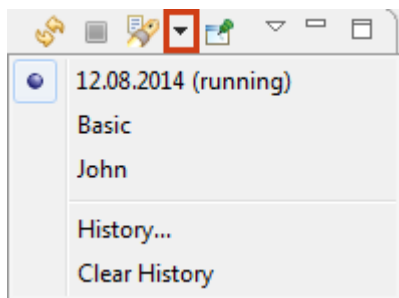
Search History

DBeaver stores the history of search queries made during the current session. You can reopen the Search view with results of a previous search query. You can also remove individual queries and clear the history. To manage the

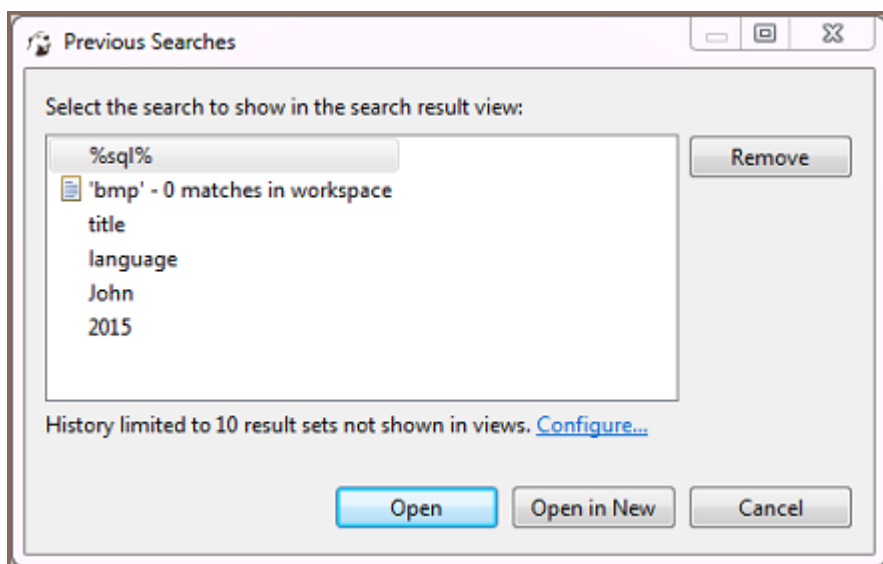
search history, use the **Show Previous Searches** button in the toolbar ().

To open the results of a previous search query, do one of the following:

- Click the arrow next to the **Show Previous Searches** button in the toolbar and then click the query in the dropdown list:



- Click the **Show Previous Searches** button itself or the arrow next to it and then **History** on the dropdown menu to open the Previous Searches window. Then, in the window, click the query and then either click **Open** to open it in the active Search view or click **Open in New** to open it in a new view:



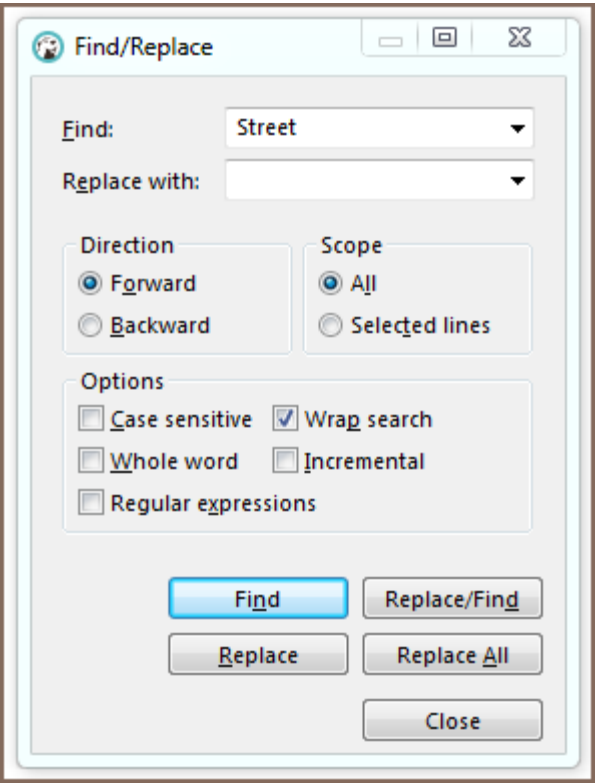
To remove one or more of the previous search queries:

1. Click the **Show Previous Searches** button in the toolbar or click the arrow next to it and then **History** on the dropdown menu. The Previous Searches window opens.
2. Click the query to remove or select several of them by clicking and simultaneously holding the **Ctrl** key.
3. Click **Remove**.

To clear the history by removing all previous queries, click the arrow next to the **Show Previous Searches** button on the toolbar and then click **Clear History** on the dropdown menu.

Data Search

To search for data in the result set, press `CTRL+F` . The standard Find/Replace search dialog box will open:



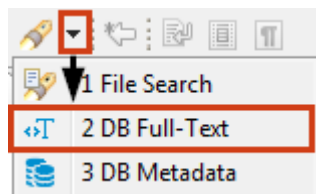
You can also use the Find and Replace feature.

NOTE: The system searches only in already fetched rows.

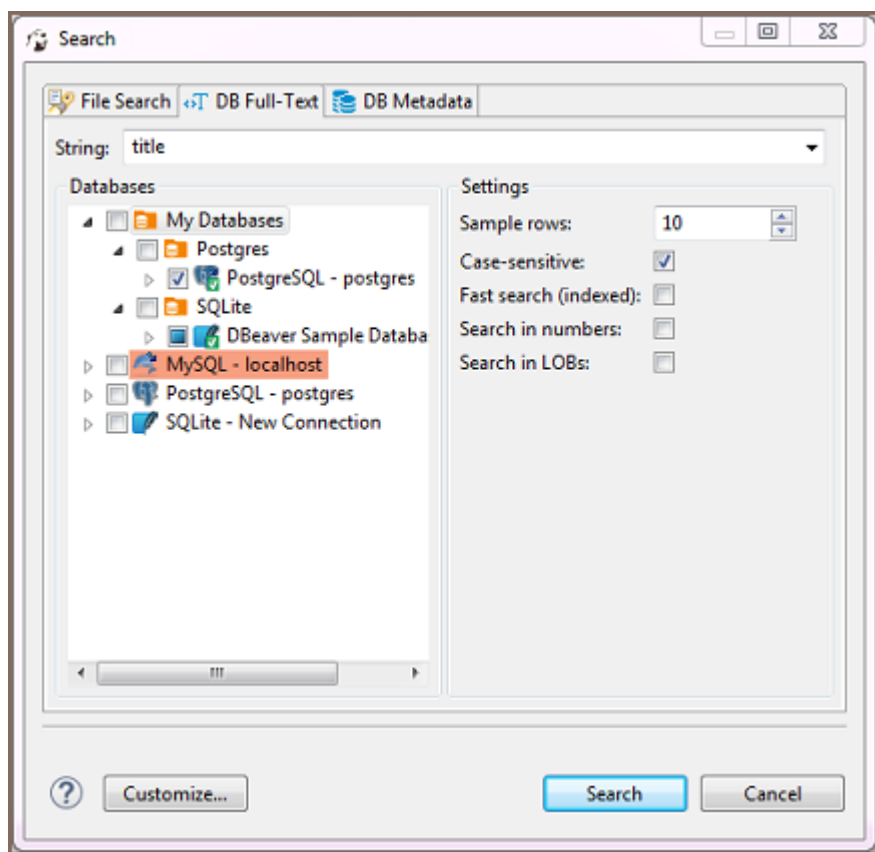
Setting	Description
Case sensitive	By default, the search is case insensitive.
Whole word	By default, the word specified in the search field can be found in the case when the word is part of another word. Enabling this setting will lead to a particular word search.
Regular expressions	Enable to use regular expressions in the search. In the search field, you can use <code>Ctrl+Space</code> for autocomplete regular expressions templates.
Wrap search	Enable this setting to find matches throughout the object from the beginning, and not only from the focus point to the end of the object.
Incremental	Matches are found incrementally as you type, if this setting enabled.

Database full text Search

To do a full text search in the database, click the arrow next to the Search icon in the main toolbar and then click **File Search** on the dropdown menu:



Alternatively, you can click the **Search** button on the main menu and then click the **DB Full-Text** tab in the Search window:

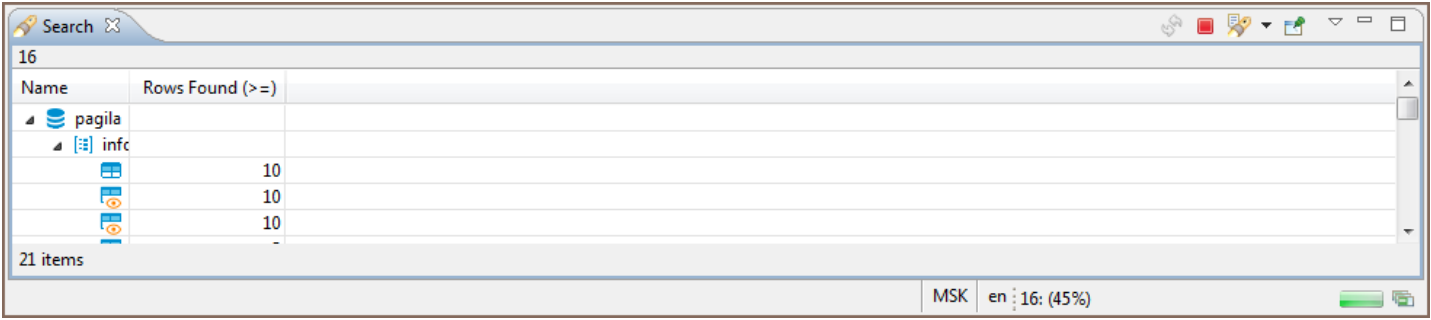


Now you need to choose the database connection or database objects against which to run the search – expand the tree in the **Databases** field to the database connections level or further down and select the checkboxes next to the required connections or database objects.

NOTE: The **Search** button is enabled only when you select the right level of checkboxes – database connections or lower nodes.

You can apply a case-sensitive search, fast search and search in numbers and LOBs.

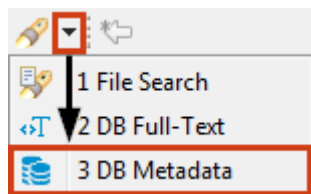
After you click **Search**, the results will open in [Search](#) view:



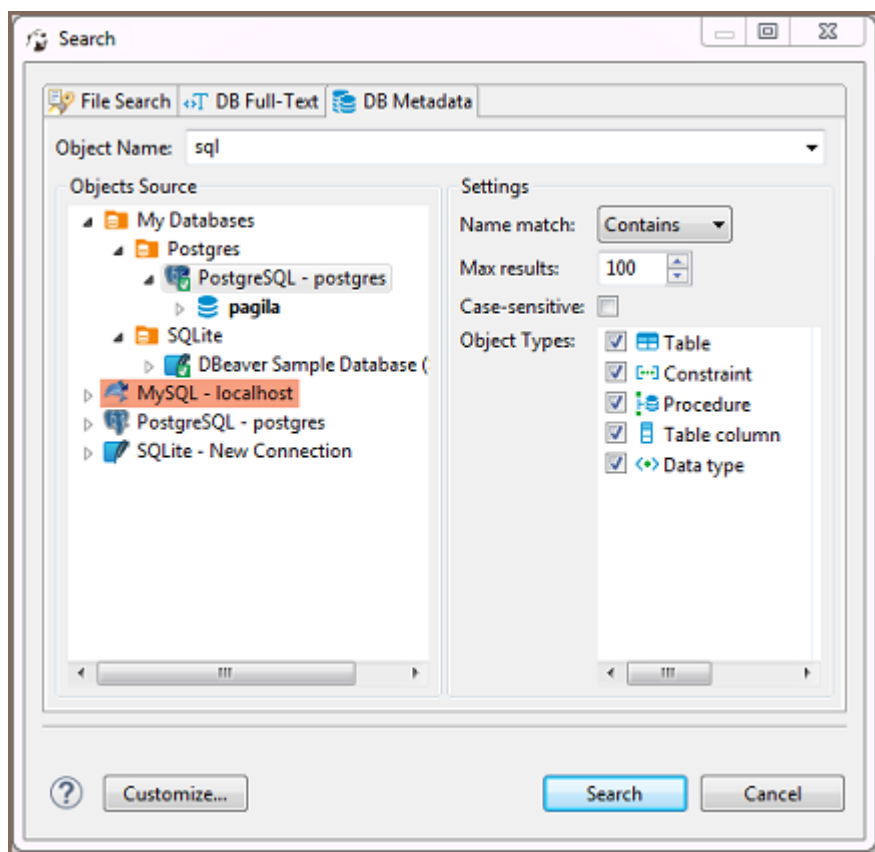
Double-clicking a row in the Search view opens the respective object in a dedicated [Database Object editor](#).

Database metadata Search

To search for database metadata, click the arrow next to the **Search** button in the main toolbar and then click **DB Metadata Search** on the dropdown menu:



Alternatively, you can click the **Search** button on the main menu and then click the **DB Metadata** tab in the Search window:

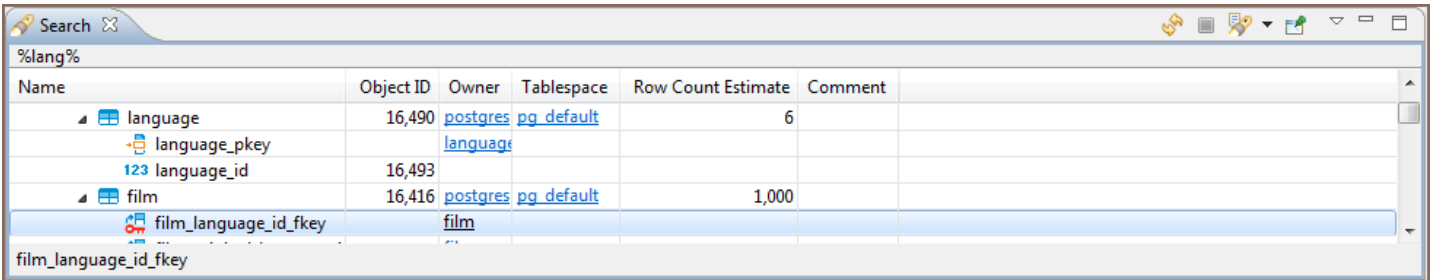


Now you need to choose the database connection against which to run the search. You only need to select the database connection(s) in the **Objects Source** field.

In the **Object Types** field, you can select the database objects among which DBeaver will run the metadata search – select or clear the checkboxes.

You can specify that the object name starts with, contains or is similar to the search combination (**Name match** field). You can also set the maximum number of results to display (**Max results** field) and apply **Case-sensitive** search.

After you click **Search**, the results will open in a [Search](#) view:



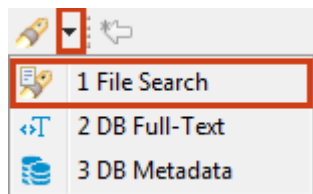
The screenshot shows the 'Search' window in DBeaver. The search criteria is '%lang%'. The results are displayed in a table with columns: Name, Object ID, Owner, Tablespace, Row Count Estimate, and Comment. The results are grouped by object type, with expandable icons on the left of the 'Name' column. The 'language' group is expanded, showing 'language' (Object ID 16,490, Owner postgres, Tablespace pg_default, Row Count Estimate 6) and 'language_pkey' (Object ID 16,493, Owner postgres, Tablespace pg_default, Row Count Estimate 1,000). The 'film' group is also expanded, showing 'film' (Object ID 16,416, Owner postgres, Tablespace pg_default, Row Count Estimate 1,000) and 'film_language_id_fkey' (Object ID 16,416, Owner postgres, Tablespace pg_default, Row Count Estimate 1,000). The 'film_language_id_fkey' row is highlighted.

Name	Object ID	Owner	Tablespace	Row Count Estimate	Comment
language	16,490	postgres	pg_default	6	
language_pkey	16,493	postgres	pg_default	1,000	
film	16,416	postgres	pg_default	1,000	
film_language_id_fkey	16,416	postgres	pg_default	1,000	

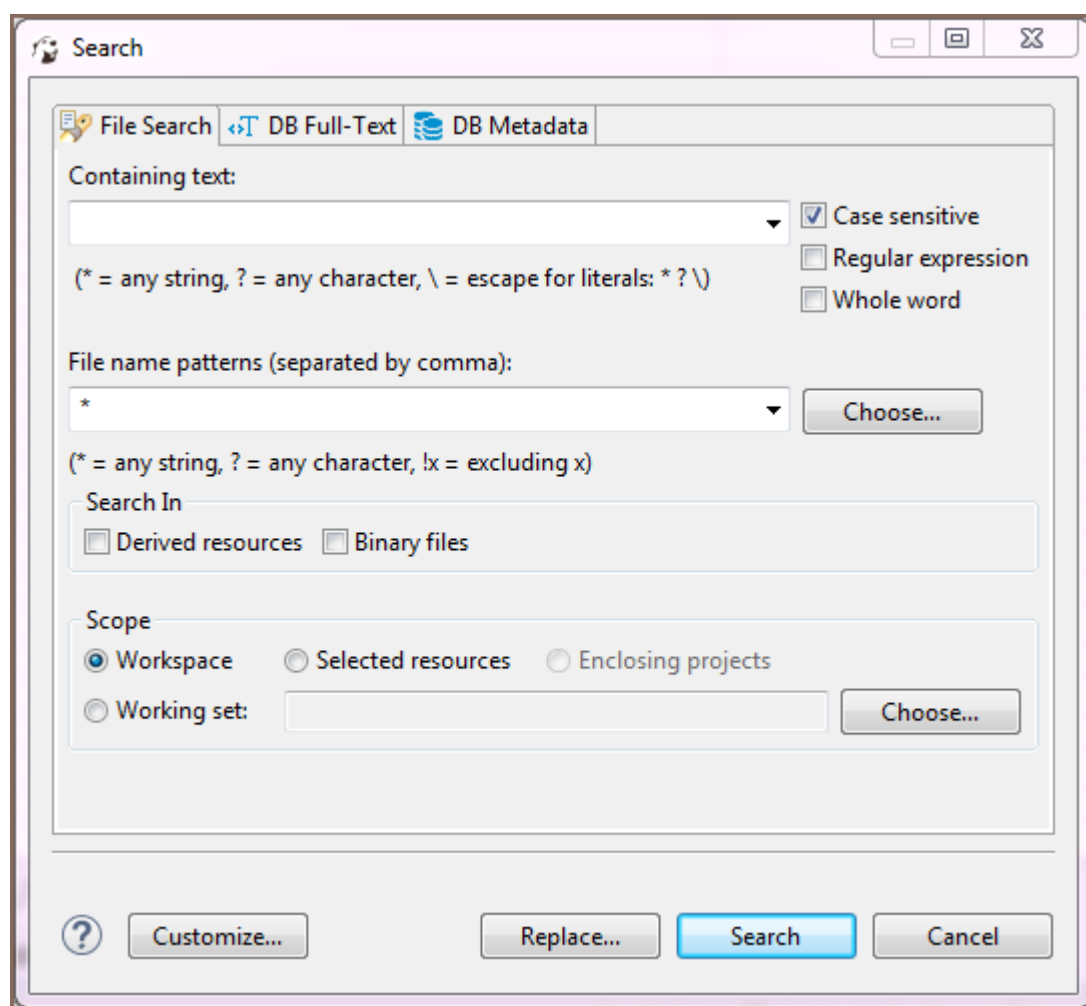
Double-clicking a row in the Search view opens the respective object in a dedicated [Database Object editor](#).

File Search

To search file contents for a string, click the Search button on the main toolbar or the arrow next to the Search button and then **File Search** on the dropdown menu:

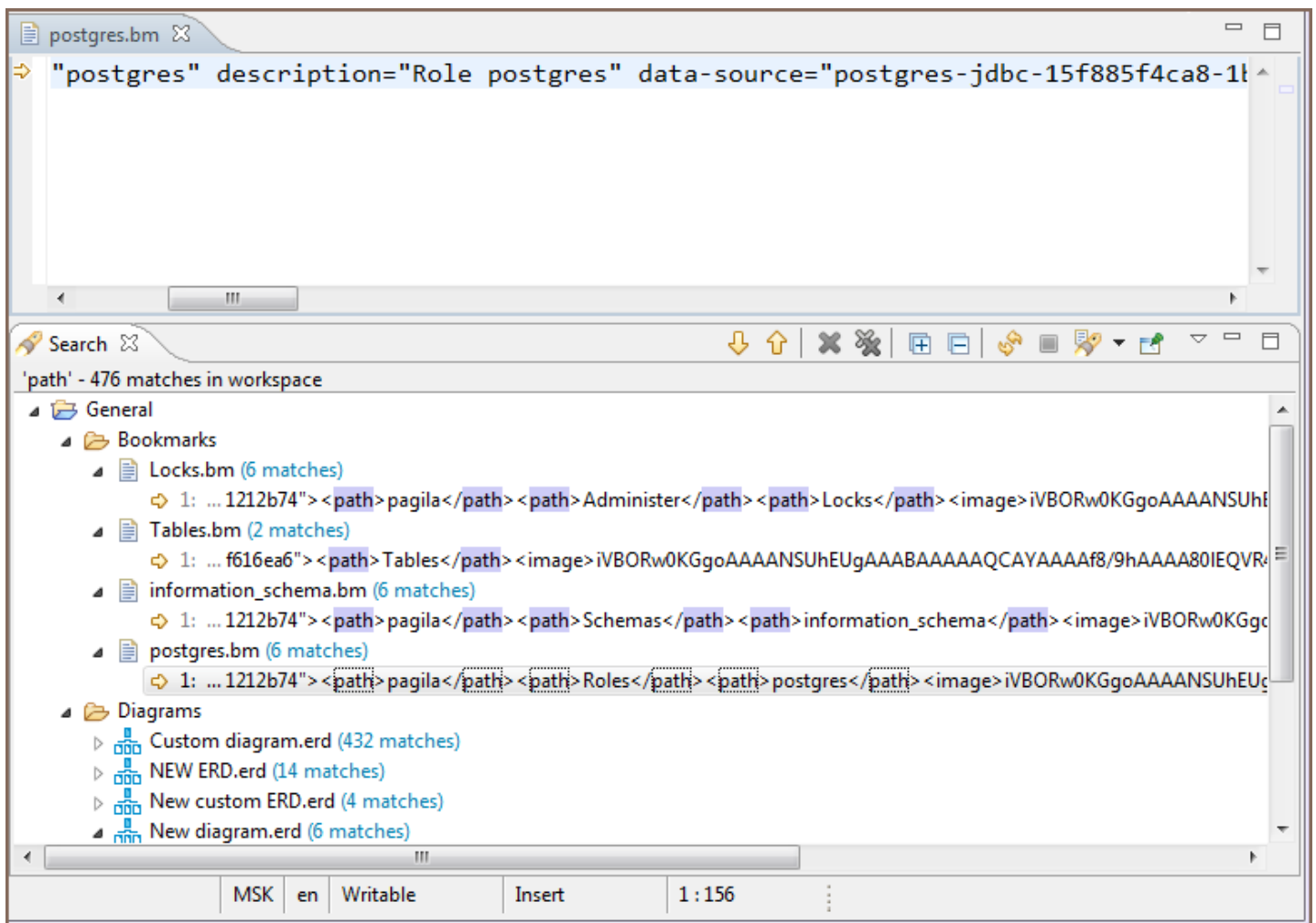


The Search window opens displaying the File Search tab:



You can apply a case sensitive search, search by regular expressions, search among particular file types (**File name patterns** field), and use the find and replace function.

After you click **Search**, the results appear in a [Search](#) view. The results represent a tree or list of files with the search combination highlighted:



The toolbar of the Search view for File search provides more tools in addition to those available for all search types:

Button	Name	Description
	Show Next / Previous Match	Open the file in a separate viewer and move the highlight to the next/previous match, respectively
	Remove Selected Matches	Removes selected row (row in focus) of the results
	Remove All Matches	Removes all results in the view
	Expand/Collapse All	Expand/collapse the tree of results

The view also provides a view menu (click the **View Menu** button () in the upper-right corner of the view) that contains the following items:

Menu item	Description
Show as List	Presents the results in the form of list
Show as Tree	Presents the results in the form of tree

Filters	Opens Search Filters dialog box
Preferences	Opens the Preferences window on the Search page

Double-clicking a results row opens it in a separate viewer.

Task management

Note: This feature is available in Community, [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Creating tasks](#)

[Create a task from tool configuration](#)

[Task editing and execution](#)

[Adjusting task configurations](#)

[Create a task from task management view](#)

[Tasks folders](#)

[Scheduling tasks](#)

Creating tasks

The task is a saved configuration of a database tool. It can be started from the task management view or from the menu by a single click. You can create tasks for frequently used tools.

Create a task from tool configuration

You can save the tool configuration into a task and run your task later with a single click. For example, you can start the [Data Transfer](#) wizard and configure the data export from several tables in the SQLite database into CSV files:

Data Transfer

Confirm
 Check results

Objects

Source Container	Source	Target Container	Target
DBever Sample Databa...	Album	C:\temp\csv-export	Album_202110011321.c...
DBever Sample Databa...	Artist	C:\temp\csv-export	Artist_202110011321.csv
DBever Sample Databa...	Customer	C:\temp\csv-export	Customer_20211001132...
DBever Sample Databa...	Employee	C:\temp\csv-export	Employee_20211001132...

Source settings

Table settings:

Open new connection(s): Yes
 Extract type: SINGLE_QUERY
 Select row count: Yes
 Selected rows only: No
 Selected columns only: No

Target settings

Files settings:

Write to the single file: No
 Directory: C:\temp\csv-export
 File name pattern: \${table}_\${timestamp}
 Encoding: UTF-8
 Timestamp pattern: yyyyMMddHHmm
 Insert BOM: No
 Compress: No

Variables ... Save task [Open Tasks view](#)

< Back

Next >

Start

Cancel

Click on the **Save configuration as task** button and fill the task properties:

Create task

Type:

Name:


Task folder:

Description:

OK






Cancel

Now click on the **Open Tasks view** link to open the task list:


Database Tasks 							
Tasks: type a part of task name here							
Name	Created	Last Run	Last Result	Last Duration	Type	Project	
▼ exportSummer							
Export Many Tables to Incorrect Path	2021-10-01 13:22:49	N/A	N/A	N/A	Data export	General	
Export20210108	2021-10-01 11:31:05	2021-10-01 11:31:22	Success	385ms	Data export	General	
ExportArt2021	2021-10-01 11:31:54	2021-10-01 11:32:05	Success	163ms	Data export	General	
ExportToJSON	2021-10-01 11:32:43	2021-10-01 11:32:57	Success	100ms	Data export	General	
Sample Task	2021-10-01 14:02:52	2021-10-01 14:02:55	Success	251ms	Data export	General	
▼ ImportAutumn							
ImportNovember	2021-10-01 11:43:16	N/A	N/A	N/A	Data import	General	
ImportOctober	2021-10-01 11:34:16	2021-10-01 11:40:19	Can't insert row	7.529s	Data import	General	
ImportSeptember	2021-10-01 11:42:22	2021-10-01 11:42:24	Success	258ms	Data import	General	
▼ Report							
Dump Report Database	2021-10-01 11:45:19	2021-10-01 11:46:20	Success	22.698s	MySQL dump	General	
Report Script	2021-10-01 12:33:04	2021-10-01 12:56:54	Success	3.482s	MySQL script	General	
Restore Report	2021-10-01 13:09:59	2021-10-01 13:10:00	Success	4.520s	MySQL restore	General	
> test							
▼ Work_With_MySQL_Tables							
Analyze2021September	2021-10-01 13:10:41	2021-10-01 13:11:02	Success	47ms	Analyze table	General	
Check Table State	2021-10-01 13:11:26	2021-10-01 13:11:48	Success	5ms	Check table	General	
Optimize Important Tables October	2021-10-01 13:13:41	2021-10-01 13:13:59	Success	78ms	Optimize table	General	

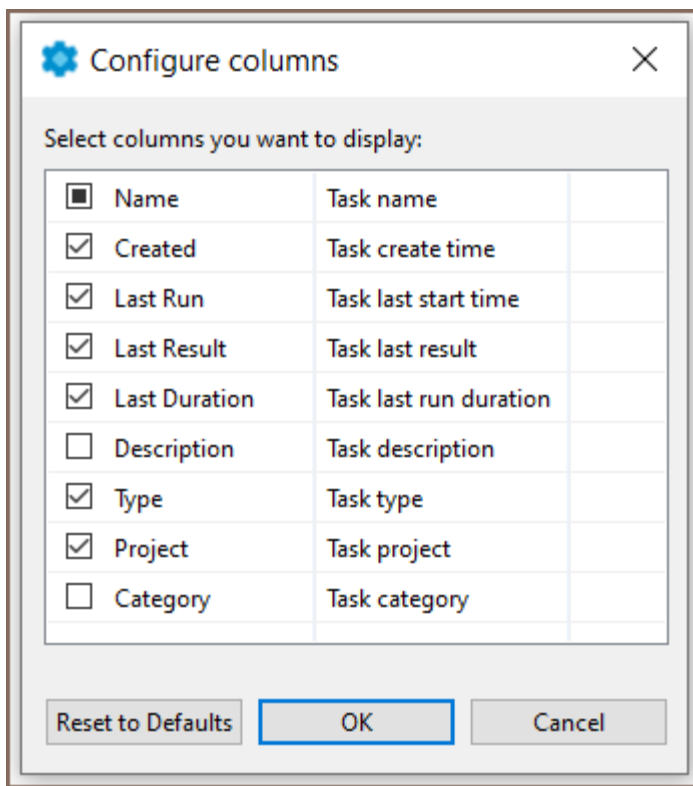
You can configure the Database Tasks View to see more or fewer View columns. Right-click inside the Database Tasks tab and choose the **Configure columns** button from the menu.

Last Duration	Type	Project
N/A	Data export	General
385ms	Data export	General
163ms	Data export	General
100ms	Data export	General
N/A	Data import	General
7.529s	Data import	General
258ms	Data import	General
22.698s	MySQL dump	General
3.482s	MySQL script	General
4.520s	MySQL restore	General
47ms	Analyze table	General
5ms	Check table	General

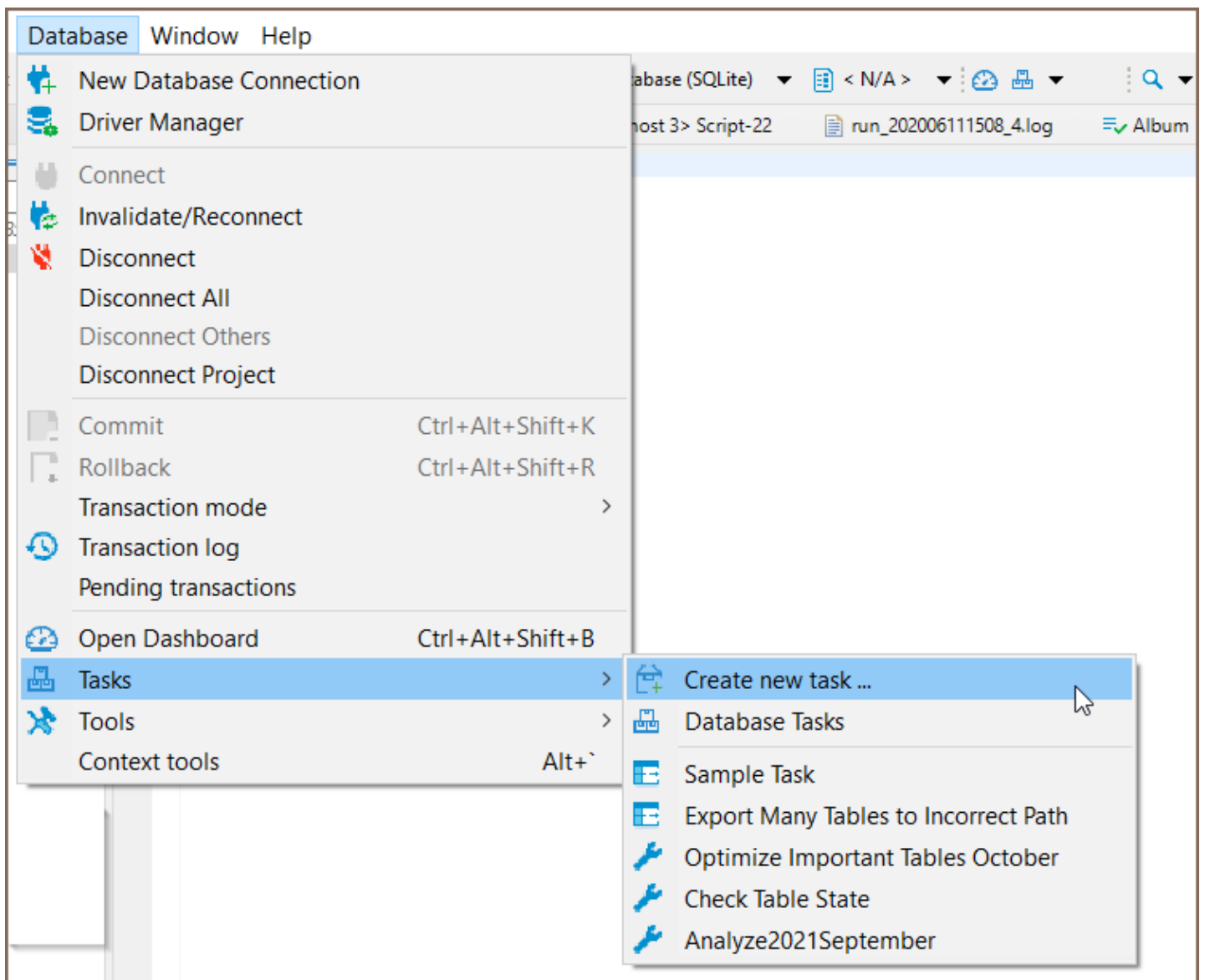
 Run task
 Edit task Enter
 Create new task ...
Copy task
 Delete Delete
 Create new task folder

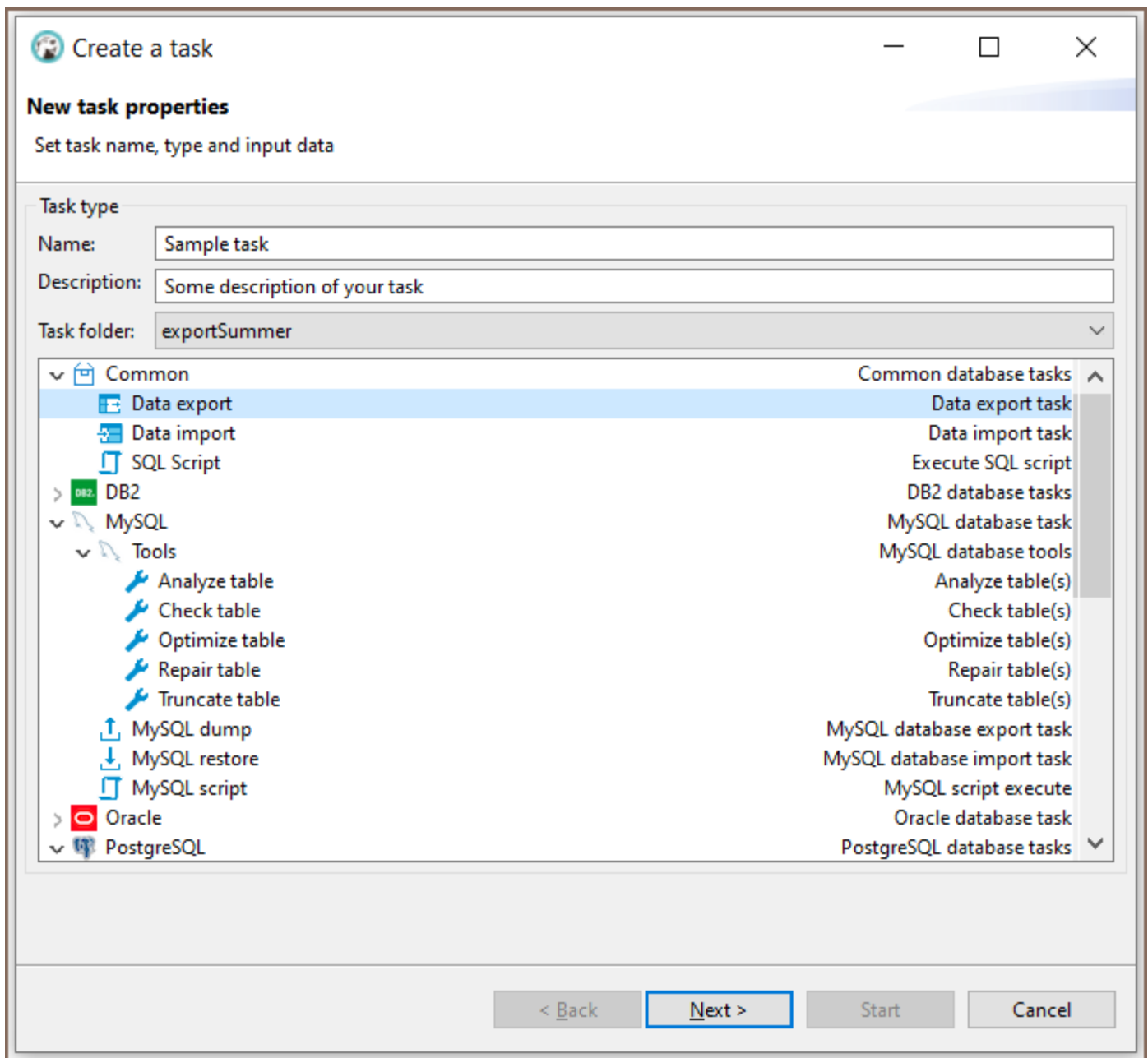
Group tasks by project
Group tasks by category
Group tasks by type

Copy
 **Configure columns**



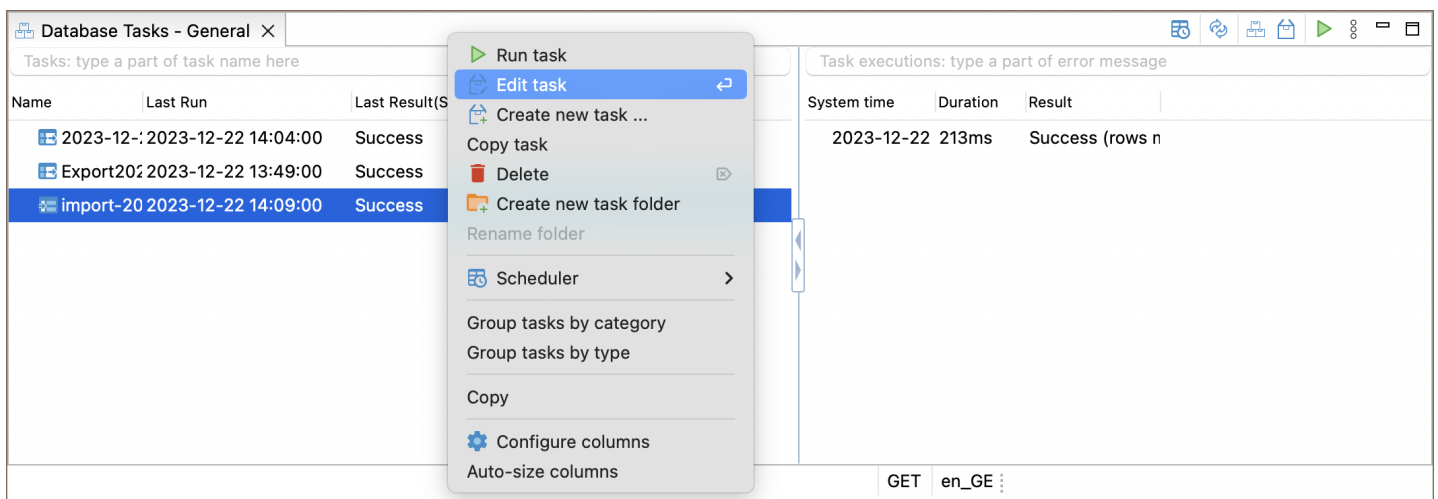
Also, you can create a new task from the main menu `Database -> Tasks -> Create new task...`.





Task editing and execution

From the task view, you can add, edit, remove, and execute saved tasks. You can use the context menu or view tools for these actions.



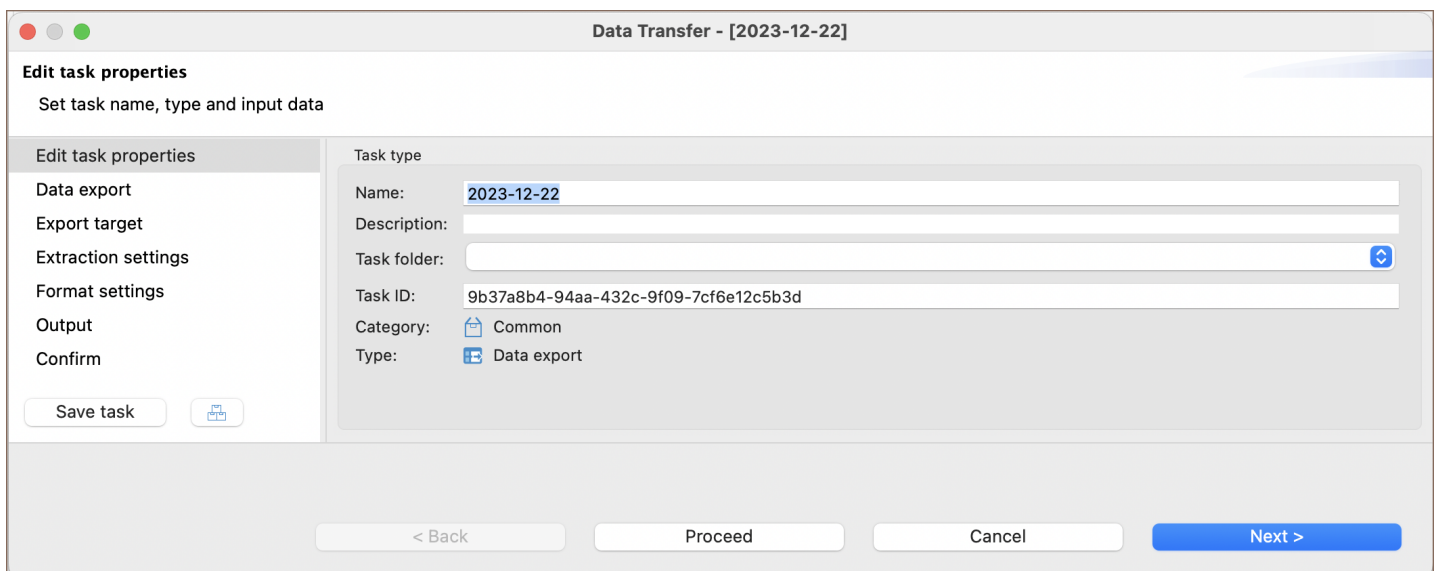
Adjusting task configurations

The process of editing a task is similar to the process of creating a task.

By clicking on **Edit** or double-clicking on a task, the task edit wizard will be opened. In this wizard, you can:

- Change the task properties.
- Change the file format, either output for an export task or input for an import task.
- Alter the set of objects for data transfer.
- Adjust any export/import configuration.

After modifying the task settings, click the **Save task** button. This button is located on the left side of the task configuration wizard.

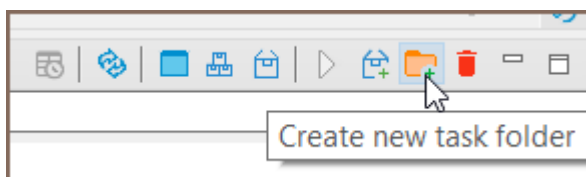
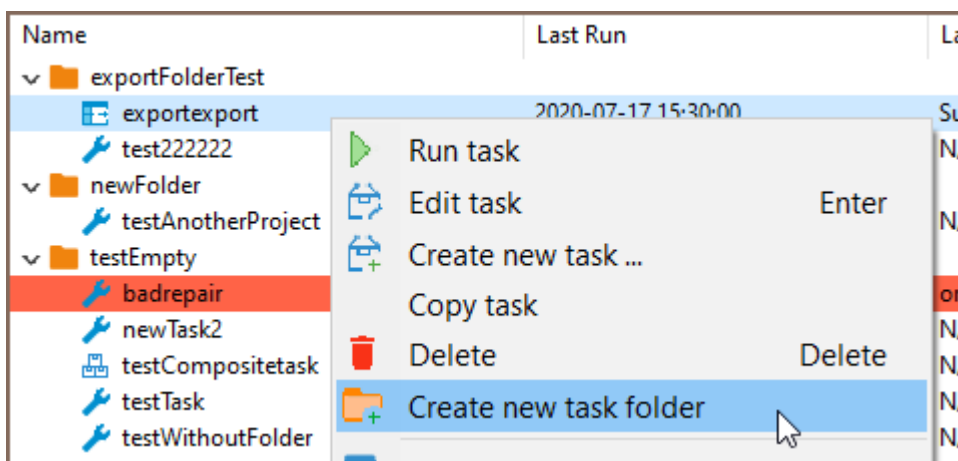


Create a task from task management view

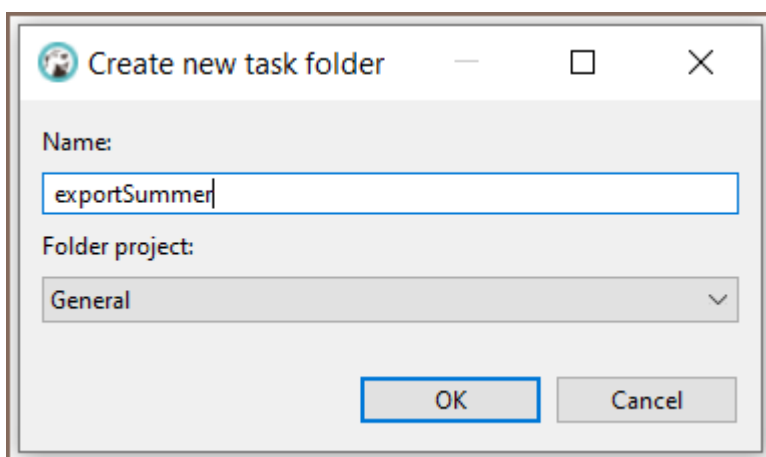
You can create a task from scratch using the tasks view. Open tasks view and click on the **Create new task** button in the View toolbar or in the context menu. In the task wizard, you can choose the task category, task type, task folder, and name. On the next wizard pages, actual tool configuration pages will be shown (they depend on the chosen task type).

Tasks folders

For better structuring, you can store your tasks in the folders. Create folder can also be from the context menu Database Tasks View.



You can choose a project for the new task folder and add a new unique name.



You can change the task folder for an existing task in the task Edit dialog (use button **Back**). Or you can drag your task to another task folder in the View.

The task folder as a task can be deleted with the button **Delete** .

Scheduling tasks

You can schedule tasks for later/regular execution. See the [Task Scheduler](#) article.

Task Scheduler

Note: This feature is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Task Scheduling from the Tasks View](#)

[For Windows Users](#)

[For macOS or GNU and Linux Users](#)

[Viewing Schedule Details](#)

[Windows](#)

[macOS or Linux](#)

[Monitoring Task Execution](#)

[Running tasks from the command line](#)

[Troubleshooting](#)

[User Authentication and Corporate Security](#)

[Overview of Windows Scheduler](#)

[Windows Task Scheduler](#)

[Excluding Legacy Scheduler](#)

[Windows Task Scheduler - Incorrect Date Format](#)

[Applicable to Legacy Scheduler](#)

[macOS - Unable to read or write to crontab](#)

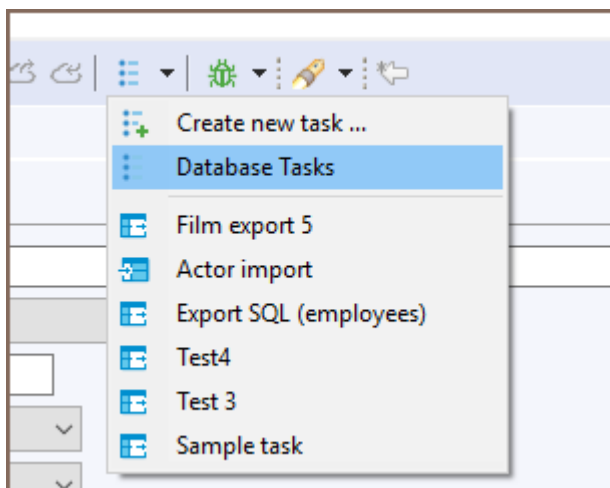
[Tasks from password-protected projects cannot be run](#)

DBeaver can schedule execution for regular tasks. DBeaver supports **Windows Task Scheduler** on Windows and **cron** on macOS and GNU/Linux. In addition, you can manually configure schedulers [using command line](#).

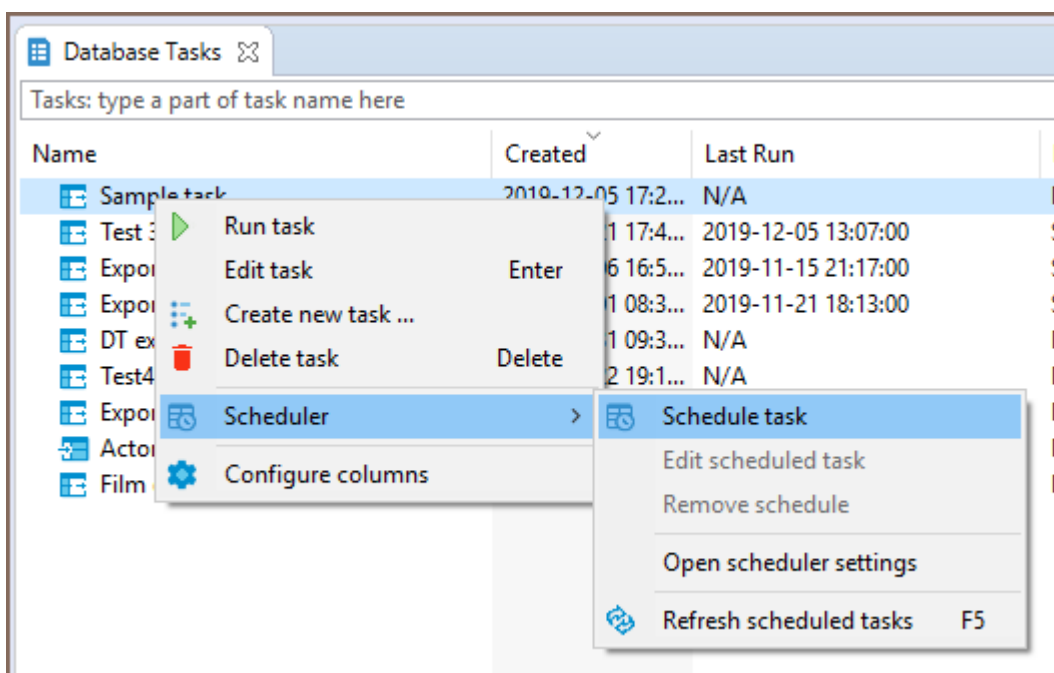
Task Scheduling from the Tasks View

For Windows Users

You can open the tasks view from the main toolbar:



or from the main menu Window. Select a task that you want to schedule in the tasks view and open the context menu:



DBeaver will open the scheduler configuration dialog. You can configure task frequency, recurrence period and start time there:

Schedule task 'Sample task'

Scheduler parameters

Set task 'Sample task' scheduler parameters

Scheduler: Local Windows Task Scheduler

Frequency

☐ Minutely

☒ Hourly

☐ Daily

☐ Weekly

☐ Monthly

☐ One Time

Settings

Start time: 12/ 5/2019 5:38:15 PM

Recur every 2 hours

Schedule Cancel

Schedule task 'Sample task'

Scheduler parameters

Set task 'Sample task' scheduler parameters

Scheduler: Local Windows Task Scheduler

Frequency

☐ Minutely

☐ Hourly

☐ Daily

☐ Weekly

☒ Monthly

☐ One Time

Settings

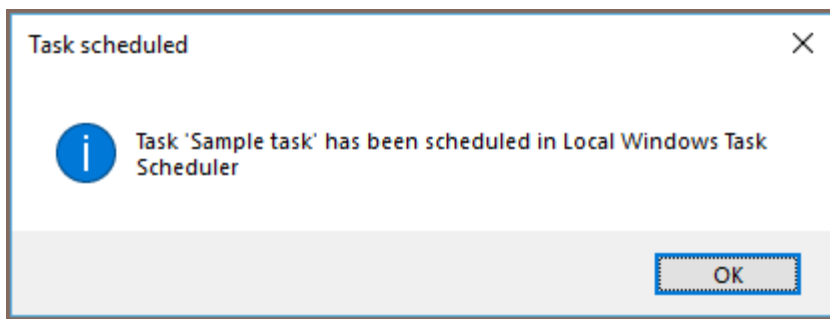
Start time: 12/ 5/2019 5:38:00 PM

Months: ☒ January ☐ July ☒ February ☐ August ☐ March ☐ September ☐ April ☐ October ☐ May ☐ November ☐ June ☐ December ☐ <All months>

Days: ☒ 1 ☐ 9 ☐ 17 ☐ 25 ☐ 2 ☐ 10 ☐ 18 ☐ 26 ☐ 3 ☐ 11 ☐ 19 ☐ 27 ☐ 4 ☐ 12 ☐ 20 ☐ 28 ☐ 5 ☐ 13 ☐ 21 ☐ 29 ☐ 6 ☐ 14 ☐ 22 ☐ 30 ☐ 7 ☐ 15 ☐ 23 ☐ 31 ☐ 8 ☐ 16 ☐ 24 ☐ Last

Update Cancel

To schedule the task, click on the Schedule button. If everything is OK, you will see the confirmation dialog:



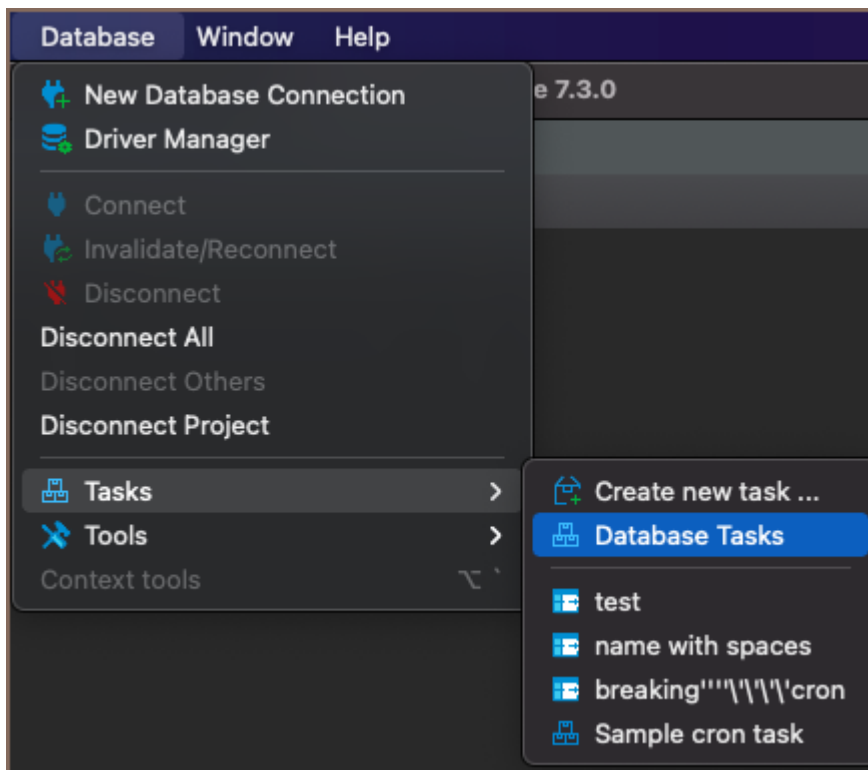
If anything goes wrong, you will see an error message dialog. You can view error details in the [Error Log](#) view.

You can change the scheduler settings at any moment by choosing Edit scheduled task command from the context menu. You can also cancel the schedule by clicking on Remove schedule.

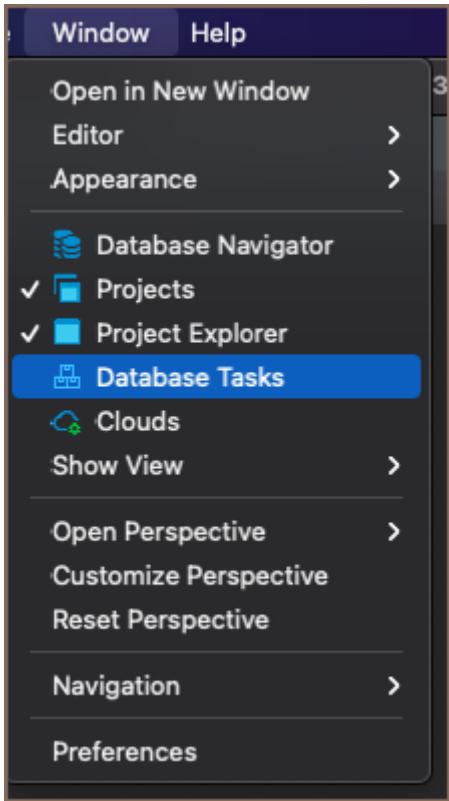
For macOS or GNU and Linux Users

You first need to open the tasks view. There are three ways to do that:

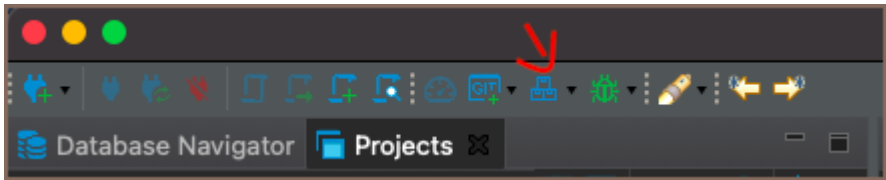
1. Database -> Tasks -> Database Tasks



2. Window -> Database Tasks

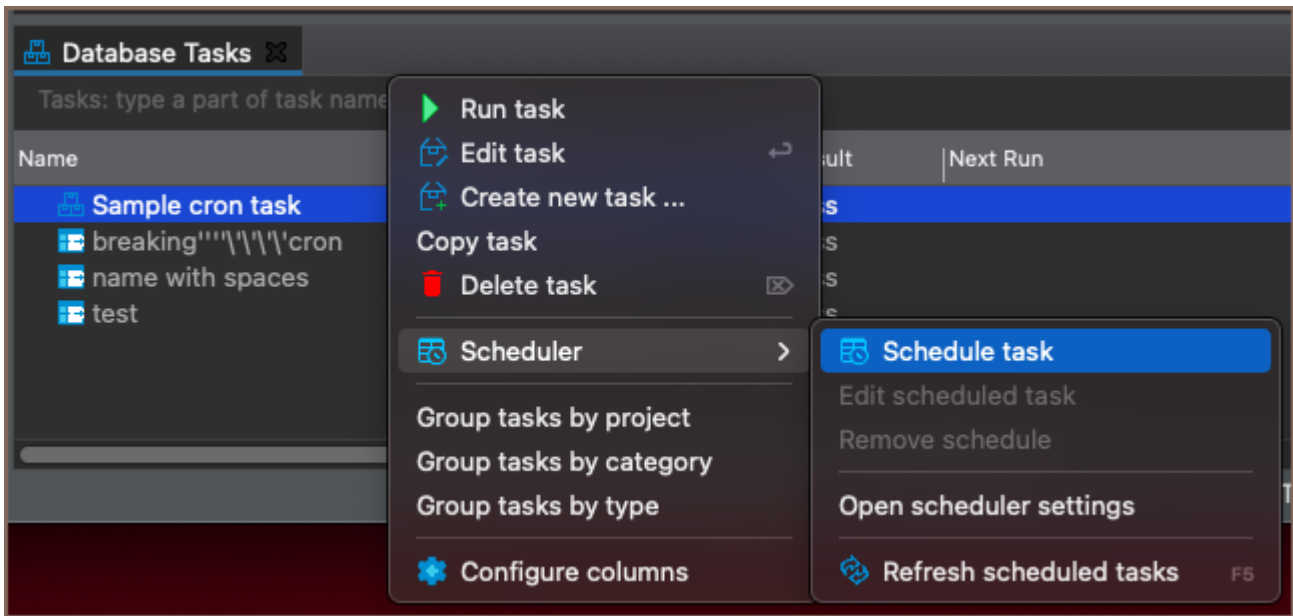


3. Click on 'Show View (Database Tasks)' icon

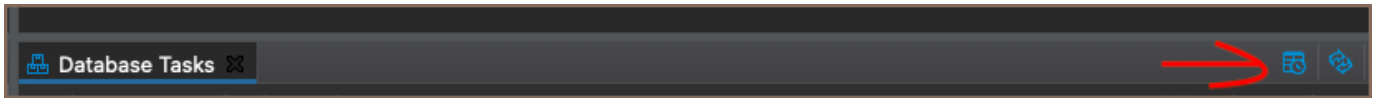


Select a task you want to schedule in the tasks view. To open the scheduler dialog, either:

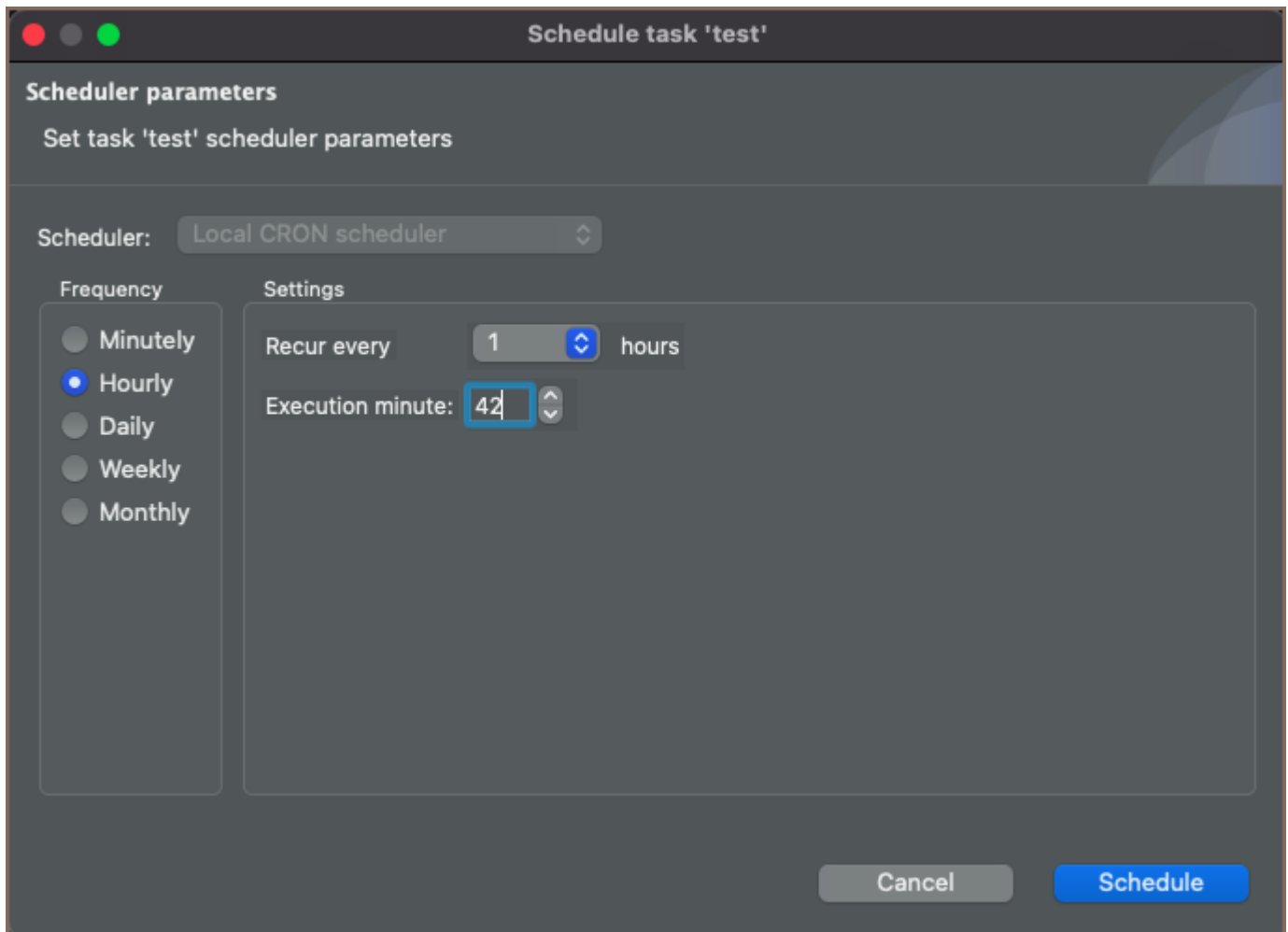
1. Open the context menu with right-click -> Scheduler -> Schedule task



2. or click on the 'Schedule task' icon

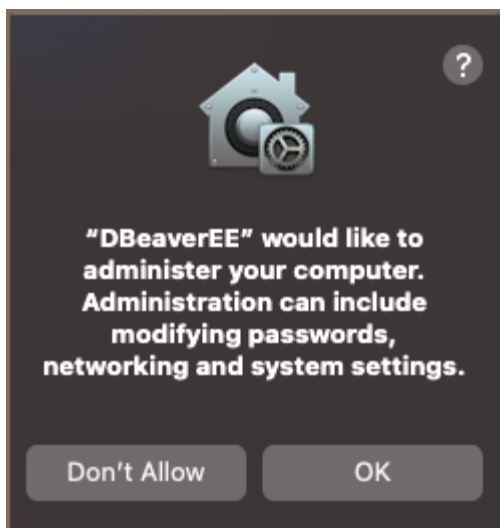


DBeaver will open the scheduler dialog. It has a lot of similarities with the corresponding dialog in Windows, but unfortunately, there are fewer settings on macOS and GNU/Linux due to the limitations of `cron`. For instance, when configuring an hourly task, you can only choose the minute at which the scheduler executes the task. In the example below, the task executes at 1:42 PM, 2:42 PM, 3:42 PM, and so on:



There is also no start date option and, in the case of minutely tasks, no start time either. The scheduler will execute the task at the specified time, but there are no guarantees when the execution starts. It is also worth pointing out that even though you can specify the seconds in the start time selector, the scheduler will ignore them. Even though we try to be compliant with as many cron implementations as possible, most cron implementations do not support this type of granularity.

On macOS 10.15 or newer versions, when scheduling a task for the first time, you will be prompted with something like this:



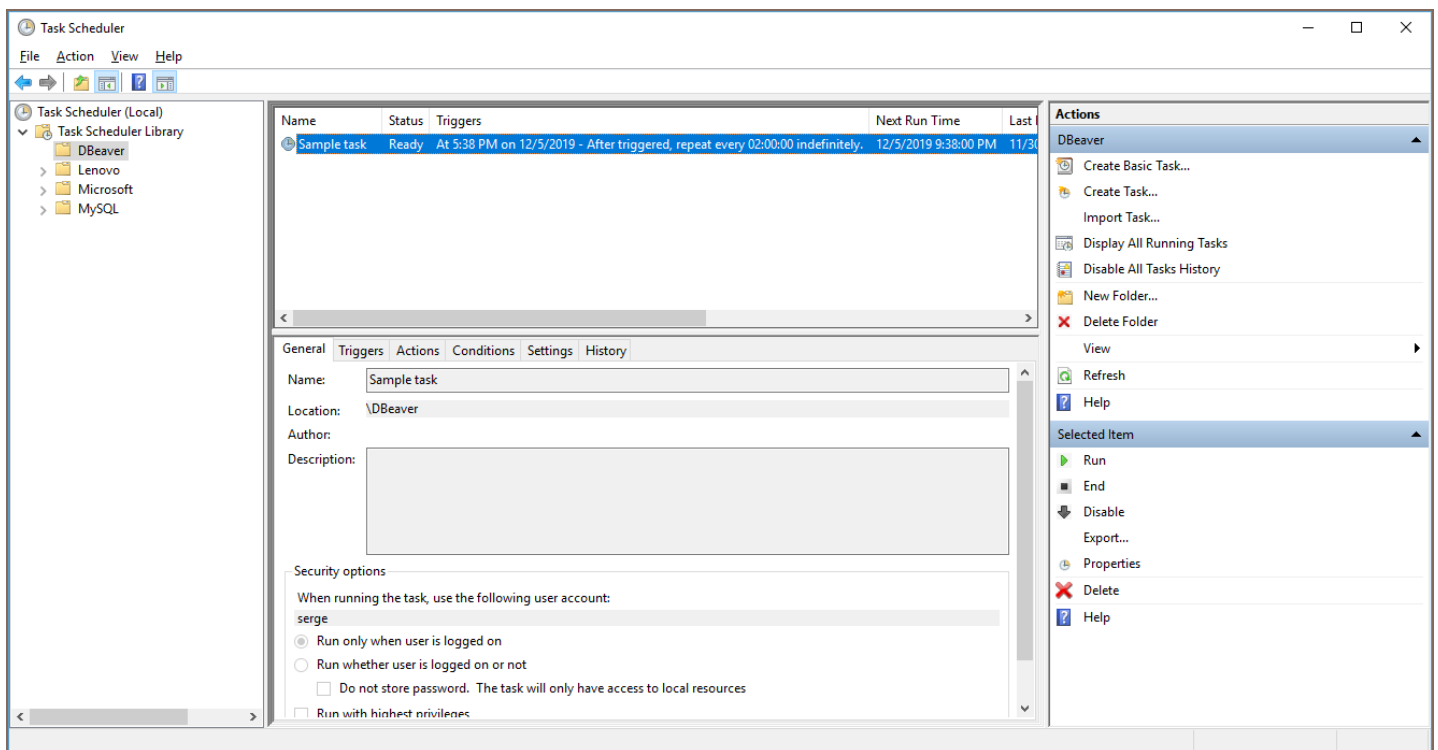
Click 'Yes' to proceed. The reason for that prompt is that macOS considers the `cron` settings (crontabs) to be system settings, and DBeaver will not be able to change them without your permission.

After that, you will see the confirmation message. Just like in Windows, you can change the scheduler settings at any moment by choosing the 'Edit scheduled task' command from the context menu, or cancel the schedule by clicking on 'Remove schedule'.

Viewing Schedule Details

Windows

You can see and change the scheduled task details in the Windows Task Scheduler. Click on the Open scheduler settings command in the task view context menu:



DBeaver stores all tasks in a folder called **DBeaver**.

macOS or Linux

You can take a look at the crontab DBeaver uses to schedule tasks in **crontab** by clicking the 'Open scheduler settings' command in the task view context menu. You can also do it in the terminal by using the command **crontab -l**. Although you can also edit the crontab by using **crontab -e**, we strongly do not recommend it.

Monitoring Task Execution

You can look through the task execution logs on the right side of the tasks view. By double-clicking on a task run item, you can see the full log with all details, errors, and warnings:

The screenshot shows the DBeaver interface. The top window displays a log of database operations, including schema loading, data transfer initialization, and a SQL query execution. The bottom window shows a table of database tasks and their execution history.

Name	Created	Last Run	Last Result	Type	Next Run	Pr	Time	Duration	Result
Sample task	2019-12-05 17:2...	N/A	N/A	Data export	12/5/2019 9:...	db	2019-11-15 21:17:00	1.225s	Success
Test 3	2019-11-21 17:4...	2019-12-05 13:07:00	Success	Data export		db	2019-11-15 21:14:00	1.216s	Success
Export payment data (daily)	2019-11-06 16:5...	2019-11-15 21:17:00	Success	Data export		Te	2019-11-15 21:13:00	1.350s	Success
Export Sample	2019-11-01 08:3...	2019-11-21 18:13:00	Success	Data export		Ge	2019-11-15 21:08:00	1.284s	Editors not sup...
DT example 1	2019-10-31 09:3...	N/A	N/A	Data export		Ge	2019-11-15 21:00:00	1.273s	Editors not sup...
Test4	2019-09-22 19:1...	N/A	N/A	Data export		db	2019-11-15 20:54:00	1.385s	Editors not sup...
Export SQL (employees)	2019-09-22 09:4...	N/A	N/A	Data export		db	2019-11-15 20:52:00	1.313s	Editors not sup...
Actor import	2019-09-21 08:3...	N/A	N/A	Data import		db	2019-11-15 20:38:00	1.309s	Editors not sup...
Film export 5	2019-09-20 20:5...	N/A	N/A	Data export		db	2019-11-15 20:30:00	1.271s	Success
							2019-11-11 08:48:00	3.307s	Success
							2019-11-06 16:56:00	3.187s	Success

DBeaver keeps the task run logs in the [workspace](#) directory, subfolder `.metadata/task-stats`.

Running tasks from the command line

The task scheduler uses the DBeaver [command line](#) interface to perform task executions. Command-line parameter

`-runTask TASK_ID` launches saved task executions (immediately). `TASK_ID` has the form `@projectName:taskName`.

You can omit the project name part if you have only one project in your workspace. In Windows, you can use the

`dbeaver-cli` executable to run tasks. Please note that if you use `dbeaver` executable (for any reason), you will

need to add the command line parameter `-nosplash` to avoid a splash screen appearance.

Troubleshooting

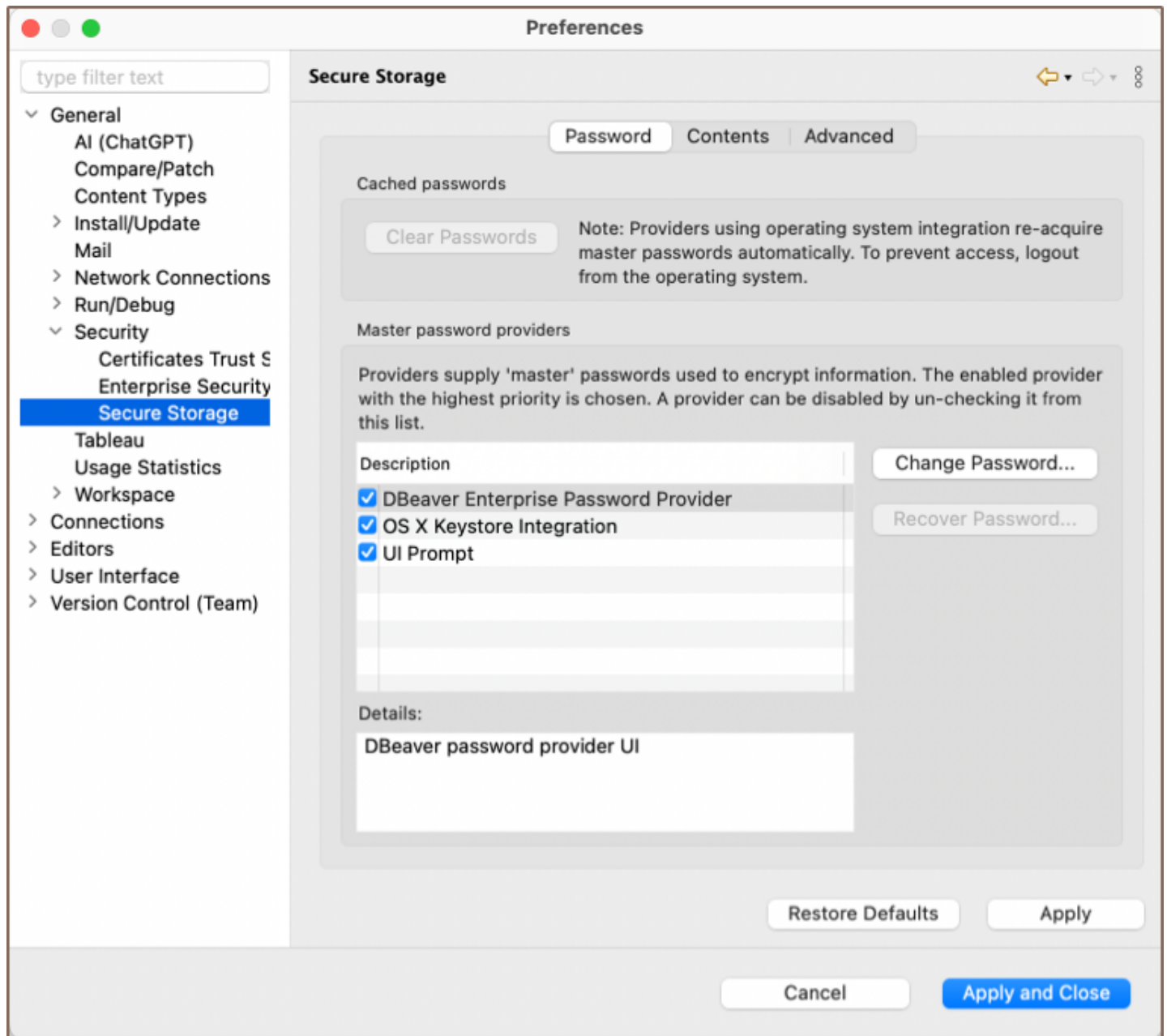
User Authentication and Corporate Security

When you

- Enable enterprise security for database credentials
- Use Mail server authentication to send data transfer results over email

Then these settings won't be accessible in scheduled tasks by default. But you can use default OS-specific encryption for that. Master password won't be used, so it is a bit less secure configuration (credentials are still encrypted, but they may be stolen if hacker will gain access to your computer).

To enable this you need to:



- Go to Preferences->General->Security->Secure Storage
- Disable "DBeaver Enterprise Password provider" and "UI prompt"
- Update credentials you want to use for scheduled task:

- For Mail profile you can open mail settings and click Ok
- For database credentials got to **Window -> Preferences -> General -> Security -> Enterprise security** and toggle "Use secure password storage" on and off to trigger stored credentials re-encryption

Overview of Windows Scheduler

There are two implementations of the Windows scheduler present:

1. CLI-based (**Legacy**): uses `schtasks.exe` to communicate with the scheduler; sensitive to locale-dependent data, such as Unicode names and date-time format.
2. COM-based (**New**): uses COM API to communicate with the scheduler; more flexible and provides more features than the CLI version.

COM-based implementation is used by default, starting from the 21.1 version of DBeaver EE.

Windows Task Scheduler

Excluding Legacy Scheduler

If you encounter an error in Windows which contains the following text:

```
com.sun.jna.platform.win32.COM.COMException ,
```

do the following:

1. Open the file `dbeaver.ini`.

Tip: Detailed instructions on finding `dbeaver.ini` are available in [our article](#).

2. Place the line `-Ddbeaver.scheduler.windows.legacy=true` below the `-vmargs` line.

Windows Task Scheduler - Incorrect Date Format

Applicable to Legacy Scheduler

If you encounter an error in Windows which looks like this:

```
ERROR: Invalid Start Date (Date should be in %some_format% format). ,
```

do the following:

1. Open the file `dbeaver.ini` in the directory with your DBeaver installation
2. Place the line `-Ddbeaver.scheduler.windows.dateFormat=%some_format%` (where %some_format% is a format from the error message) below the `-vmargs` line.

This flag is available starting from the 7.3.4 EA version of DBeaverEnterprise and might be removed in the future.

macOS - Unable to read or write to crontab

When scheduling tasks on macOS 10.15 or newer versions, the OS will prompt you to elevate DBeaver's permissions to administer your computer. If you do not grant these permissions, DBeaver will fail to schedule your tasks with an error `Unable to read or write to crontab`. To bypass this, simply restart DBeaver and try to schedule the task again. The operating system will prompt you to elevate the permissions again. If macOS never prompted to do that in the first place, you could grant `Full disk access` permissions in the macOS settings. Here is how to do that:

1. Open `System Preferences`.
2. Click on `Security & Privacy`.
3. Choose the `Privacy` tab.
4. Choose the `Full Disk Access` folder.
5. Unlock the preferences lock to the bottom if it is locked.
6. Click the + button.
7. Select DBeaverEE in the file picker that opens.
8. Click `Open`.
9. Close the lock.

Tasks from password-protected projects cannot be run

You need to pass a password for one or more projects via the command-line interface.

To do so, you need to set the `dbeaver.project.password` parameter in [the external configuration file](#) like so:

```
# You can specify a single password for all projects:
dbeaver.project.password=p4$$w0rd

# Otherwise, you can specify a list of passwords for given projects:
dbeaver.project.password=@General:p4$$w0rd,@Other:12345
```

The syntax for a single entry is `@ <name of the project> : <password of the project>`; others are separated by the `,` symbol.

Please note that `@` and `:` symbols are mandatory.

Composite Tasks

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Creating a composite task](#)

[Setting up a composite task](#)

As the name suggests, the *composite task* is a type of task that consists of other tasks. Just like the other type of tasks, the composite tasks can be scheduled via [Task Scheduler](#). Let's take a look at what they can offer.

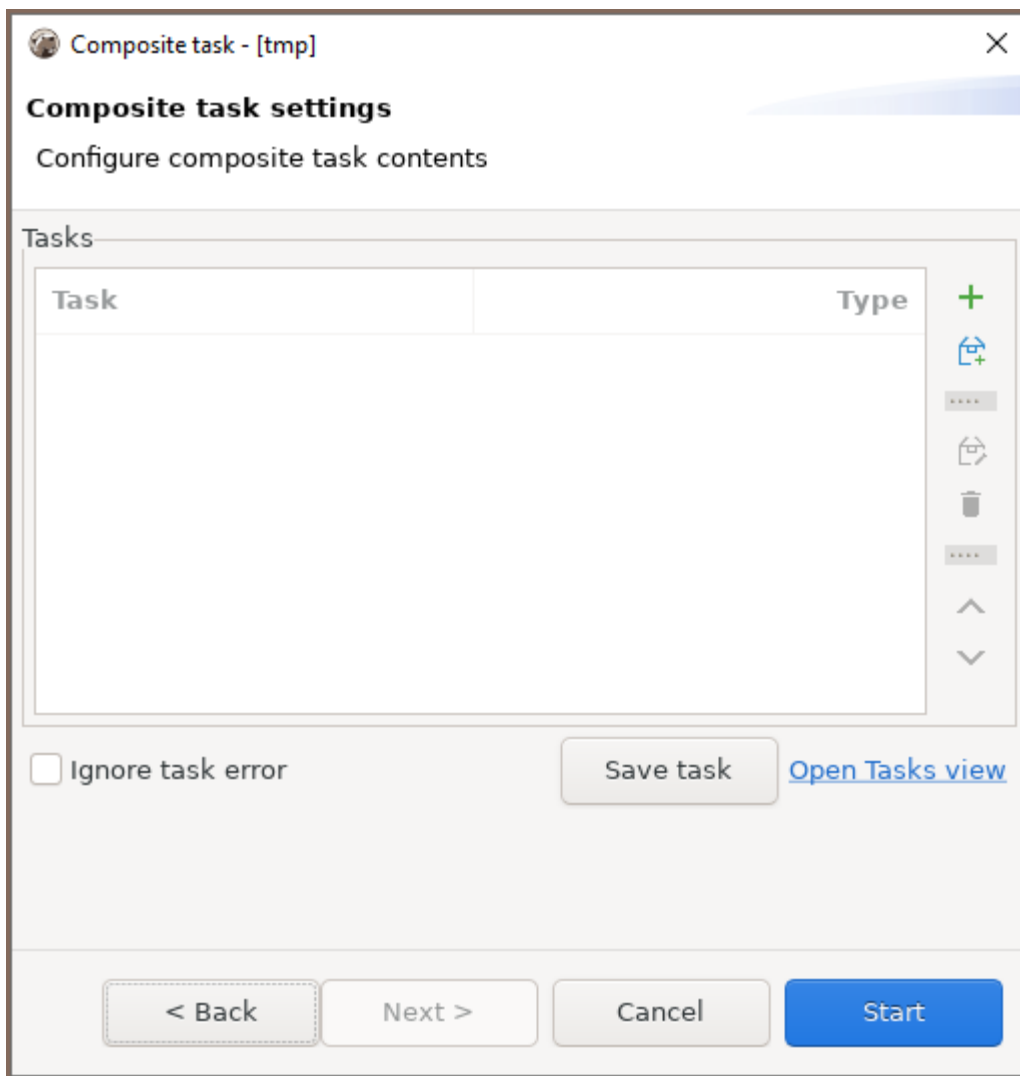
Creating a composite task

The first thing we need to open is the *Create a task* dialog. You can do it in multiple ways:

- From the context menu in the database navigator -> Tools -> Create new task... -> Composite task
- By clicking Database -> Tasks -> Create new task... -> Composite task
- From the context menu in the Database Tasks view.

Choose *Composite task*, enter the task name, description (optional), and hit *Next*.

You will be presented with the following dialog:



Setting up a composite task

When creating a composite task, you need to specify which tasks the composite task consists of.

This can be done:

1. By adding an existing task. To do that, click the button with the plus sign
2. By creating a new task and adding it simultaneously. To do that, click the button below the aforementioned button with the plus sign
3. By drag-and-dropping a task from the _Database tasks panel.

As a side note, you can add a composite task to your new composite task.

You can edit tasks in the same dialog, delete a task from a composite task, and change the execution order.

There is also a very important checkbox, *Ignore task error*. The tasks from the *composite task* are executed in the order they appear in the settings dialog. Executing a task from a *composite task* might produce an error that will block the next tasks from proceeding. The *Ignore task error* checkbox can be used to bypass this behavior.

GIT integration

Note: This feature is available in Community, [Lite](#), [Enterprise](#) and [Ultimate](#) editions only. For Git functionality specific to the Team Edition, please refer to [Team Edition Git documentation](#). For users of the Community edition to access **Git** feature, first [install the required extension](#).

Table of contents

[Overview](#)

[Create and share a Project repository](#)

[Cloning Existing Repositories](#)

[Updating remote repository with changes](#)

[Pulling new changes from remote branch](#)

[Installing the Git extension](#)

Overview

DBeaver supports full integration with Git, a version control system. This integration allows users to maintain project configurations, scripts, diagrams, bookmarks, and other artifacts in a Git repository. It facilitates project management by enabling version control and collaboration within DBeaver.

In this guide, we will cover the foundational Git operations in DBeaver, such as creating and cloning repositories, committing changes, and pulling updates.

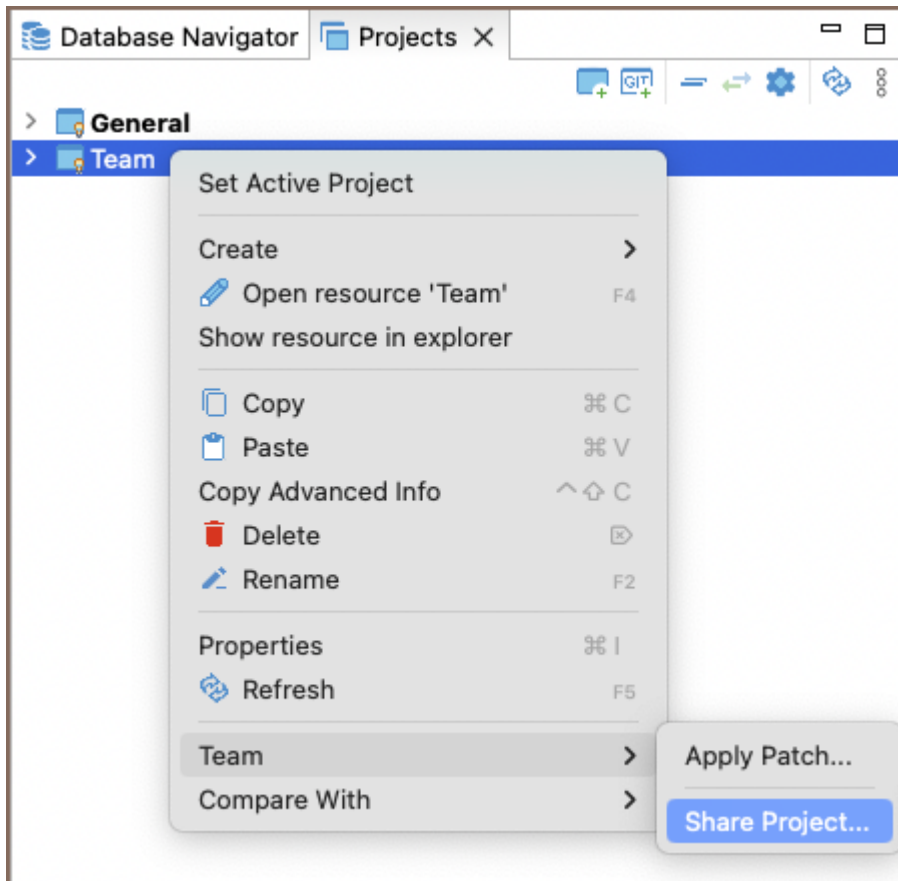
DBeaver's Git support is built on the Eclipse platform. For advanced Git functionalities like branching, and merging conflicts, refer to the [Eclipse EGit documentation](#).

Important: DBeaver's Git integration is designed to work only with GitHub.

Create and share a Project repository

To share an existing project, follow these steps:

1. Navigate to the **Projects tab**, select the project you want to share, and choose **Team -> Share project** from the context menu.



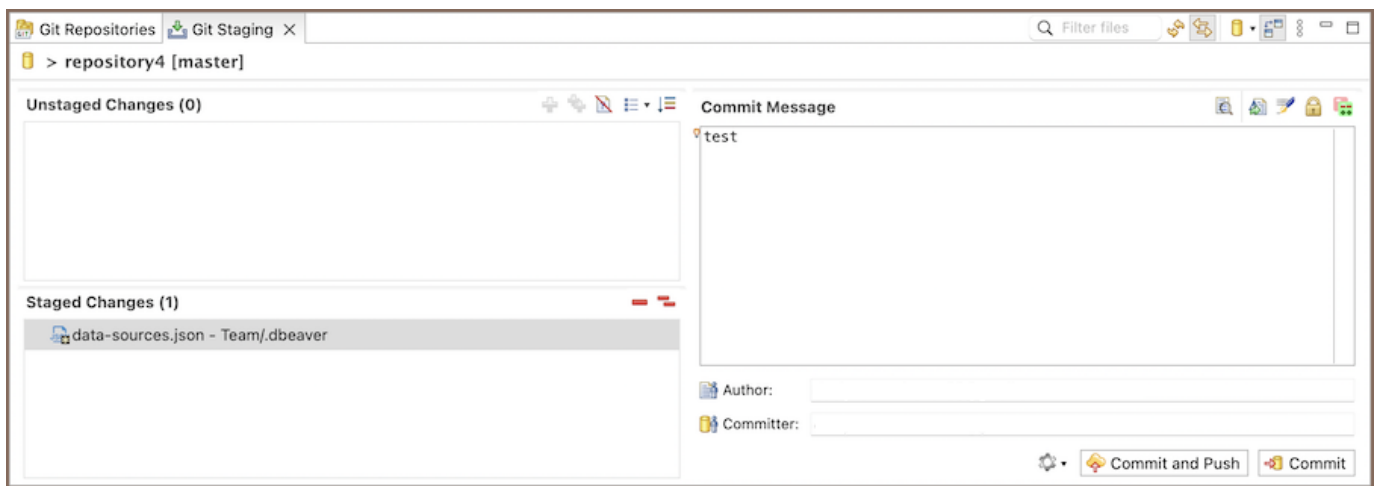
2. Create a new local repository.
3. On the repository page on GitHub copy the repository URL/SSH.

Tip: For more details on creating GitHub repository you can refer to the official [GitHub documentation](#).

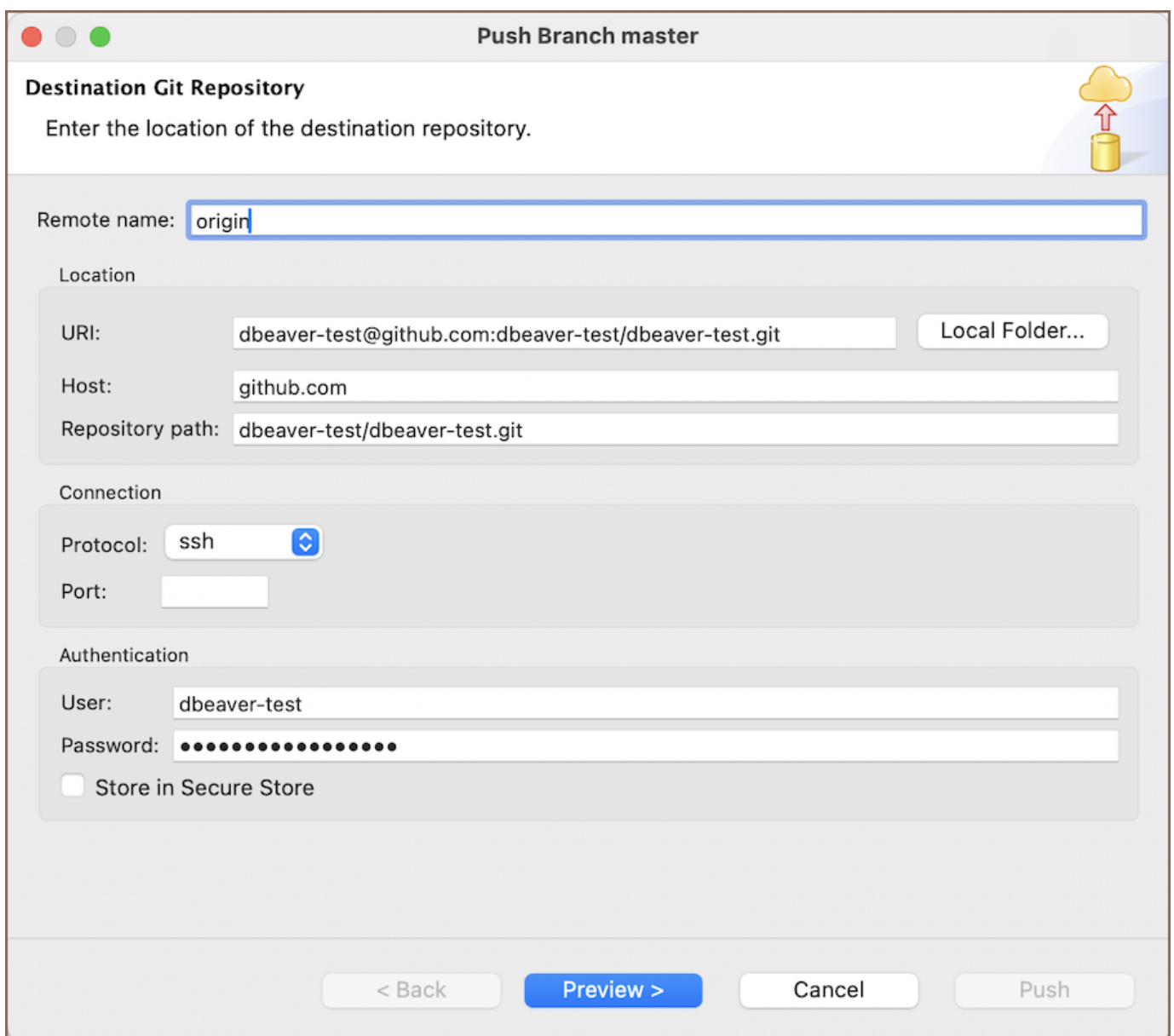
4. Generate a GitHub **access token** in **Settings -> Developer Settings -> Personal access tokens -> Tokens (classic) -> Generate new token (classic)**. Select all options in the **'repo'** section and **'read:org'** under **admin:org**.

Important: DBeaver supports all types of Git tokens, including fine-grained and Personal access tokens (classic).

5. In DBeaver, right-click the project node and choose **Team -> Commit...** or use the shortcut **Ctrl+#**. Stage the files you wish to include in the commit by clicking **Add**. Enter a clear and concise commit message, then click **Commit** to finalize your changes.




6. Use the **Push HEAD** button and paste the GitHub repository URL/SSH and provide your GitHub **User** and **access token** as the **Password**.



7. Push changes to the remote repository by using the **Push** button.

Cloning Existing Repositories

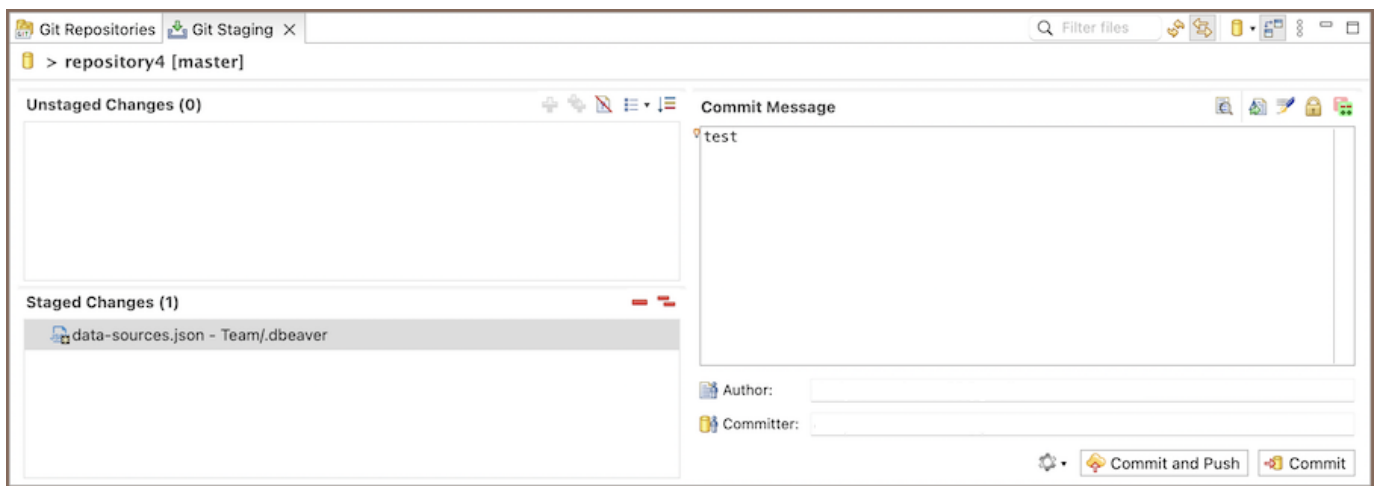
To clone an existing repository, ensure you have a GitHub account and follow these steps:



1. Click on **Create Project from Git**  at the **Projects** Tab or navigate through **File -> Git -> Create Project from Git** in the [Main menu](#).
2. Obtain the remote repository's URL from its GitHub page by clicking the **Code** button and copying the URL /SSH.
3. Generate a GitHub **access token** in **Settings -> Developer Settings -> Personal access tokens -> Tokens (classic) -> Generate new token (classic)**. Select all options in the **'repo'** section and **'read:org'** under **admin:org**.
4. In the wizard, enter the remote repository location. Use your GitHub **User** (it is commonly **git**) and **access token** instead of a **Password**.
5. Choose which branches to clone from the remote repository.
6. Define the local file system location for the repository storage and set initial configurations.
7. Select the desired wizard type for the project setup and complete the cloning process. The project will now be visible in the [Project Explorer](#).

Updating remote repository with changes

After making changes to your project, such as creating, deleting, or modifying connections, scripts, or ERDs, follow these steps to update your remote repository:

1. Right-click on the project in the **Projects** tab and select **Team -> Commit** to open the commit interface.



2. In the **Git Staging** tab at the bottom, you will see **Unstaged Changes**. Use the **Add Selected files to the index**  or **Add all files to the index**  button to stage the changes you want to commit. Enter a commit message and then execute the commit using the **Commit** button or **Ctrl+Enter**. The commit will then be visible in the **History** tab (**Window -> Show view -> Other -> Version Control(Team) -> History**).
3. To push your commit to the remote repository, you can use the **Commit and Push** button. Alternatively, to push changes manually, navigate through the project's context menu (**Team -> Remote -> Push**) or from the **History** tab by right-clicking on the desired commit and selecting **Push commit**. This updates your git repository with the new changes.

Pulling new changes from remote branch

To incorporate new changes from the remote branch into your local project, follow these steps:

1. Right-click on the project in the **Project** tab and select **Team -> Pull**. Alternatively, if you are in the Git Repositories view, you can right-click on the repository and select **Pull**. This action fetches and merges any new changes from the branch your local repository is tracking into your current branch.
2. If you have not made any local changes, or if you wish to discard your local changes and synchronize with the remote branch, you can use the **Team -> Reset** command. This will reset your local repository to the state of the remote repository, discarding any local changes that have not been committed.

This process ensures your project is up to date with the latest changes from the remote branch, facilitating collaboration and avoiding potential merge conflicts.

Installing the Git extension

Community Edition users can manually install the Git extension by following these steps:

1. Via Eclipse Marketplace:

- Visit the Eclipse Marketplace [URL](#)
- Follow the instructions to install the Git support extension directly through your DBeaver installation.

2. Via P2 Repository:

- Open DBeaver, and navigate to **Help -> Install New Software**.
- Enter the P2 repository [URL](#) in the **Work with** field.
- Select the Git extension from the list of available software and follow the on-screen instructions to complete the installation.

After installation, restart DBeaver for the changes to take effect, and you will have access to Git functionality within your DBeaver environment.

Tableau integration

Note: This feature is available in [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Overview](#)

[Powering Tableau with DBeaver](#)

[Easy connection to existing databases](#)

[Working with custom SQL queries](#)

[Filling profile for Tableau connection](#)

[Adding a Tableau Profile in DBeaver](#)

[Creating a database connection](#)

[Associate DBeaver and Tableau connection](#)

[Editing Custom SQL Queries](#)

Overview

[Tableau](#) is a leading data visualization and analytics platform that empowers you to connect, visualize, and share data in a way that makes it easier to understand. With its intuitive drag-and-drop interface, Tableau enables users to create interactive and shareable dashboards, providing insights into complex data sets.

Having a registered Tableau account is a prerequisite for those looking to leverage Tableau's capabilities. This article explores how DBeaver integrates with Tableau, enhancing the data manipulation process.

Powering Tableau with DBeaver

DBeaver's integration is compatible with both the On-premise and Cloud versions of Tableau, offering the following key features:

Easy connection to existing databases

DBeaver's integration with Tableau allows you to connect effortlessly to databases already present in Tableau. This seamless connection enables you to access and manage data without complex configurations or additional tools.

[Learn more about creating a connection.](#)




Working with custom SQL queries

A standout feature of DBeaver's integration with Tableau is the ability to edit custom SQL queries stored in Tableau. You can modify these queries directly within DBeaver, providing a streamlined workflow for advanced data analysis. [Learn more about editing custom SQL queries.](#)

Filling profile for Tableau connection

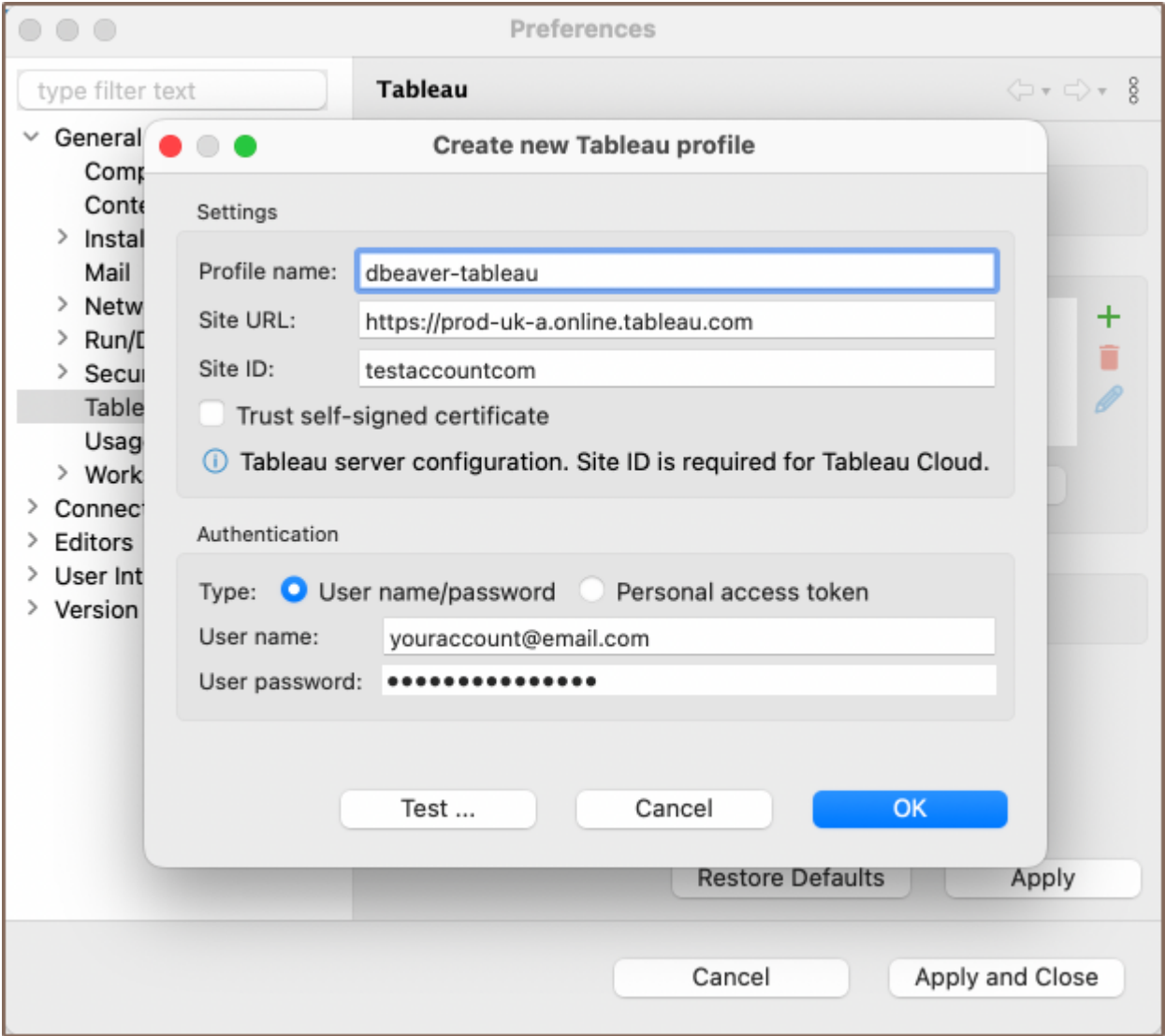
To configure Tableau credentials in DBeaver, follow these steps:

1. Navigate to **Window -> Preferences -> General -> Tableau**.
2. In the Tableau settings, you will find the following options:

Option	Description
Enable Tableau Integration	Enables Tableau in connections and in UI.
Tableau Profile	Manage Tableau profiles. User profiles are displayed here, and you can work with them using the buttons:  - Add profile  - Remove profile  - Edit profile
Test Connection	Test the connection between DBeaver and Tableau.
Automatically associate Tableau connections	Automatically associate Tableau connections with DBeaver data sources. Learn more about associate connection.

Adding a Tableau Profile in DBeaver

When you click on **Add Profile**, a **Create new Tableau profile** window will appear. This is how you fill out the profile:




Field	Description
Profile name	The name of the profile in DBeaver (choose a convenient name for yourself).
Site URL	The URL should look like this: <code>https://prod-uk-a.online.tableau.com</code> .
SITE ID (required for Tableau Cloud)	Enter the correct <code>SITE ID</code> used during Tableau registration (e.g., copy <code>testaccountcom</code> from your Tableau home page URL like <code>https://prod-uk-a.online.tableau.com/#/site/testaccountcom/home</code>).
Trust self- signed certificate	Allows not to validate the server certificate. This option is useful if you are using a self-signed certificate and need it to pass validation. For more information on managing certificates, you can refer to our article about trust stores .
Authentication	Use the credentials of the account you use (login/password or token name/value).

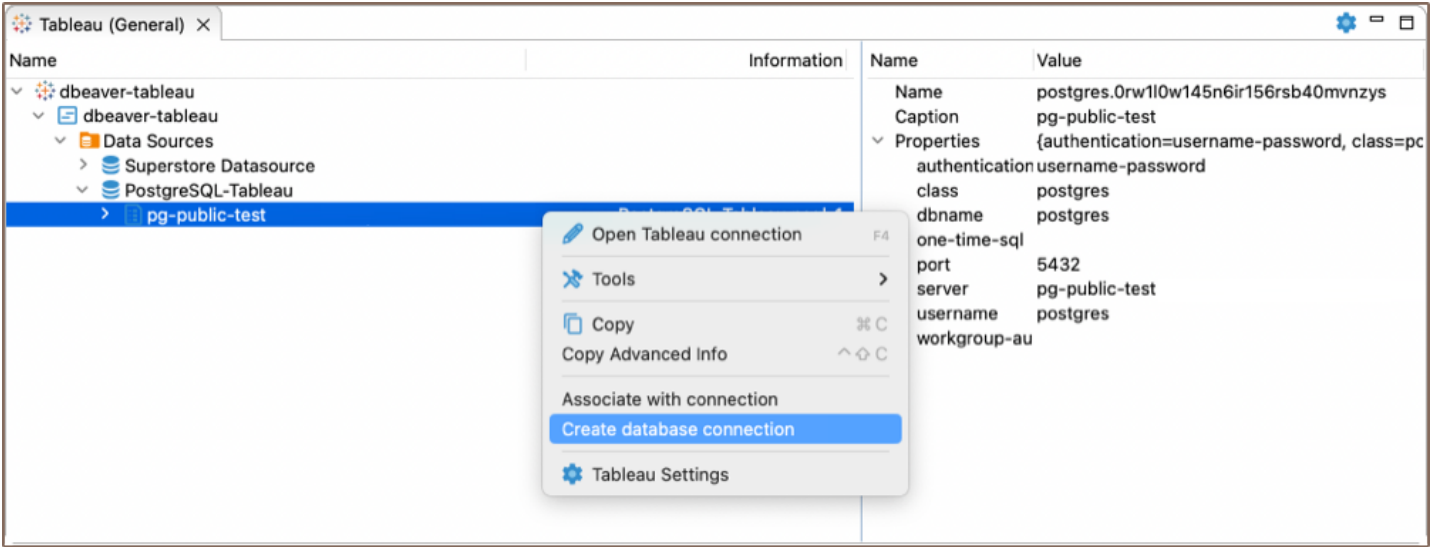
After filling out the profile details, you can test the connection to Tableau to ensure that everything is set up correctly. This step verifies that the provided information is accurate and that DBeaver can successfully communicate with Tableau.

Creating a database connection

Important: Before creating a connection in DBeaver, a connection must already be configured in Tableau. More details can be found [in the official Tableau documentation](#).

Creating a connection between DBeaver and Tableau is a straightforward process:

1. In the DBeaver [toolbar](#), click on the **Show View (Tableau)** button . The Tableau view will open in the lower part of DBeaver, displaying your profiles.
2. In your profile, select the desired database, right-click, and choose **Create Database Connection**.
3. DBeaver will prompt you to name the connection, and it will appear in the [Database Navigator](#).

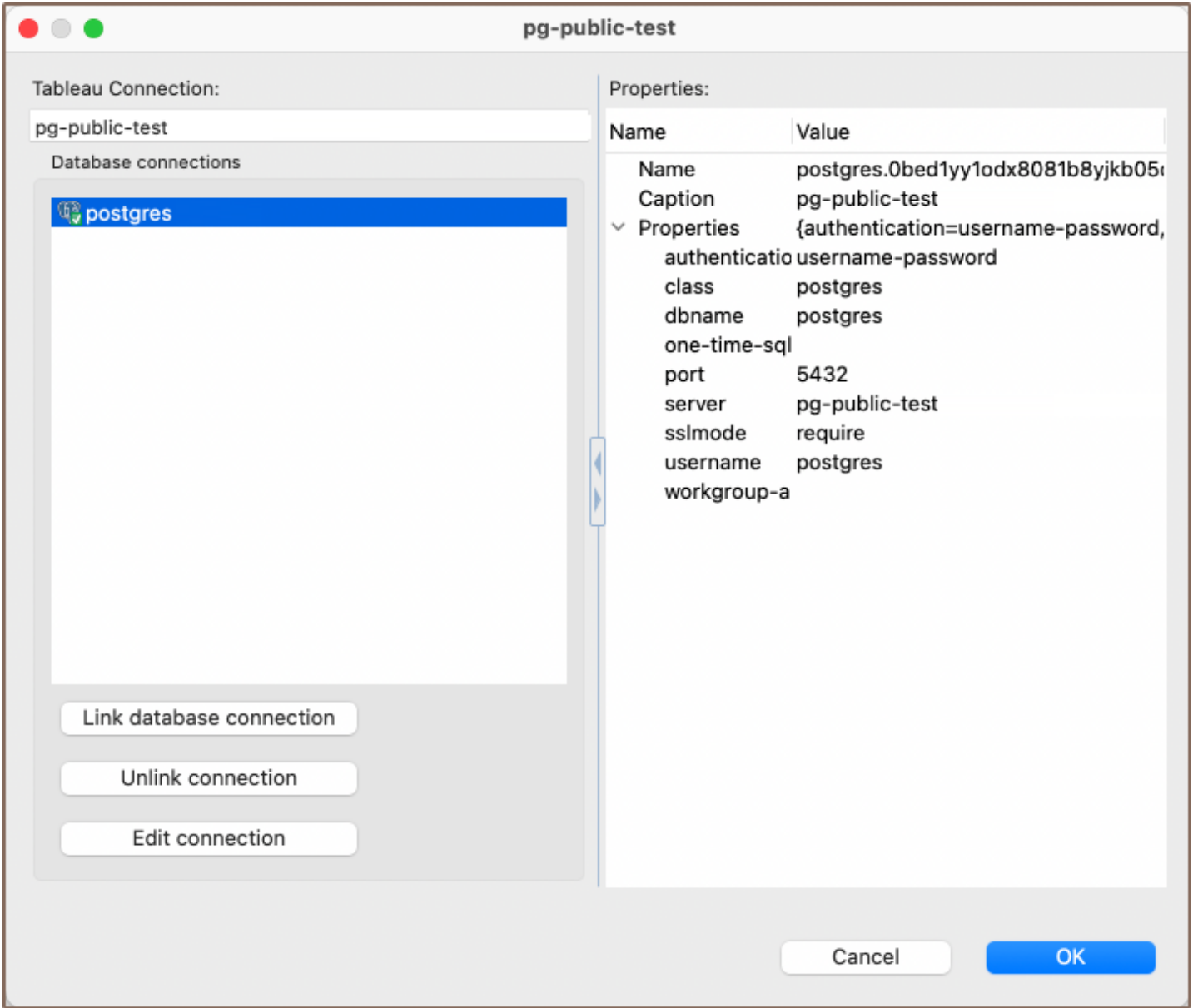


Associate DBeaver and Tableau connection

The **Associate with connection** button allows you to view a list of databases that can be associated with Tableau in DBeaver. It can automatically find the database based on the hosts.

Tip: If the **Automatically associate Tableau connections** option is selected in the preferences, clicking on a table will automatically open it in DBeaver.

To configure the database that is associated with Tableau, you can use the **Open Tableau connections** option or double-click the database in Tableau view. A window with connections that are associated with Tableau will appear. From this window, you can choose to **Link database connection**, **Unlink connection**, or **Edit connection** as needed.



Editing Custom SQL Queries

Working with custom SQL queries in Tableau can be challenging, especially when it comes to complex data manipulation. DBeaver's integration with Tableau offers a solution, allowing you to edit and manage custom SQL

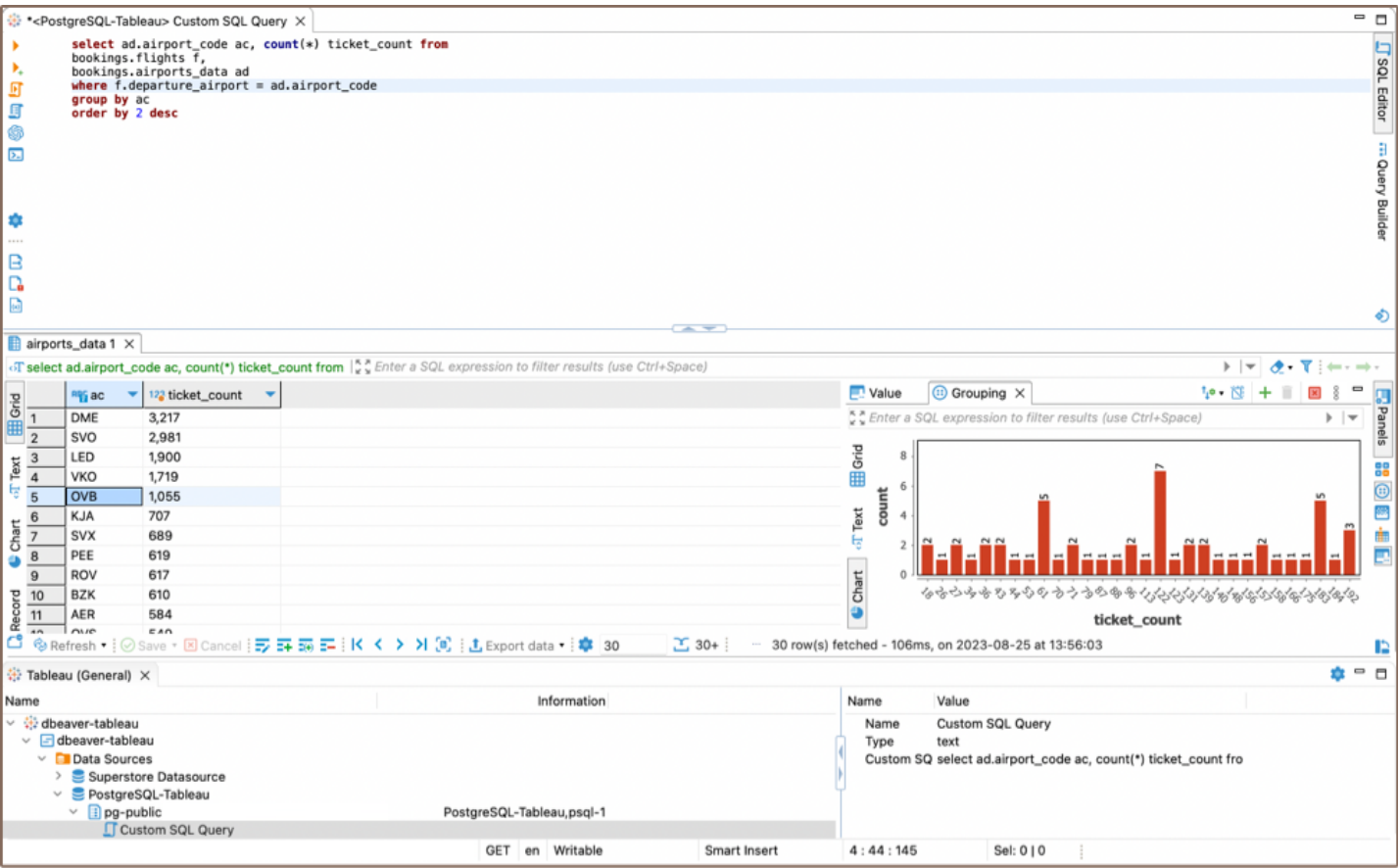
queries with ease. By connecting Tableau to DBeaver, you can leverage DBeaver's robust SQL editing tools to enhance your data analysis workflow in Tableau.

Important: The custom SQL query must already be configured in Tableau. More details can be found [in the official Tableau documentation](#).

This is how you can edit custom SQL queries using DBeaver:

1. In the Tableau view in DBeaver, select the Custom SQL query in your profile. You can double-click it, right-click and choose **Open Custom SQL**, or press the shortcut `F4` .
2. The [SQL Editor](#) will open, where you can work with the query just like any other query in DBeaver. You can make necessary changes, utilize [grouping](#), [filters](#), and other functionalities, and save and execute the query to see the changes reflected in Tableau.

Note: It is essential to save the query in DBeaver to ensure that the changes are also reflected and saved in Tableau.



This integration simplifies the process of working with custom SQL queries in Tableau, making it more efficient and user-friendly.

PGDebugger

Table of contents

[Prerequisites for Debugging](#)

[Installation](#)

[PostgreSQL 12 on Ubuntu based distros](#)

[Installation from source code](#)

[Running debugger in the DBeaver interface](#)

Prerequisites for Debugging

To enable interactive debugging of PL/pgSQL procedures on a Postgres server, you need to use the `_plugin_debugger`. The `_plugin_debugger` is a typical interactive debugger delivered as an extension. It requires a shared library preload in Postgres to operate the `shared_preload_libraries` parameter in the settings. The debugger is developed and maintained by EDB. Its source code is available for examination and improvement.

The debugger provides the required server API for debugging PL/SQL procedures with:

- Breakpoint management;
- Step-by-step tracing;
- Variable acquisition and management.

Installation

PostgreSQL 12 on Ubuntu based distros

If you happen to have a PostgreSQL 12 installed via `apt`, then the procedure is quite straightforward:

```
sudo apt install postgresql-12-pldebugger
sudo service postgresql restart
```

After that, run the following command in the database or databases that you wish to debug functions in:

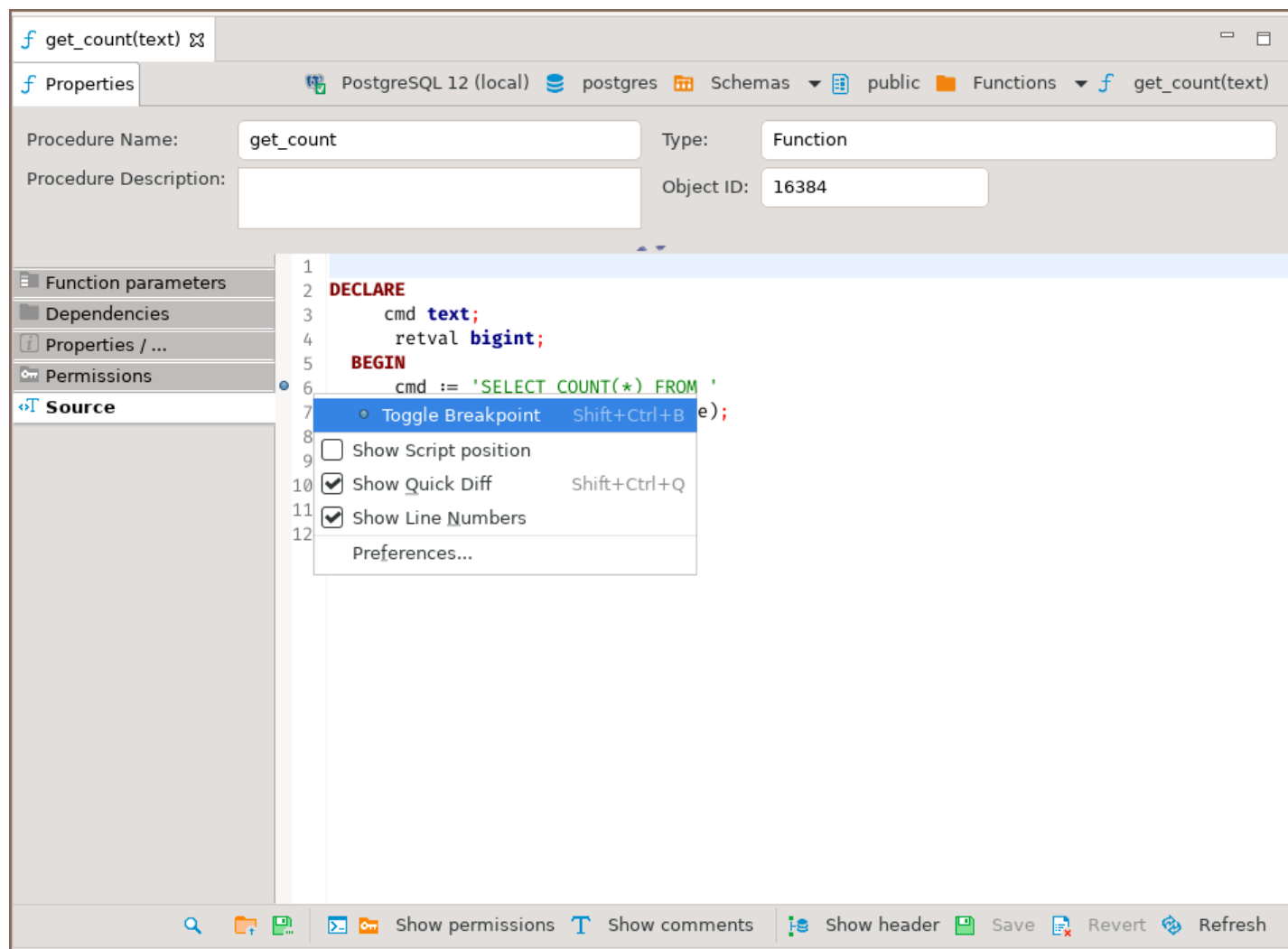
```
CREATE EXTENSION pldbgapi;
```

Installation from source code

You can find the source code in [this repository](#). Installation instructions are located in the README file.

Running debugger in the DBeaver interface

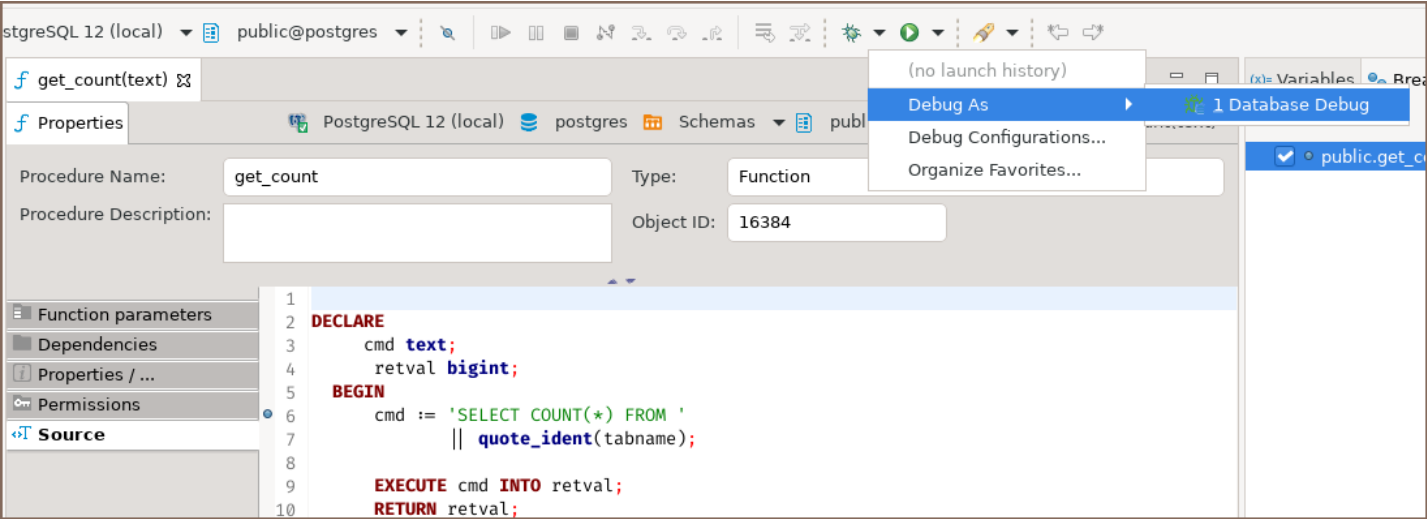
Open the source code of the function you want to debug. To toggle breakpoints, place the caret on the line you want the function to be stopped at and use a shortcut **Shift + Control + B**. Alternatively, you can toggle the breakpoint with your mouse by clicking on a ruler, as demonstrated in the screenshot below:



NB. You must only toggle the breakpoints when *Show header* option is **not** on.

Then you need to set up a debugging configuration. Locate the downward-facing arrow right to the bug icon, click on it, then

Debug As -> Database Debug:



The *Edit Configuration* dialog opens. Set up input values in the table *Function parameters*.

Edit Configuration

Edit launch configuration properties

Create a configuration to debug database code

Name: get_count

Settings

Common

Connection settings

DataSource: PostgreSQL 12 (local)Driver: PostgreSQL

Type

☒ Function ☐ Trigger

Attach type

☒ Local ☐ Global

Function

Function: public.get_count(text)Process ID:

Function parameters

Name	Value	Type	Kind
tabname	'test'	text	IN

Revert

Apply

Cancel

OK

Click on *OK* button, and you are ready to go!

The usual buttons essential for debugging such as *Step Over* and *Continue* are located here:

PostgreSQL 12 (local) public@postgres

get_count(text)

Properties PostgreSQL 12 (local) postgres Schemas public Functions get_count(text)

Procedure Name: get_count Type: Function

Procedure Description: Object ID: 16384

Function parameters
Dependencies
Properties / ...
Permissions
Source

```
1  
2 DECLARE  
3     cmd text;  
4     retval bigint;  
5 BEGIN  
6     cmd := 'SELECT COUNT(*) FROM '  
7         || quote_ident(tabname);  
8  
9     EXECUTE cmd INTO retval;  
10    RETURN retval;  
11 END;  
12
```

Optional extensions

Table of contents

[Install Process](#)

[Extension installation in CE version](#)

[Extension installation in EE](#)

[Extensions](#)

[Office integration](#)

[Debugger support](#)

[SVG format support](#)

[Git integration](#)

[AI integration](#)

You can install optional extensions (plugins) in DBeaver.

Install Process

Extension installation in CE version

1. Main menu `Help -> Install New Software`
2. Paste extension P2 repository URL into `Work with` field and press `Enter`
3. Check items you wish to install
4. Click Next->Finish. Restart DBeaver.

IMPORTANT: The installation will not work if DBeaver is installed in a write-protected folder (like `Program Files`, `/Applications`, `/usr/`). To proceed you need to run DBeaver from some non write-protected folder or run it as the Administrator/root.

Extension installation in EE

You can do this from the online Eclipse Marketplace (open Marketplace UI from main menu). OR you can install it manually the same way as in the CE version (see above)

Extensions

Office integration

This extension supports data export in Office formats (XLS). It works for all platforms and all popular Offices - MS Office, Open Office, Libre Office, MacOS X.

It is included in the EE version.

Marketplace URL: <https://marketplace.eclipse.org/content/dbeaver-office-integration>

P2 repository URL: <https://dbeaver.io/update/office/latest/>

30		30	Edward	Francis	[NULL]	230 Elgin Street	Or
31		31	Martha	Silk	[NULL]	194A Chain Lake Drive	Ha
32		32	Aaron	Mitchell	[NULL]	696 Osborne Street	W
33		33	Ellie	Sullivan	[NULL]	5112 48 Street	Ye
34		34	João	Fernandes	[NULL]	Rua da Assunção 53	Lis
35		35	Madalena	Sampaio	[NULL]	Rua dos Campeões Europeus de Viena	Pa
36		36	Hannah	Schneider	[NULL]	Tauentzienstraße 8	Be
37		37	Fynn	Zimmermann	[NULL]	Berger Straße 10	Fr
38		38	Niklas	Schröder	[NULL]	Barbarossastraße 19	Be
39		39	Camille	Bernard	[NULL]	4, Rue Milton	Pa
40		40	Dominique	Lefebvre	[NULL]	8, Rue Hanovre	Pa
41		41	Marc	Dubois	[NULL]	11, Place Bellecour	Ly
42		42	Wyatt	Girard	[NULL]	9, Place Louis Barthou	Bc
43		43	Isabelle	Merrier	[NULL]	68, Rue Louvance	Ni

Save Cancel Script

Record

Panels

Grid

Text

59 row(s) fetched - 2ms (+2ms)

Open results in Excel

Debugger support

This extension supports SQL debugger in PostgreSQL.

It is included in the EE version.

Marketplace URL: <https://marketplace.eclipse.org/content/dbeaver-sql-debugger> P2 repository URL:

<https://dbeaver.io/update/debug/latest/>

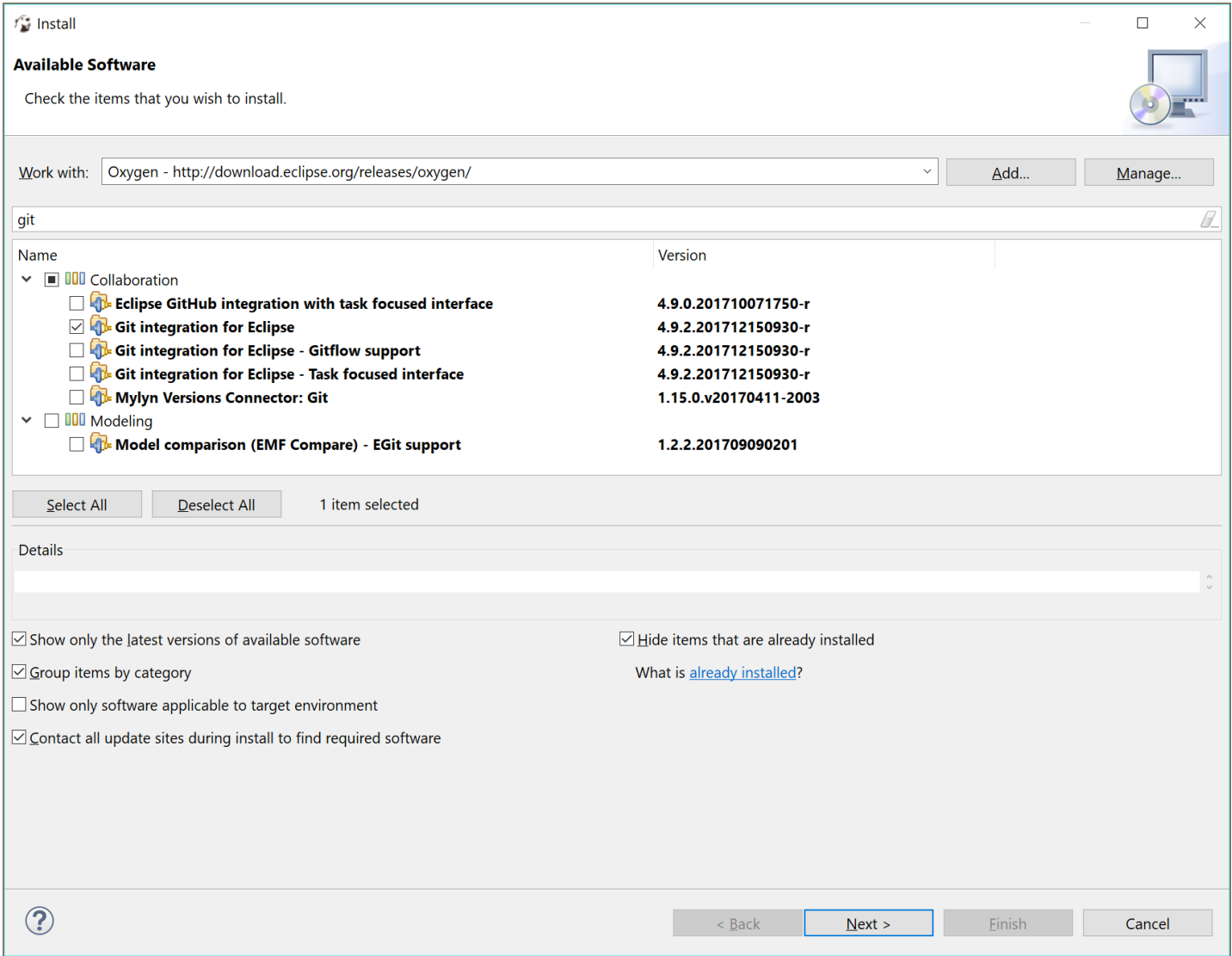
SVG format support

This extension supports ERD export in SVG (vector) format.
It is included in the EE version.

Marketplace URL: <https://marketplace.eclipse.org/content/dbeaver-svg-format-support>
P2 repository URL: <https://dbeaver.io/update/svg/latest/>

Git integration

Choose Oxygen (or other Eclipse version item) in available items:



AI integration

With this extension, you can create working SQL queries using human language in the SQL editor.

[Learn more about AI integration](#)

Marketplace URL: <https://marketplace.eclipse.org/content/dbeaver-chatgpt>

P2 repository URL: <https://dbeaver.io/update/ai/latest/>

Accessibility

DBeaver offers you the ability to use most of its features with a keyboard and screen readers. Here you will find a guidelines on how to use DBeaver without a mouse in some common cases.

Note: DBeaver is tested primarily with standard Windows keyboard shortcuts for accessibility.

Table of contents

[General](#)

[Screen readers support](#)

[Structure and navigation](#)

[Shortcuts](#)

[Colors and fonts](#)

[DBeaver Views and Wizards](#)

[Database Navigator](#)

[New Connection](#)

[Object Properties](#)

[SQL Editor](#)

[Result Set](#)

[ER Diagrams](#)

General

Screen readers support

DBeaver provides support for various screen readers:

- **NVDA.** Find more details in the [NVDA User Guide](#).
- **JAWS.** Access comprehensive information through the [JAWS Documentation](#).
- **Narrator.** More information available at the [Narrator User Guide](#).

To change the screen reader in DBeaver, navigate to **Window -> Preferences -> User Interface -> Accessibility**.

Structure and navigation

DBeaver is an application consisting of many parts, so-called Views, such as [Database Navigator](#), [SQL Editor](#), [Data Editor](#), etc. A [Toolbar](#) with many buttons inside each **View** and horizontal and vertical tabs allows you to switch between **Views** and inside them.

Main navigation shortcuts

- You can switch Views by using `Ctrl + F7` | `#F7` .
- You can switch between buttons on the toolbars and menus using **Tab** and the **arrows**.

How to switch tabs

- You can switch horizontal tabs by using `Ctrl + PageUp / PageDown` .
- Sub-tabs can be switched by using `Alt + PageUp / PageDown` .
- If there are vertical tabs inside the view, you can switch them by using `Ctrl + Alt + PageUp / PageDown` or `Ctrl + Shift + PageUp / PageDown` .

When you focus on any interface object, such as a table, column, tab, or cell, you can open the **Context menu** to see all the actions available for that object.

To open the Context menu, use `Shift + F10` | `#F10` .

Shortcuts

Most of DBeaver's functions are accessible through special shortcuts. You can find them in **Window -> Preferences -> User Interface -> Keys**. If you mostly use the keyboard, you can switch to the **DBeaver Keyboard Only** scheme, which has additional shortcuts.

[Learn more about DBeaver shortcuts](#)

Colors and fonts

Themes: DBeaver supports theme customization to enhance your user experience. Among the available options, DBeaver provides integration for the High-Contrast theme native to Windows systems. For more details on this and other themes, see our comprehensive [theme guide](#).

Fonts: To change fonts in DBeaver, navigate to **Window -> Preferences -> User Interface -> Appearance -> Color and fonts**, and open DBeaver fonts. The **Main font** allows you to change the font family, size, and color in most elements of DBeaver's interface, including the **Database Navigator** and the **Data Grid**. The Monospace font changes the font in the **SQL Editor** and other **Views** with monospaced fonts.

Data Colorization: You can [color rows in the Data Grid](#) by following some rules or even by [selecting colors for particular data types](#).

DBeaver Views and Wizards

Let's see some comments on using the most popular **Views** in DBeaver with the keyboard and screen reader.

Database Navigator

In the **Database Navigator**, you can view and open all the database connections and objects inside them. You can move between database objects and open nodes using arrows.

- To view object properties, use `F4` .
- To open **SQL Editor** associated with the selected object, use `F3` .

[See all Database Navigator shortcuts](#)

New Connection

To [create a new connection](#), open the **Wizard window** using `Ctrl+N` | `#N` , then type **connect** and select **Database connection**.

[See Connection shortcuts](#)

Object Properties

The [Properties Editor](#) is used to see all the information about database objects. When you focus on the first row of the table, NVDA reads the contents of the row with the corresponding column names.

- You can open object properties with `F4` .
- To switch between left-side tabs, use `Alt + Shift + Up` | `Alt + Shift + Down` .

[See Properties Editor shortcuts](#)

SQL Editor

The [SQL Editor](#) allows you to create and execute SQL scripts associated with a database connection. To open the **SQL Editor**, focus on the connection or table in the **Database Navigator** and press `F3` . To execute the SQL script under the cursor, press `Ctrl + Enter` .

- [See SQL Editor shortcuts](#)

Result Set

When you place your focus on the first cell in the data table, the NVDA screen reader says the table name and the name of a column consisting of this cell. If you move from one cell to another, the screen reader announces the column names when you change focus.

If you need to work with columns using the keyboard, you can select an entire column, copy the column name, and resize the column using special keyboard shortcuts. Some of them are only available in the **DBeaver Keyboard Only** scheme.

[See Result Set shortcuts](#)

ER Diagrams

[ER Diagrams](#) are fully accessible with the keyboard and the NVDA screen reader. You can move inside the table, move or resize it, and listen to the screen reader pronounce the table name and column names inside that table.

[Learn more about ER Diagram shortcuts](#)

User interface themes

Table of contents

[Change the DBeaver theme](#)

[Managing the High Contrast theme in Windows](#)

[Activation of High Contrast theme](#)

[Deactivation of High Contrast theme](#)

[High Contrast theme switching limitations](#)

The interface themes governs the look and feel of the user interface's windows, dialogs, buttons, and all other visual components. By default, DBeaver is configured to use the **Light** theme.

Change the DBeaver theme

1) Go to **Window -> Preferences -> User Interface -> Appearance**.

2) Select from the following available themes:

- Light
- Dark
- Classic

Note: To fully apply the selected theme, you must restart DBeaver.

Managing the High Contrast theme in Windows

Windows users have the additional option of using **High Contrast** themes. This section provides details on how DBeaver interacts with these themes.

Activation of High Contrast theme

When the **High Contrast** theme is activated in Windows, DBeaver automatically switches its theme to **High Contrast**. If DBeaver is already running when the system theme changes, a notification will appear suggesting a restart for the theme to apply correctly.

Deactivation of High Contrast theme

If you deactivate the **High Contrast** theme in Windows, it will automatically be removed from DBeaver's list of available themes after the restart.

Note: After disabling the **High Contrast** theme in Windows, DBeaver will revert to the theme that was previously selected.

High Contrast theme switching limitations

The theme switcher in DBeaver is disabled when the **High Contrast** theme is active in Windows. You cannot switch to a **non-High Contrast** theme in DBeaver unless you disable the **High Contrast** theme in Windows.

UI language

Table of contents

[Overview](#)

[Changing interface language in preferences](#)

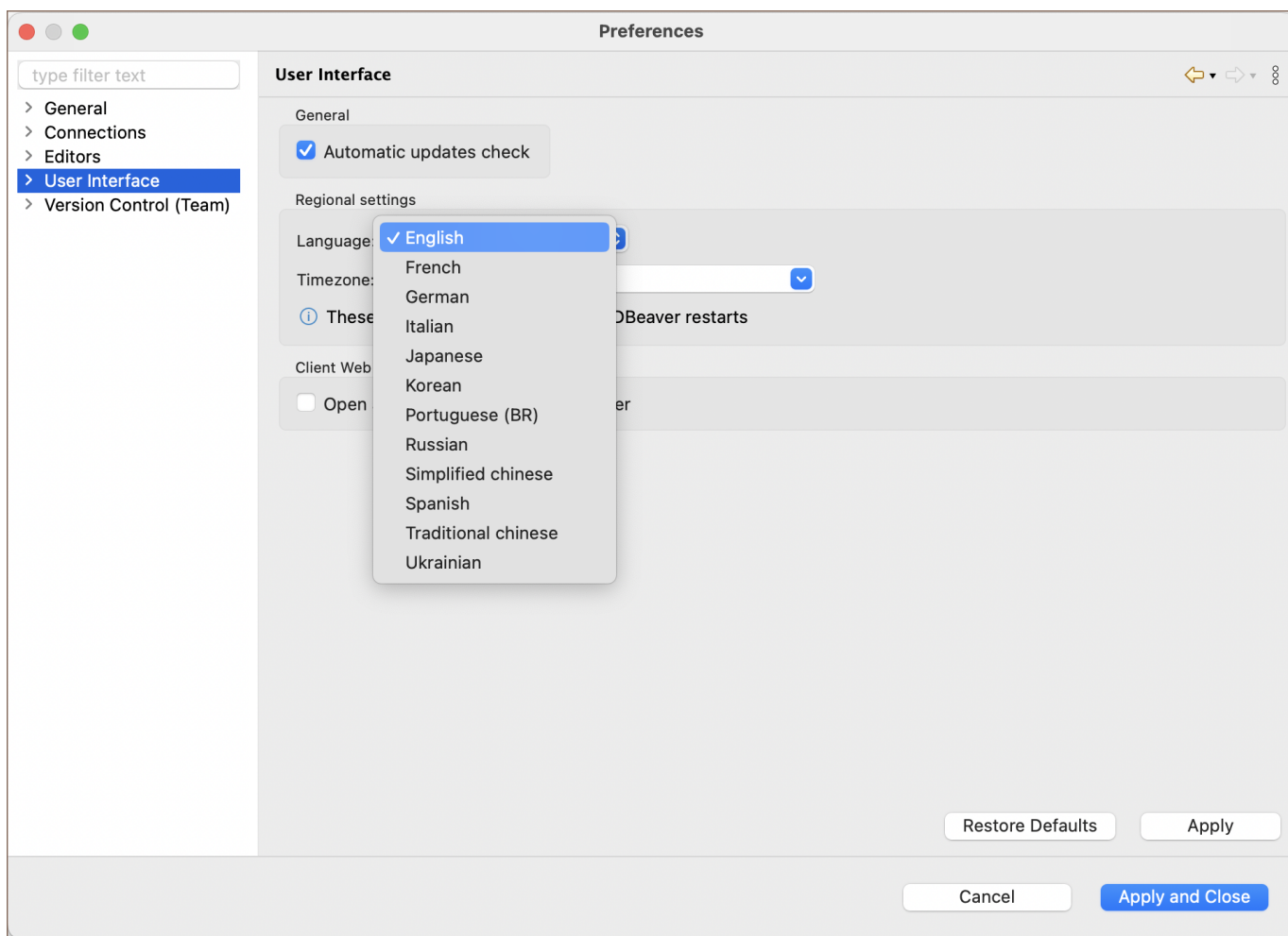
[Changing interface language in configuration file](#)

Overview

The interface language of DBeaver can be customized to suit your preferences. There are two methods to achieve this:

Changing interface language in preferences

1. Navigate to **Window -> Preferences -> User Interface**.



2. Choose your desired language from the drop-down menu.
3. Click the **Apply and Close** button to save your settings.

Note: If DBeaver is installed in a directory without write permissions, you may not be able to change the language directly through the interface. If this occurs, please proceed to modify the configuration file as described below.

Changing interface language in configuration file

1. Locate the `dbeaver.ini` file.

Tip: Detailed instructions on finding `dbeaver.ini` are available in [our article](#).

2. Using a text editor, open this file to make changes.
3. Insert the following lines above the `-vmargs` line:

```
-nl
XX
```

Replace `XX` with the language code from the list below:

Language	Code
English	en
French	fr
German	de
Italian	it
Japanese	jp
Korean	ko
Portuguese (BR)	pt_BR
Romanian	ro
Russian	ru
Simplified Chinese	zh
Spanish	es
Traditional Chinese	zh
Ukrainian	uk

Toolbar customization

Table of contents

[Top toolbar customization](#)

[SQL Editor toolbar customization](#)

[Data Editor toolbar customization](#)

DBeaver offers highly customizable toolbars to enhance your workflow:

1. **Top toolbar:** This toolbar is the primary control panel located at the top of the DBeaver interface. It provides quick access to various database actions, transactions, and utilities.

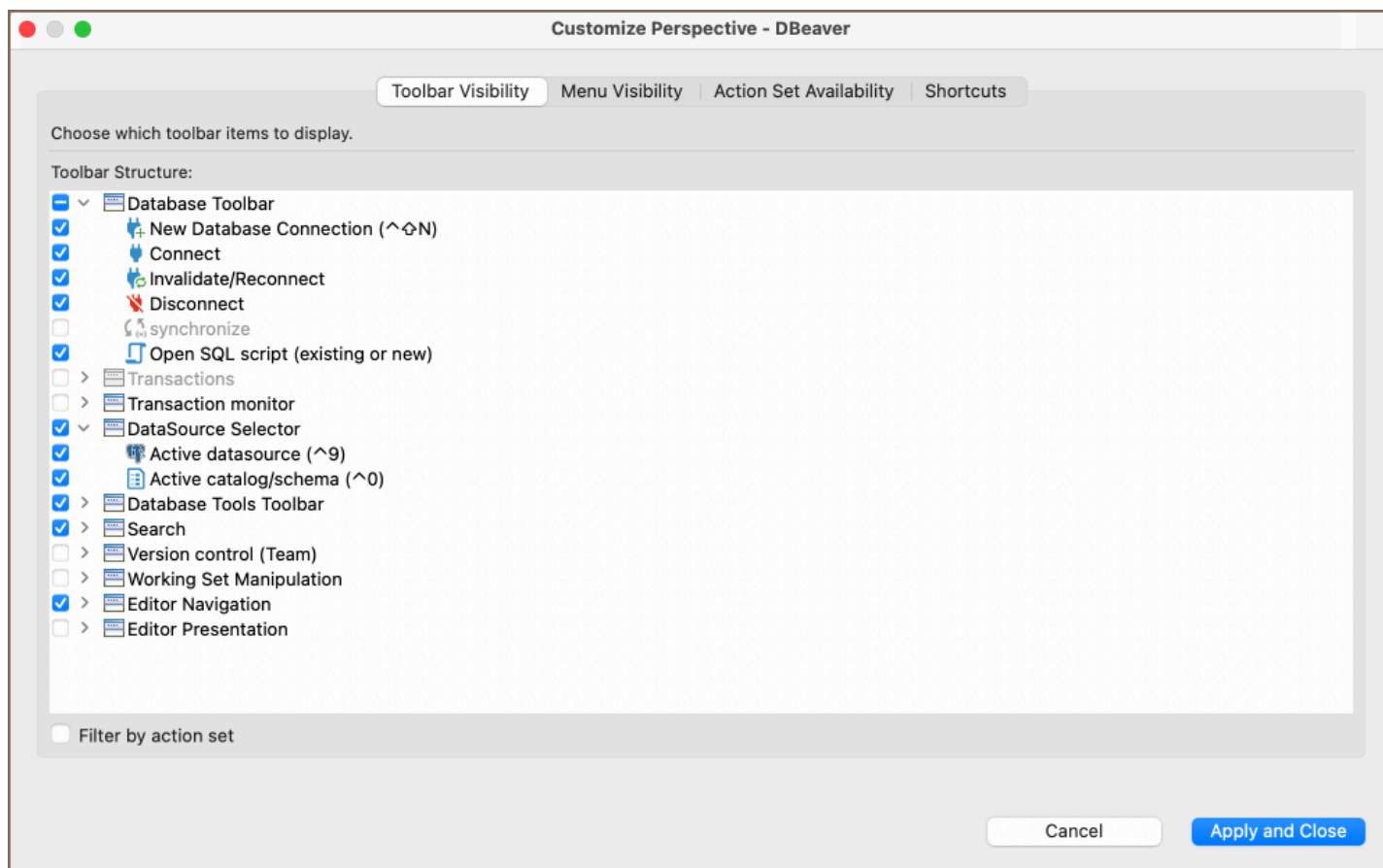


2. **SQL Editor toolbar:** Specifically designed for SQL editing, this toolbar appears when you are working within the SQL editor. It provides functionalities for executing SQL queries, navigating through your code, and other SQL-specific tasks.
3. **Data Editor toolbar:** This toolbar provides abilities for adding, editing, and exporting rows, for navigating and filtering data.

Both toolbars are designed for ease of use and can be customized to fit your specific needs. From basic database operations to advanced debugging and transaction controls, these toolbars can be tailored to streamline your database management activities.

Top toolbar customization

To access toolbar customization, navigate to **Window -> Customize Perspective -> Toolbar Visibility** tab.



You can customize the Top toolbar by adding or removing various sections or individual elements based on your specific needs. Below is a table summarizing the sections available for customization:

Section	Description
Database Toolbar	Manage database connections and navigation. For further details, see our Database Navigator article.
Transaction	Control database transactions.
Transaction Monitor	Monitor transaction status and logs.
DataSource Selector	Choose active data sources and schemas. Additional information can be found in our article on Active Database .
Database Tools Toolbar	Access dashboards and views. Consult our article on Dashboards for more insights.
Launch	Execute, debug, and profile code.
Debug	Manage debug sessions and control execution.
Git	Perform Git-related activities.
Search	Enable quick search functionalities in code. Further reading is available in our Search article.

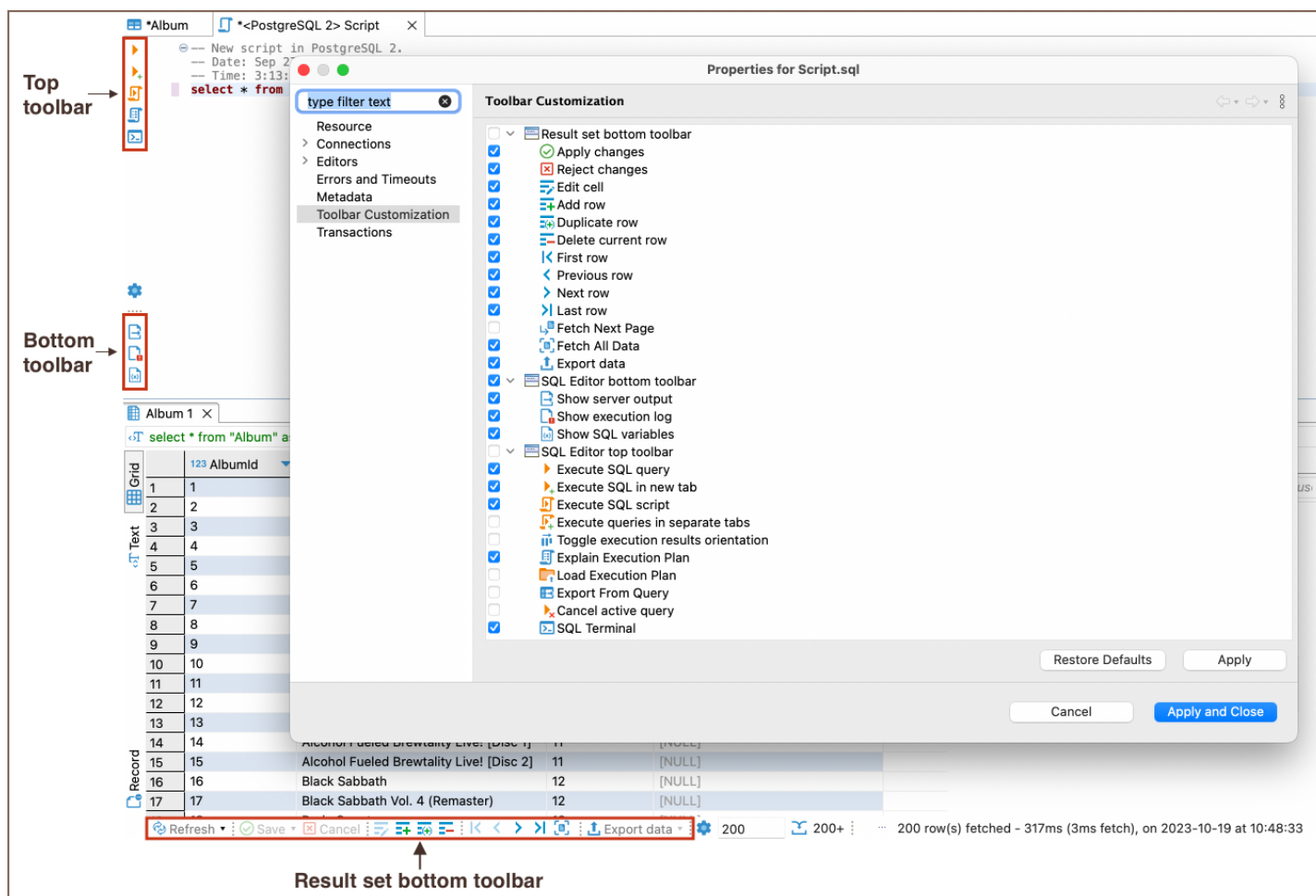
Version Control (Team)	Manage and synchronize with version control system. Learn more in our article about Project Team Work .
Working Set Manipulation	Manage working sets for organization.
Editor Navigation	Navigate through editor locations.
Editor Presentation	Customize text display settings in the editor.

Note: Some toolbar sections may not be visible because all their child elements are in an unavailable action set. To make them available, go to the **Action Set Availability** tab and check the checkbox next to the required section.

SQL Editor toolbar customization

The SQL Editor toolbar in DBeaver can be customized to streamline your SQL development workflow. To access toolbar customization options, follow one of these methods:

- Navigate to **Window -> Preferences -> User Interface -> Toolbar Customization**.
- Right-click within the SQL Editor window and choose **Preferences -> Toolbar Customization**.




In the customization window, you can add or remove various sections and buttons. Below is a table summarizing the available sections:

Section	Description
SQL Editor bottom toolbar	Modify technical options at the bottom of the SQL Editor.
SQL Editor top toolbar	Customize query-related options at the top of the SQL Editor.

Note: For detailed explanations of the button functionalities in the SQL Editor, consult our [SQL Editor](#) article.

Data Editor toolbar customization

Access the [Data Editor](#) toolbar customization options through the following steps:

- Go to **Window -> Preferences -> User Interface -> Toolbar Customization**.
- Alternatively, execute an SQL query in the SQL Editor, then in the Results tab, click on the Configure button  and choose Toolbar Customization.

Within the customization menu, you can tailor the **Result set bottom toolbar** to fit your workflow. This section allows you to alter the tools for interacting with the displayed data sets.

Workspace location

Table of contents

- [Default location of DBeaver workspace](#)
- [Folders](#)
- [Custom location](#)
- [Old default workspace location](#)
- [Ancient default workspace location](#)

Default location of DBeaver workspace

By default DBeaver stores all its files (configurations, scripts, diagrams, etc) in the following folder:

OS	Location
Windows	%APPDATA%\DBeaverData\
MacOS	~/Library/DBeaverData/
Linux	\$XDG_DATA_HOME/DBeaverData/ (\$XDG_DATA_HOME= ~/.local/share if not set)

Folders

Folder	Location
workspace6	Workspace files for DBeaver 6.1+
drivers	Auto downloaded database drivers

Custom location

You can specify a custom workspace location by passing parameter `-data <path>` in the command line. `<path>` can be an absolute or relative directory path.

Old default workspace location

Old - means before DBeaver 6.1.3

OS	Location
Windows	<code>C:\Users\YourName\.dbeaver4</code> .
Linux	<code>~/ .dbeaver4/</code>
MacOS	<code>~/ .dbeaver4/</code>

Ancient default workspace location

Ancient - means before DBeaver 4.

OS	Location
Windows	<code>C:\Users\YourName\.dbeaver</code> .
Linux	<code>~/ .dbeaver/</code>
MacOS	<code>~/ .dbeaver/</code>

Resetting your workspace

The article has been moved to [Reset UI settings](#).

Keyboard shortcuts

Table of contents

[Connection](#)

[Result Set](#)

[Data Editor](#)

[Text Editor](#)

[SQL Editor](#)

[ER Diagrams](#)

[Visual Query Builder](#)

[Search](#)

[Data viewer](#)

[Window](#)

[Database Navigator](#)

[Properties Entity Editor](#)

[Navigation tree](#)

[File](#)

[Run/Debug](#)

[Git](#)

[Project](#)

[Oracle](#)

[Utility](#)

[Help](#)

[References](#)

Here is a complete list of default hotkeys in DBeaver UI for Windows, Linux, and macOS users. It will help you work in DBeaver faster and more efficiently. Remember that you can always change the keyboard shortcut in the DBeaver settings. Go through: **Window -> Preferences -> User Interface -> Keys**. Select command and add a keyboard shortcut to the **Binding** row.

If you mostly use keyboard, to get more shortcuts you can switch default key scheme to **DBeaver Keyboard Only** key scheme in **Window -> Preferences -> User Interface -> Keys** settings.

Connection

Shortcut for Windows /Linux	Shortcut for macOS	Action
F4	F4	Open object editor
Ctrl+Shift+N	Ctrl+Shift+N	Open Connect to a database window
Ctrl+Shift+D	^⇧D	Open database meta-object
Ctrl+Alt+Enter	^⌘↶	Open a new SQL console. No script file will be created.
Ctrl+]	^]	Create a new SQL script(***)
F3	F3	Open existing SQL script (***)
Ctrl+Enter	^↶	Open most recent SQL script(***) for an active connection
Ctrl+Alt+Shift+K	^4	Commit changes in current session
Alt+`	^V	Shows context menu with database tools
Ctrl+Alt+Shift+U	^⌘⇧U	Generates UUID/GUID and inserts into an active text editor or data editor
Ctrl+Alt+Shift+B	^⌘⇧B	Open database dashboard
Ctrl+Alt+Shift+R	^8	Rollback changes in current session
Ctrl+0	^0	Select active schema for current database
Ctrl+9	^9	Select connection for current editor
Alt+Shift+Down/Up	Alt+⌘+Down/Up	Switch vertical tabs

Result Set

Shortcut for Windows	Shortcut for	

/Linux	macOS	Action
Alt+Insert	⌘Insert	Add new row
Ctrl+D	^D	Copy values from row above to current row
Ctrl+Alt+D	^⌘D	Copy values from row below to current row
Alt+Delete	⌘⌫	Delete current row
Ctrl+Alt+Insert	^⌘Insert	Duplicate current row
Enter	↵	Edit cell value with inline editor
Esc	Esc	Reset cell to original value
Ctrl+S	^S	Apply data changes
Ctrl+Alt+Shift+T	^⌘⇧T	Set focus to filter editor/data editor
Ctrl+Shift+7	^⇧7	Switch focus to results viewer/active panel
Ctrl+Tab	Ctrl+Tab	Switch focus between the Grid view and the Value panel
Ctrl+Alt+Shift+Enter	^⌘⇧↵	Apply cell changes
Shift+Enter	⇧↵	Edit cell value in separate dialog/editor
Ctrl+Shift+=	^⇧=	Fetch all rows
Ctrl+Alt+N	^⌘N	Fetch next page of results
Ctrl+F11	^F11	Filter by unique attribute values
F11	F11	Filter context menu
Ctrl+Alt+Shift+Left	^⌘⇧←	Move to first row
Ctrl+Alt+Shift+Right	^⌘⇧→	Move to last row
Alt+Space	⌘Space	Follow foreign key link
Ctrl+Alt+Right	^⌘→	Move to next row
Ctrl+Alt+Left	^⌘←	Move to previous row
Ctrl+Shift+1	^⇧1	Foreign keys and tables referencing current table
Ctrl+R	^R	Reject data changes

Ctrl+Backspace	^⌫	Set cell to default value
Ctrl+`	^`	Switch results presentation
Tab	Tab	Toggle results Grid/Record view
F7	F7	Toggle extra result panels
Ctrl+2	^2	Toggle results sort order (ascending/descending/default)
Ctrl+Alt+C		Select column(s)
Ctrl+Alt+R		Select row(s)
Shift+F11		Show context menu for column
Ctrl+G		Go to row
Ctrl+Shift+G		Go to column

Result Set (Additional shortcuts for *DBeaver Keyboard Only* key scheme)

Shortcut for Windows/Linux	Shortcut for macOS	Action
Alt+Shift+Right		Move column(s) right
Alt+Shift+Left		Move column(s) left
Alt+Shift+C		Copy column names
Alt+Shift+H		Hide columns
Alt+Shift+T		Show hidden columns
Alt+Shift+F		Open filters settings
Ctrl+Alt+F6		Show/hide results panel (Calc)
Ctrl+Alt+F5		Show/hide results panel (Grouping)
Ctrl+Alt+F4		Show/hide results panel (Metadata)
Ctrl+Alt+F3		Show/hide results panel (References)
Ctrl+Alt+F2		Show/hide results panel (Value)

Data Editor

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Space	⌘Space	Enable autocomplete
Ctrl+F	⌘F	Find and replace text
Ctrl+Shift+Space	⌘⇧Space	Show Context Information in Dialogs and Windows
Ctrl+Alt+Space	⌘⌥Space	Show Context Information (SQL Editor Context)
Ctrl+X	⌘X	Cut the selection to the clipboard
Ctrl+C	⌘C	Copy the selection to the clipboard
Ctrl+V	⌘V	Paste from the clipboard
Delete	⌘⌫	Delete the selection
Ctrl+O	⌘O	Export Diagram
Ctrl+K	⌘K	Find next item
Ctrl+Shift+K	⇧⌘K	Find previous item
Ctrl+J	⌘J	Incremental find
Ctrl+Shift+J	⇧⌘J	Incremental find reverse
Ctrl+Shift+Q	⌘⇧Q	Toggles quick diff information display on the line number ruler
Ctrl+1	⌘1	Suggest possible fixes for a problem
Ctrl+Y	⇧⌘Z	Redo the last operation
Ctrl+Z	⌘Z	Undo the last operation
Shift+Delete	⇧⌘⌫	Remove selected Pictogram Elements
Ctrl+A	⌘A	Select all
Alt+Shift+A	⌥⌘A	Toggle block/column selection in the current text editor
Ctrl+Shift+Insert	⇧⌘Insert	Toggle insert mode
Alt+Shift+Y	⌥⌘Y	Toggle word wrap in the current text editor

Ctrl+F5	⌘F5	Update selected Pictogram Elements
Alt+/ 	^.	Context insensitive completion
Ctrl+Shift+Space	^⇧Space	Context insensitive completion (SQL Editor Context)

Text Editor

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Shift+Y	⇧⌘Y	Changes the selection to lowercase
Ctrl+Shift+X	⇧⌘X	Changes the selection to uppercase
Ctrl+Numpad_Subtract	⌘Numpad_Subtract	Collapses the folded region at the current selection
Ctrl+Shift+Numpad_Divide	⇧⌘Numpad_Divide	Collapses all folded regions
Ctrl+Alt+Down	⌘⇧↓	Duplicates the selected lines and moves the selection to the copy
Ctrl+D	⌘D	Delete a line of text
Ctrl+Delete	⌘⇧⌫	Delete the next word
Ctrl+Backspace	⌘⇧〉	Delete the previous word
Ctrl+Shift+Delete	⇧⌘⌫	Delete to the end of the line of text
Ctrl+Alt+Up	⌘⇧↑	Duplicates the selected lines and leaves the selection unchanged
Ctrl+Numpad_Add	⌘Numpad_Add	Expands the folded region at the current selection
Ctrl+Numpad_Multiply	⌘Numpad_Multiply	Expands all folded regions
Ctrl+Shift+Enter	⇧⌘↵	Adds a new line above the current line
Shift+Enter	⇧↵	Adds a new line below the current line
Ctrl+Alt+J		Join lines of text (Editing Text)
Ctrl+Shift+J	^⇧J	Join lines of text (SQL Editor Context)
End	⌘→	Go to the end of the line of text
Home	⌘←	Go to the start of the line of text
Alt+Down	^⇧↓	Moves the selected lines down (Editing Text)

Ctrl+Shift+Down	⇧↓	Moves the selected lines down (SQL Editor Context)
Alt+Up	^⇧↑	Moves the selected lines up (Editing Text)
Ctrl+Shift+Up	⇧↑	Moves the selected lines up (SQL Editor Context)
Ctrl+Right	⇧→	Go to the next word
Ctrl+Left	⇧←	Go to the previous word
Ctrl+Shift+Numpad_Multiply	⇧⌘Numpad_Multiply	Resets the folding structure
Ctrl+Down		Scroll down one line of text
Ctrl+Up		Scroll up one line of text
Shift+End	⇧⌘→	Select to the end of the line of text
Shift+Home	⇧⌘←	Select to the beginning of the line of text
Ctrl+Shift+Right	⇧⇧→	Select the next word
Ctrl+Shift+Left	⇧⇧←	Select the previous word
	⇧↘	Select to the end of the text
	⇧↖	Select to the beginning of the text
F2	F2	Displays information for the current caret location in a focused hover
Ctrl+End	↘	Go to the end of the text
Ctrl+Home	↖	Go to the beginning of the text
Ctrl+Numpad_Divide	⌘Numpad_Divide	Toggles folding in the current editor
Insert	Insert	Toggle overwrite mode
Ctrl++	⌘+	Zoom in text, increase default font size for text editors
Ctrl+=	⌘=	Zoom in text, increase default font size for text editors
Ctrl+-	⌘-	Zoom out text, decrease default font size for text editors

SQL Editor

Shortcut for Windows	Shortcut for	
----------------------	--------------	--

/Linux	macOS	Action
Alt+X	⌘X	Execute SQL script(**)
Alt+N	⌘N	Execute SQL script natively
Ctrl+Enter	^↵	Execute SQL statement(*)
Ctrl+	^	Execute SQL statement in a new tab
Ctrl+Alt+Shift+X	^⌘⇧X	Execute script's statements in separate results tabs
Ctrl+ /	^ /	Add or remove single line comment
Ctrl+Shift+ /	^⇧ /	Add or remove multi line comment
Ctrl+Alt+T	^⌘T	Switch active SQL editor panel
Ctrl+T	^T	Show/hide results panel
Ctrl+Shift+ \	^⇧ \	Close results tab
Ctrl+Shift+F	^⇧F	Format text
Ctrl+Alt+'	^⌘'	Select value of the selected SQL expression
Ctrl+Shift+E	^⇧E	Explain execution plan
Ctrl+Shift+[^⇧[Position cursor on the matching bracket
Ctrl+Alt+Shift+O	^⌘⇧O	Load SQL script from file system
Ctrl+Shift+T	^⇧T	Maximize/normalize results panel
Alt+Down	⌘↓	Switch to the next query
Alt+Up	⌘↑	Switch to the previous query
Ctrl+F2	^F2	Rename current SQL script
F4	F4	Open editor of current (highlighted) database object
Ctrl+Alt+Shift+A	^⌘⇧A	Select and show all rows (no fetch size limit)
Ctrl+Alt+Shift+C	^⌘⇧C	Select row count for query under cursor
Ctrl+Shift+.	^⇧.	Set active connection from database navigator selection
Ctrl+Shift+O	^⇧O	Show server output console

Ctrl+Alt+Shift+W	^⌘⇧W	Toggles text editor soft word wrap
Ctrl+Shift+I	^⇧I	Switch presentation to (AI Chat)

ER Diagrams

See [ER Diagrams documentation](#) for a list of shortcuts.

Visual Query Builder

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+B	^B	Open Visual Query Builder
Ctrl+Shift+B	^⇧B	Show/hide generated SQL query text
Ctrl+Alt+PageUP/PageDown Ctrl+Shift+PageUP /PageDown	Ctrl+⌘+PageUP /PageDown Ctrl+⇧+PageUP/PageDown	Switch vertical tabs
Alt+1	⌘1	Changes focus to the main tab
Alt+2	⌘2	Changes focus to the palette
Alt+3	⌘3	Changes focus to the outline
Alt+4	⌘4	Changes focus to parameters
Alt+5	⌘5	Switches to Visual Query Builder settings part
Alt+6	⌘6	Switches to query preview

Important: The shortcuts Alt+1 through Alt+6 are only effective in the DBeaver Keyboard Scheme.

Search

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+H	^H	Open the Search dialog

Ctrl+Alt+G	⌘G	Searches the files in the workspace for specific text
Ctrl+Alt+Shift+F	⌘⇧⌘L	Quick search in Windows

Data viewer

Shortcut for Windows/Linux	Shortcut for macOS	Action
Tab	Tab	Cycle through hand tool and previously used tool
Ctrl+=	⌘=	Zoom In
Ctrl+-	⌘-	Zoom Out
Alt+Shift+Q, Q	⌘⇧Q Q	Shows a particular view
Alt+Shift+Q, B	⌘⇧Q B	Shows a particular view (Breakpoints)
Alt+Shift+Q, C	⌘⇧Q C	Shows a particular view (Console)
Alt+Shift+Q, L	⌘⇧Q L	Shows a particular view (Error Log)
Alt+Shift+Q, Z	⌘⇧Q Z	Shows a particular view (History)
Alt+Shift+Q, O	⌘⇧Q O	Shows a particular view (Outline)
Alt+Shift+Q, X	⌘⇧Q X	Shows a particular view (Problems)
Alt+Shift+Q, S	⌘⇧Q S	Shows a particular view (Search)
Alt+Shift+Q, Y	⌘⇧Q Y	Shows a particular view (Synchronize)
Alt+Shift+Q, V	⌘⇧Q V	Shows a particular view (Variables)

Window

Shortcut for Windows /Linux	Shortcut for macOS	Action
F12	⌘F12	Activate Editor
	⌘W	Closes the active Dialog
Ctrl+3	⌘3	Find actions. Quickly access UI elements

Ctrl+M	^M	Maximize/restore a state of active view or editor
Ctrl+F6	⌘F6	Switch to the next editor
Ctrl+F8	⌘F8	Switch to the next perspective
Ctrl+F7	⌘F7	Switch to the next view
Ctrl+Shift+F6	⇧⌘F6	Switch to the previous editor
Ctrl+Shift+F8	⇧⌘F8	Switch to the previous perspective
Ctrl+Shift+F7	⇧⌘F7	Switch to the previous view
Ctrl+E	⌘E	Open the editor drop-down list
	⇧F10	Show the context menu
Alt+Shift+F3	⌘⇧F3	Shows contribution information for the currently selected element
Ctrl+Shift+L	⇧⌘L	Show the key assist dialog
Ctrl+F10	⌘F10	Show the context menu for the ruler
Alt+-	⇧⌘F10	Show the system menu
Ctrl+F10	⌘F10	Show the view menu
Ctrl+Shift+E	⇧⌘E	Switch to an editor
Alt+F11	^⌘F	Toggles the window between full screen and normal
Ctrl+_	⇧⌘-	Split or join the currently active editor (Horizontal)
Ctrl+_	⇧⌘[Split or join the currently active editor (Vertical)

Database Navigator

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Shift+,	^⇧,	Link with editor
Ctrl+Alt+Shift+D	^⌘⇧D	Add Bookmark
Enter	↵	Opens task configuration
Ctrl+Shift+A	^⇧A	Set as default object

Properties Entity Editor

Shortcut for Windows/Linux	Shortcut for macOS	Action
Alt+Shift+Up	Alt+Shift+Up	Switch to the next entity property left-side tab
Alt+Shift+Down	Alt+Shift+Down	Switch to the previous entity property left-side tab
Ctrl+Alt+PageUP/PageDown Ctrl+Shift+PageUP /PageDown	Ctrl+⌘+PageUP/PageDown Ctrl+Shift+PageUP /PageDown	Switch vertical tabs

Navigation tree

Shortcut for Windows/Linux	Shortcut for macOS	Action
Alt+Left	⌘[Move backward in the editor navigation history
Ctrl+Shift+Numpad_Divide	⇧⌘Numpad_Divide	Collapse the current tree
Ctrl+Shift+Numpad_Multiply	⇧⌘Numpad_Multiply	Expand the current tree
Alt+Right	⌘]	Move forward in the editor navigation history
Ctrl+G	^G	Go to a specified line of text in Windows
Ctrl+L	⌘L	Go to a specified line of text (Editing Text)
Ctrl+.	⌘.	Navigate to the next item
Ctrl+Alt+Right	^⌘→	Next edit location
Alt+F7	⌘F7	Switch to the next page
Alt+PageDown	⌘⇓	Switch to the next sub-tab
Ctrl+PageDown	^⇓	Switch to the next tab
Ctrl+Shift+R	⇧⌘R	Open an editor on a particular item
Ctrl+,	⇧⌘.	Navigate to the previous item
Ctrl+Alt+Left	^Q	Previous edit location

Alt+Shift+F7	⇧⌘F7	Switch to the previous page
Alt+PageUp	⇧⌘↑	Switch to the previous sub-tab
Ctrl+PageUp	⌘↑	Switch to the previous tab
Alt+Shift+W	⇧⌘W	Open the Show In menu

File

Shortcut for Windows/Linux	Shortcut for macOS	Action
F2	F2	Rename the selected item
F5	F5	Refresh the selected items
Ctrl+S	⌘S	Save the changes in current file
Ctrl+Shift+S	⇧⌘S	Save changes in all open files
Ctrl+F4	⌘W	Close the active editor
Ctrl+Shift+F4	⇧⌘W	Close all editors
Ctrl+N	⌘N	Open the New item wizard
Alt+Shift+N	⇧⌘N	Open the New menu
Ctrl+O	⌘O	Open a file
Ctrl+P	⌘P	Print
Alt+Enter	⌘I	Display the properties of the selected item

Run/Debug

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Alt+M	⇧⌘M	Add memory block
Ctrl+W	⌘W	Close the selected rendering
F11	⌘F11	Launch in debug mode

Ctrl+Z	^D	Send end of file
Ctrl+G	⌘G	Go to Address
Ctrl+N	⌘N	Add a new rendering
Ctrl+Alt+N	⌘⌘N	Show renderings from next memory monitor.
Ctrl+Shift+.	⇧⌘.	Load next page of memory
Ctrl+Shift+,	⇧⌘,	Load previous page of memory
Ctrl+F11	⇧⌘F11	Launch in run mode
Ctrl+R	⌘R	Resume and break when execution reaches the current line
Ctrl+Alt+B	^⌘B	Sets whether or not any breakpoint should suspend execution
F5	F5	Step into
F6	F6	Step over
F7	F7	Step return
F8	F8	Resume
Ctrl+F2	⌘F2	Terminate
Ctrl+Shift+B	⇧⌘B	Creates or removes a breakpoint
Ctrl+T	⌘T	Toggle visibility of the Memory Monitors Pane
Shift+F5	⇧F5	Toggles enablement of debug step filters

Git

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Left	⌘←	Collapse Working Tree
Ctrl+#	⌘⌘3	Commit
Ctrl+C	⌘C	Copy Path to Clipboard
Ctrl+O	⌘O	Show the quick outline for a unified diff
F2	F2	Rename Branch

Ctrl+Shift+K	^⇧K	Commit changes to Git
Ctrl+Shift+U	^⇧U	Update changes from Git

Project

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+B	⌘B	Build all projects

Oracle

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+F9	^F9	Compile

Utility

Shortcut for Windows&Linux	Shortcut for macOS	Action
Ctrl+Shift+C	^⇧C	Advanced Copy
Ctrl+Shift+V	^⇧V	Paste with extra settings
CTRL+ALT+PAGE_UP CTRL+ALT+PAGE_DOWN		Switch between tabs in database object Properties view
Alt+Insert	⌘+N	Create Object It is available both in the Database Navigator and the Properties Editor .

Help

Shortcut for Windows&Linux	Shortcut for macOS	Action

F1	F1	Open the documentation
----	----	------------------------

References

- * - Current query is the query under cursor or the selected text. Query is separated from other script queries by delimiter (; by default) or by empty lines.
- ** - Current script is a set of all queries in the current SQL file. If there is a text selection then only queries in this selection are processed. Queries are separated from each other with a delimiter (; by default).
- *** - Current connection detected from active window and selection. If active (focused) window is SQL editor or database object editor then the current connection is the same as in this editor. If the active window is the database navigator then the active connection is the "owner" connection of the currently selected element. In other cases there is no current connection and DBeaver will ask you to choose the connection explicitly.

Certificate management

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[System-specific Features](#)

[Managing Certificates](#)

[Import Certificates](#)

[Viewing and Deleting Certificates](#)

You can manage trust store settings in DBeaver to enable secured driver downloads and establish SSL connections. This is achieved by adding or removing certificates from the **Custom Truststore**. Certificates are integral to establishing secure connections, allowing necessary driver files to be downloaded, and enabling SSL connections to databases. Without the appropriate certificates, you might encounter an error message.

System-specific Features

On Windows operating systems, a system-specific trust store, known as the **Windows Truststore**, is available. However, this is a read-only trust store.

Note: You cannot import any certificates into this system trust store, it will appear empty. This Windows-specific option does not display on MacOS or Linux systems.

Viewing certificates is permitted for the **Java Truststore**, but modifications are impossible.

To modify or manage certificates, DBeaver provides the **Custom Truststore**.

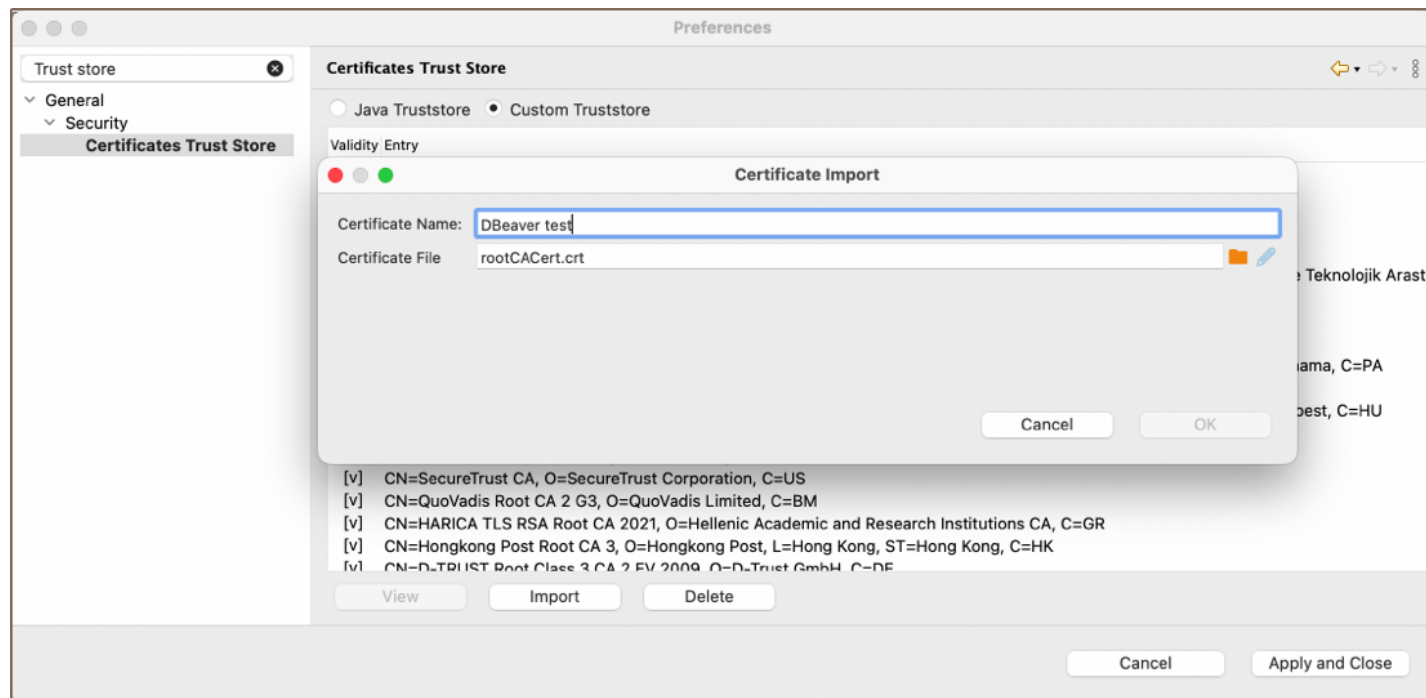
Managing Certificates

Import Certificates

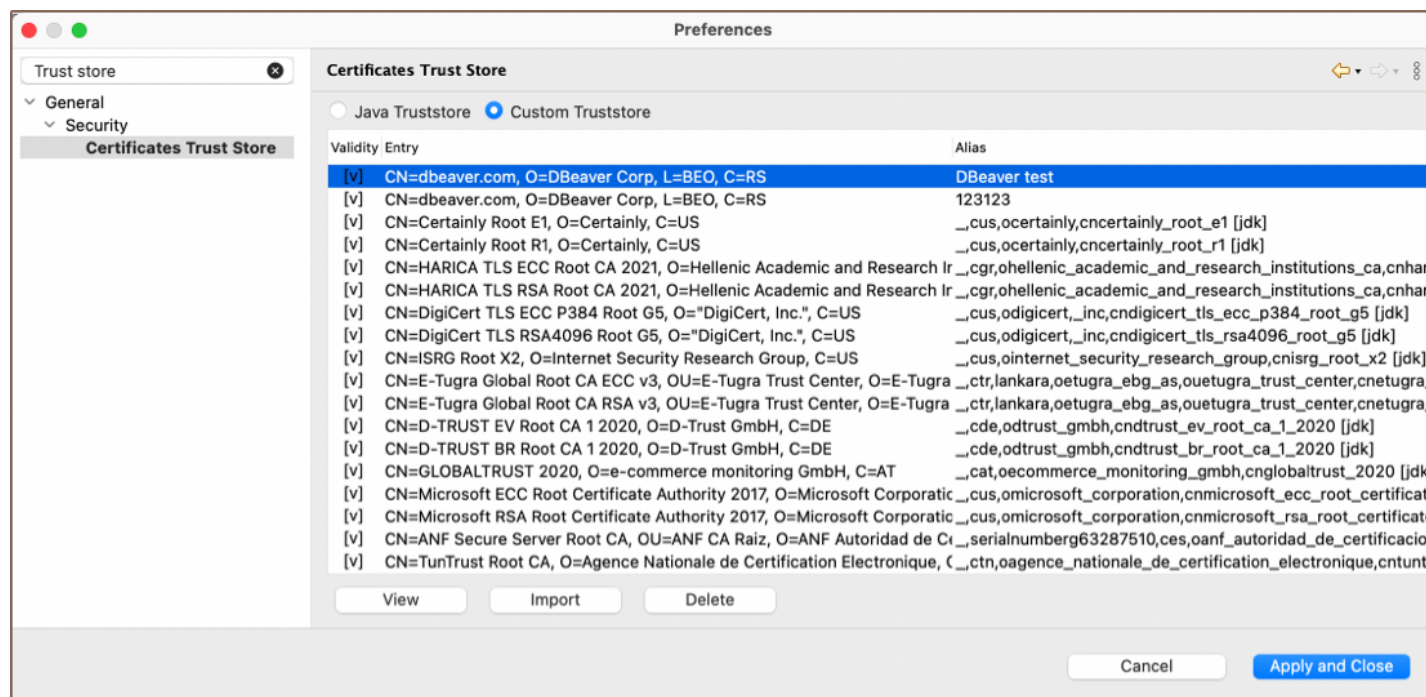
To import a certificate into the **Custom Truststore**, follow the steps below:

1) Navigate to **Windows -> Preferences -> General -> Security -> Certificates Trust Store** and click the **Import** button.

2) Browse your file system, locate the certificate file (**.cer** or **.crt**), and select it.



3) Assign an alias to the certificate and click **OK**.



4) After applying changes to the trust store and clicking **Apply and Close**, DBeaver prompts a dialog suggesting a restart for the trust store changes to take effect.

Once a certificate is added to the trust store, it enables secure connections and authentication. You can verify the validity of a certificate based on its expiration date.

Viewing and Deleting Certificates

In the **Custom Truststore** section, select the certificate and click the **View** button to see detailed information about the certificate. If you need to remove a certificate from the trust store, select the appropriate certificate and click the **Delete** button.

Configuration files

Table of contents

- [Overview](#)
- [Configuration files](#)
 - [dbeaver.ini](#)
 - [How to Locate the dbeaver.ini](#)
 - [Other configuration files](#)
 - [Location of the other configuration files](#)

Overview

DBeaver utilizes several configuration files to tailor the application to user needs. These files include:

- `dbeaver.ini`
- `data-sources.json`
- `tasks.json`

Configuration files

Each of the listed files serves a specific purpose, from storing data sources and tasks to overall application settings.

dbeaver.ini

Through modifications to `dbeaver.ini`, you can apply licenses, manage drivers, enhance security, adjust user interface settings, and more.

Configuration	Description
License Management	Configure DBeaver Enterprise Edition with a specific license.
Driver Management	Manage and automate the provisioning of database drivers.
Administrative Preference	Implement restrictions to enhance application security and control over available

Restrictions	functionalities.
AI-Smart Assistance	Control the AI-Smart Assistance feature, including the option to disable it.
Command-Line Parameters	Customize DBeaver startup and operational parameters through the command line.
JDBC Tracing	Enable tracing for JDBC operations to diagnose and troubleshoot issues.

Tip: DBeaver also utilizes preference files to further customize and optimize the user experience. These preference files allow users to fine-tune the application settings and behaviors according to their specific needs.

For detailed information, refer to the [Admin manage preferences](#) article.

How to Locate the dbeaver.ini

The `dbeaver.ini` file's location varies depending on the operating system and the method of installation. As a general rule, you can find `dbeaver.ini` in the directory where DBeaver is installed. For macOS, specifically, it is located in the `Contents/Eclipse` subfolder of the application.

Here are the common locations based on the operating system and installation method:

Operating System	Location of <code>dbeaver.ini</code>
macOS	<code>./Applications/[DBeaver Edition].app/Contents/Eclipse/dbeaver.ini</code>
Linux	<code>/usr/share/dbeaver-[edition]/dbeaver.ini</code> for installations via <code>.deb</code> , <code>.rpm</code> , or PPA packages.
Windows	<code>C:\Program Files\[DBeaver Edition]\dbeaver.ini</code> for a default installation.

Replace `[DBeaver Edition]` with your specific installation, such as `Dbeaver` for Community Edition, `DBeaverLite` for Lite Edition, `DBeaverEE` for Enterprise Edition, `DBeaverUltimate` , and `DBeaverTeam` for Ultimate and Team Editions respectively.

Other configuration files

The `data-sources.json` and `tasks.json` configuration files in DBeaver serve crucial roles, particularly when it comes to the portability of database connections and tasks configurations across different DBeaver installations.

--	--

Configuration file	Capabilities
<code>data-sources.json</code>	The value of <code>data-sources.json</code> lies in its ability to facilitate the migration of database connections from one DBeaver installation to another. This is particularly useful when moving to a new workstation, setting up a team member with the same database connections, or ensuring consistency across different environments for testing or development purposes.
<code>tasks.json</code>	This file stores the configuration for various tasks, such as SQL scripts execution, data transfer operations, and other automated tasks. Similar to <code>data-sources.json</code> , <code>tasks.json</code> allows for the easy migration of configured tasks from one DBeaver instance to another.

Location of the other configuration files

The location of `data-sources.json` and `tasks.json` files is consistent across all operating systems, relative to the DBeaver workspace:

- **data-sources.json:** `./workspace6/General/.dbeaver/data-sources.json`
- **tasks.json:** `./workspace6/General/.dbeaver/tasks.json`

Tip: Detailed instructions on finding your workspace are available in [Workspace Location](#) article.

Enterprise edition

Table of contents

[Overview](#)

[Enterprise Edition features](#)

[Subscription model](#)

[Integrated database drivers](#)

[Supported databases](#)

[Relational databases](#)

[Non-Relational databases](#)

Overview

[DBeaver Enterprise Edition](#) is ideal for users who need a powerful set of tools for complex data environments. This edition goes beyond simple database tasks, enhancing data visualization, query development, security features, and administrative tasks. It is a vital tool for professionals looking to improve their productivity and manage data more effectively.

Note: The DBeaver Enterprise Edition is designed for individual use only.

Enterprise Edition features

The Enterprise Edition provides specialized tools and features that cater to the needs of businesses:

- **Data visualization**
 - **Data filtering and sorting:** Provides tools to filter and sort data directly within the database interface.
[Learn more.](#)
 - **Color coding and highlighting:** Simplifies data analysis by color-coding and highlighting table rows and fields based on conditions. [Learn more.](#)

- **Functional panels:** Enhance data manipulation by offering additional space in the Data Editor for managing complex data types, accessible via tabs in a dedicated panel. [Learn more.](#)
- **Tableau integration:** Facilitates integration with Tableau for advanced data visualization and analysis. [Learn more.](#)
- **Query development**
 - **SQL Editor:** Provides code completion for easier writing, auto-formatting, spelling checks, and support for variables. [Learn more.](#)
 - **Execution plan visualization:** Provides detailed visualizations of SQL execution plans. [Learn more.](#)
 - **Additional tools for query creation:**
 - **Visual Query Builder (VQB):** Provides a graphical interface to construct SQL queries without writing code. [Learn more.](#)
 - **AI assistant:** Integrates an AI assistant and AI Chat to offer guidance and automation within the SQL Editor. [Learn more.](#)
 - **SQL debugger:** Offers an SQL debugger specifically designed for PostgreSQL, aiding in query development and troubleshooting. [Learn more](#)
- **Administrative tasks**
 - **Reverse engineering (ERD):** Provides capabilities for reverse engineering databases into entity-relationship diagrams. [Learn more.](#)
 - **Forward engineering:**
 - **Metadata editor:** Enables the management and editing of database metadata.
 - **Edit mode for ER-Diagrams:** Offers functionalities for editing entity-relationship diagrams to aid in database design. [Learn more.](#)
 - **Administrative tools:**
 - **Backup and Restore:** Offers functionalities for backing up and restoring databases, ensuring data safety. [Learn more.](#)

- **Analyze and vacuum:** Tools for optimizing database performance by analyzing and vacuuming database objects.
- **Permission and user management:** Manages database permissions and user accounts.
- **Schema comparison:** Provides tools for comparing database schemas either within the same database or across different databases. [Learn more.](#)
- **Data comparison:** Offers tools for comparing data across databases or within different parts of the same database. [Learn more.](#)
- **Mock data generator:** Provides tools for generating mock data, useful for testing and development purposes. [Learn more.](#)
- **Data migration**
 - **Export and Import from files:** Supports data export and import in up to 10 different file formats. [Learn more.](#)
 - **Direct transfer between databases:** Provides tools for efficient data migration directly between different databases or systems. [Learn more.](#)
- **Security**
 - **Database authentication:** Implements database authentication mechanisms to secure access.
 - **Master and Project Password:**
 - **Master password:** Enables the use of a master password to enhance security across the DBeaver. [Learn more.](#)
 - **Project password:** Facilitates the use of project-specific password to secure projects. [Learn more.](#)
- **Automation**
 - **Task manager:** A tool for managing and monitoring database tasks, enhancing productivity and organization. [Learn more.](#)
 - **Task scheduler:** Allows for scheduling database tasks, automating routine operations. [Learn more.](#)

- **Git integration:** Supports integration with Git for version control, and change tracking. [Learn more.](#)

Subscription model

The DBeaver Enterprise Edition is offered as an annual subscription, which should be renewed to continue receiving updates and support.

[Discover more about the DBeaver Enterprise Edition and subscribe.](#)

Integrated database drivers

The Enterprise Edition has a comprehensive set of JDBC drivers, offering connectivity to various databases without needing separate downloads and configurations. Additionally, a versatile ODBC driver is supported for broader database compatibility.

For additional details on utilizing the ODBC driver, please consult the [ODBC-JDBC driver](#) article.

Supported databases

The Enterprise Edition provides support for a wide range of databases. Below are tables listing the databases available and specifying whether the Enterprise Edition offers enhanced features compared to the Community Edition.

* Enhanced features may include additional metadata capabilities, such as the ability to manage and interact with specific database objects unique to certain systems. For instance, managing Datashares in Redshift or handling Sequences** in Netezza. These features also cover various advanced authentication methods, and numerous other functionalities designed to improve security and data management.

Relational databases

Database name	Enhanced features* compared to the Community Edition
Altibase	-
Apache Arrow	Not available in Community Edition

Apache Calcite Avatica	✓
Apache Kylin	-
Azure Databricks	✓
Azure SQL Server	✓
Babelfish via TDS (beta)	✓
CUBRID	✓
Cache	-
ClickHouse	-
ClickHouse (Legacy)	✓
CloudSQL - MySQL	Not available in Community Edition
CloudSQL - PostgreSQL	Not available in Community Edition
CloudSQL - SQL Server	Not available in Community Edition
CockroachDB	-
CrateDB	-
CrateDB (Legacy)	-
Dameng	-
Data Virtuality	-
Db2 for IBM i	✓
Db2 for LUW	-
Db2 for z/OS	✓
Denodo 8	-
Derby Embedded	-
Derby Server	-
Dremio	-
DuckDB	-

EDB	✓
Exasol	✓
Firebird	✓
Fujitsu Enterprise Postgres	Not available in Community Edition
Google AlloyDB	Not available in Community Edition
Google Cloud Spanner	✓
Greenplum	✓
H2 Embedded	✓
H2 Embedded V.2	-
H2 Server	✓
H2GIS Embedded	-
H2GIS Server	-
HANA	✓
HSQL Embedded	-
HSQL Server	-
Informix	✓
Ingres	-
InterSystems IRIS	-
JDBCX	-
Jennifer	-
Kognitio	-
MS Access (UCanAccess)	-
MariaDB	✓
Materialize	-
MaxDB	-

Mimer SQL	-
MonetDB	-
MySQL	✓
MySQL 5 (Legacy)	✓
NDB Cluster	-
NetSuite	Not available in Community Edition
Netezza	✓
NuoDB	-
ODBC	Not available in Community Edition
OceanBase	-
Ocient	-
OmniSci (formerly MapD)	-
OpenEdge	-
OpenSearch	✓
Oracle	✓
Pervasive SQL	-
PostgreSQL	✓
PrestoDB	-
PrestoSQL	-
Raima	Not available in Community Edition
Redshift	✓
RisingWave	-
SAP ASE jConnect	-
SQL Server	✓
SQL Server (Old driver, jTDS)	✓

SQLite	✓
SQLite Crypt	Not available in Community Edition
SQream DB	-
Salesforce	Not available in Community Edition
Salesforce Data Cloud	-
SingleStore	✓
Snowflake	✓
StarRocks	-
Sybase jConnect	✓
Sybase jTDS	✓
Teradata	✓
TiDB	-
Trino	-
Vertica	✓
Virtuoso	-
Yellowbrick	-
YugabyteDB	-

Non-Relational databases

Database name	Enhanced features [*] compared to the Community Edition
Apache Drill	✓
Apache Hive	✓
Apache Ignite	-
Apache Kyuubi	-
Apache Phoenix	-

Apache Spark	-
Athena	✓
CSV	-
Cassandra	Not available in Community Edition
Cloudera Impala	-
CosmosDB (Cassandra)	Not available in Community Edition
CosmosDB (MongoDB)	Not available in Community Edition
CouchDB	Not available in Community Edition
Couchbase	Not available in Community Edition
Couchbase 5+	Not available in Community Edition
DBF	-
DocumentDB	Not available in Community Edition
DynamoDB	Not available in Community Edition
Elasticsearch	✓
Firestore	Not available in Community Edition
Gemfire XD	-
Google BigQuery	✓
Google Cloud Bigtable	Not available in Community Edition
InfluxDB	Not available in Community Edition
InfluxDB 2	Not available in Community Edition
InfluxDB 3	Not available in Community Edition
Keyspaces	Not available in Community Edition
Machbase	-
MongoDB	Not available in Community Edition
Neo4j	✓

Neptune	Not available in Community Edition
Open Distro Elasticsearch	✓
OrientDB	-
Redis	Not available in Community Edition
ScyllaDB	Not available in Community Edition
SnappyData	-
Solr	-
TDengine	✓
TDengine Cloud	✓
TimescaleDB	-
Timestream	Not available in Community Edition
Yugabyte CQL	Not available in Community Edition

Lite edition

Table of contents

[Overview](#)

[Lite Edition features](#)

[Subscription model](#)

[Integrated database drivers](#)

[Supported databases](#)

[Relational databases](#)

[Non-Relational databases](#)

Overview

[DBeaver Lite Edition](#) is tailored for users who need basic tools to work with databases. This version offers a streamlined interface that focuses on simple database tasks like viewing data and running queries. It is especially useful for business users who need an easy-to-use tool.

Note: The DBeaver Lite Edition is designed for individual use only.

Lite Edition features

Key features included in the Lite Edition are designed to support everyday database tasks with minimal hassle:

- **Data Visualization**
 - **Color coding and highlighting:** Simplifies data analysis by color-coding and highlighting table rows and fields based on conditions. [Learn more.](#)
 - **Data filtering and sorting:** Provides tools to filter and sort data directly within the database interface. [Learn more.](#)
 - **Analytical charts:** Offers tool for creating analytical charts to represent data visually. [Learn more.](#)

- **Entity relationship diagrams:** Visualize database schemas through entity-relationship diagrams. [Learn more.](#)
- **Data import/export**
 - **Office formats support (XLS):** Enables users to import and export data in Office formats, specifically Excel (**XLS**). [Learn more.](#)
- **Query development**
 - **AI assistant in SQL Editor:** Integrates an AI assistant and AI Chat to offer guidance and automation within the SQL Editor. [Learn more.](#)
 - **Search in query history:** Enables searching through previous queries for easier management and reuse.
 - **Spelling:** Provides spelling verification to improve the quality and accuracy of SQL scripts. [Learn more.](#)
 - **Visual query builder:** Provides a graphical interface to construct SQL queries without writing code. [Learn more.](#)
- **Comprehensive security support:** Ensures secure access to databases within your environment, even if they are protected by Single Sign-On (SSO), Kerberos, SSL, and other security measures.

Subscription model

The DBeaver Lite Edition is offered as an annual subscription, which should be renewed to continue receiving updates and support.

[Discover more about the DBeaver Lite Edition and subscribe.](#)

Integrated database drivers

The Lite Edition has a comprehensive set of JDBC drivers, offering connectivity to various databases without needing separate downloads and configurations. Additionally, a versatile ODBC driver is supported for broader database compatibility.

For additional details on utilizing the ODBC driver, please consult the [ODBC-JDBC driver](#) article.

Supported databases

The Lite Edition provides support for a wide range of databases. Below are tables listing the databases available and specifying whether the Lite Edition offers enhanced features compared to the Community Edition.

* Enhanced features may include additional metadata capabilities, such as the ability to manage and interact with specific database objects unique to certain systems. For instance, managing Roles** in Redshift. These features also cover various advanced authentication methods, and numerous other functionalities designed to improve security and data management.

Relational databases

Database name	Enhanced features* compared to the Community Edition
Altibase	-
Apache Arrow	Not available in Community Edition
Apache Calcite Avatica	✓
Apache Kylin	-
Azure Databricks	✓
Azure SQL Server	✓
Babelfish via TDS (beta)	✓
CUBRID	✓
Cache	-
ClickHouse	-
ClickHouse (Legacy)	✓
CloudSQL - MySQL	Not available in Community Edition
CloudSQL - PostgreSQL	Not available in Community Edition
CloudSQL - SQL Server	Not available in Community Edition
CockroachDB	-

CrateDB	-
CrateDB (Legacy)	-
Dameng	-
Data Virtuality	-
Db2 for IBM i	✓
Db2 for LUW	-
Db2 for z/OS	✓
Denodo 8	-
Derby Embedded	-
Derby Server	-
Dremio	-
DuckDB	-
EDB	✓
Exasol	✓
Firebird	✓
Fujitsu Enterprise Postgres	Not available in Community Edition
Google AlloyDB	Not available in Community Edition
Google Cloud Spanner	✓
Greenplum	✓
H2 Embedded	✓
H2 Embedded V.2	-
H2 Server	✓
H2GIS Embedded	-
H2GIS Server	-
HANA	✓

HSQL Embedded	-
HSQL Server	-
Informix	✓
Ingres	-
InterSystems IRIS	-
JDBCX	-
Jennifer	-
Kognitio	-
MS Access (UCanAccess)	-
MariaDB	✓
Materialize	-
MaxDB	-
Mimer SQL	-
MonetDB	-
MySQL	✓
MySQL 5 (Legacy)	✓
NDB Cluster	-
NetSuite	Not available in Community Edition
Netezza	✓
NuoDB	-
ODBC	Not available in Community Edition
OceanBase	-
Ocient	-
OmniSci (formerly MapD)	-
OpenEdge	-

OpenSearch	✓
Oracle	✓
Pervasive SQL	-
PostgreSQL	✓
PrestoDB	-
PrestoSQL	-
Raima	Not available in Community Edition
Redshift	✓
RisingWave	-
SAP ASE jConnect	-
SQL Server	✓
SQL Server (Old driver, jTDS)	✓
SQLite	✓
SQLite Crypt	Not available in Community Edition
SQream DB	-
Salesforce	Not available in Community Edition
Salesforce Data Cloud	-
SingleStore	✓
Snowflake	✓
StarRocks	-
Sybase jConnect	✓
Sybase jTDS	✓
Teradata	✓
TiDB	-
Trino	-

Vertica	✓
Virtuoso	-
Yellowbrick	-
YugabyteDB	-

Non-Relational databases

Database name	Enhanced features [*] compared to the Community Edition
Apache Drill	✓
Apache Hive	✓
Apache Ignite	-
Apache Kyuubi	-
Apache Phoenix	-
Apache Spark	-
Athena	✓
CSV	-
Cassandra	Not available in Community Edition
Cloudera Impala	-
CosmosDB (Cassandra)	Not available in Community Edition
CosmosDB (MongoDB)	Not available in Community Edition
CouchDB	Not available in Community Edition
Couchbase	Not available in Community Edition
Couchbase 5+	Not available in Community Edition
DBF	-
DocumentDB	Not available in Community Edition
Elasticsearch	✓

Gemfire XD	-
Google BigQuery	✓
InfluxDB	Not available in Community Edition
InfluxDB 2	Not available in Community Edition
InfluxDB 3	Not available in Community Edition
Keyspaces	Not available in Community Edition
Machbase	-
MongoDB	Not available in Community Edition
Neo4j	✓
Neptune	Not available in Community Edition
Open Distro Elasticsearch	✓
OrientDB	-
Redis	Not available in Community Edition
ScyllaDB	Not available in Community Edition
SnappyData	-
Solr	-
TDengine	✓
TDengine Cloud	✓
TimescaleDB	-
Timestream	Not available in Community Edition
Yugabyte CQL	Not available in Community Edition

Ultimate edition

Table of contents

[Overview](#)

[Ultimate Edition features](#)

[Advanced Enterprise features in Ultimate Edition](#)

[Subscription model](#)

[Integrated database drivers](#)

[Cloud Explorer databases](#)

[Supported databases](#)

[Relational databases](#)

[Non-Relational databases](#)

Overview

[DBeaver Ultimate Edition](#) is perfect for users who require a powerful set of tools for managing complex data environments and are also seeking enhanced cloud capabilities. This edition builds on the features of the Enterprise Edition by adding specialized tools for cloud database management and Kafka data streams. It offers seamless integration and efficient management of cloud data sources, making it an excellent choice for users who need to extend their data handling into the cloud.

Note: The DBeaver Ultimate Edition is designed for individual use only.

Ultimate Edition features

The Ultimate Edition offering unique capabilities designed for complex, multi-platform data management tasks with a strong emphasis on enhanced cloud integration:

- **Cloud Explorer:** A tool for managing and connecting cloud-hosted databases. [Learn more](#)

It offers integration with various cloud services:

- **Amazon Web Services** [Learn more](#)
- **Google Cloud Platform** [Learn more](#)
- **Microsoft Azure** [Learn more](#)
- **Cloud Storage support:** Enables file management within cloud environments of Amazon Web Services and Google Cloud Platform. [Learn more](#)
- **Kafka support:** Provides integration with Apache Kafka.

Advanced Enterprise features in Ultimate Edition

In addition to its unique features, the Ultimate Edition utilizes all the advanced capabilities from the Enterprise Edition, including:

- **Data visualization**
 - **Data filtering and sorting:** Provides tools to filter and sort data directly within the database interface. [Learn more.](#)
 - **Color coding and highlighting:** Simplifies data analysis by color-coding and highlighting table rows and fields based on conditions. [Learn more.](#)
 - **Functional panels:** Enhance data manipulation by offering additional space in the Data Editor for managing complex data types, accessible via tabs in a dedicated panel. [Learn more.](#)
 - **Tableau integration:** Facilitates integration with Tableau for advanced data visualization and analysis. [Learn more.](#)
- **Query development**
 - **SQL Editor:** Provides code completion for easier writing, auto-formatting, spelling checks, and support for variables. [Learn more.](#)
 - **Execution plan visualization:** Provides detailed visualizations of SQL execution plans. [Learn more.](#)
 - **Additional tools for query creation:**

- **Visual Query Builder (VQB):** Provides a graphical interface to construct SQL queries without writing code. [Learn more.](#)
- **AI assistant:** Integrates an AI assistant and AI Chat to offer guidance and automation within the SQL Editor. [Learn more.](#)
- **SQL debugger:** Offers an SQL debugger specifically designed for PostgreSQL, aiding in query development and troubleshooting. [Learn more](#)
- **Administrative tasks**
 - **Reverse engineering (ERD):** Provides capabilities for reverse engineering databases into entity-relationship diagrams. [Learn more.](#)
 - **Forward engineering:**
 - **Metadata editor:** Enables the management and editing of database metadata.
 - **Edit mode for ER-Diagrams:** Offers functionalities for editing entity-relationship diagrams to aid in database design. [Learn more.](#)
- **Administrative tools:**
 - **Backup and Restore:** Offers functionalities for backing up and restoring databases, ensuring data safety. [Learn more.](#)
 - **Analyze and vacuum:** Tools for optimizing database performance by analyzing and vacuuming database objects.
 - **Permission and user management:** Manages database permissions and user accounts.
 - **Schema comparison:** Provides tools for comparing database schemas either within the same database or across different databases. [Learn more.](#)
 - **Data comparison:** Offers tools for comparing data across databases or within different parts of the same database. [Learn more.](#)
 - **Mock data generator:** Provides tools for generating mock data, useful for testing and development purposes. [Learn more.](#)

- **Data migration**

- **Export and Import from files:** Supports data export and import in up to 10 different file formats. [Learn more.](#)
- **Direct transfer between databases:** Provides tools for efficient data migration directly between different databases or systems. [Learn more.](#)

- **Security**

- **Database authentication:** Implements database authentication mechanisms to secure access.
- **Master and Project Password:**
 - **Master password:** Enables the use of a master password to enhance security across the DBeaver. [Learn more.](#)
 - **Project password:** Facilitates the use of project-specific password to secure projects. [Learn more.](#)

- **Automation**

- **Task manager:** A tool for managing and monitoring database tasks, enhancing productivity and organization. [Learn more.](#)
- **Task scheduler:** Allows for scheduling database tasks, automating routine operations. [Learn more.](#)
- **Git integration:** Supports integration with Git for version control, and change tracking. [Learn more.](#)

Subscription model

The DBeaver Ultimate Edition is offered as an annual subscription, which should be renewed to continue receiving updates and support.

[Discover more about the DBeaver Ultimate Edition and subscribe.](#)

Integrated database drivers

The Ultimate Edition has a comprehensive set of JDBC drivers, offering connectivity to various databases without needing separate downloads and configurations. Additionally, a versatile ODBC driver is supported for broader database compatibility.

For additional details on utilizing the ODBC driver, please consult the [ODBC-JDBC driver](#) article.

This edition also includes the Cloud Explorer feature, which provides access to cloud databases such as those on Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure. Cloud Explorer enables centralized authentication and automatic configuration, simplifying the management of cloud database connections.

Cloud Explorer databases

For a list of supported databases, please refer to the [related section](#) in the Cloud Explorer guide.

Supported databases

The Ultimate Edition provides support for a wide range of databases. Below are tables listing the databases available and specifying whether the Ultimate Edition offers enhanced features compared to the Community Edition.

* Enhanced features may include additional metadata capabilities, such as the ability to manage and interact with specific database objects unique to certain systems. For instance, managing Datashares in Redshift or handling Sequences** in Netezza. These features also cover various advanced authentication methods, and numerous other functionalities designed to improve security and data management.

Relational databases

Database name	Enhanced features* compared to the Community Edition	Available through Cloud Explorer
Altibase	-	-
Apache Arrow	Not available in Community Edition	-

Apache Calcite Avatica	✓	-
Apache Kylin	-	-
Azure Databricks	✓	-
Azure SQL Server	✓	✓
Babelfish via TDS (beta)	✓	-
CUBRID	✓	-
Cache	-	-
ClickHouse	-	-
ClickHouse (Legacy)	✓	-
CloudSQL - MySQL	Not available in Community Edition	✓
CloudSQL - PostgreSQL	Not available in Community Edition	✓
CloudSQL - SQL Server	Not available in Community Edition	✓
CockroachDB	-	-
CrateDB	-	-
CrateDB (Legacy)	-	-
Dameng	-	-
Data Virtuality	-	-
Db2 for IBM i	✓	-
Db2 for LUW	-	-
Db2 for z/OS	✓	-
Denodo 8	-	-
Derby Embedded	-	-
Derby Server	-	-
Dremio	-	-
DuckDB	-	-

EDB	✓	-
Exasol	✓	-
Firebird	✓	-
Fujitsu Enterprise Postgres	Not available in Community Edition	-
Google AlloyDB	Not available in Community Edition	✓
Google Cloud Spanner	✓	✓
Greenplum	✓	-
H2 Embedded	✓	-
H2 Embedded V.2	-	-
H2 Server	✓	-
H2GIS Embedded	-	-
H2GIS Server	-	-
HANA	✓	-
HSQL Embedded	-	-
HSQL Server	-	-
Informix	✓	-
Ingres	-	-
InterSystems IRIS	-	-
JDBCX	-	-
Jennifer	-	-
Kognitio	-	-
MS Access (UCanAccess)	-	-
MariaDB	✓	-
Materialize	-	-
MaxDB	-	-

Mimer SQL	-	-
MonetDB	-	-
MySQL	✓	✓
MySQL 5 (Legacy)	✓	-
NDB Cluster	-	-
NetSuite	Not available in Community Edition	-
Netezza	✓	-
NuoDB	-	-
ODBC	Not available in Community Edition	-
OceanBase	-	-
Ocient	-	-
OmniSci (formerly MapD)	-	-
OpenEdge	-	-
OpenSearch	✓	-
Oracle	✓	✓
Pervasive SQL	-	-
PostgreSQL	✓	✓
PrestoDB	-	-
PrestoSQL	-	-
Raima	Not available in Community Edition	-
Redshift	✓	✓
RisingWave	-	-
SAP ASE jConnect	-	-
SQL Server	✓	✓
SQL Server (Old driver, jTDS)	✓	-

SQLite	✓	-
SQLite Crypt	Not available in Community Edition	-
SQream DB	-	-
Salesforce	Not available in Community Edition	-
Salesforce Data Cloud	-	-
SingleStore	✓	-
Snowflake	✓	-
StarRocks	-	-
Sybase jConnect	✓	-
Sybase jTDS	✓	-
Teradata	✓	-
TiDB	-	-
Trino	-	-
Vertica	✓	-
Virtuoso	-	-
Yellowbrick	-	-
YugabyteDB	-	-

Non-Relational databases

Database name	Enhanced features* compared to the Community Edition	Available through Cloud Explorer
Apache Drill	✓	-
Apache Hive	✓	-
Apache Ignite	-	-
Apache Kyuubi	-	-

Apache Phoenix	-	-
Apache Spark	-	-
Athena	✓	✓
CSV	-	-
Cassandra	Not available in Community Edition	-
Cloudera Impala	-	-
CosmosDB (Cassandra)	Not available in Community Edition	✓
CosmosDB (MongoDB)	Not available in Community Edition	✓
CouchDB	Not available in Community Edition	-
Couchbase	Not available in Community Edition	-
Couchbase 5+	Not available in Community Edition	-
DBF	-	-
DocumentDB	Not available in Community Edition	✓
DynamoDB	Not available in Community Edition	✓
Elasticsearch	✓	-
Firestore	Not available in Community Edition	✓
Gemfire XD	-	-
Google BigQuery	✓	✓
Google Cloud Bigtable	Not available in Community Edition	✓
InfluxDB	Not available in Community Edition	-
InfluxDB 2	Not available in Community Edition	-
InfluxDB 3	Not available in Community Edition	-
Kafka (ksq DB)	Not available in Community Edition	-
Keyspaces	Not available in Community Edition	✓
Machbase	-	-

MongoDB	Not available in Community Edition	-
Neo4j	✓	-
Neptune	Not available in Community Edition	✓
Open Distro Elasticsearch	✓	-
OrientDB	-	-
Redis	Not available in Community Edition	-
ScyllaDB	Not available in Community Edition	-
SnappyData	-	-
Solr	-	-
TDengine	✓	-
TDengine Cloud	✓	-
TimescaleDB	-	-
Timestream	Not available in Community Edition	✓
Yugabyte CQL	Not available in Community Edition	-

How to migrate to Team edition

Table of contents

[Problem Overview](#)

[DBeaver Team Edition](#)

[Authentication](#)

[Working in Desktop client](#)

[Working in the Web client](#)

[Projects, Teams, and Roles in DBeaver Team Edition](#)

[Working with database credentials](#)

[Database Tasks and Scheduling](#)

[Version Control](#)

[Migration overview](#)

[Migration steps](#)

This article explains the key differences between single-user DBeaver products and DBeaver Team Edition. We will look at vital information for users considering or undergoing migration.

Problem Overview

DBeaver desktop, which is a versatile database management suite, includes editions like Community, Lite, Enterprise, and Ultimate. These editions are primarily designed for individual use. All configurations, database connections, SQL scripts, diagrams, and other related files are stored on the local machine.

While these single-user editions provide powerful tools for individual database management, they are not optimized for team collaboration. Sharing resources between multiple users via email, message applications, or disk shares can lead to inconveniences and the need to constantly synchronize.

Using shared version control systems like Git is a potential solution, but this approach has its own challenges, particularly for non-programmers who might find Git concepts a bit complex. In addition, merge conflicts can sometimes disrupt the configuration.

On the security front, the single-user editions of DBeaver do not mandate authentication. This means that anyone who gains access to a computer can also access all saved configurations and sensitive data, such as database server credentials. Though workarounds like workspace master passwords or project-based passwords with configuration encryption exist, they can be especially cumbersome, especially in larger organizations.

Recognizing these challenges, DBeaver has introduced the **DBeaver Team Edition**, tailored for multi-user collaboration.

DBeaver Team Edition

DBeaver Team Edition is a multi-user application designed for distributed team work. The solution comprises:

- DBeaver Team Edition server (For deployment details, refer to [Deployment](#))
- DBeaver Team Edition desktop client (With functionality similar to classic DBeaver)
- DBeaver Team Edition web client (Comparable to [CloudBeaver](#))

Important: DBeaver Team Edition requires users to deploy the server on their own infrastructure. It is not provided as a hosted solution or a Software as a Service (SaaS) offering.

This article does not discuss server installation and configuration. For that information, consult the links provided above. Instead, this document focuses on guiding users through the process of migrating their existing configurations to adapt to the multi-user environment.

Authentication

To utilize DBeaver Team Edition, each user is required to authenticate within the system. Various authentication methods can be configured by system administrators to suit their organizational needs:

- Username and Password
- Cloud provider authentication (AWS, Google, Azure)
- Windows NTLM

The authentication dialog appears on the initial screen when launching the desktop client or accessing the web application.

Working in Desktop client

The DBeaver Team Edition client is very similar to DBeaver Ultimate in terms of user interface, editors, and views. However, there are some key differences:

- Database connections, SQL scripts, and other resources are stored on a remote server, fetched to the local machine on demand.
- Saved database credentials are securely stored and encrypted on the server. In the case of a desktop client, these credentials are requested on-demand and are never saved on the local disk.
- The database task scheduler initiates tasks on the remote server, not the local machine.
- The User interface varies slightly depending on the user's roles (see below).
- Git integration is omitted from desktop clients as files are not saved on the local file system. [Git Integration](#) is available through the administration panel.
- Quick file search functionality is not available in the DBeaver Team Edition client.

Working in the Web client

The Web client is very similar to the classic CloudBeaver, with the following notable differences:

- DBeaver Team Edition supports an infinite number of global projects (in classic CloudBeaver there is only one global project).
- The user interface varies based on the user's roles (see below).

Projects, Teams, and Roles in DBeaver Team Edition

In DBeaver Team Edition, the concept of projects is similar to that in classic desktop versions:

- A project contains database connections, SQL scripts, datasets, and other resources.
- Every user has access only to some projects (it is controlled by the administrator).

A team functions as a group of users, and users can belong to multiple teams. Access to projects can be granted either to a team or to individual users.

Roles serve as access levels for users, and each user can have only one role:

Capability \ Roles	Administrator	Developer	Manager	Editor	Viewer
Full system administration	+	-	-	-	-
Manage users and licenses	+	-	-	-	-
Create/delete projects	+	-	-	-	-
Edit connections	+	+	-	-	-
View connections	+	+	+	-	-
Create/edit/delete resources and scripts	+	+	+	-	-
Create/edit datasets	+	+	+	-	-
Create/edit folders	+	+	+	-	-
Save datasets to private project	+	+	+	-	-
Use DBeaver desktop	+	+	+	-	-
In-line data editing	+	+	+	+	-
View, browse, filter, export datasets	+	+	+	+	+

Working with database credentials

Unlike other products, in DBeaver Team Edition, credentials are entered at the application's startup.

Users can reuse the credentials used for logging in when connecting to databases. This approach also streamlines the process of connecting to various databases, especially those hosted on cloud servers, by using the same set of credentials for both the DBeaver Team Edition and database access.

Database Tasks and Scheduling

In DBeaver Team Edition, tasks are scheduled on the server, which differs from other DBeaver products. As a result, task outcomes are stored exclusively on the server. To access these results on a local machine, download them from the server.

Version Control

In DBeaver Team Edition, Git integration is engineered for collaborative work:

- **Admin-Configured credentials:** Administrators establish Git credentials, which means all changes made by the users are committed on behalf of the administrator-set user, not the individual users themselves.
- **Local repository management:** The local Git repository is not directly accessible for standard Git operations. Team Edition automates Git interactions, streamlining version control for collaborative work.

For further details on version control within DBeaver Team Edition, refer to the [Git TE integration documentation](#).

For Desktop DBeaver, see the [Project team work documentation](#).

Migration overview

The decision to migrate entire projects in DBeaver Team Edition is driven by two main factors:

- **Comprehensive content:** A project encapsulates all necessary elements, such as connections, scripts, and datasets. Migrating the project as a whole keeps these components together.
- **Role-based access control:** In DBeaver Team Edition, permissions are assigned at the project level, and the number of permissions available varies based on user roles. This makes it important to migrate entire projects to match the unique structure in Team Edition. For more details about roles, see in [related section](#).

Below, we explore the steps for migrating your project, focusing on a scenario where DBeaver Team Edition has been deployed using Docker. We offer a range of deployment options for DBeaver Team Edition, each with its own specific steps and requirements. For detailed information and a complete list of these deployment methods, please refer to the [DBeaver Team Edition deployment guide](#).

Important If you have set a [Project password](#) in the DBeaver Desktop version or enabled the `use secure password storage` option (located at **Window -> Preferences -> General -> Security**), you may encounter issues during migration. To avoid these issues, consider temporarily removing these security measures before migration.

Migration steps

1. In your terminal find the container ID and Name of `dbeaver/cloudbeaver-rm`. You can do this by running the `docker ps` command. Copy the container Name (you will need it later).
2. Find the path to your project in DBeaver Desktop:
 - In DBeaver Desktop, go to the Projects tab (**Window -> Projects**), select your project, and then, for Windows, click on the project and press **Alt+Enter** or for macOS, press **⌘I**. This action will open a dialog where you can copy the **Location** of your workspace path.
3. In another terminal window, copy your project folder into the Docker container using the following command (replace `project_path` with the actual path to your project and `container_name` with the copied container Name): `docker cp project_path container_name:/opt/resource-manager/workspace/shared-projects`
4. If everything is done correctly, you will see a message confirming the successful copy.

New table creation

Table of contents

[Overview](#)

[Create table](#)

[Saving changes](#)

[Modify table](#)

[Delete table](#)

[Restrictions on table creation](#)

Overview

This guide provides detailed instructions on creating a new table in DBeaver. This is a beginner-friendly guide, meaning it is written for individuals who may not have previous experience with databases or DBeaver.

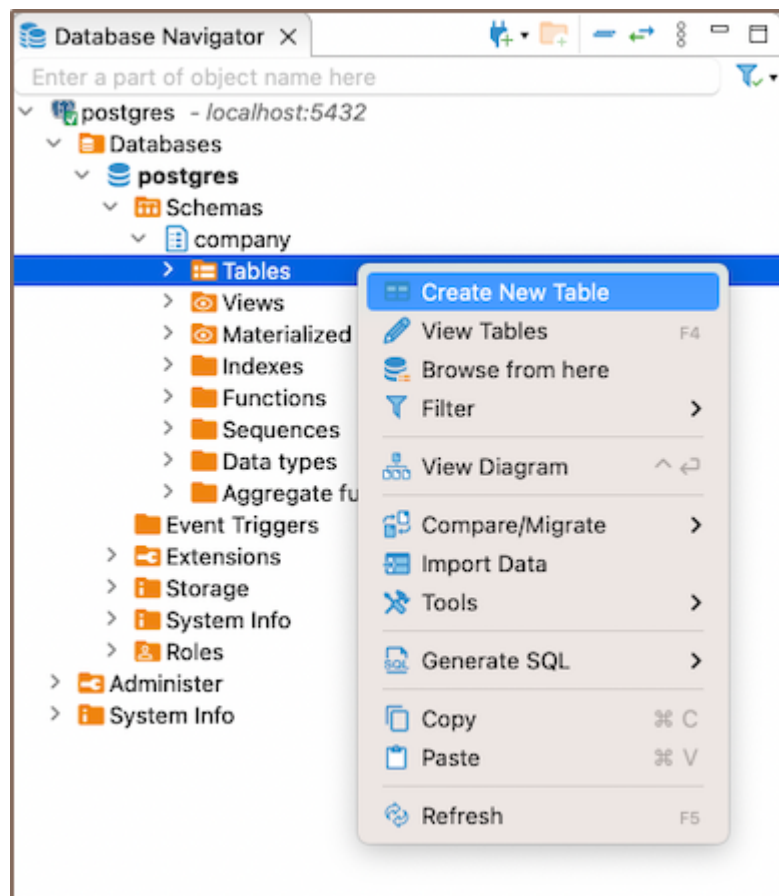
A database table, the focal point of our guide, is a systematic arrangement of data. It consists of rows and columns, forming cells where the data resides. While the number of columns is fixed for a table, it can encompass an unlimited number of rows. The maximum number of columns a table can have in a database depends on the database management system. DBeaver facilitates various operations, including data manipulation and data definition, with these tables.

Initially, the guide will cover the essential step of creating a table. After successfully creating a table, there are several enhancements you can add, such as columns, constraints, indexes, and triggers. The creation of a table is a fundamental operation in database management, and while there are many aspects to consider, not all of them are mandatory. The only mandatory step is the creation of the table itself and the definition of [columns](#). The rest are optional and can be used to enhance the functionality and integrity of your database.

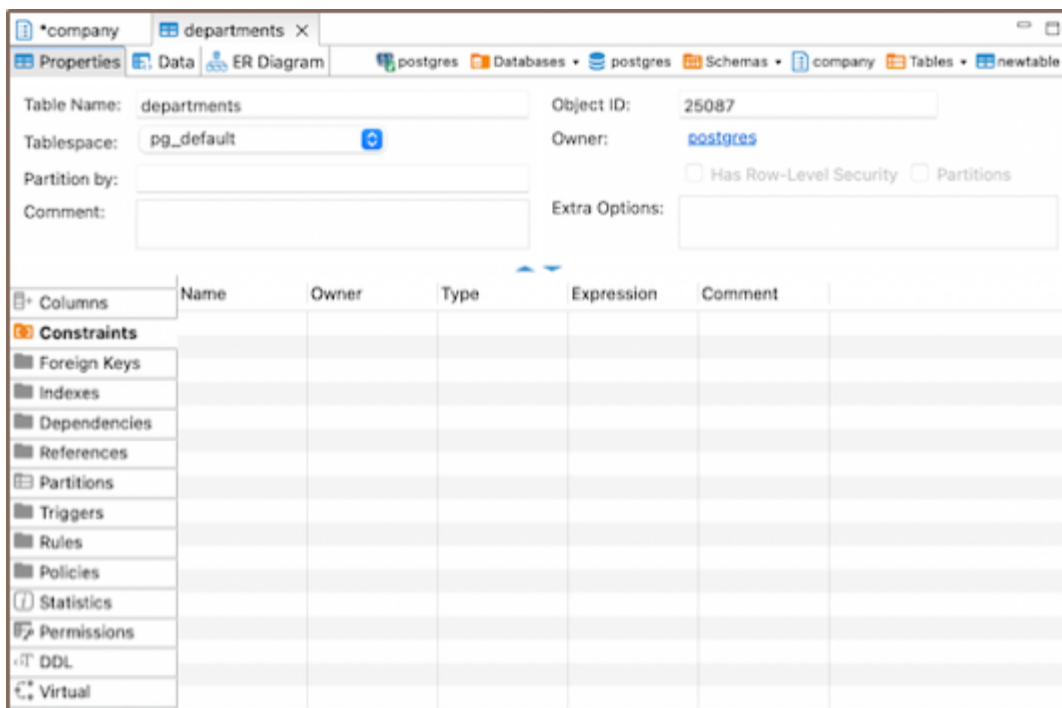
While this guide is designed around PostgreSQL within DBeaver, the processes can vary slightly with other database systems.

Create table

In DBeaver, creating a table is a straightforward process. Once you've [connected to your database](#), you can initiate the table creation process by right-clicking on the Tables folder in the [Database Navigator](#) and selecting **Create New Table**.



Upon creating a new table in DBeaver, a window will open presenting the [Properties editor](#). This is the dedicated interface where you can manage and manipulate your tables. The specific options available to you will depend on the capabilities of your chosen database system.




There are also various ways to create a table: from the **Properties editor** and from the **ER diagram**.

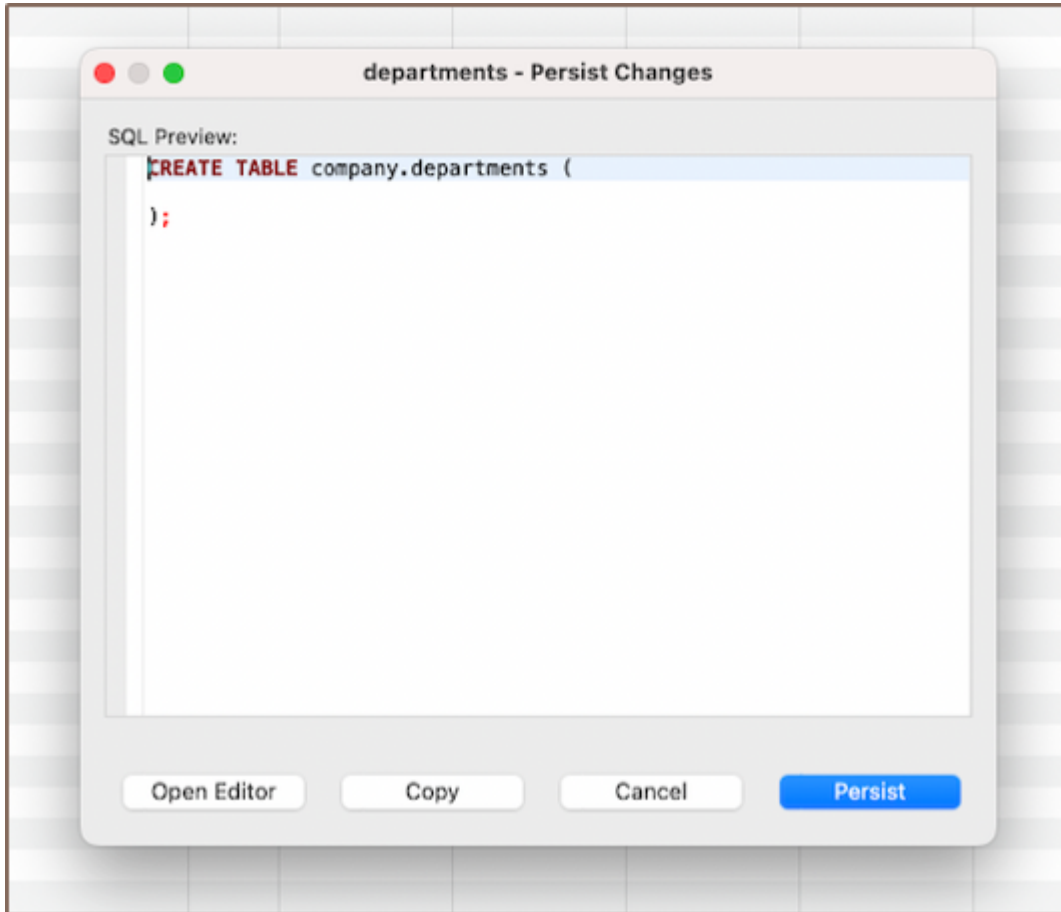
- From **Properties editor**: In the **Database Navigator**, double-click on the database name or table name, or select **View schema** from the context menu. In the window that opens, navigate to the **Properties** tab, select the **Tables** section, right-click on the window, and choose **Create New Table**.
- From **ER Diagram**: In the **Database Navigator**, double-click on the database name or table name. In the window that opens, navigate to the **ER Diagram** tab, right-click on the window, and choose **Create New Table**. Alternatively, in the **Database Navigator**, right-click on the database/table name and click **View Diagram**. In the window that opens, you can right-click and choose **Create New Table**.

Saving changes

Saving a table in DBeaver is a crucial yet straightforward process. Once you have defined the properties of the table, it's important to commit these changes to the database. Until you save your changes, your new table exists only within DBeaver and hasn't been created in the actual database. Here are the three options for saving the changes:

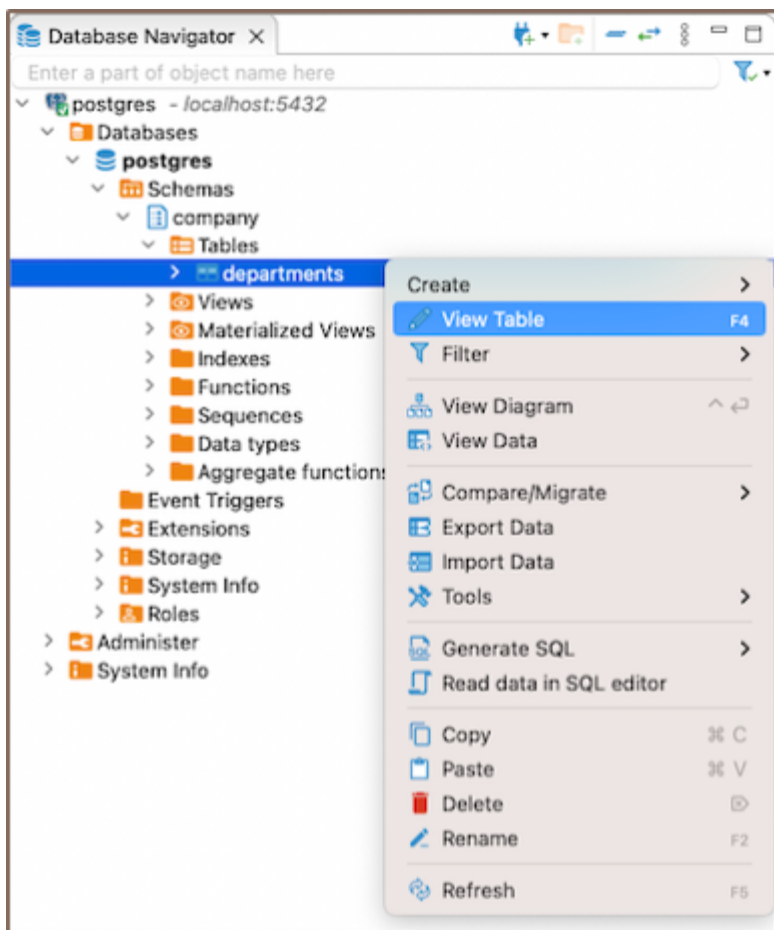
- Click on **File -> Save -> Persist**.
- Select the desired table in the **Database Navigator** and press **Ctrl+S** (or **CMD+S** for Mac OS), choose **Persist** to save the changes.
-

Utilize the **Save** button  **Save ...** located at the bottom of the **Properties editor** and press **Persist** to save the changes.



Modify table

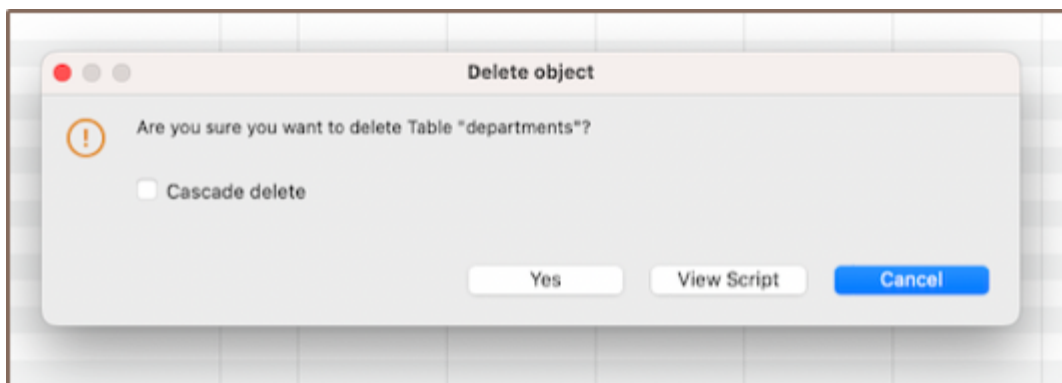
Modifying a table in DBeaver involves adjusting its structure or properties, such as adding, removing, or changing columns, adjusting constraints, or altering table settings. To make adjustments, navigate to the **Properties editor**. Access this by right-clicking the table's name in the **Database Navigator** menu and selecting **View Table**.



The specific options and capabilities available to you when modifying a table will depend on the database system you're working with. Always refer to the documentation of your specific database system for detailed information.

Delete table

Deleting a table in DBeaver involves a few simple steps. Firstly, navigate to the table you wish to delete in the **Database Navigator**. Once you've located the table, right-click on it and select the **Delete** option from the context menu. A confirmation dialog box will appear, allowing you to review the action before it's performed.



Upon confirming your decision, DBeaver will execute the necessary SQL command to remove the table from your PostgreSQL database. Please be aware that this action is permanent, and any data stored in the table will be irretrievably lost. Therefore, it's crucial to ensure that you've adequately backed up any important data before proceeding with the deletion.

Important: The confirmation dialog box, which appears when you are deleting a table in DBeaver, offers an important feature called the **Cascade delete** option. This option, when selected, automatically removes all related records in child tables, aligning with the deletion of a record in the parent table. While this can be useful for maintaining data integrity, it should be used with caution. If not handled properly, it can lead to unintended data loss by deleting records that you may not have intended to remove. Always ensure you understand the relationships and dependencies in your data before using the **Cascade delete** option.

Restrictions on table creation

DBeaver is a database management tool, and as such, it doesn't impose its own restrictions on table creation. Instead, the restrictions that apply when creating tables are determined by the specific database system you're using.

However, there are some general considerations to keep in mind when creating tables:

- **Naming Restrictions:** The name of the table must be unique within its schema. Most database systems also have restrictions on the length of the table name and the characters it can contain.
- **Storage Restrictions:** The total amount of data that can be stored in a table is determined by the storage capacity of the database system.
- **Performance Considerations:** While not a restriction per se, it's important to note that the structure of a table can have significant impacts on the performance of data retrieval and manipulation operations. For example, having a large number of columns, especially ones that are rarely used, can slow down query performance.

Remember that the specific restrictions and limitations can vary depending on the database system you're using. Always refer to the documentation of the specific database system for detailed information.

Further Reading:

New Table Creation | [Creating columns](#) | [Implementing constraints](#) | [Utilizing foreign keys](#) | [Creating indexes](#) | [Incorporating triggers](#)

Incorporating triggers

Table of contents

[Create](#)

[Function](#)

[Trigger](#)

[Saving changes](#)

[Modify](#)

[Delete](#)

[Restrictions](#)

This article focuses on triggers, essential components in database management. Triggers are procedures that are automatically executed in response to certain events in a database. They are integral to the creation and manipulation of [tables](#), enforcing [constraints](#), and automating [tasks](#).

The process involves creating a function, which is then used in a trigger. For example, a function can check if a new salary value is negative and raise an exception. This function can be used in a trigger that runs before any **INSERT** or **UPDATE** operations on a salary table.

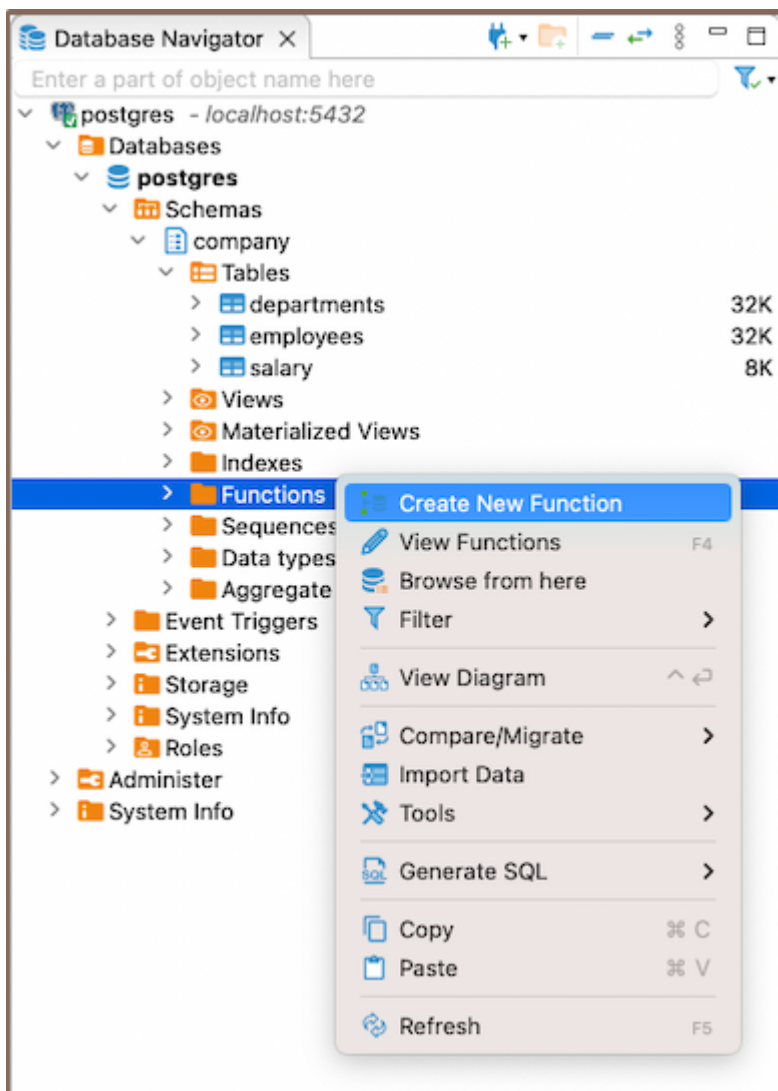
However, triggers come with restrictions and can impact performance, so their use should be optimized. The behavior of triggers can vary based on the database system in use, so always refer to the specific documentation for your system.


Create

Function

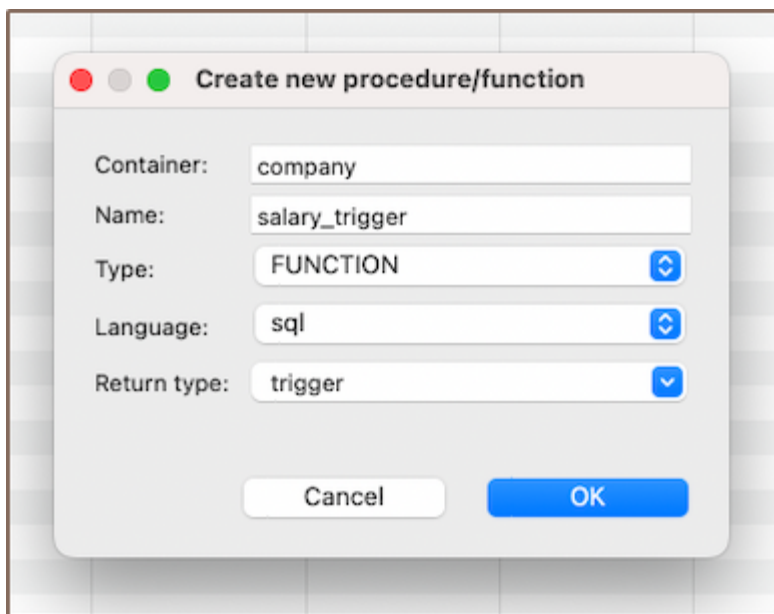
1) To create a trigger, start by creating a function. Here are three ways you can do it:

- Open the [Properties editor](#) of the database and navigate to the **Functions** tab of the corresponding database.
- Navigate to the database where you want to add a function in the [Database Navigator](#). Within the selected database, you'll find a folder named **Functions**.



- Additionally, to create a new function, you can utilize the **Create New Function** button  located at the bottom of the **Properties editor**.

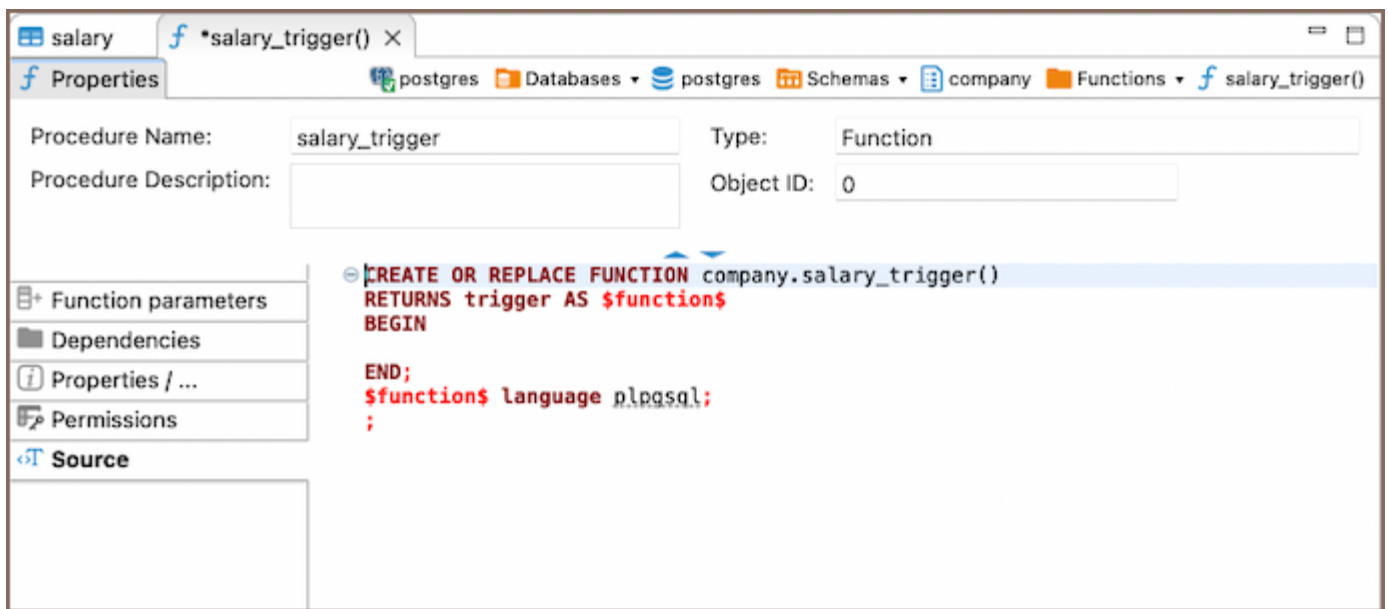
2) Depending on the method you chose in step 1, either right-click on the window (in the **Properties Editor**) or within the **Functions** folder (in the **Database Navigator**), and select **Create New Function**. If you're using the third method, simply click the **Create New Function** button. This will open a new window where you'll be able to customize the function's settings to suit your needs.



Field	Description
Function Name	This is the name you choose for the function. It should be unique within the schema where the function is being created.
Type	In this case, you would select Function .
Language	This is the programming language in which the function is written. For most database systems, this would be SQL or a variant of SQL, such as PL/pgSQL for PostgreSQL or PL/SQL for Oracle.
Return Type	This specifies the type of value the function will return. For a trigger function, this is usually specified as trigger .

Note: These settings can vary depending on the specific database system you're using. Always refer to the documentation for your database system for the most accurate information.

3) After you've chosen the needed settings for your function, you'll need to write the actual code for the function. This is done in the **Source** section.




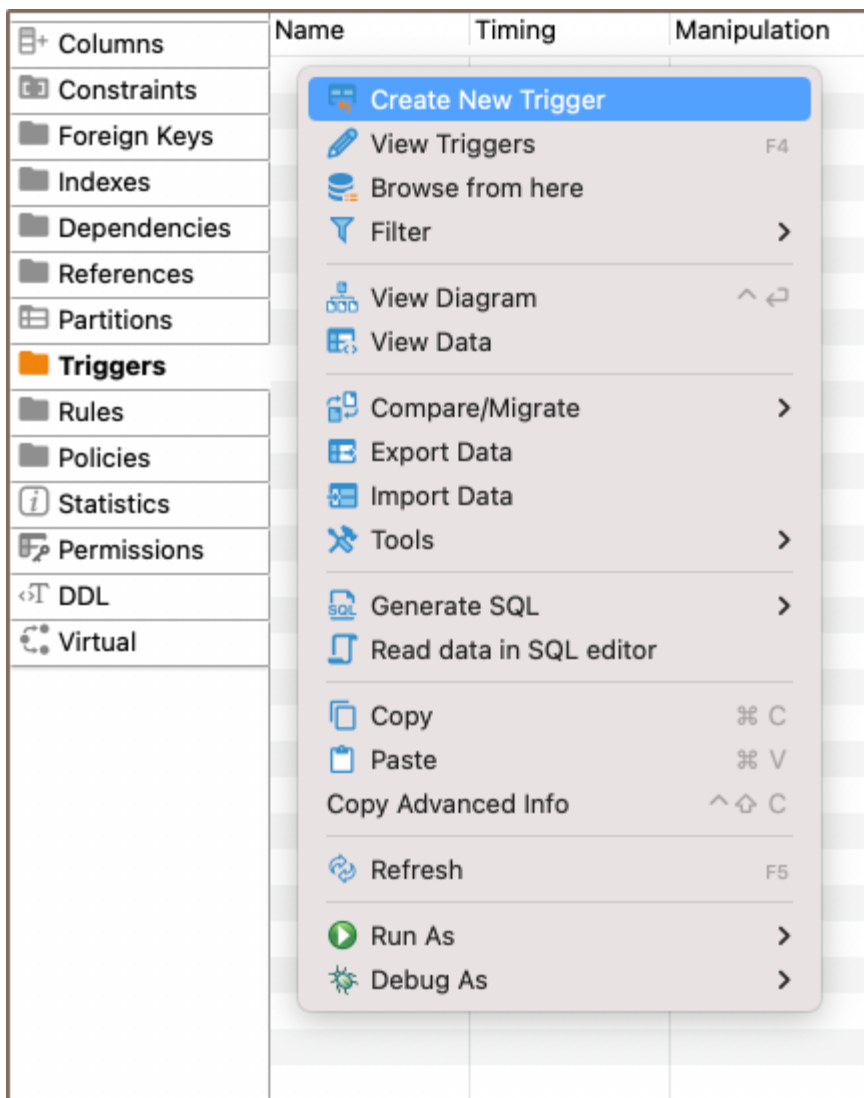
This function will be called when the trigger is activated. The function you write will depend on what you want the trigger to do. For example, if you want the trigger to check if a new salary value is negative and raise an exception if it is, you might write a function like this:

```
CREATE OR REPLACE FUNCTION company.salary_trigger()
RETURNS trigger AS $function$
BEGIN
    IF NEW.salary_amount < 0 THEN
        RAISE EXCEPTION 'Error: Salary cannot be negative';
    END IF;
    RETURN NEW;
END;
$function$ language plpgsql
```

Trigger

1) Creating a trigger is a process that mirrors the creation of a function. Once your function is defined, you can move on to create the trigger. This can be done through the **Properties Editor** or the **Database Navigator**.

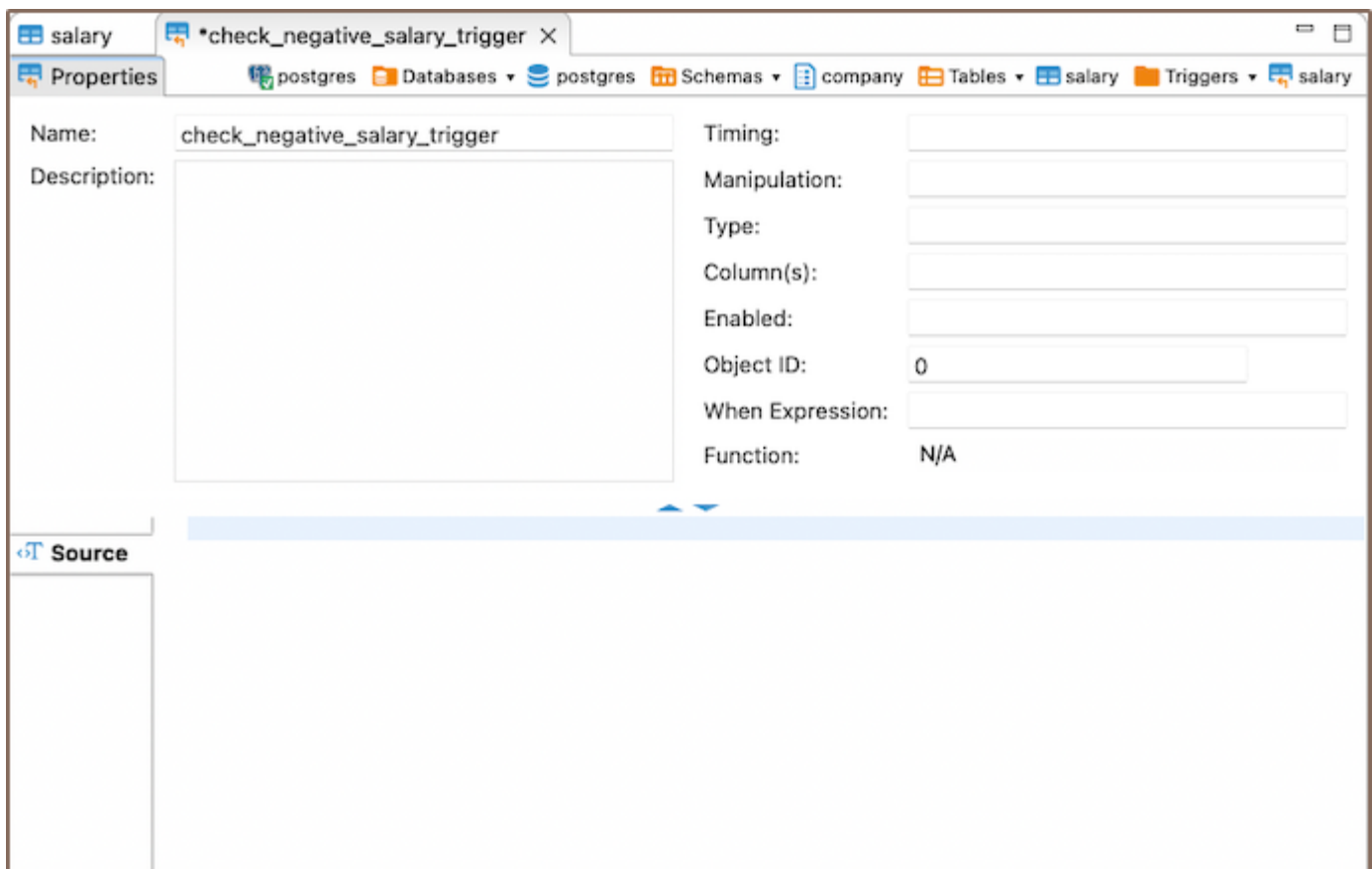
- In the **Properties Editor**, you can navigate to the **Triggers** section associated with the table for which you want to create the trigger, right-click on the window and select **Create New Trigger**. Here, you can also select **Create New Trigger** button  from the **Bottom menu**.



- Similarly, in the **Database Navigator**, you can right-click on the relevant table, select **Triggers**, and then choose **Create New Trigger**.

2) Next, you need to provide a name for the trigger and specify the function that it should call when it's activated. The trigger's function is typically written to perform a specific task when certain conditions are met. For example, if you've written a function to check if a new salary value is negative, you can create a trigger that calls this function before any **INSERT** operations on the salary table.

3) Finally, describe the code for the trigger.



For instance:


```
CREATE TRIGGER check_negative_salary_trigger
BEFORE INSERT
ON company.salary
FOR EACH ROW
EXECUTE function company.salary_trigger();
```

In this example, a `salary_trigger()` function is created that evaluates the new salary value (NEW.salary). If the salary is negative, it throws an exception with an error message. Then, a `check_negative_salary_trigger` is established to run this function before any `INSERT` operations on the "salary" table. Should the salary value be negative, an exception is triggered and the operation is stopped.

Saving changes

Once you have configured the trigger and function properties, it is essential to save these changes to the database. Until you commit your modifications, the new trigger and function will only exist within DBeaver and will not be added to the actual database. Here are three options for committing the changes:


- Click on **File -> Save -> Persist**.

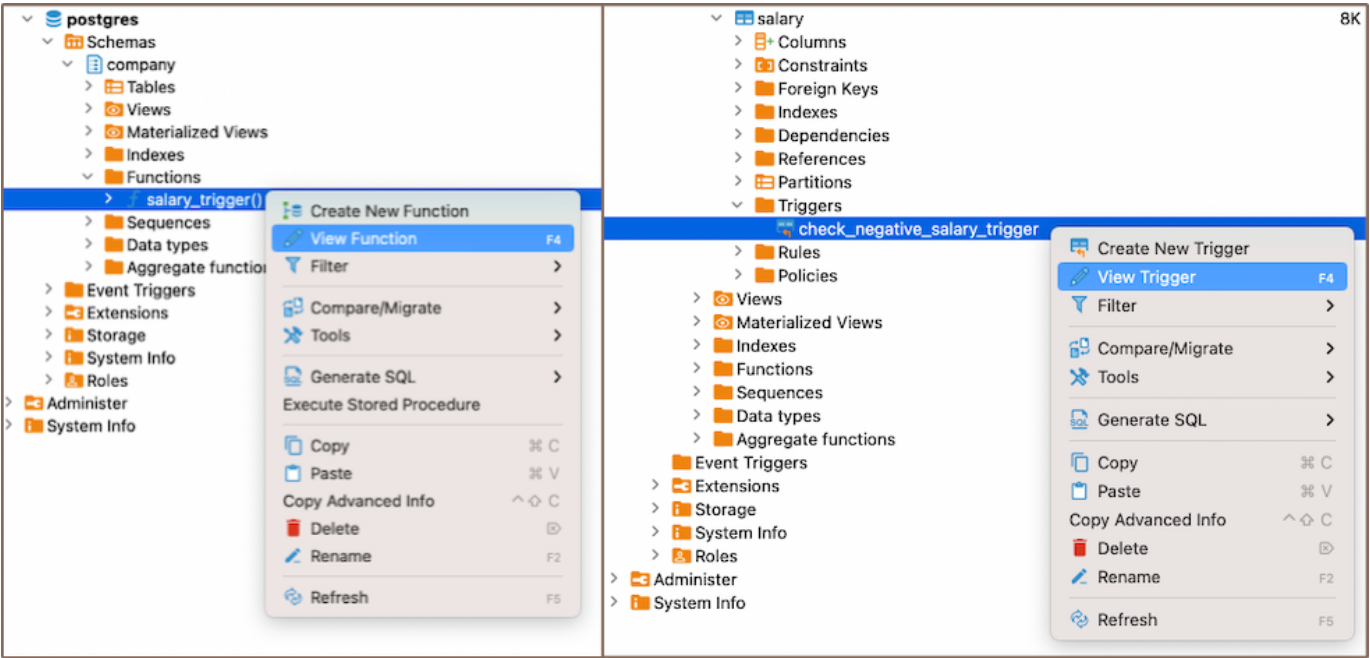
- Select the desired table in the **Database Navigator** and press **Ctrl+S** (or **CMD+S** for Mac OS), choose **Persist** to save the changes.
- Utilize the **Save** button  **Save ...** located at the bottom of the **Properties editor** and press **Persist** to save the changes.

Modify

In the realm of database management, the need to modify existing functions or triggers is a common occurrence. Whether it's to correct an error, optimize performance, or adapt to new requirements, the ability to modify these elements is crucial. DBeaver offers intuitive and efficient ways to make these modifications.

The DBeaver interface provides multiple pathways to access and modify functions or trigger:

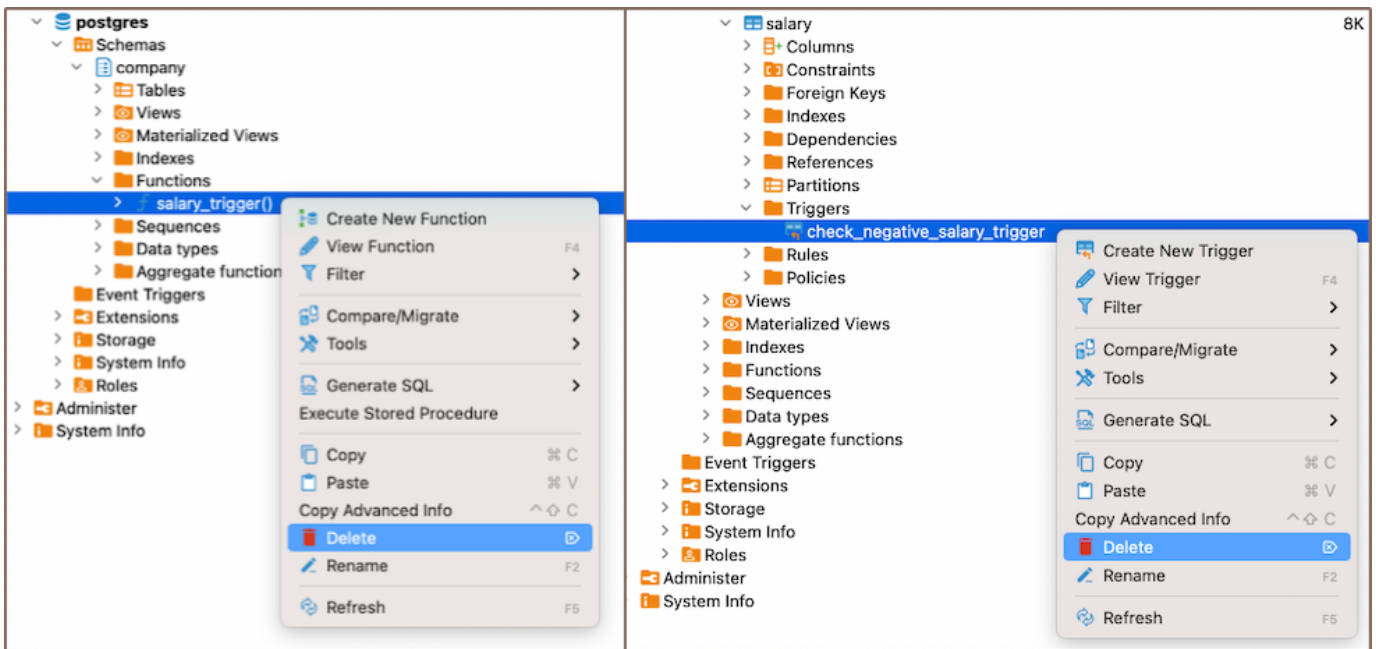
- **Properties Editor:** Open the **Properties Editor** by double-clicking on the database name (for functions) and table name (for triggers). Within the selected database or table, you'll find a folder named **Triggers** or **Functions**. Right-click on the trigger or function you want to modify and select **View Trigger** or **View Function**.
- **Bottom Menu:** In the **Properties Editor** interface, use the **View Indexes** or **View Function** button .
- **Database Navigator:** In the **Database Navigator**, select the appropriate database or table. Within the selected database or table, you'll find a folder named **Triggers** or **Functions**. Right-click the trigger or function and select **View Index** or **View Function**.




Delete

You can delete an index either through the **Properties Editor** or the **Database Navigator**:

- Using the **Properties Editor**: Open the Properties Editor, navigate to the **Triggers** or **Functions** tab of the corresponding database (for functions) and table (for triggers), and find the trigger you want to delete.
- Using the **Database Navigator**: Navigate to the database where the table with the trigger or function is located. Open the **Triggers** or **Functions** folder, and find the trigger or function you want to delete.



2) To delete the trigger or function, right-click on the trigger's or function's name and select **Delete**, or you can select the necessary trigger or function and press the **Delete** key, or **Delete button**  (in the **Bottom Menu** of the **Properties Editor**). 3) A confirmation dialog will appear, asking you to confirm the deletion. Before proceeding, ensure that you've selected the correct trigger or function for deletion. 4) After confirming the deletion, you'll need to persist the changes to apply them to the database.

Restrictions

- **Recursive Triggers**: Some databases allow recursive triggers, where a trigger can call itself. This can lead to infinite loops and excessive system load if not handled correctly.

- **Data Modification:** Triggers can't be used to modify a table that is already being used (for reading or modifying) in the statement that invoked the function or trigger.
- **Execution Order:** The order in which multiple triggers are fired isn't guaranteed. This can lead to unexpected results if triggers have interdependencies.
- **Performance:** Triggers can slow down data modification operations because they add extra processing. They should be used sparingly and optimized for performance.
- **Transaction Control Statements:** In many SQL databases, transaction control statements (like COMMIT and ROLLBACK) are not allowed within trigger code. However, exceptions apply depending on the database system.

Further Reading:

[New Table Creation](#) | [Creating columns](#) | [Implementing Constraints](#) | [Utilizing foreign keys](#) | [Creating indexes](#) |

Incorporating triggers

Creating columns

Table of contents

[Create](#)

[Saving changes](#)

[Modify](#)

[Delete](#)

[Restrictions](#)

As you delve deeper into the intricacies of table structures, understanding columns and their functionalities becomes crucial. In DBeaver, columns are fundamental components of tables, which in turn are housed within databases. Before you can create columns, you must first establish a [database](#) and a [table](#).

Columns in DBeaver are versatile and customizable. You can create new columns, modify their settings, and even delete them when necessary. Each column in a table has a specific data type, dictating the kind of data it can store. You can also enforce rules on columns such as nullability, unique constraints, check constraints, and default values. These rules are essential to maintain data integrity, accuracy, and reliability in your database.

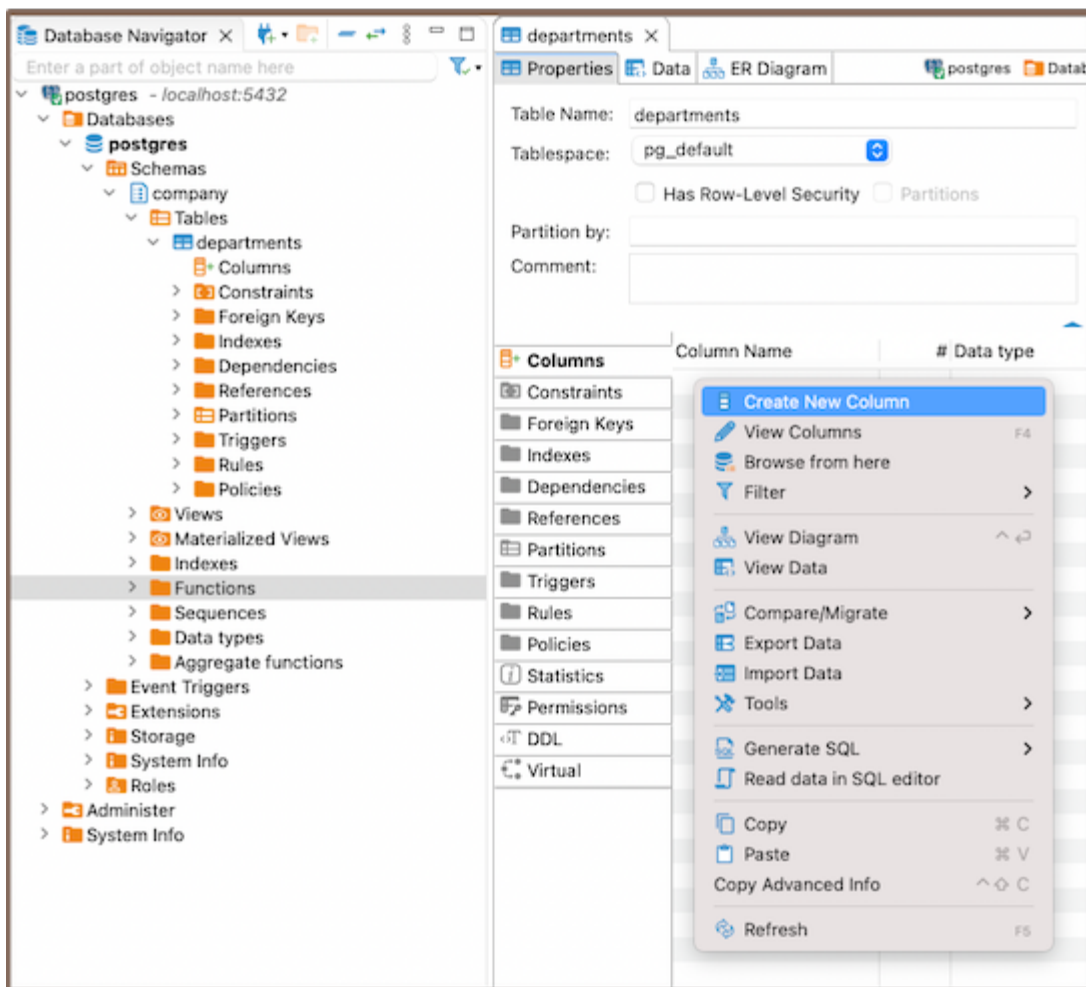
Beyond these basic operations, you can perform more advanced tasks with columns. You can rename columns, change their data types, adjust permissions, and add comments. All databases may not support some of these operations, so it's essential to be aware of the specific capabilities of your database system.

This guide will walk you through the process of creating, modifying, and deleting columns, as well as customizing their settings to suit your needs.

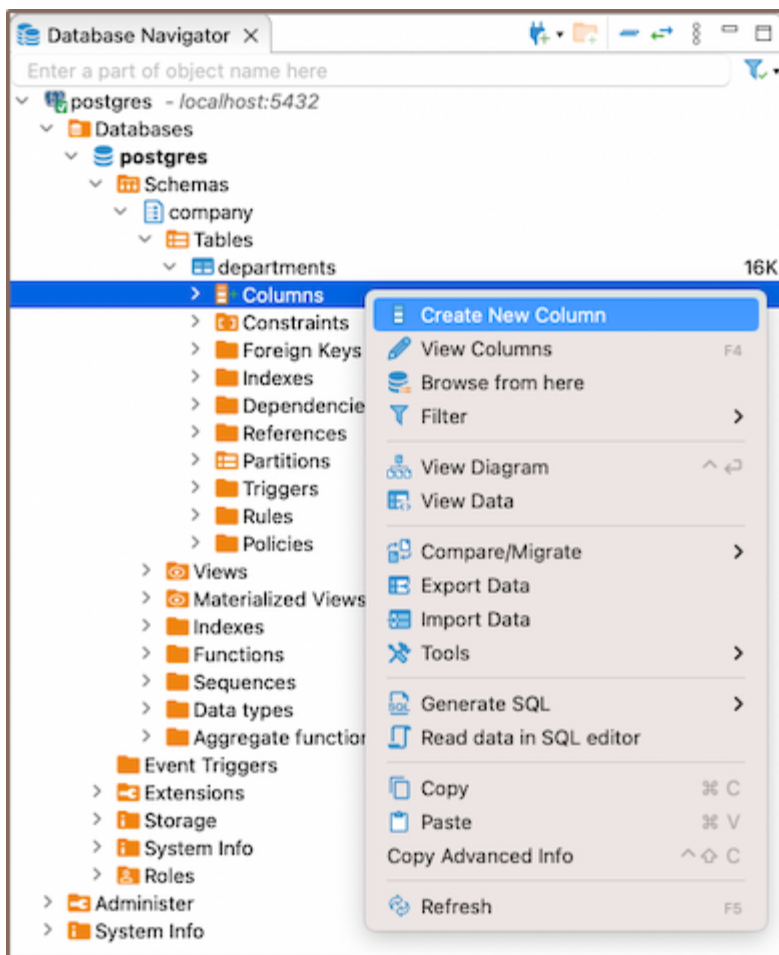
Create


Creating a new column in DBeaver involves a series of steps that guide you through the process. Here's how you can do it:

- 1) Go to the **Columns** tab in the [Properties editor](#), right-click on the pane of the object's sub-entities, and select **Create New Column** from the context menu.



- Also you can also add a new column by expanding the table view in the [Database Navigator](#) and selecting **Create New Column**.



- Another method to create a new column is by utilizing the **Create New Column** button  located at the bottom of the [Editor panel](#).

2) When creating a new column, you'll be presented with the **Edit Attribute** window. Here, you can customize the column's settings to suit your needs. You can adjust the **Name**, **Data type**, **Identity**, **Collation**, **Not null**, **Default**, add a **Comment** as needed, and configure **Keys** for the column.

Table column

Edit attribute column1

Properties

Name:

id

Data type:

varchar

Identity:

Always

Collation:

default

☒ Not Null

Default:

Comment:

Keys

☒ Unique

Type:

Primary Key

Name:

articles_pk

Cancel


OK

Field	Description
Name	Specify the name of the column.
Data type	Select the type of data the column will store.
Identity	Set the column as an identity column, which will auto-increment its value with each new row.
Collation	Set the collation for the column, which determines how data in the column is sorted and compared.
Not null	Enforce that the column must always contain a value.
Default	Set a default value for the column, which will be used when no value is specified for the column during insertion.
Comment	Add a comment for the column, which can be helpful for documentation purposes.
Unique	Select this checkbox to enforce that the values in the column must be unique.
Type	Choose the type of key for the column: Primary Key , or Unique Key .

Name	Specify the name of the key, which is useful for identification in constraints and indexing.
------	--

Saving changes

Saving a column in DBeaver is an essential and simple operation. After you've configured the column properties, it's vital to persist these changes to the database. Until you save your modifications, your new column only exists within DBeaver and hasn't been added to the actual database table. Here are the three options for committing the changes:

- Click on **File** -> **Save** -> **Persist**.
- Select the desired table in the **Database Navigator** and press `Ctrl+S` (or `Cmd+S` for Mac OS), choose **Persist** to save the changes.
- Utilize the **Save** button  **Save ...** located at the bottom of the [Editor panel](#) and press **Persist** to save the changes.


Modify

As your database evolves and your needs change, you may find it necessary to modify the properties of your columns. Whether it's renaming a column, changing its data type, adjusting its nullability, or updating its permissions, DBeaver provides you with the flexibility to adapt your columns to your changing requirements.

Modifying columns is a crucial aspect of database management, allowing you to ensure that your columns continue to serve their intended purpose effectively.

Here's how you can modify a column in DBeaver:

- Via the **Properties Editor**: You can double-click on the table name to open the **Properties Editor**. In this window, you can see all the columns of the table. Right-click on the column you want to modify and select **View Column**.
- Via the **Database Navigator**: Go to the **Database Navigator** menu, and choose the proper database and column to modify its settings. Right-click the column and select **View Column**.


- Via the **Bottom Menu**: In the bottom menu of the **Properties Editor** interface, there is a **View Column** button . Clicking this button will open the properties of the currently selected column in the **Editor panel**, allowing you to modify its settings.

Note: Not all databases support all types of column modifications, so always check the specific capabilities of your database system.

Delete

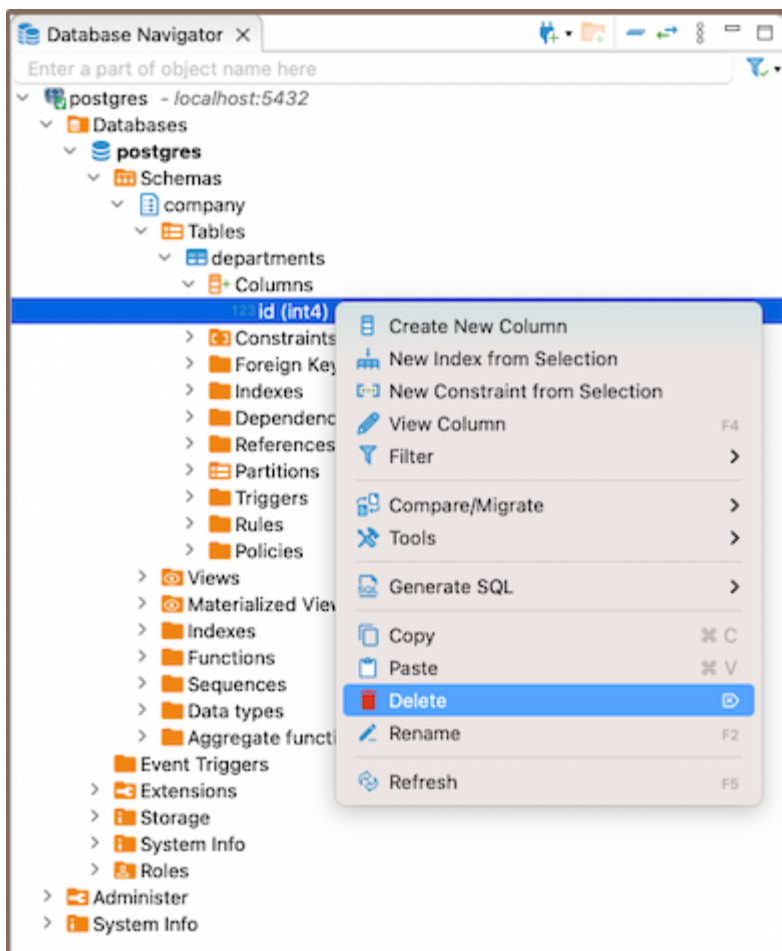
1) You can delete a column either through the **Properties Editor** or the **Database Navigator**:

- Using the **Properties Editor**: Open the Properties Editor, navigate to the **Constraints** tab of the corresponding table, and find the column you want to delete.
- Using the **Database Navigator**: Navigate to the database where the table with the column is located. Find and select the table, open the **Columns** folder, and find the column you want to delete.

2) To delete a column, right-click on the column's name and select **Delete**, or you can select the necessary column and press the **Delete** key, or **Delete button**  (in the **Bottom Menu** of the **Properties Editor**).

3) A confirmation dialog will appear, asking you to confirm the deletion. Before proceeding, ensure that you've selected the correct column for deletion.

4) After confirming the deletion, you'll need to persist the changes to apply them to the database.



Once a column is deleted, all the data stored in that column will be lost. Furthermore, there are several restrictions and considerations to keep in mind when deleting columns:

Limitations	Description
Data Loss	As mentioned, deleting a column will permanently remove all data stored in that column. Make sure to back up any important data before proceeding with the deletion.
Primary Key Columns	If the column you want to delete is a primary key column, you must first remove its primary key constraint within the table properties. Primary key columns are used to identify each row in the table uniquely, and deleting them without removing the constraint can cause issues.
Referential Integrity	If the column you want to delete is referenced by other tables in the database (i.e., it's a foreign key), you may not be able to delete the column without first deleting or modifying the referencing tables. This is to maintain referential integrity in the database.
Database Support	Not all databases support the ability to delete columns. Always check the specific capabilities of your database system.

Restrictions

When working with columns in DBeaver, it's important to understand the various restrictions that can be applied to them. These restrictions are not limitations but rather rules that help ensure the integrity, accuracy, and reliability of the data stored in your database. They dictate what data can be stored in a column, how it's treated, and how it interacts with other data in the database. Let's take a closer look at these restrictions:

- **Data Type:** Each column in a table has a specific data type. The type of data that can be stored in a column is dictated by its data type.
- **Nullability:** By default, columns can hold **NULL**. However, by specifying **NOT NULL** during column creation, you can enforce that every row must contain a value for that column.
- **Unique Constraint:** A unique constraint ensures that all values in a column are distinct. Duplicate entries are not permitted.
- **Check Constraint:** A check constraint enables you to specify a condition on a column. Every value added to the column must satisfy this condition.
- **Default Value:** A column can be given a default value. If no value is specified during data insertion, the column uses the default.

Further Reading:

[New Table Creation](#) | [Creating columns](#) | [Implementing constraints](#) | [Utilizing foreign keys](#) | [Creating indexes](#) | [Incorporating triggers](#)

Creating Indexes

Table of contents

[Create](#)

[Saving changes](#)

[Modify](#)

[Delete](#)

[Restrictions](#)

In the process of creating [tables](#) in a database, an integral part is the creation of indexes. Indexes are tools that can significantly enhance the performance of your database. They are designed to speed up the retrieval of data from the tables, making them particularly useful in large databases where without them, searches could be time-consuming.

Indexes work by providing a more direct path to the data, reducing the need for the database to scan the entire table. They can be created on one or more [columns](#) of a table, and when a query is made, the database can use the indexes to find the data more quickly.

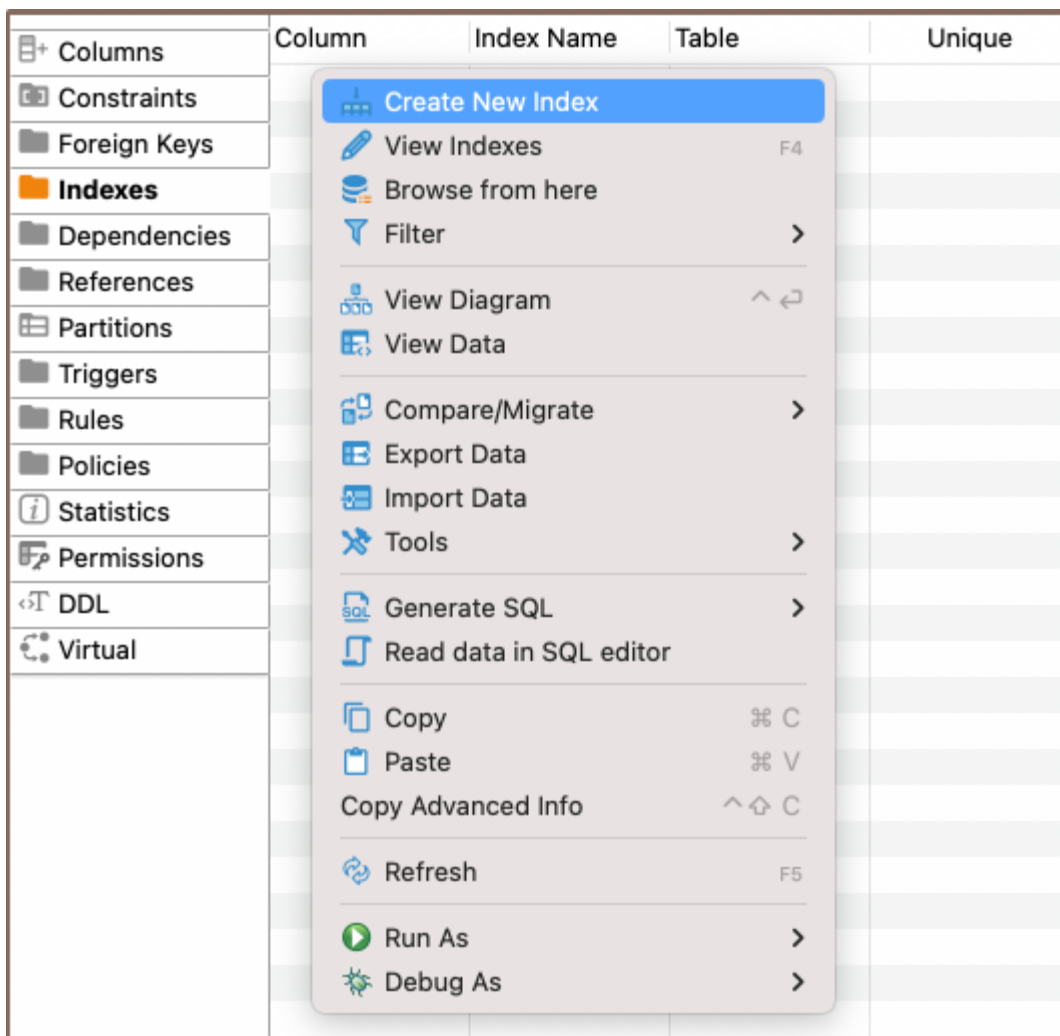
However, while indexes can improve the speed of data retrieval, they can also slow down data modification operations such as `INSERT`, `UPDATE`, and `DELETE`, as the index must also be updated. Therefore, it's crucial to strike a balance between the number of indexes and the frequency of data modification in your database.


This article will guide you through the process of creating, modifying, and deleting indexes using the DBeaver. It will also discuss the restrictions you need to be aware of when working with indexes, such as the impact on performance, storage considerations, and restrictions related to null values and certain data types.

Create

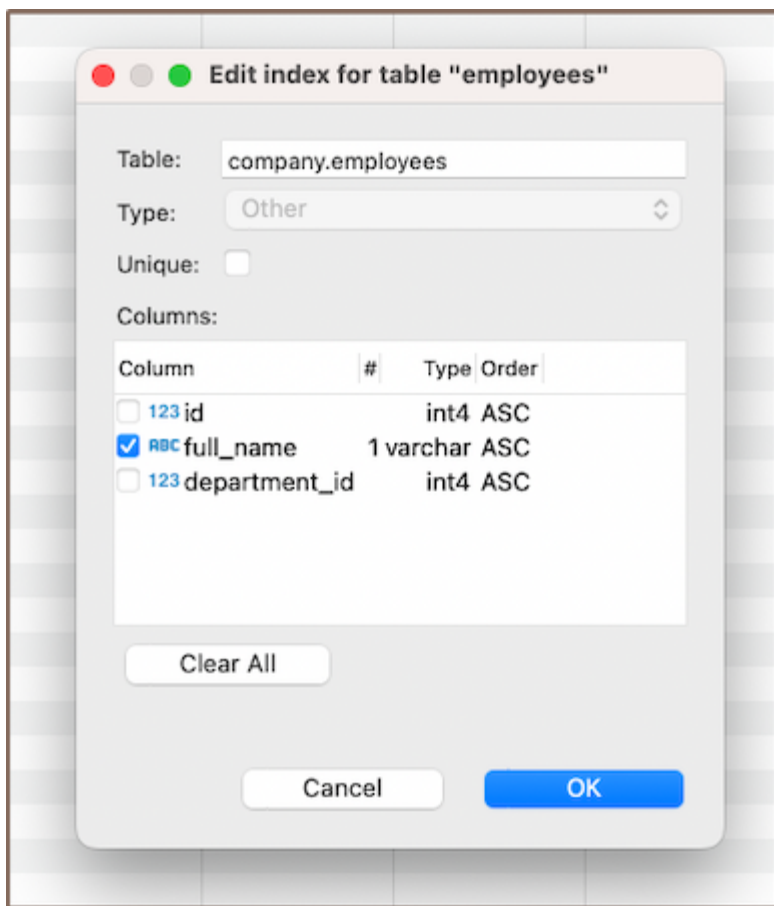
1) You can start the process of creating a new index in three ways:

- Using the [Properties editor](#): Open the Properties Editor and navigate to the **Indexes** tab of the corresponding table.



- Using the [Database Navigator](#): Navigate to the database where the table you want to add a index to is located. Find and select the table for which you want to create a new index. Within the selected table, you'll find a folder named **Indexes**.
- Additionally, to create a new index, you can utilize the **Create New Index** button  located at the bottom of the **Properties editor**.

2) Depending on the method you chose in step 1, either right-click on the window (in the **Properties Editor**) or within the **Foreign Keys** folder (in the **Database Navigator**), and select **Create New Index**. If you're using the third method, simply click the **Create New Index** button. This will open a new window where you'll be able to customize the index's settings to suit your needs.




Field	Description
Unique	If set to unique, the index ensures that all entries in the indexed column(s) are distinct. This can limit data entry if values must be unique.
Columns	This setting allows you to select one or more columns for the index.

Note: In addition to creating a simple index on a single column, DBeaver also provides the option to create a composite index. A composite index is an index that is built on several table columns. This type of index can significantly boost query execution speed when dealing with multiple-column conditions. It's recommended to put unique fields at the end of the composite index to maximize its efficiency.

3) After you've chosen the column(s) for your index and configured any additional settings, you can finalize the creation of the index by clicking **OK**.


Saving changes

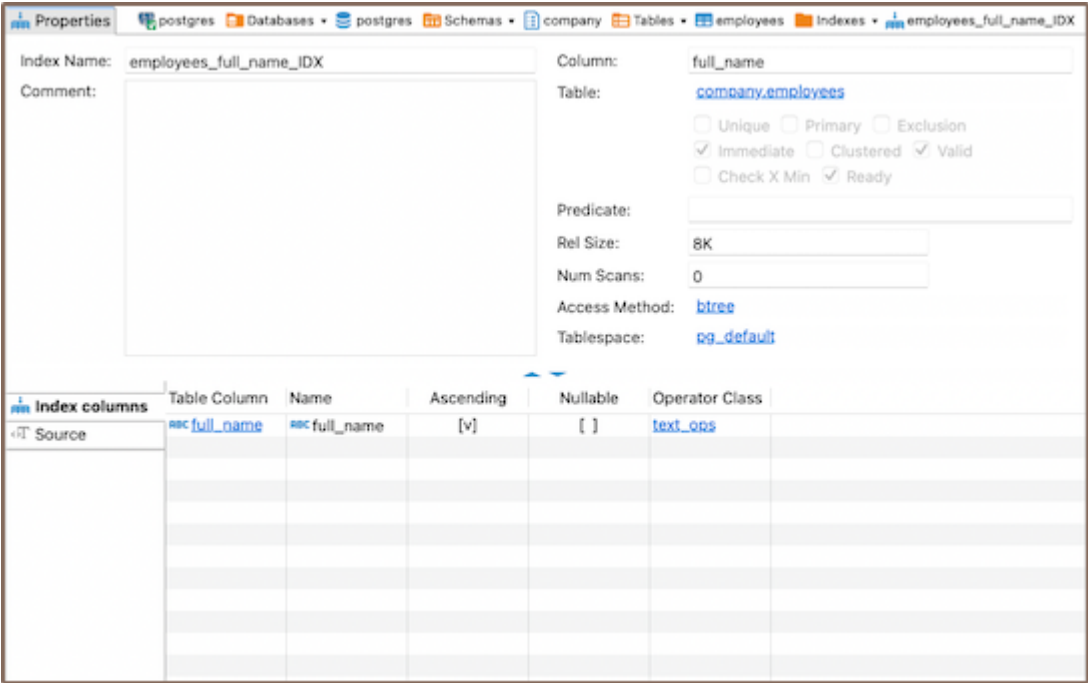
Once you have configured the index properties, it is essential to save these changes to the database. Until you commit your modifications, the new index will only exist within DBeaver and will not be added to the actual database table. Here are three options for committing the changes:

- Click on **File** -> **Save** -> **Persist**.
- Select the desired table in the **Database Navigator** and press **Ctrl+S** (or **CMD+S** for Mac OS), choose **Persist** to save the changes.
- Utilize the **Save** button  **Save ...** located at the bottom of the **Properties editor** and press **Persist** to save the changes.

Modify

As your database evolves, there might be a need to modify the properties of your indexes. This could involve changing the indexed columns, adjusting the uniqueness constraint, or renaming the index. DBeaver provides the flexibility to make these modifications as required. Here's the process:

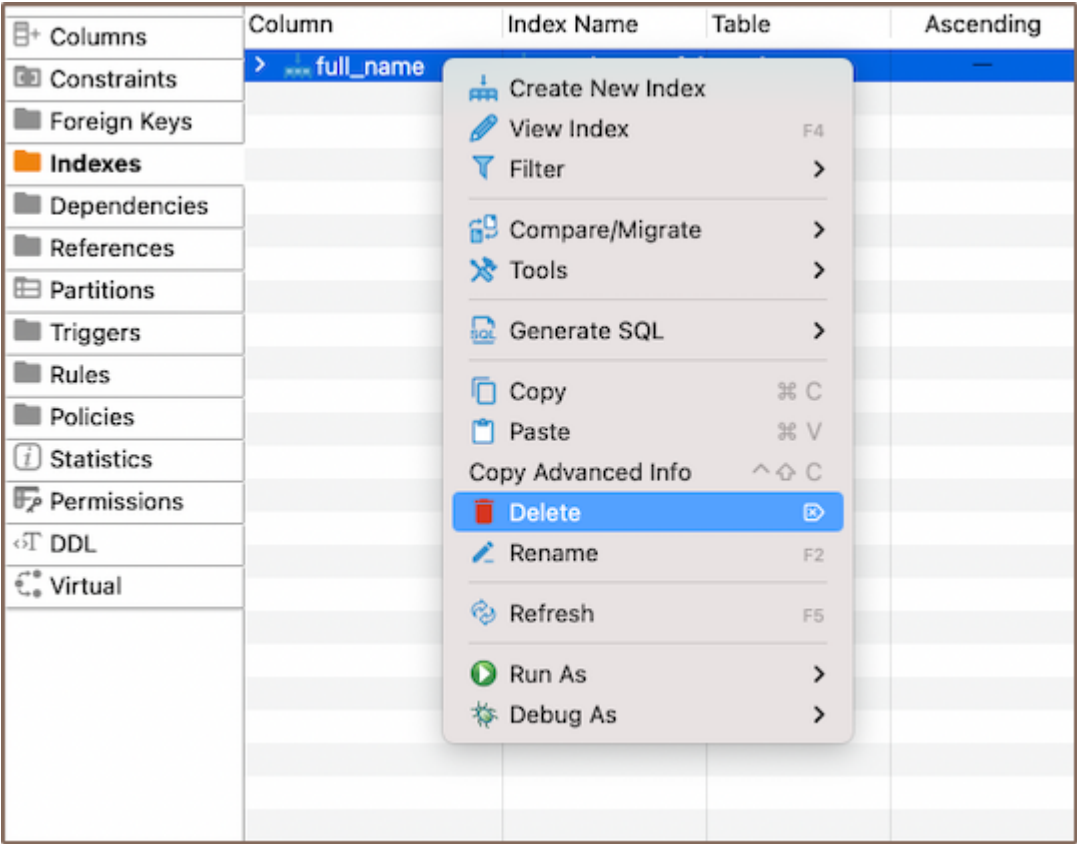
- **Properties Editor:** Open the **Properties Editor** by double-clicking on the table name. Within the selected table, you'll find a folder named **Indexes**. Right-click on the index you want to modify and select **View Index**.
- **Database Navigator:** In the Database Navigator, select the appropriate database and table. Within the selected table, you'll find a folder named **Indexes**. Right-click the index and select **View Index**.
- **Bottom Menu:** In the **Properties Editor** interface, use the **View Indexes** button  to open the properties of the selected index in the **Editor panel**.




Delete

1) You can delete an index either through the **Properties Editor** or the **Database Navigator**:

- Using the **Properties Editor**: Open the **Properties Editor**, navigate to the **Indexes** tab of the corresponding table, and find the index you want to delete.



- Using the **Database Navigator**: Navigate to the database where the table with the index is located. Open the **Indexes** folder, and find the index you want to delete.

2) To delete an index, right-click on the index's name and select **Delete**, or you can select the necessary index and press the **Delete** key, or **Delete button**  (in the **Bottom Menu** of the **Properties Editor**). 3) A confirmation dialog will appear, asking you to confirm the deletion. Before proceeding, ensure that you've selected the correct index for deletion. 4) After confirming the deletion, you'll need to persist the changes to apply them to the database.

Restrictions

- Unique indexes created due to a primary key or unique constraint can't be deleted. Instead, remove the constraint. This action removes the uniqueness requirement for column values involved in the constraint expression and deletes the corresponding unique index.
- **Unique Indexes:** These enforce uniqueness on the indexed column(s), which can limit data entry if values must be distinct.
- **Performance Impact:** Indexes can improve query speed, but they slow down data modification operations (`INSERT` , `UPDATE` , `DELETE`) as the index must also be updated.
- **Storage:** Indexes consume disk space. This might become significant if there are many indexes or the table is large.
- **Null Values:** Depending on the database system, there may be restrictions related to indexing null values.
- **Data Types:** Certain data types might not be indexable or may require specific index types.

Further Reading:

[New Table Creation](#) | [Creating columns](#) | [Implementing Constraints](#) | [Utilizing foreign keys](#) | [Creating indexes](#) | [Incorporating triggers](#)

Implementing constraints

Table of contents

[Primary key](#)

[Create](#)

[Saving changes](#)

[Modify](#)

[Delete](#)

[Unique key](#)

[Check expression](#)

[Restrictions](#)

As you continue to explore the intricacies of [table](#) structures, understanding the role and functionality of constraints becomes essential. Constraints are rules that we apply to data [columns](#) in a table when we want to limit the type of data that can go into a table. This can help ensure the accuracy and reliability of our data.

While creating constraints is not always necessary, it can be a powerful tool in the table creation process. Just like you create columns, you can also define constraints that uphold the integrity of your data. Constraints can be as simple as ensuring uniqueness in a column or as complex as a check expression that defines acceptable values for a column.

In this guide, you'll learn how to create, modify, and delete constraints, as well as how to customize their settings to suit your needs. You'll discover how to create [primary keys](#), [unique keys](#), and [check expressions](#). You'll also learn about the restrictions and considerations you need to keep in mind when working with constraints.

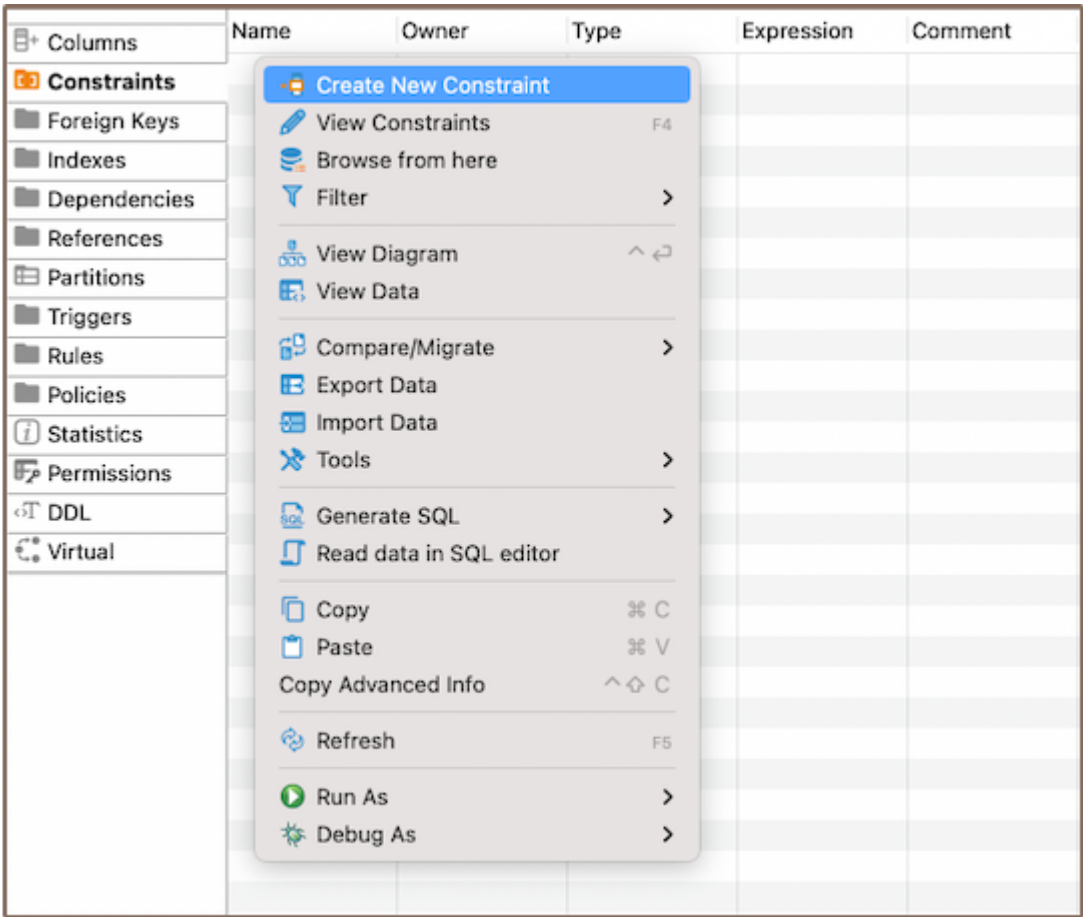
Primary key


Primary keys are a crucial part of your database structure, uniquely identifying each record in a table. When you define a primary key, the database ensures no duplicate values exist in the specified column(s), maintaining data integrity and improving performance. DBeaver simplifies this process, providing an intuitive interface for creating, modifying, and deleting primary keys.

Create

Creating a new primary key in DBeaver involves a series of steps that guide you through the process. Here's how you can do it:

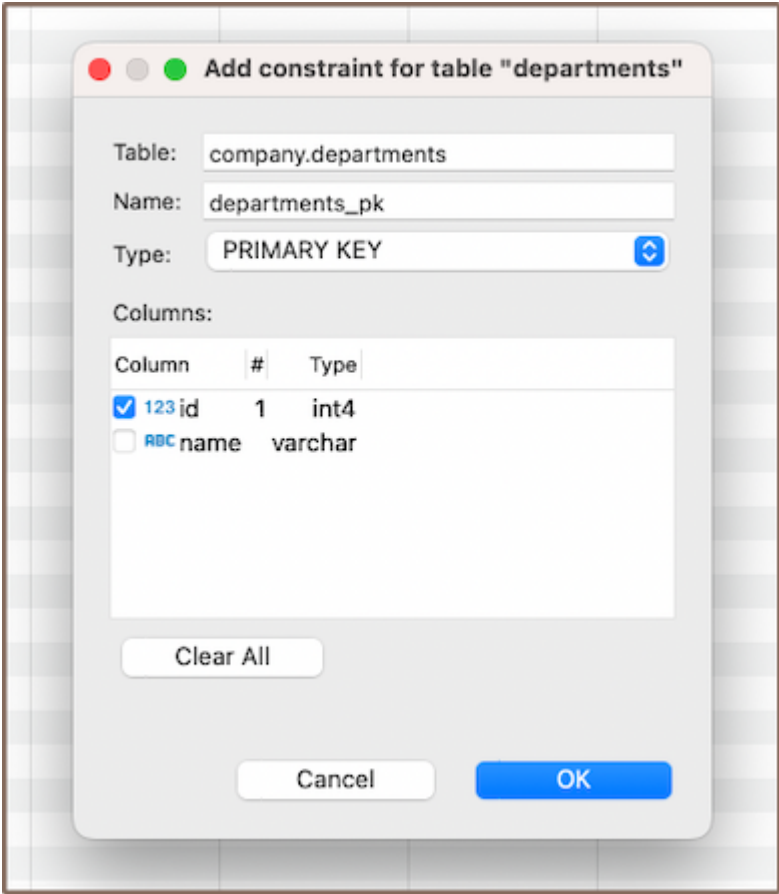
- 1) You can start the process of creating a new constraint in three ways:
 - Using the [Properties editor](#): Open the Properties Editor and navigate to the **Constraints** tab of the corresponding table. This is where you'll find all the existing constraints for the table and where you'll create new ones.



- Using the [Database Navigator](#): Navigate to the database where the table you want to add a constraint to is located. Find and select the table for which you want to create a new constraint. Within the selected table, you'll find a folder named **Constraints**.
- You can also utilize the **Create New Constraint** button  located at the bottom of the **Properties editor**.

Tip: You can create primary key during the column creation process as described in the article [Creating columns](#).

2) Depending on the method you chose in step 1, either right-click on the window (in the **Properties Editor**) or within the **Constraints** folder (in the **Database Navigator**), and select **Create New Constraint**. If you're using the third method, simply click the **Create New Constraint** button. This will open a new window where you'll be able to define the details of your new constraint.




Field	Description
Name	Here, you can specify the name of the constraint. It's a good practice to choose a name that clearly indicates the purpose of the constraint.
Type	This is where you select the type of constraint. Options include Primary Key, Unique Key, Check. The type of constraint you choose will depend on the rule you want to enforce on the column.
Columns	In this section, you can select the column or columns that the constraint will apply to. If you're creating a composite constraint, you can select multiple columns.

3) After selecting the column, click **OK** to save your new constraint. A window will appear displaying the newly created primary key. This is a good time to double-check that everything is correct.

Saving changes

Once you have configured the constraint properties, it is essential to save these changes to the database. Until you commit your modifications, the new constraint will only exist within DBeaver and will not be added to the actual database table. Here are three options for committing the changes:

- Click on **File -> Save -> Persist**.
- Select the desired table in the **Database Navigator** and press **Ctrl+S** (or **Cmd+S** for Mac OS), choose **Persist** to save the changes.
- Utilize the **Save** button  **Save ...** located at the bottom of the **Properties editor** and press **Persist** to save the changes.

Modify

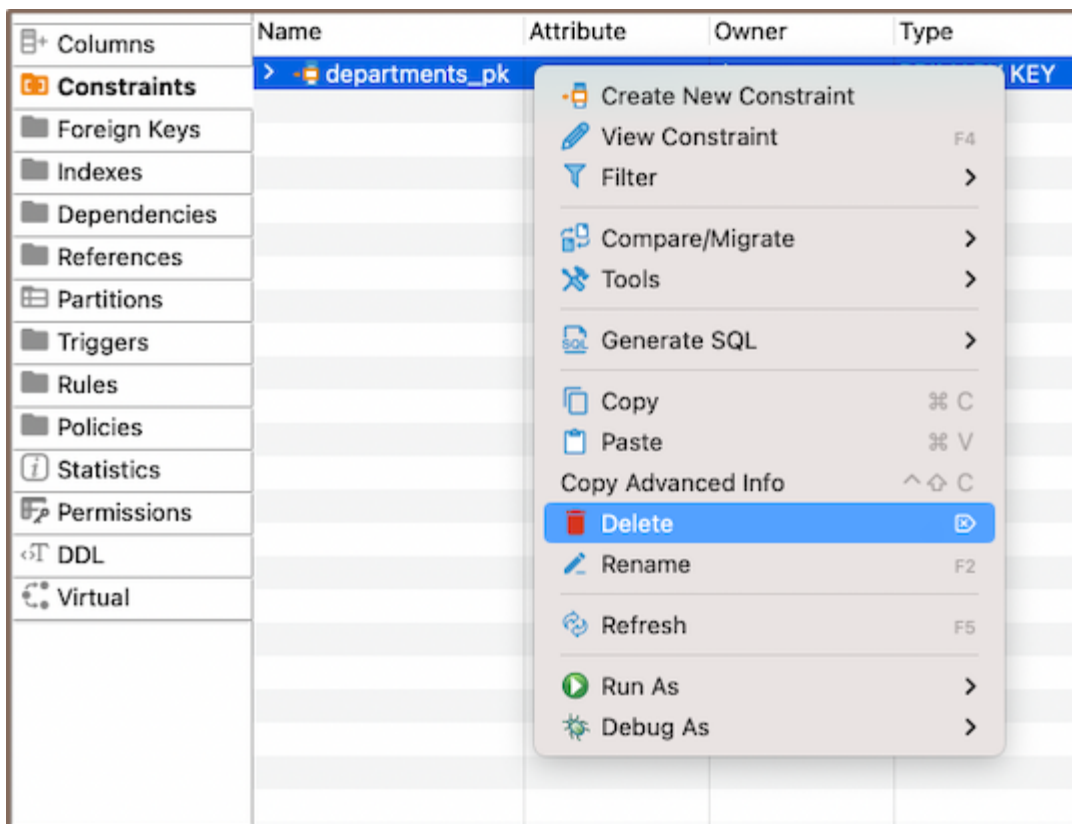
To modify an existing primary key, you need to first delete it and then add a new primary key with the updated parameters.

Note: Remember, modifying primary keys should be done with care. Primary keys are integral to maintaining the integrity of your data, and changing them can have significant effects on your database. Always double-check your changes and make sure they align with your data requirements.


Delete

1) You can delete a primary key either through the **Properties Editor** or the **Database Navigator**:

- Using the **Properties Editor**: Open the Properties Editor, navigate to the **Constraints** tab of the corresponding table, and find the constraint you want to delete.



- Using the **Database Navigator**: Navigate to the database where the table with the constraint is located. Find and select the constraint, open the **Constraints** folder, and find the constraint you want to delete.

2) To delete a primary key, right-click on the key's name and select **Delete**, or you can select the necessary primary key and press the **Delete** key, or **Delete button**  (in the **Bottom Menu** of the **Properties Editor**).

3) A confirmation dialog will appear, asking you to confirm the deletion. Before proceeding, ensure that you've selected the correct constraint for deletion.

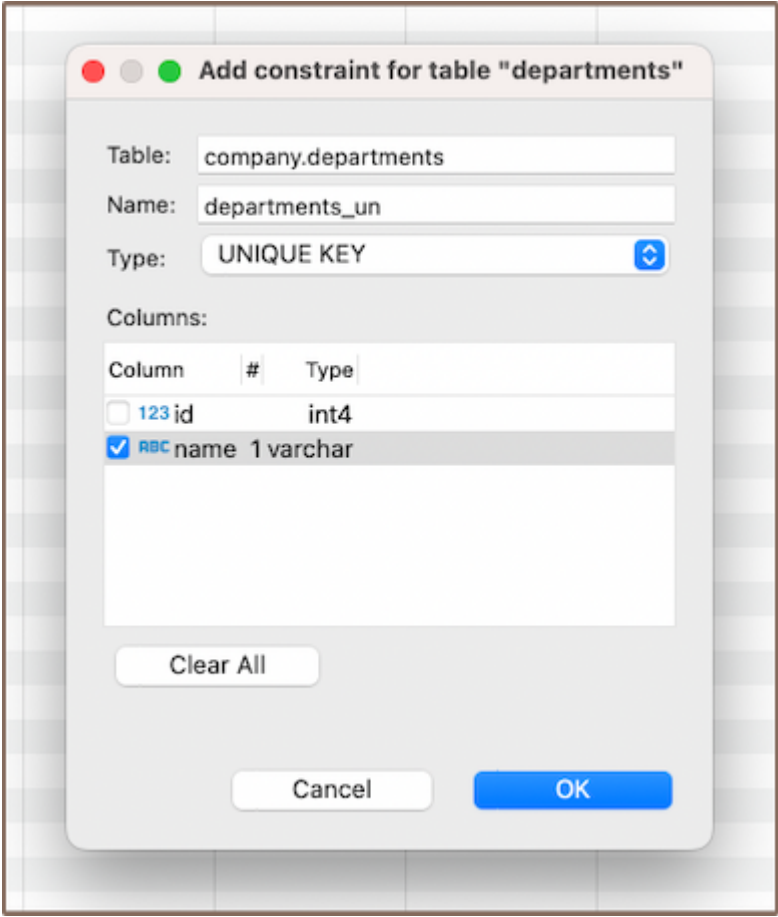
4) After confirming the deletion, you'll need to persist the changes to apply them to the database.

Important: Be particularly cautious if you're considering using the **Cascade delete** option. This option will not only delete the primary key but also any dependent records in your database. This can lead to significant data loss if not handled carefully.

Unique key

A unique key is a tool that ensures distinct values within a column or a group of columns in a table. While it can identify unique records, it doesn't have to be the primary key. You can create a unique key on a single column or multiple columns, ensuring that each value is unique. This is crucial for maintaining data integrity within your table. If you attempt to insert a duplicate value into a column with a unique key constraint, you'll encounter an error.

The processes of creating, deleting and modifying a unique key are similar to those for a [primary key](#).



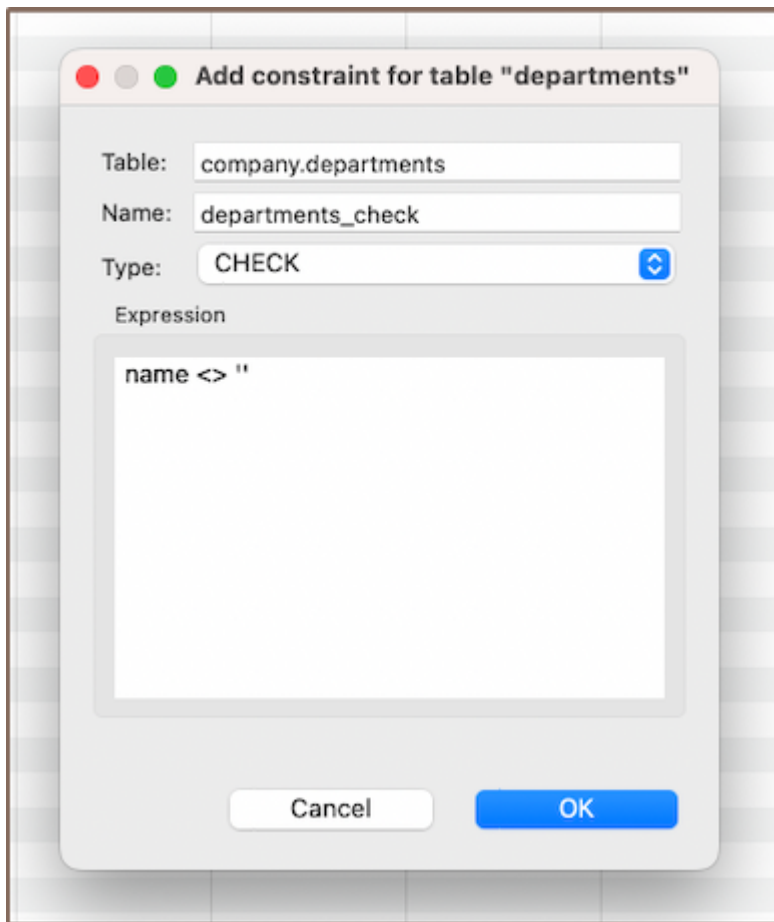
Check expression

A check expression in a database is also a constraint, it defines acceptable values for a column. It enforces specific conditions, ensuring that the data in the column complies with the defined rules. The processes of creating, deleting and modifying a check expression are similar to those for a [primary key](#).

For instance, suppose you have a column named "name" and you want to ensure that its value is never an empty string. You could create a check expression as follows:

```
name <> ''
```

This would prevent any records from being inserted or updated with an empty string for the "name" column.



Note: Check expressions are a powerful tool for maintaining data integrity. They enforce specific conditions on your data, preventing invalid entries. Always consider the implications of your actions when working with check expressions.

Restrictions

Working with constraints comes with certain restrictions that you need to be aware of. These restrictions are in place to maintain the integrity of your data and prevent unwanted changes. Here's what you need to know:

- **Unique Values:** When creating a primary key or unique constraint, the column(s) you choose must contain unique values. If there are duplicate values in the column(s), the constraint creation will fail.
- **Null Values:** Primary keys cannot contain null values. If the column(s) you choose for a primary key contains null values, you will need to either remove or replace these before creating the constraint. Unique constraints, on the other hand, typically allow for multiple null values.

- **Data Type Compatibility:** The type of constraint you want to create may have specific data type requirements. For example, check constraints require a Boolean expression, so the column(s) involved must be able to participate in such an expression.
- **Existing Data Compliance:** When creating a check constraint, all existing data in the table must comply with the check condition. If any rows violate the condition, the constraint creation will fail.
- **Performance Impact:** Creating constraints, especially on large tables, can have a performance impact. This is because the database needs to check the constraint condition for all existing rows. This might take considerable time and resources for large tables.
- **Modifying Constraints:** Constraints cannot be directly modified. This is because constraints are defined with specific parameters that can't be changed directly.
- **Deleting Constraints with Dependencies:** You can't delete a primary key in a table with records unless you first remove all dependent foreign keys. This is because the primary key is likely being referenced by these foreign keys, and deleting it would break these references and potentially cause data integrity issues.

Further Reading:

[New Table Creation](#) | [Creating columns](#) | [Implementing Constraints](#) | [Utilizing foreign keys](#) | [Creating indexes](#) | [Incorporating triggers](#)

Utilizing foreign keys

Table of contents

[Create](#)

[Saving changes](#)

[Modify](#)

[Delete](#)

[Restrictions](#)

This article provides a comprehensive guide on the creation and management of Foreign Keys within DBeaver. While not always necessary, foreign keys can be used to establish relationships between [tables](#), which can help maintain data consistency and integrity.

In DBeaver, the creation of a foreign key involves referencing a [unique key](#) in another table. This process forms a relationship that is essential for maintaining data integrity. Additionally, DBeaver allows you to define specific rules for the behavior of foreign keys when deleting or updating rows in the main table.

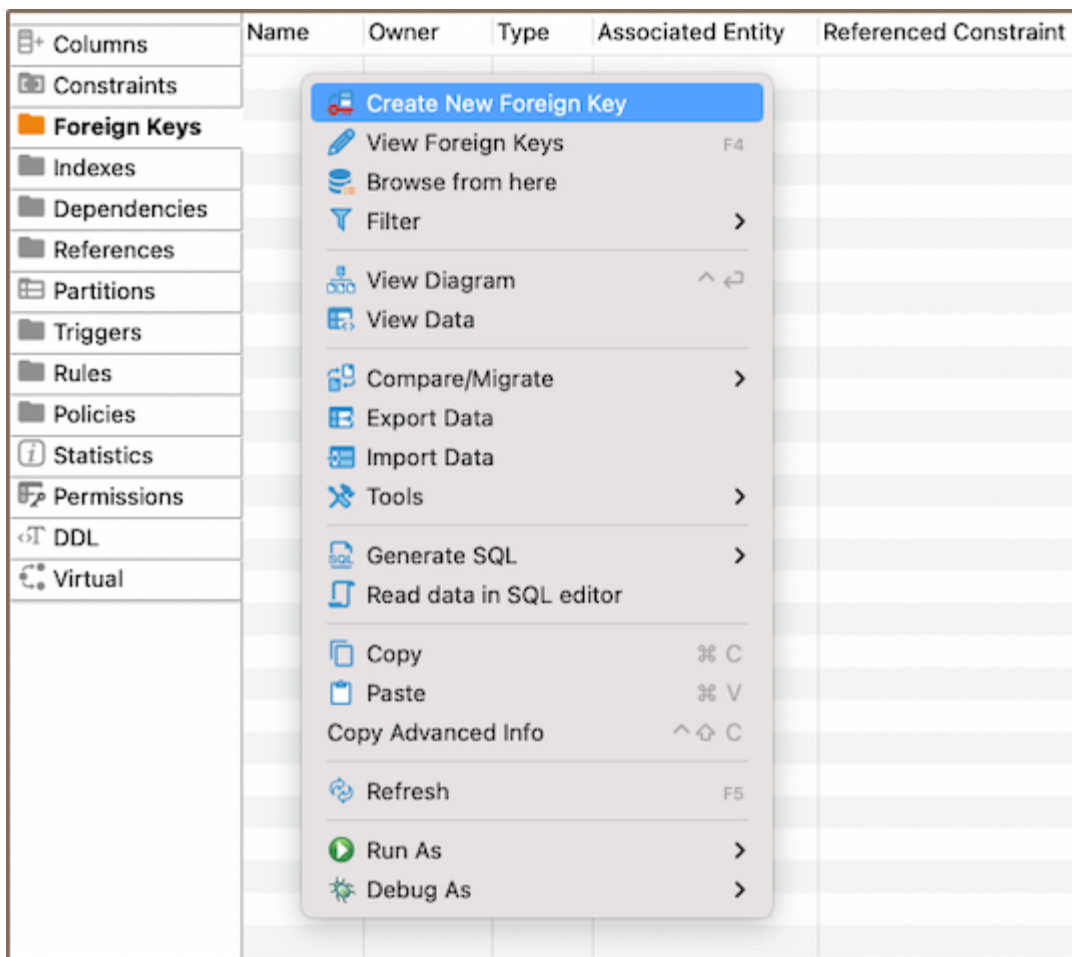
This guide will detail the steps involved in creating, modifying, and deleting foreign keys. It will also explain how to set up rules for their behavior. It is important to note that the management of foreign keys requires careful consideration due to their significant impact on data relationships.


Create

Creating a new foreign key in DBeaver involves a series of steps that guide you through the process.

1) You can start the process of creating a new foreign key in three ways:

- Using the [Properties editor](#): Open the Properties Editor and navigate to the **Foreign Keys** tab of the corresponding table.



- Using the [Database Navigator](#): Navigate to the database where the table you want to add a foreign key to is located. Find and select the table for which you want to create a new foreign key. Within the selected table, you'll find a folder named **Foreign Keys**.
- Additionally, to create a new foreign key, you can utilize the **Create New Foreign Key** button  located at the bottom of the **Properties editor**.

2) Depending on the method you chose in step 1, either right-click on the window (in the **Properties Editor**) or within the **Foreign Keys** folder (in the **Database Navigator**), and select **Create New Foreign Key**. If you're using the third method, simply click the **Create New Foreign Key** button. This will open a new window where you'll be able to customize the foreign key's settings to suit your needs.

Constraint

Edit foreign key | Edit foreign key for table "actor"

Table: public.actor

Container: public

Reference table:

film_category	
inventory	
language	
payment	
rental	
staff	

Unique Key: language_pkey (Primary Key)

Custom Name: language_fk

Columns:

Column	Column Type	Ref Column	Ref Column Type
<language_id>	serial4	language_id	serial4

On Delete: No Action

On Update: No Action


Column options ...

☐ Deferrable
☐ Deferred

Cancel


OK

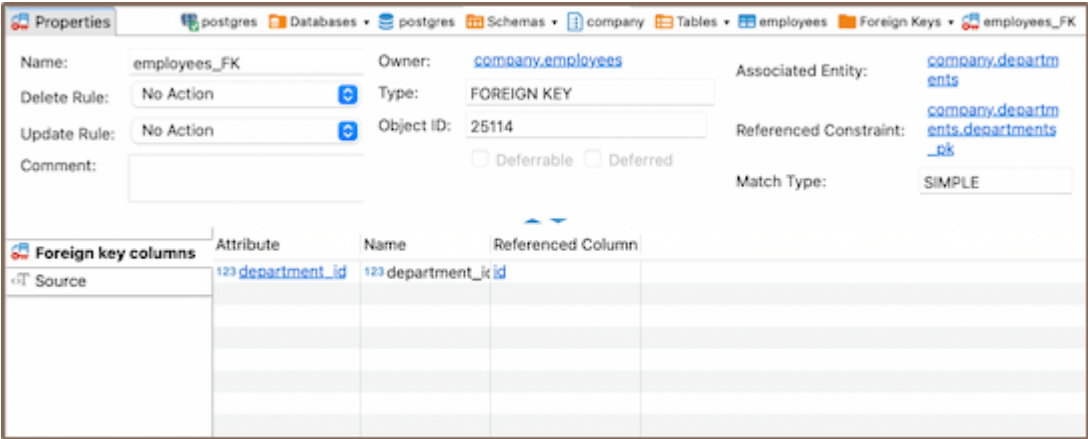
Field	Description
Reference table	Select the table that the foreign key references.
Unique Key	Choose the unique key in the reference table that the foreign key will point to.
Columns	<p>Select the columns in your table that will become the foreign key.</p> <p>You can start creating a foreign key even if the specified column does not exist. In such a situation, you can open the dropdown menu in the column field and select <new>.</p>
Deferrable	Specify whether the enforcement of the foreign key constraint can be deferred until the end of the transaction.
Deferred	If the foreign key is deferrable, this option sets whether the constraint is deferred by default.

- Click on **File** -> **Save** -> **Persist**.
- Select the desired table in the **Database Navigator** and press **Ctrl+S** (or **CMD+S** for Mac OS), choose **Persist** to save the changes.
- Utilize the **Save** button  **Save ...** located at the bottom of the **Properties editor** and press **Persist** to save the changes.

Modify

Over time, you may need to adjust the properties of your foreign keys. This could involve renaming a foreign key, changing its referenced table or unique key, adjusting its columns, or updating its rules for deletion or update. DBeaver allows you to make these modifications as needed. Here's how:

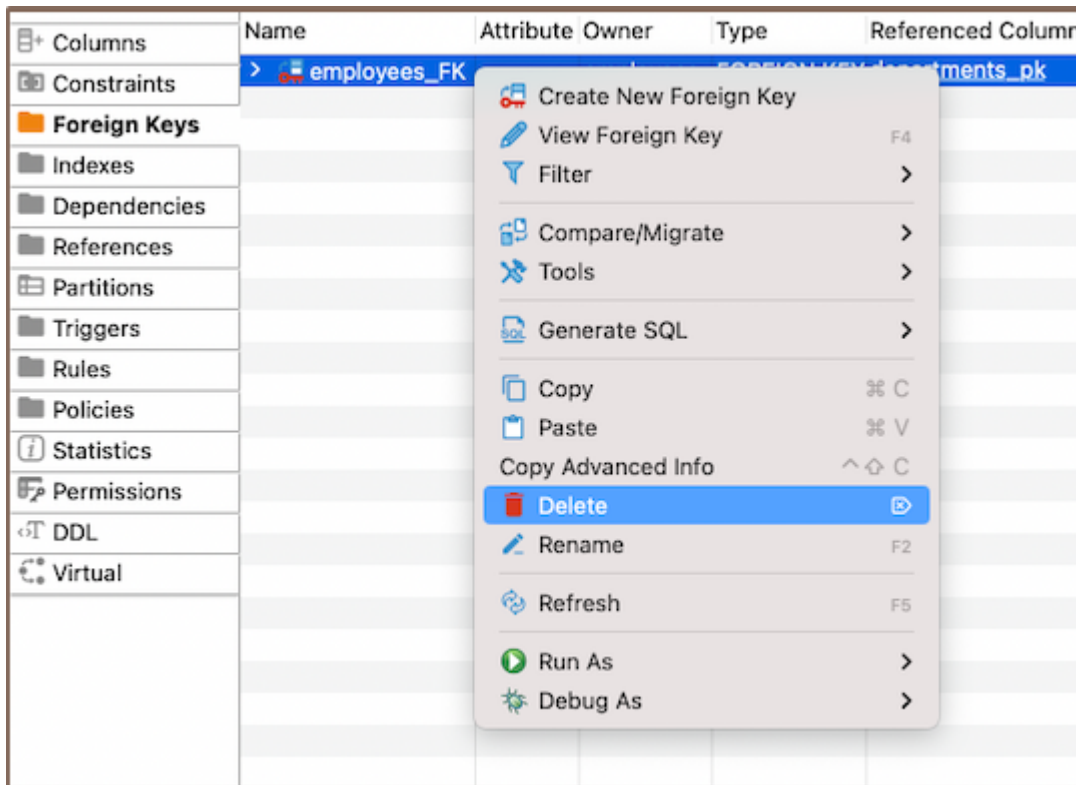
- **Properties Editor:** Open the Properties Editor by double-clicking on the table name. Within the selected table, you'll find a folder named **Foreign Keys**. Right-click on the foreign key you want to modify and select **View Foreign Key**.
- **Database Navigator:** In the Database Navigator, select the appropriate database and table. Within the selected table, you'll find a folder named **Foreign Keys**. Right-click the foreign key and select **View Foreign Key**.
- **Bottom Menu:** In the **Properties Editor** interface, use the **View Foreign Key** button  to open the properties of the selected foreign key in the **Editor panel**.




Delete

1) You can delete a foreign key either through the **Properties Editor** or the **Database Navigator**:

- Using the **Properties Editor**: Open the Properties Editor, navigate to the **Foreign Keys** tab of the corresponding table, and find the foreign key you want to delete.



- Using the **Database Navigator**: Navigate to the database where the table with the foreign key is located. Open the **Foreign Keys** folder, and find the foreign key you want to delete.

2) To delete a foreign key, right-click on the key's name and select **Delete**, or you can select the necessary foreign key and press the **Delete** key, or **Delete button**  (in the **Bottom Menu** of the **Properties Editor**). 3) A confirmation dialog will appear, asking you to confirm the deletion. Before proceeding, ensure that you've selected the correct foreign key for deletion. 4) After confirming the deletion, you'll need to persist the changes to apply them to the database.

Restrictions

- **Referential Integrity**: Foreign keys enforce referential integrity. The values in the foreign key column must match those in the referenced primary key.
- **No Circular References**: Circular foreign key references are not permitted in most databases where two or more tables refer to each other.

- **Delete and Update Operations:** Deleting or updating rows from the referenced table can be complicated because of the need to preserve referential integrity.
- **Schema Modifications:** Changes to the schema of a table (like altering data types or column deletions) can be restricted if a foreign key references the table.

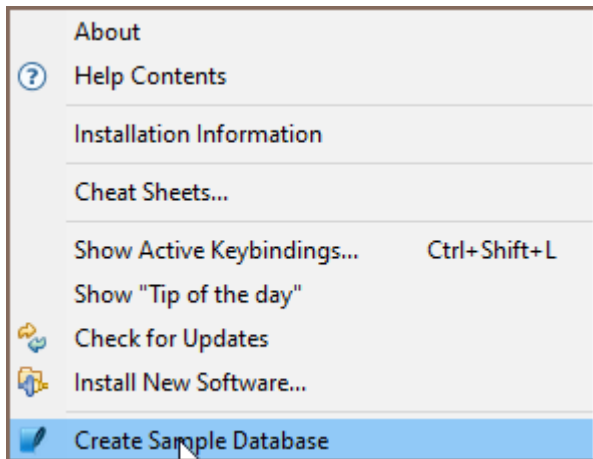
Further Reading:

[New Table Creation](#) | [Creating columns](#) | [Implementing Constraints](#) | Utilizing foreign keys | [Creating indexes](#) | [Incorporating triggers](#)

How to create a sample database

If you want to test **DBeaver** features, you can create a demo database. There is a specialized database that can be created. You can create a sample *SQLite* database at the first launch. The database can be added at any time by the following sequence. Chinook is used as a reference database

Help -> Create Sample Base



How to work with database Partitions

Table of contents

[Creating Partitions](#)

[Creating Partitions using SQL Editor](#)

[Saving changes](#)

[Creating Partitions in Oracle](#)

[Step 1: Create a new table and columns](#)

[Step 2: Modify a partitioning type](#)

[Step 3: Create a new partition](#)

[Creating a subpartition](#)

[Step 4: Save the table](#)

[Working with partitioned tables](#)

[Deleting Partitions](#)

[Database Navigator](#)

[Properties Editor](#)

[SQL Editor](#)

Partitioning is a database design technique when data is split across multiple tables or databases but is logically still one table. This technique is proper when dealing with large tables, as it can improve query performance, manageability, and availability.

Note: Not all databases support partitioning, and the level of support and specific functionalities may vary among those that do. Always refer to your specific DBMS documentation to understand the partitioning capabilities it offers.

Creating Partitions

1) The first step to creating Partitions is to [create a table](#). Consider a table with the following columns:


```
orderid int4,  
customerid int4,  
orderdate date,  
year int4
```

2) Next, go to the **Properties** tab, and find the **Partition by** field. Here, you need to specify the Partition criteria. For instance, to Partition the data based on the `year` column, you can use `RANGE (year)`.

3) Navigate to the **Partitions** in tabbed Editors. Right-click and select **Create New Partition**. This action will open a new **Partition** table window.

4) In the new window, specify the **Partition Expression**. This expression defines the boundaries for the Partition. For example, to create a Partition for the years `2023` and `2024`, you can use `FOR VALUES FROM (2023) TO (2024)`.

Creating Partitions using SQL Editor

You can also create a Partition by using SQL statements. The specific SQL command may vary slightly depending on the type of your database.

1) Here's an example of how to create a partitioned table `orders` based on a range of Partition on the `year` column using PostgreSQL syntax:

```
CREATE TABLE orders (  
    orderid int NOT NULL,  
    customerid int NOT NULL,  
    orderdate date NOT NULL,  
    "year" int NOT NULL  
) PARTITION BY RANGE ("year");
```


Note: The example provided above uses PostgreSQL syntax for partitioning. For other databases, such as Oracle, MySQL, or SQL Server, it is necessary to use the syntax that corresponds to the specific database system.

2) Next, you would create Partitions within this table. For instance, if you want to create a Partition for the year `2023`:

```
CREATE TABLE orders_2023 PARTITION OF orders FOR VALUES FROM (2023) TO (2024);
```

Saving changes

Once you have configured the Partition properties, saving these changes to the database is essential. Until you commit your modifications, the new Partition will only exist within DBeaver and will not be added to the actual database table. Here are three options for committing the changes:

- Click on **File** -> **Save** -> **Persist**.
- Select the desired table in the **Database Navigator** and press `Ctrl+S` (or `Cmd+S` for macOS), choose **Persist** to save the changes.
- Utilize the **Save** button  **Save ...** located at the bottom of the **Properties editor** and press **Persist** to save the changes.

Creating Partitions in Oracle

Partitions in Oracle are useful for managing large tables by dividing them into smaller, more manageable pieces. For a detailed guide on creating partitions and understanding the benefits of each partitioning type, refer to the [Oracle Database Partitioning Documentation](#).

To create partitions in Oracle via DBeaver, follow these steps:

Tip: Besides using the GUI for creating partitions, you can also create partitions through the SQL Editor. For instructions on using the SQL Editor for partitioning, refer to the [Creating Partitions using SQL Editor](#) section.

Step 1: Create a new table and columns

1. Create a new table including the columns that you need.
2. Specify the column(s) that will be used for partitioning in the **Partition keys** field.
3. If necessary, specify any subpartition keys by entering the relevant column name(s) in the **Subpartition Keys** field.

Tip: You can list multiple columns separated by commas. The case of the column names does not matter, you can use `column1` or `CoLumN1`.

[illegible]

Step 2: Modify a partitioning type

Optionally, navigate to the **Statistics** tab for partitioning customization.

Columns	Name	Value
Constraints	Statistics	
Foreign Keys	Stat Row Count	0
References	Last Statistics Update	
Triggers	Free %	0
Indexes	Used %	0
Partitions	Ini Trans	0
Privileges	Max Trans	0
Dependencies	Ini Extent	0
Statistics / ...	Next Extent	0
DDL	Min Extents	0
Virtual	Max Extents	0
	Increase %	0
	Freelists	0
	Freelist Groups	0
	Blocks	0
	Empty Blocks	0
	Avg Space	0
	Chain Count	0
	Avg Row Len	0
	Avg Space Freelist Blocks	0
	Freelist Blocks	0
	Size	0
	Partitioning	
	Partition Type	RANGE
	Subpartition Type	NONE
	Partition Interval	
	Partition Count	0
	Partition Tablespace	

Setting	Description	Default Value
Partition Type	Determines the method used for partitioning the table.	RANGE
Subpartition Type	Defines the method for subpartitioning, if subpartitions are used.	RANGE

Note: You can only change the **Partition Type** and **Subpartition Type** for a new table, not an existing one.

To customize the partition and subpartition types:

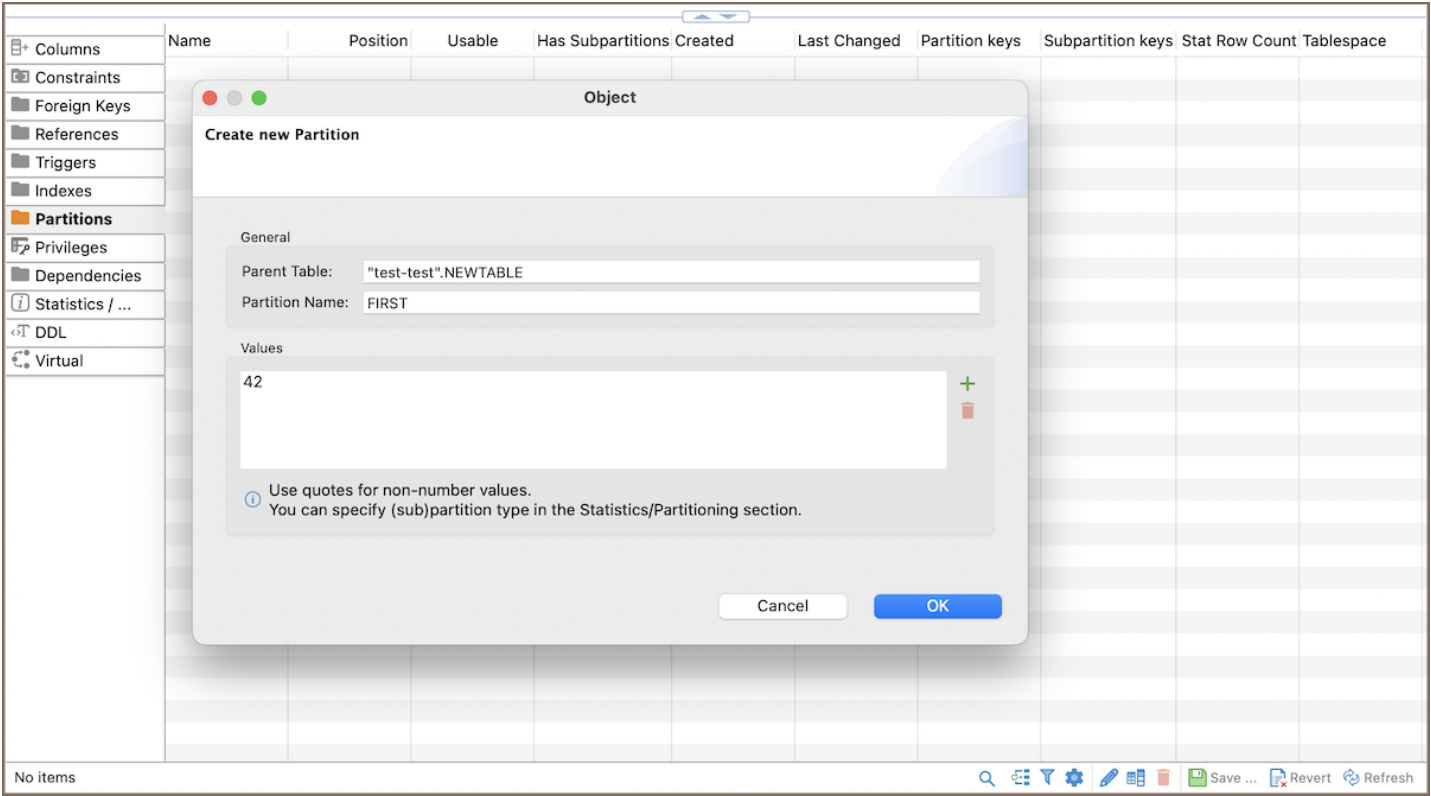
1. Click on the **Partition Type** or **Subpartition Type** value field.
2. A dropdown menu will appear where you can select from options such as **RANGE** , **HASH** , **LIST** , or **SYSTEM** .

Partitioning Type	Description
RANGE	Organizes data into partitions based on a range of values.
HASH	Distributes data across partitions based on a hash key.
LIST	Groups data into partitions based on a list of explicit values.
SYSTEM	Allows manual control over which partition stores each row.

Each type serves a different purpose and can be selected based on the specific requirements of your data organization strategy.

Step 3: Create a new partition

- 1. Go to the **Partitions** tab.
- 2. Create a new partition with the desired values.



Creating a subpartition

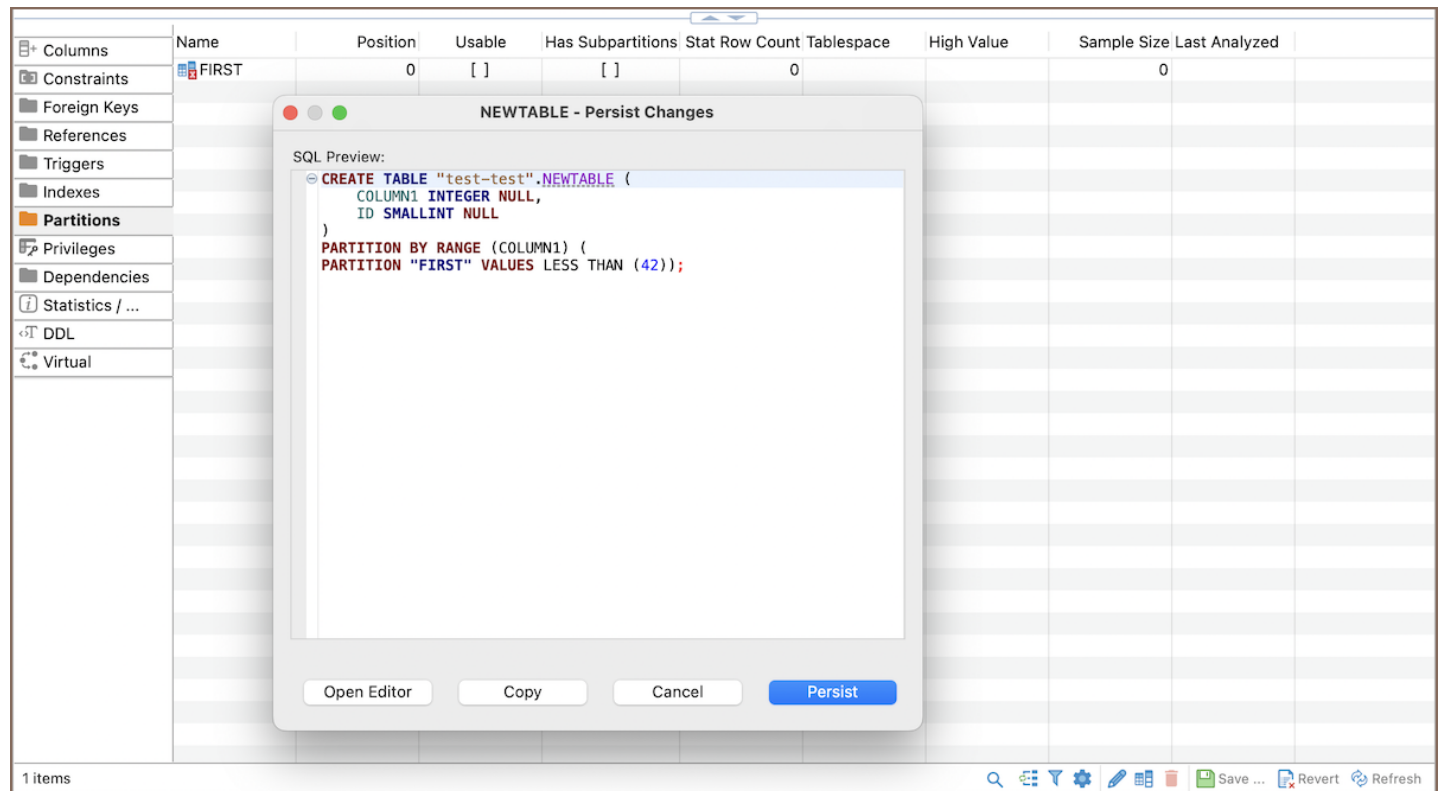
Once you have defined the partition keys and, if needed, subpartition keys, you can create a subpartition by following these steps:

- 1. Navigate to the **Partitions** tab.
- 2. Right-click on the partition within which you want to create a subpartition.
- 3. From the context menu, select **Create** and then **Subpartition** from the subsequent options.

Important: Ensure that the subpartitioning keys and types are already defined as these will dictate how the subpartitions are organized.

Step 4: Save the table

After setting up the partitions, save the table to finalize the creation process.



Working with partitioned tables

After you've created a partitioned table in DBeaver, you can interact with it just like any other table. Remember, though, that the Partition expression will impact which data goes into which Partition.

Consider the table you created before, partitioned by `RANGE (year)` with a Partition for the years `2023` to `2024`.

1) To insert data into the table, you would use a regular `INSERT` statement:

```
INSERT INTO orders (orderid, customerid, orderdate, "year")
VALUES (1, 101, '2023-01-01', 2023),
       (2, 102, '2023-06-01', 2023),
       (3, 103, '2024-01-01', 2024);
```

Each row of data would be automatically directed to the appropriate Partition based on the year value. For instance, the first two rows would be stored in the Partition for `2023`, and the third would be stored in the Partition for `2024`.

2) To retrieve data from your partitioned table, you use a standard `SELECT` statement:

```
SELECT *  
FROM orders  
WHERE year = 2023;
```

This will return all rows from the `2023` Partition.

Deleting Partitions

Deleting a Partition in DBeaver is simple and can be done via the [Database Navigator](#), the [Properties Editor](#), or the [SQL Editor](#).

Warning: When a Partition is deleted, all the data stored in that Partition is permanently lost. The Partition is also removed from the table's Partitioning scheme. It is always recommended to back up the data before deleting a Partition.

Database Navigator

- 1) In the **Database Navigator**, expand the table which contains the Partition you want to delete.
- 2) Right-click on the Partition, then click **Delete**.

Properties Editor

- 1) Go to the **Partitions** tab of the Properties Editor for your table.
- 2) Right-click on the Partition, then click **Delete**.

SQL Editor

You can also delete a Partition by using SQL statements. The specific SQL command may vary slightly depending on the type of your database.

How to import connections from external tools

Table of contents

[Overview](#)

[How to import connections](#)

[Custom connections format](#)

Overview

DBeaver provides the capability to import your connection settings from several commercial tools, making it easy to switch whether you are working simultaneously with multiple tools or planning a full move to DBeaver.

This feature allows you to migrate all necessary database connections, saving time and eliminating the need to manually recreate each connection.

Note: Secure data, such as SSH keys and passwords, are not imported automatically for security reasons. You will need to enter this information manually in DBeaver after the import process.

You can import connection settings from the following tools:

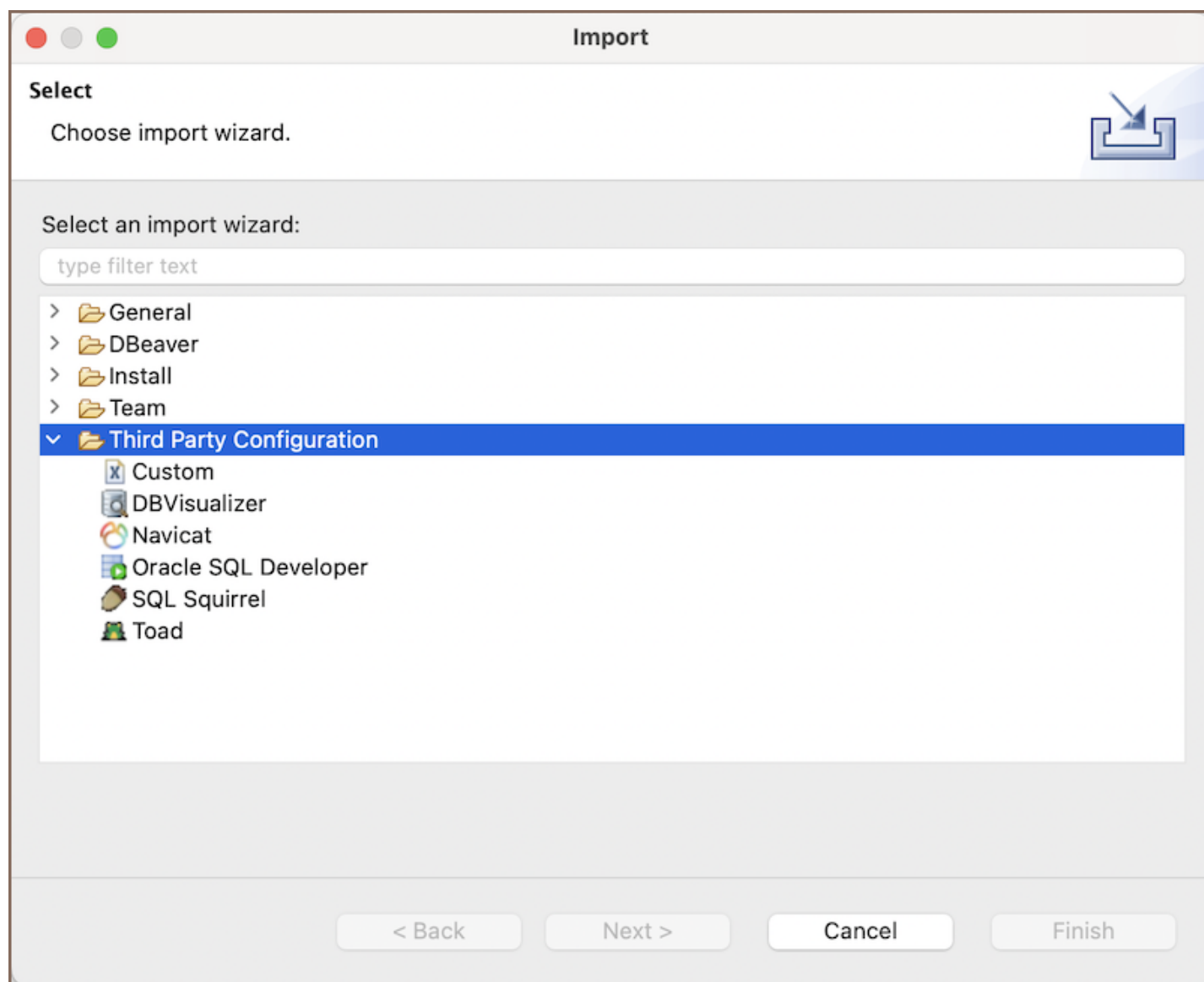
- DBVisualizer
- Navicat
- Oracle SQL Developer
- SQL Squirrel
- Toad

Note: Import functionality from DBVisualizer is supported only for versions up to version 7. Versions newer are not currently supported for import.

How to import connections

To import your database connections from the tools above into DBeaver, follow these steps:

1. Go to **File -> Import**.
2. Select the **Third Party Configuration** tab.
3. Choose the tool from which you want to import your connections.
4. Follow the specific steps provided for each tool to complete the import process.



Custom connections format

If you need to import connections from **CSV** / **XML** format, follow these steps:

1. In the list of tools, select **Custom**.
2. A window will open where you can select the required database driver.

3. In the following window, choose the input type (**XML** / **CSV**), specify the path to the file, and select the encoding from the dropdown menu.
4. Click the **Finish** button to complete the import process.

Admin manage preferences

Table of contents

[Overview](#)

[Locating preference files](#)

[Importing and Exporting workspace configuration](#)

[Reading and modifying preference files](#)

[Preferences](#)

[Default navigator view for new connections](#)

[Miscellaneous navigator settings](#)

[Default transaction settings](#)

[Metadata settings](#)

[Query manager settings](#)

[Database Navigator settings](#)

[Error Logs settings](#)

[User Interface settings](#)

[SQL Editor settings](#)

[Data Editor settings](#)

[Drivers settings](#)

Overview

This guide describes how to set up an application preferences through configuration files without even booting it up.

Locating preference files

Preference files have the extension `.prefs` and can be found in the [workspace](#) directory under the following path:

```
.metadata/.plugins/org.eclipse.core.runtime/.settings
```

You might not find a preference file you are interested in, if it has not been created yet. In this case, you will need to create one yourself. See the examples below to figure out which preference file you need.

Importing and Exporting workspace configuration

You can easily import and export your workspace configuration. Follow the steps below to perform these actions:

1. Go to **File** -> **Import** (for importing) or **Export** (for exporting).
2. In the opened window, select **DBeaver**, then choose **Preferences** from the nested menu.
3. In the new window:
 - For importing, navigate to the location where the configuration file is stored, select the file, and click **Finish**.
 - For exporting, choose the destination where you want to save the configuration file, and click **Finish**.
4. If you are importing, DBeaver will prompt you to restart the application to apply the changes. Confirm the restart to complete the import process.

Reading and modifying preference files

Preference files consist of key-value pairs glued together with `=` and separated with a line break:

```
# file org.jkiss.dbeaver.test.prefs
some.key=value1
some.other.key=value2
```

Preferences

You can place the following preferences in the `org.jkiss.dbeaver.core.prefs` file:

Default navigator view for new connections

		Default	Allowed	
--	--	---------	---------	--

Name	Type	Value	Values	Description
navigator.settings.default.preset	string	simple	simple , advanced	Sets the default view mode for a new connections. Do not specify this preference if you want to configure a custom preset.
navigator.settings.default.showSystemObjects	boolean	false	true , false	Controls whether system objects must be shown. Used if preset is not specified.
navigator.settings.default.showUtilityObjects	boolean	false	true , false	Controls whether utility objects must be shown. Used if preset is not specified.
navigator.settings.default.showOnlyEntities	boolean	false	true , false	Controls whether only schemas and tables must be shown. Used if preset is not specified.
navigator.settings.default.mergeEntities	boolean	false	true , false	Controls whether all tables must be shown in a single list (without any catalog or database). Used if preset is not specified.

For more information, please see [Simple and Advanced View](#).

Miscellaneous navigator settings

Name	Type	Default Value	Allowed Values	Description
navigator.settings.default.connectionPattern	string	host_or_database	expression with variables	Pattern for new connections title.
navigator.show.folder.placeholders	boolean	true	true , false	Displays placeholders for special folders. Show special folders (e.g. Scripts) even if they were not created yet.
navigator.sort.case.insensitive	boolean	true	true , false	Sorts items in case-insensitive mode.
navigator.sort.folders.first	boolean	true	true , false	Shows folders first.

Default transaction settings

Name	Type	Default Value	Allowed Values	Description
transaction.smart.commit	boolean	false	true , false	Enables smart commit mode.
transaction.smart.commit.recover	boolean	true	true , false	Returns to auto-commit mode after transaction end.
transaction.auto.close.enabled	boolean	false	true , false	Automatically ends (rollback) transaction after idle period.
transaction.auto.close.ttl	integer	900	integer value	Timeout before transaction close.
transaction.show.notifications	boolean	true	true , false	Shows transaction and notification.

Metadata settings

Name	Type	Default Value	Allowed Values	Description
database.meta.separate.connection	string	DEFAULT	ALWAYS , NEVER	Controls whether to open a separate connection for metadata read. Do not specify this preference if you want to use default settings.
database.meta.casesensitive	boolean	false	true , false	Specifies the usage of case-sensitive names in DDL statements.
database.props.expensive	boolean	true	true , false	Enables display of row count for tables.
database.meta.server.side.filters	boolean	true	true , false	Enables filtering on the server side.

Query manager settings

Name	Type	Default Value	Allowed Values	Description
qm. queryTypes	string	<code>USER</code> , <code>USER_FILTERED</code> , <code>USER_SCRIPT</code>	<code>USER</code> , <code>USER_FILTERED</code> , <code>USER_SCRIPT</code> , <code>UTIL</code> , <code>META</code> , <code>META_DDL</code>	View for different query types.
qm. objectTypes	string	<code>session</code> , <code>txn</code> , <code>query</code>	<code>session</code> , <code>txn</code> , <code>query</code>	Displays different object types. <code>txn</code> - Transactions.
qm. maxEntries	integer	<code>200</code>	integer value	Number of entries displayed per page.
qm. storeLogs	boolean	<code>false</code>	<code>true</code> , <code>false</code>	Enables the saving of logs.
qm. logDirectory	string	workspace directory	string value	Folder location for saving logs.
qm. historyDays	integer	<code>90</code>	integer value	Number of days for storing logs.

Database Navigator settings

Name	Type	Default Value	Allowed Values	Description
navigator.show. connection.host	boolean	<code>true</code>	<code>true</code> , <code>false</code>	Displays the conn
navigator.show. objects. description	boolean	<code>false</code>	<code>true</code> , <code>false</code>	Displays the objec
navigator.show. statistics.info	boolean	<code>true</code>	<code>true</code> , <code>false</code>	Displays the statis
navigator.show. node.actions	boolean	<code>true</code>	<code>true</code> , <code>false</code>	Displays the actio

navigator.show. folder. placeholders	boolean	true	true , false	Displays placeholder
navigator.sort. folders.first	boolean	true	true , false	Displays folders fi
navigator.sort. case.insensitive	boolean	false	true , false	Orders elements a
navigator.color. nodes.all	boolean	false	true , false	Sets connection c
navigator.show. objects.tips	boolean	true	true , false	Displays object br
navigator.show. tooltips	boolean	true	true , false	Displays informati
navigator.show. tooltips.file. contents	boolean	false	true , false	Allows to show a
navigator.object. doubleClick	string	EDIT	EXPAND	The behavior of d
navigator. connection. doubleClick	string	EXPAND	EDIT , CONNECT , SQL_EDITOR , SQL_EDITOR_NEW	The behavior of d
navigator.object. defaultEditorPage	string	""	default.object.editor , org.jkiss.dbeaver.ui.editors.data.DatabaseDataEditor , org.jkiss.dbeaver.erd.ui.editor.ERDEditorEmbedded org.jkiss.dbeave	Defines which edi Properties Editor (Editor (org.jkiss.dbeave) , or the ERD Editc org.jkiss.dbeave
navigator.expand. on.connect	boolean	false	true , false	Expands navigato
navigator.restore. filters	boolean	false	true , false	Determines wheth upon reopening.

navigator.long.list.fetch.size	integer	5000	integer values	Determines nodes
navigator.restore.state.depth	integer	0	integer values	Restores the navig depth.

Error Logs settings

Name	Type	Default Value	Allowed Values	Description
logs.debug.enabled	boolean	true	true , false	Enables debug logs.
logs.debug.location	string	\${workspace}/.metadata/dbeaver-debug.log	string values	Location of the log file.
logs.files.output.maxSize	integer	10485760	integer values	Maximum log file size (KB).
logs.files.output.maxCount	integer	3	integer values	Maximum backup log files count.

User Interface settings

Name	Type	Default Value	Allowed Values	Description
ui.auto.update.check	boolean	true	true , false	Enables automatic updates check.
platform.language	string	en	en , zh_CN , ru , fr , de , it , ja , es , pt , ko , zh_TW ,	Platform language setting.
java.client.timezone	string	operating system time zone	time zones	Time zone setting.
notifications.				Determines whether notification popups

enabled	boolean	true	true , false	are allowed to appear.
notifications. closeDelay	integer	3000	integer values	Defines the timeout (in ms) before a notification will disappear.

SQL Editor settings

Name	Type	Default Value	Allowed Values	Description
database.editor. separate.connection	string	DEFAULT	ALWAYS , NEVER	Opening a separate connection for each editor.
database.editor.connect. on.activate	boolean	true	true , false	Connects on editors activation.
database.editor.connect. on.execute	boolean	false	true , false	Connects on query execution.
SQLEditor. autoSaveOnChange	boolean	false	true , false	Auto-saves after any modifications.
SQLEditor. autoSaveOnClose	boolean	false	true , false	Auto-saves editor on close.
SQLEditor. autoSaveOnExecute	boolean	false	true , false	Saves editor on query execution.
SQLEditor. autoSaveActiveSchema	boolean	true	true , false	Saves/restores active schema.
SQLEditor.resultSet. closeOnError	boolean	false	true , false	Closes results tab on error.
SQLEditor.resultSet. replaceCurrentTab	boolean	true	true , false	Replaces active result tab on single query execution.
SQLEditor.resultSet. orientation	string	HORIZONTAL	VERTICAL	Result view orientation.
SQLEditor.outputPanel. autoShow	boolean	true	true , false	Opens output viewer on new messages.

SQLEditor.outputPanel. autoShow	integer	20	integer values	Sets maximum of result tabs for single query.
SQLEditor. ContentAssistant.auto. activation.enable	boolean	true	true , false	Enables SQL Assistant auto-activation.
SQLEditor. ContentAssistant. activate.hippie	boolean	false	true , false	Uses Hippie Engine for autocompletion.
SQLEditor. ContentAssistant.auto. activation.delay	integer	0	integer values	Auto activation delay (ms).
SQLEditor. ContentAssistant.auto. keystrokes.activation	boolean	true	true , false	Activates SQL Assistant on typing.
SQLEditor. ContentAssistant.insert. single.proposal	boolean	true	true , false	Auto inserts proposal in SQL assistant if there is only one option.
SQLEditor. ContentAssistant. autocompletion.tab	boolean	true	true , false	Use Tab for autocompletion in SQL Assistant.
SQLEditor. ContentAssistant.insert. case	integer	0	1 , 2	SQL Assistant insertion case (1- Upper case, 2 - Lower case).
SQLEditor. ContentAssistant. replace.word	boolean	false	true , false	Sets up whether to replace the current word on accepting autocompletion proposal.
SQLEditor. ContentAssistant. proposals.short.name	boolean	false	true , false	Uses short object names in SQL Assistant.
SQLEditor. ContentAssistant.show. helpTopics	boolean	false	true , false	Shows server help topics in SQL Assistant.

SQLEditor. ContentAssistant.show. values	boolean	true	true , false	Proposes values for the given attribute. Note that it can drastically degrade performance when working with large tables, as it selects distinct values.
sql.proposals.insert. table.alias	string	PLAIN	NONE , EXTENDED	Inserts table aliases (in FROM clauses).
SQLEditor.format. activeQuery	boolean	true	true , false	Enables formatting of the query under cursor only.
script.auto.folders	boolean	false	true , false	Creates script folder for each connection.

Data Editor settings

Name	Type	Default Value	Allowed Values	Description
resultset.maxrows	integer	200	integer values	Determines the number of rows fetched from the database in a single batch.
resultset.autofetch.next. segment	boolean	true	true , false	Automatically loads the next data segment when the current one is fully processed or viewed.
resultset.reread.on. scroll	boolean	true	true , false	Refreshes the data when moving to the next page.
resultset.maxrows.sql	boolean	false	true , false	Modifies the source SQL query to scroll through or limit results. This is typically achieved using the LIMIT/OFFSET SQL clause.
resultset.order.mode	string	SMART	CLIENT_SIDE , SERVER_SIDE , SMART	Specifies the method of sorting returned query results.
				Retrieves information

resultset.read.metadata	boolean	true	true , false	about the table structure, specifically the unique keys.
resultset.read.references	boolean	true	true , false	Retrieves information about the table structure, specifically the foreign keys.
resultset.cancel.timeout	integer	5000	integer values	Defines the duration (in milliseconds) the system waits before cancelling a non-responsive query.
sql.query.filter.force.subselect	boolean	false	true , false	Forces the usage of subqueries when applying filters to the data.
resultset.column.label.ignore	boolean	false	true , false	Prioritizes column names over column labels when referencing data.
resultset.edit.key.use_all_columns	boolean	false	true , false	Forms a virtual composite key using all table columns for data saving. If set to true , this will determine the action taken if a key element is missing. If set to false , no action will be taken when a key element is missing.
resultset.edit.disableIfKeyMissing	boolean	false	true , false	Prevents data modification when a primary key is absent. This setting defines the action taken when the assumed primary key is missing.
keep.statement.open	boolean	false	true , false	Closes the result set after a selection. To view a cursor column value, the option should be enabled.

resultSet.datetime. editor	boolean	false	true , false	Enables the advanced datetime editor in the data editor.
resultSet.edit.new.row. after	boolean	true	true , false	Places newly inserted rows immediately following the currently selected row.
resultSet.edit. refreshAfterUpdate	boolean	true	true , false	Automatically refreshes the data after making changes.
resultSet.filter.use. navigator	boolean	true	true , false	Employs filters in the navigator to refine the displayed data.
resultSet.confirm. beforeSave	boolean	false	true , false	Displays a prompt asking for a confirmation before saving changes.
resultSet.show. errorDialog	boolean	false	true , false	Presents error messages in a separate dialog box.
resultSet.mark.cell.value. occurrences	boolean	false	true , false	Highlights instances of the selected cell's value within the data set.
resultSet.image. browser	boolean	true	true , false	Displays images in the web browser.
resultSet.behavior. autoSwitchMode	boolean	false	true , false	Allow to switch between Record mode and Grid mode when dealing with single or multiple rows. When set to <code>true</code> , the system will automatically switch modes based on the

				number of rows selected (Record mode for a single row, Grid mode for multiple rows).
resultset.calc. columnWidthByValues	boolean	false	true , false	Displays the description for each column in the header.
resultset.show. columnDescriptio	boolean	false	true , false	Adjusts the width of each column based on the content of its cells.
resultset.show. connectionName	boolean	false	true , false	Displays the name of the connection in the status section.
resultset.transform. complex.type	boolean	true	true , false	Enables the visualization of complex columns (such as objects, structures and arrays).
resultset.show. rightJustifyNumbers	boolean	true	true , false	Allows right-justify numbers.
resultset.show. rightJustifyDateTime	boolean	true	true , false	Aligns date and time values to the right side of their respective cells or columns.
resultset.behavior. autoCompleteProposial	boolean	true	true , false	Activates the auto-completion feature in the data filter text field.
resultset.show. oddRows	boolean	true	true , false	Marks odd and even rows in the grid.
resultset.show. colorizeDataTypes	boolean	false	true , false	Colorized data types in the grid.
resultset.show.attlcons	boolean	true	true , false	Displays icons representing the data types of

				each column in the grid.
resultset.show.attFilters	boolean	true	true , false	Enables the display of attribute filters in the grid.
resultset.show.attrOrder	boolean	false	true , false	Shows asc/desc order icon in column header.
resultset.show.collections.inline	boolean	false	true , false	Enables the inline display of collections within the grid.
resultset.show.boolean.checkbox	boolean	true	true , false	Represents boolean values in the grid as checkboxes.
resultset.show.whitespace.characters	boolean	false	true , false	Displays whitespace characters within the grid.
resultset.grid.value.format	string	UI	EDIT , NATIVE , UI	Determines the format in which values are displayed within the grid.
resultset.grid.value.format	integer	30	integer values (from 0 to 100)	Determines the maximum percentage of width that a column can occupy in the grid.
resultset.highlight.selectedRows	boolean	true	true , false	Enables the highlighting of entire rows when one or more cells within those rows are selected.
resultset.hScroll.smooth	boolean	true	true , false	Enables smooth scrolling within the DBeaver.
resultset.show.boolean.toggleOnClick	boolean	true	true , false	Allows the boolean values to be toggled or switched between true and false with a single click.

resultset.behavior. inlineEnter	boolean	false	true , false	Selects the next cell after the inline edit completed (or finished).
resultset.behavior. doubleClick	string	INLINE_EDITOR	NONE , EDITOR , INLINE_EDITOR , COPY_VALUE , COPY_PASTE_VALUE	Defines the action taken when a cell in a grid is double-clicked (Options: EDITOR, INLINE_EDITOR, COPY_VALUE, COPY_PASTE_VALUE, NONE).
resultset.show.row. batch.size	integer	1	integer values	Number of consequent rows with the same background color. Default = 1 means odd/even rows coloring.
resultset.text.tab.size	integer	4	integer values	Determines tab size in plain text.
resultset.text.max. column.size	integer	255	integer values	Determines the maximum length of a column's content when viewing data in plain text.
resultset.text.value. format	string	EDIT	EDIT , UI , NATIVE	Determines how plain text values are displayed in DBeaver.
resultset.text.show. nulls	boolean	false	true , false	Shows NULL if this setting is enabled or nothing if disabled.
resultset.text.line. number	boolean	false	true , false	Enables the display of line numbers.
resultset.text.delimiter. leading	boolean	false	true , false	Enables trailing delimiter in plain text for Grid Representation.
				Enables trailing delimiter in

resultset.text.delimiter. trailing	boolean	true	true , false	plain text for Grid Representation.
resultset.text.delimiter. top	boolean	false	true , false	Enables top delimiter in plain text for Grid Representation.
resultset.text.delimiter. bottom	boolean	false	true , false	Enables bottom delimiter in plain text for Grid Representation.
resultset.text.extra. spaces	boolean	false	true , false	Enables extra spaces in plain text for Grid Representation.
resultset.string.use. content.editor	boolean	false	true , false	Opens separate string editor instead of dialog.
resultset.binary. representation	string	string	hex , base64 , string	Determines the format in which binary data is displayed.
resultset.binary.editor	string	EDITOR	EDITOR , PANEL	Enables to change the binary data editor. Choose between EDITOR (dedicated binary data editor) or PANEL (dialog window for viewing/editing binary data).
resultset.binary. stringMaxLength	integer	32	integer values (max: 3200)	Determines the maximum length of binary strings that can be displayed in the binary editors.
content.memory. maxsize	integer	10000	integer values (max: 1000000000)	Determines the maximum length of LOB (Large Object) data that

				will be kept in memory when working with binary editors.
content.hex.encoding	string	UTF-8	All available encodings from the current JVM	Allows to specify the character encoding used when interpreting binary data as strings.
hex.font.name	string	Courier New	Courier New , Andale Mono , Menio , Monaco , PT Mono	Allows to specify the name of the font to be used in the data editor.
hex.font.style	integer	0	0 - Regular, 1 - Bold, 2 - Italic, 3 - Bold Italic	Allows to choose the font style for the data editor.
hex.font.size	integer	10	6 - 14 , 16 , 18 , 22 , 32 , 72	Allows to choose the font style for the data editor.
default.hex.width	integer	8	4 , 8 , 16	Allows to choose the default width for the data editor.
resultset.json. maxObjectMembers	integer	200 (max: 2000000000)	integer values	Specifies the threshold for collapsing JSON objects in the data editor.
resultset.json. maxArrayElements	integer	200 (max: 2000000000)	integer values	Determines the threshold for collapsing arrays in the JSON data editor.
resultset.json. maxElementNesting	integer	4 (max: 4000000000)	integer values	Determines which objects in JSON will be collapsed (hidden) if their nesting

				level exceeds a specified threshold.
dataformat.type. number. nativeSpecialValues	boolean	false	true , false	Instructs the data editor to utilize the native formatting for special values.
dataformat.type. number.useGrouping	boolean	false	true , false	Indicates whether grouping is used to display numbers.
dataformat.type. number.useTypeScale	boolean	false	true , false	Uses column/attribute data type scale as minimum fraction digits number.
dataformat.type. number.roundingMode	string	UNNECESSARY	UP , DOWN , CEILING , FLOOR , HALF_UP , HALF_DOWN , HALF_EVEN , UNNECESSARY	Determines the rounding mode used for numerical values.
dataformat.type. number. minIntegerDigits	integer	1	integer values	Determines the minimum number of digits to display for integer values.
dataformat.type. number. minFractionDigits	integer	0	integer values	Determines the minimum number of fractional digits displayed.
dataformat.type. number. maxIntegerDigits	integer	2147483647	integer values	Defines the maximum number of digits allowed for an integer value.
dataformat.type. number. maxFractionDigits	integer	10	integer values	Specifies the maximum number of fraction digits allowed.

dataformat.type. number.groupingSize	integer	3	integer values	Determines the grouping size for displaying numeric values.
dataformat.type. timestamp.pattern	string	yyyy-MM-dd HH:mm:ss.SSS	timestamp format	Determines the format for displaying and editing timestamp data.
dataformat.type. timestamptz.pattern	string	yyyy-MM-dd HH:mm:ss.SSS Z	timestamp with timezone format	Determines the format for displaying timestamp values with timezone information.
dataformat.type.date. pattern	string	yyyy-MM-dd	date format	Determines the format used for displaying and editing dates.
dataformat.type.time. pattern	string	HH:mm:ss	time format	Determines the format used for displaying and editing time.
dataformat.type.timetz. pattern	string	HH:mm:ss Z	time with timezone format	Determines the format used for displaying and editing time with timezone.

The preferences of **GIS Viewer** from **Data Editor** you can find in the `org.jkiss.dbeaver.data.gis.view.prefs` file:

Name	Type	Default Value	Allowed Values	Description
gis.view. minZoomLevel	integer	0	integer values	Determines the maximum level of zoom.
gis.view. maxObjectsRender	integer	10000	integer values	Determines the maximum number of objects that can be displayed.
gis.view.defaultSRID	integer	4326	integer values	Determines the default Spatial Reference Identifier (SRID).

Drivers settings

Name	Type	Default Value	Allowed Values	Description
ui.drivers.home	string	<code>path/to/DBeaverData/drivers</code>	string values	Specifies the local folder for driver storage.
ui.proxy.host	string	""	string values	Identifies the host for the HTTP proxy when downloading drivers.
ui.proxy.port	integer	<code>1080</code>	integer values	Specifies the port number used by the HTTP proxy during driver downloads.
ui.proxy.user	string	""	string values	Determines the username for accessing proxy during driver downloads.
ui.proxy.password	string	""	string values	Sets the password required for the proxy during driver downloads.
ui.drivers.sources	string	<code>https://dbeaver.io/files/jdbc/</code>	string values	Provides the URL where driver files can be downloaded.
ui.drivers.global.libraries	string	""	string values	Defines the global classpath from which global libraries can be loaded.
ui.drivers.version.update	boolean	<code>false</code>	<code>true</code> , <code>false</code>	Determines whether the system checks for new driver versions.

Admin manage connections

Table of contents

[Provide predefined connections](#)

[Importing connections from CSV or XML](#)

[Supported names](#)

[Sample CSV](#)

[Sample XML](#)

[Secure connections from editing](#)

[Using environment variables](#)

This guide describes how to manage/secure the DBeaver database connections. It is designed for System administrators. Regular users should check [this](#) guide.

Provide predefined connections

DBeaver keeps connections information in the project folder. By default, all projects reside in the [workspace](#).

The default project folder is [workspace](#)\workspace6\General.

DBeaver up to 6.1.3

DBeaver keeps information about project connections in the `.dbeaver/data-sources.json` file.

All secured information (user name, password, secret keys, etc) is stored in the encrypted file,

`.dbeaver/credentials-config.json`.

DBeaver can load multiple connection files. Any files in the project folder matching the

`.dbeaver/data-sources*.json` pattern will be loaded on the startup. So you can create a file, for example,

`.dbeaver/data-sources-2.json` in the project folder and DBeaver will see it.

DBeaver < 6.1.3 (Legacy)

DBeaver keeps information about project connections in the `.dbeaver-data-sources.xml` file.

DBeaver can load multiple connection files. Any files in the project folder matching the

`.dbeaver-data-sources*.xml` pattern will be loaded on the startup. So you can create a file, for example,

`.dbeaver-data-sources-2.xml` in the project folder and DBeaver will see it.

Importing connections from CSV or XML

You can import a connection from CSV or XML files.

The CSV file must have a header row (first line of file) with column names (see list of supported columns below).

The XML file should contain a top-level element and a set of nested elements. The connections config must be specified in the attributes of the nested elements. Attribute names are the same as the CSV column names.

Supported names

Name	Meaning
name	Connection name
url	JDBC URL
host	Database server host name
port	Database server port
database	Database/schema name
user	User name
password	User password

You can only specify the URL or the host/port/etc setting.

User name/password are optional.

Sample CSV

```
name,host,port,server,database,url,user,password,type
Postgre Import XML 1,localhost,5432,,postgres,jdbc:postgresql://localhost:5432/postgres,postgres,postgres,
Postgre Import XML 2,localhost,5432,,postgres2,jdbc:postgresql://localhost:5432/postgres2,postgres2,postg
```


Sample XML

```
<connections>
  <connection name="Postgre Import XML 1" host="localhost" port="5432" server="" database="postgres"
  <connection name="Postgre Import XML 2" host="localhost" port="5432" server="" database="postgres"
</connections>
```

Secure connections from editing

It is possible to set the connection settings as read-only (protected by password)

- Generate MD5 hash of your password. You can do it from the command line using Linux utility md5sum (`md5sum <<<"your password")`) or you can do it online - just google "MD5 hash online".
- Add field `lockPassword` in the connection descriptor (in `.dbeaver/data-sources.json` in `connections` element. So it will look like this:

```
"postgres-jdbc-161537836e8-3e0957d039995715": {
  "provider": "postgresql",
  "driver": "postgres-jdbc",
  "name": "PostgreSQL - postgres",
  "save-password": true,
  "show-system-objects": true,
  "read-only": false,
  "folder": "PG",
  "lockPassword": "2ba81a47c5512d9e23c435c1f29373cb"
...
}
```

- If the user will try to change connection settings now, he/she will be asked for a password.

Using environment variables

You can use references on environment variables in most of connection configuration properties.

For example:

```
"postgres-jdbc-161537836e8-3e0957d039995715": {
  "provider": "postgresql",
  "driver": "postgres-jdbc",
  "name": "PostgreSQL - postgres",
  "user": "${dbeaver.default-user}",
...
}
```

Admin manage drivers

Table of contents

- [Configure drivers with pre-installed jars](#)
- [Provide predefined drivers configuration](#)

Configure drivers with pre-installed jars

You can customize drivers configuration in the [workspace](#)/.metadata/.plugins/org.jkiss.dbeaver.core/drivers.xml file. If you have some pre-installed jar files you can reference them in drivers.xml. Example:

```
<library type="jar" path="absolute-jar-folder-path\driver-jar.jar" custom="true" />
```

Also in drivers.xml you can use the following variables to specify relative paths:

Variable	Meaning
drivers_home	Standard DBeaver drivers location - (\$workspace/drivers by default)
dbeaver_home	DBeaver installation folder
home	User home folder
workspace	DBeaver workspace path

For instance:

```
<library type="jar" path="${workspace}\drivers\my-driver.jar" custom="true" />
```

Full drivers.xml example:

```
<?xml version="1.0" encoding="UTF-8"?>
<drivers>

    <provider id="postgresql">
        <driver id="postgres-jdbc" custom="false" embedded="false" name="PostgreSQL" class="org.postgresql.Driver">
            <library type="jar" path="maven:/org.postgresql:postgresql:RELEASE" custom="false">
                <file id="org.postgresql:postgresql" version="42.2.20" path="{drivers_home}/org/postgresql/lib/postgresql-42.2.20.jar" />
            </library>
            <library type="jar" path="maven:/net.postgis:postgis-jdbc:RELEASE" custom="false">
                <file id="net.postgis:postgis-jdbc" version="2.5.0" path="{drivers_home}/net/postgis/lib/postgis-jdbc-2.5.0.jar" />
            </library>
            <library type="jar" path="maven:/net.postgis:postgis-jdbc-jtsparser:RELEASE" custom="false">
                <file id="net.postgis:postgis-jdbc-jtsparser" version="2.5.0" path="{drivers_home}/net/postgis/lib/postgis-jdbc-jtsparser-2.5.0.jar" />
            </library>
        </driver>
    </provider>

</drivers>
```

Provide predefined drivers configuration

In some cases you may need to provide a driver's configuration or driver `jar` files for a number of DBeaver installations automatically.

This can be done by adding a special parameter in the `dbeaver.ini` file:

```
-Ddbeaver.drivers.configuration-file=c:\some-path\dbeaver-drivers-config.xml
```

Tip: Detailed instructions on finding `dbeaver.ini` are available in [our article](#).

This file has the same structure as `drivers.xml` file (see above) and it will be loaded *before* `drivers.xml`.

You can specify partial driver configuration. For example if you need to configure only the jar path then it may look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<drivers>
    <provider id="generic">
        <driver id="netezza">
            <library type="lib" path="X:\jdbc-drivers\netezza-jdbc.jar" />
        </driver>
    </provider>
</drivers>
```

Admin preference restrictions

Table of contents

[Overview](#)

[Modifying the DBeaver configuration file](#)

[Using Windows Registry](#)

[Configuration options](#)

[Registry and configuration file parameters](#)

[Configuration file parameters](#)

[Master password policy settings](#)

Overview

This guide outlines the steps for enhancing security and control in DBeaver by restricting specific functionalities.

For Windows, Mac, and Linux, restrictions in the `dbeaver.ini` file can be implemented, while Windows users can also leverage the Registry for additional control.

Tip: Detailed instructions on finding `dbeaver.ini` are available in [our article](#).

Follow the appropriate steps based on your operating system and preference.

Modifying the DBeaver configuration file

1. Navigate to the directory containing the `dbeaver.ini` file.
2. Use a text editor to open the configuration file.
3. Locate the `-vmargs` line within the file.
4. Add the desired settings from the [Configuration options](#), ensuring each setting is preceded by `-D` (for instance `-Dpolicy.data.export.disabled = true`) and placed below the `-vmargs` line.
5. Save the edited file.

- Restart DBeaver to apply changes.

Using Windows Registry

- Open the Registry Editor by typing `regedit` in the Run dialog (Win + R).
- Navigate to the following path in the Registry: `HKEY_CURRENT_USER\Software\DBeaver Corp\DBeaver\policy`
- Create a new value for each setting, from the [Configuration options](#) based on desired restrictions.
- Exit the Registry Editor and restart DBeaver to apply the changes.

Important: Editing the Windows Registry can significantly affect your system. Always make sure to back up the registry before making any changes. For more information on how to work with the Windows Registry, refer to the [Microsoft Documentation on Registry Editing](#).

Configuration options

This section consolidates the configuration options, applicable through Windows Registry keys and as settings in the DBeaver configuration file.

Registry and configuration file parameters

This section outlines the parameters that can be configured either through the Windows Registry or directly in the `dbeaver.ini`.

Registry key/Variable Name	Type	Default value	Allowed values	Description
<code>policy.data.export.disabled</code>	<code>DWORD</code> / <code>Boolean</code>	<code>0</code> / <code>false</code>	<code>0</code> / <code>false</code> , <code>1</code> / <code>true</code>	Restricts ability to export data from databases.
<code>policy.software.install.disabled</code>	<code>DWORD</code> / <code>Boolean</code>	<code>0</code> / <code>false</code>	<code>0</code> / <code>false</code> , <code>1</code> / <code>true</code>	Blocks ability to install new software and access the Eclipse marketplace.

<code>policy.software.update.disabled</code>	DWORD / Boolean	0 / false	0 / false , 1 / true	Prevents ability to check for DBeaver updates.
<code>connection.credentials.save.restricted</code>	DWORD / Boolean	0 / false	0 / false , 1 / true	Disables the ability to save database credentials across all user profiles.

Configuration file parameters

This section details the parameters that can be set only in the `dbeaver.ini`.

Variable Name	Default Value	Allowed Values	Description
<code>dbeaver.disable.secure.masking</code>	false	false , true	Disables the masking of sensitive information in log files.

Master password policy settings

Registry key/Variable Name	Type	Default value
<code>policy.master.password.disable</code>	DWORD / Boolean	1 / true
<code>policy.master.password.policy</code>	REG_SZ / String	{"minLength": 8, "minLetterCount": 1, "minNumberCount": 1, "minSymbolCo

--	--	--

Admin variables

Table of contents

[Variables](#)

[What are variables](#)

[Using variables](#)

[The Variable resolution](#)

[Declare external variables in a file](#)

Variables

Disclaimer: this article does NOT cover variables used in SQL editor.


What are variables

A variable is a special template which is replaced with an associated value. Variables help keep your configuration clean and tidy by avoiding unnecessary repetitions: instead of manually replacing each occurrence of some value, you can replace the value of a variable just once and its occurrences will be kept intact.

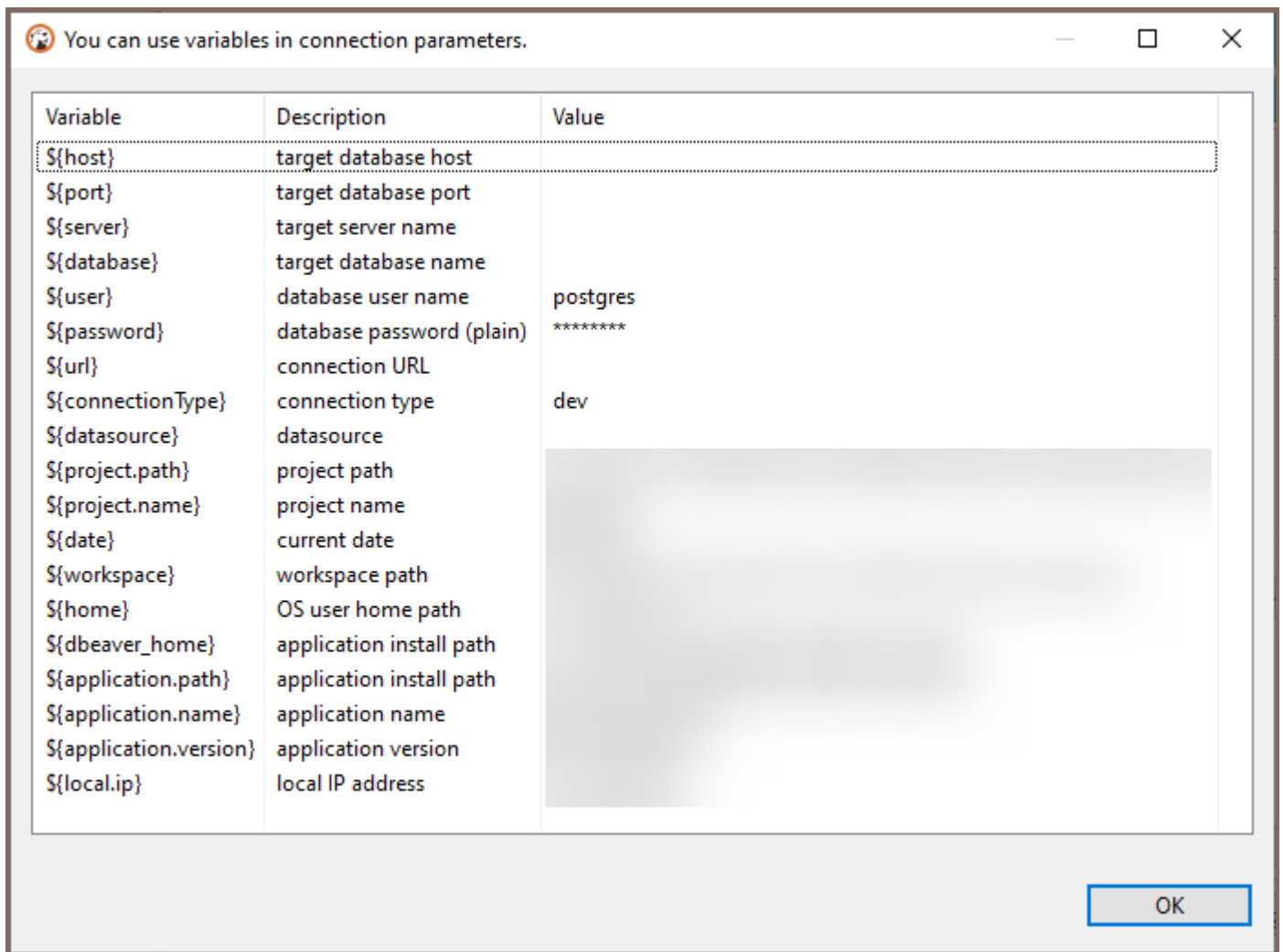
Variables are available in a lot of places. You might find them in:

1. The Connection settings
2. The SSH tunnel settings
3. [Data Transfer](#) and in other tasks
4. The Command Line Interface

Wherever you find variables available in the user interface, a hint will usually be present:

 You can use variables in connection parameters.

By clicking on it, a separate window will open showing all the available variables with their descriptions and values:



Using variables

To use the variables you need to utilize a special syntax:

```
${variable_name}
```

Where `variable_name` is a variable name which you want to use. The window shown above already contains `${` and `}` anchors so you do not need to specify them multiple times.

Variables may be a part of something bigger: e.g., a part of a path. For example, if some file lies under the user's root folder, you may want to use variable `home` : `${home}` : `${home}/path/to/file.txt` .

The Variable resolution

A value of a particular variable may be resolved from different places depending on its origin.

The resolution is performed in the following order (from the highest to the lowest priority):

1. Dialog, where a specific variable is used
2. Datasource-specific variables (`host` , `port` , `server` , `database` , etc.)
3. Application-specific variables (`application.name` , `application.version` , `home` , etc.)
4. Environment variables. For additional information see [Wikipedia](#).
5. External configuration (see [the below](#))

If you have an environment variable called `home` , it will be resolved in the application-specific variable because it has higher priority.

Declare external variables in a file

You can create a file and fill it with pairs of named values and pass it to DBeaver using the `-vars` [command-line](#) argument.

Variables from this file can be accessed by other command-line arguments, in the data transfer wizard, and in other places that support variable resolutions.

For example, you may want to put your credentials in that file to avoid showing them to everyone else:

```
# Lines that start with the `#` symbol are comments and therefore ignored.  
sampleVar1=abc  
someOtherVar=DBeaver is cool  
password=P4$$w0r3
```

You can use them as such:

```
dbeaver.exe -vars C:\secrets.properties -con 'driver=<xxx>|url=<xxx>|password=${password}'
```

Here, the `-con` argument has the `${password}` variable that will be replaced with `P4$$w0r3` defined in the example file from above.

Command line

Table of contents

[Windows](#)

[Mac](#)

[Command line parameters](#)

[DBeaver control](#)

[System parameters](#)

[VM arguments](#)

[Connection parameters](#)

[Declare external variables in a file](#)

Command line parameters can be passed directly to the DBeaver executable. The way to do this depends on your operating system:

Windows

You can use the `dbeaver-cli.exe [parameters]` executable. This executable does not spawn a new window, so you can see the output messages.

Mac

Parameters can be passed to DBeaver in one of two ways:

- Use the open command followed by the `-a` flag and the name of the application, along with any necessary arguments. Depending on the version of DBeaver, you might need to specify `DBeaver.app`, `DBeaverLite.app`, `DBeaverEE.app`, `DBeaverUltimate.app`, or `DBeaverTeam.app`. For example:

```
open -a "DBeaver.app" --args [parameters] .
```

Note: This method does not redirect logging messages, stdout, and stderr to the terminal.

- Pass parameters directly to the DBeaver executable in the terminal using the path where the `.app` file is located. For example: `/Applications/DBeaver.app/Content/MacOS/dbeaver [parameters]`. This method redirects logging messages, stdout, and stderr to the terminal.

For both Windows and Mac, parameters can also be added in the `dbeaver.ini` configuration file. These should be written at the beginning of the file, with each parameter on its own line.

Tip: Detailed instructions on finding `dbeaver.ini` are available in [our article](#).

Command line parameters

DBeaver control

Name	Value	Example
-help	Prints help message	
-stop	Quits DBeaver	
-dump	Prints DBeaver thread dump	
-f	Opens the file in DBeaver UI, if the command has -con argument, connects it to datasource	<code>-f c:\some-path\some-file.sql</code>
-con	Opens database connection in DBeaver UI	See connection parameters table
-closeTabs	Closes all open editor tabs	
-disconnectAll	Closes all open connections	
-reuseWorkspace	Forces reuse of single workspace by multiple DBeaver instances	
-newInstance	Forces new DBeaver instance creation (do not try to reuse already running one)	
-bringToFront	Brings the DBeaver window on top of other applications	

-var ★	<p>Customs variables for runTask. You can change existing variables in the task. You cannot add new task variables with this parameter. You can add several parameters at once to the command line, each starting with "-var". Used right before -runTask. Template:</p> <pre>-var variableName=variableValue</pre>	<pre>-var film=sakila.film</pre> <pre>-var actor=sakila.actor</pre> <pre>-runTask "exportFromSakila"</pre> <p>PRO versions only.</p>
-vars	Path to a property file with variables	<pre>-vars c:\path\to\file.properties</pre> <p>For more information see the main article</p>
-runTask ★	Executes specified task	<pre>-runTask "@projectName:taskName"</pre> <p>PRO versions only. See Task Scheduler.</p>
-license ★	Path to the license file	<pre>-license "/etc/licenses/dbeaver.txt"</pre> <p>PRO versions only.</p>

System parameters

Name	Value	Example
-nl	Locale	en_US
-data	Workspace path	c:\ProgramData\MyWorkspace
-nosplash	Omits splash screen	true
-clean	Clears all Eclipse caches. Use it if DBeaver fails to start after it upgrades.	
-vmargs	VM parameters	See VM arguments table

VM arguments

You can pass any advanced Java parameters supported by your local JVM. Parameters supported by HotSpot JVM (17): <https://docs.oracle.com/en/java/javase/17/docs/specs/man/java.html>

Parameters supported by all JVMs:

Name	Value	Example

-Xms	Sets initial memory available for DBeaver	<code>-Xmx1000m</code>
-Xmx	Sets maximum memory available for DBeaver	<code>-Xmx4000m</code>

Connection parameters

All connection parameters must be supplied as a single command line argument. The parameters are divided by pipe (`|`). The parameter name and value is divided by `=` .

Example:

```
-con "driver=sqlite|database=C:\db\SQLite\Chinook.db|name=SQLiteChin|openConsole=true|folder=SQLite"
```

Name	Description	Example
name	Connection name	<code>Test connection</code>
driver	Driver name or ID	<code>driver=sqlite</code> , <code>driver=mysql</code> , etc
url	Connection URL. Optional (JDBC URL may be constructed by a driver from other parameters)	<code>url=jdbc:sqlite:C:\db\SQLite\Chinook.db</code>
host	Database host name (optional)	<code>host=localhost</code>
port	Database port number (optional)	<code>port=1534</code>
server	Database server name (optional)	<code>server=myserver</code>
database	Database name or path (optional)	<code>database=db-name</code>
user	User name (optional)	<code>user=root</code>
password	User password (optional)	<code>password=mysecret</code>
auth	Authentication model ID. See Auth models	<code>auth=postgres_pgpass</code>
authProp. propName	Custom authentication parameters (depends on the driver and auth model)	<code>authProp.oracle.net.wallet_location=C:/temp/ora-wallet</code>
	Does not ask user for a	

savePassword	password on connection	<code>savePassword=true</code>
showSystemObjects	Shows/Hides system schemas, tables ,etc	<code>showSystemObjects=true</code>
showUtilityObjects	Shows/Hides utility schemas, tables ,etc	<code>showUtilityObjects=true</code>
folder	Puts a new connection in a folder	<code>folder=FolderName</code>
autoCommit	Sets connection auto commit flag (default value depends on driver)	<code>autoCommit=true</code>
prop.propName	Advanced connection parameters (depend on driver)	<code>prop.connectTimeout=30</code>
id	Connection id	<code>oracle_thin-16a88e815bd-70598e648cedd28c</code> (useful in conjunction with <code>create=false</code>)
connect	Connects to this database	<code>connect=false</code>
openConsole	Opens the SQL console for this database (sets <code>connect</code> to true)	<code>openConsole=true</code>
create	Creates new connection	<code>create=false</code> (true by default). If it is set as false, then an existing connection configuration will be used. The name or id parameter must be specified.
save	Saves new connection	When <code>create=true</code> , then <code>save=false</code> (default) makes new connection temporary, <code>save=true</code> means that new connection will be saved and accessible between DBeaver launches.

Declare external variables in a file

See the [main article](#)

Session manager

Table of contents

[Overview](#)

[Supported databases](#)

[How to use](#)

Overview

The Session Manager in DBeaver provides you with control and visibility into your sessions, allowing you to terminate, disconnect, view and search for sessions, as well as perform several other actions.

Supported databases

- Altibase
- [AlloyDB](#)
- [Db2 for LUW](#)
- Exasol
- Fujitsu Enterprise Postgres
- [Greenplum](#)
- MariaDB
- [MongoDB](#)
- [MySQL](#)
- [MySQL on Google Cloud](#)
- [Netezza](#)

- [Oracle](#)
- [PostgreSQL](#)
- [PostgreSQL on Google Cloud](#)
- [Redshift](#)
- [Microsoft SQL Server](#)
- [Microsoft SQL Server on Google Cloud](#)

How to use

To open the Session Manager:









- 1) In the [Database Navigator](#), navigate to your desired database.
- 2) Go to the **Administer** section.
- 3) Double-click on **Session Manager** or right-click and select **View Session Manager**.

The screenshot shows the DBeaver Session Manager window for a PostgreSQL database. The main table lists several sessions, with the session having PID 308 selected. Below the table, the 'Session Details' panel provides information about the selected session.

PID	User	Database	App Name	State	Query Start	Brief Query
5,354	CB_Developers	postgres	DBeaver Ultimate	idle	2023-08-02 14:35:13.488031	SHOW TRANSACTION ISOLATION LEVEL
5,531	CB_Developers	postgres	DBeaver Ultimate	active	2023-08-02 14:38:27.387666	SELECT sa.* FROM pg_catalog.pg_stat_activity sa
5,366	CB_Developers	postgres	DBeaver Ultimate	idle	2023-08-02 14:35:06.865453	SELECT t.oid,t*,c.relkind,format_type(nullif(t.ty ...
308	azuresu	azure_sys		idle	2023-08-02 14:21:13.855193	SELECT query_stats.pgms_stats_insert_data(0, '60 m ...
309	azuresu	azure_sys		idle		
310	azuresu	postgres		idle	2023-05-26 19:12:46.093725	update cron.job_run_details set status = 'failed', ...
311	azuresu	azure_sys				
312	azuresu					
5,520	azuresu	postgres		idle	2023-08-02 14:38:23.441263	SELECT pg_catalog.pg_is_in_recovery() as is_in_Rec ...
5,517	azuresu	azure_maintenance		idle	2023-08-02 14:38:26.468771	insert into public.isnmover (id, update_time) sele ...

Name	Value
General	
PID	308
Database	azure_sys
State	idle
Brief Query	SELECT query_stats.pgms_stats_insert_data(0, '60 m ...
Client	
User	azuresu
Client Host	
Client Port	
App Name	

Once you have opened the **Session Manager**, you will have access to a variety of functionalities based on your database type. To access these functionalities, you must right-click on the desired session, or select the session and use the buttons in the bottom toolbar of the Editor. Here is a rundown of the buttons and their functionalities:

Button	Image	Description
Kill Session		Terminates the selected session forcefully.
Disconnect Session		Disconnects the selected session from the server.
Show Active/All Connections		Toggles between displaying only the active sessions and all sessions in your database.
Show Background		Reveals background processes that are running in the session.
Show Inactive		Shows the sessions that are currently inactive in your database.
Refresh		Updates the session list to reflect the current state of the database.
Configure Auto-Refresh		Allows you to set up automatic refreshing of the session list.
SQL Script		Calls a SQL script that will display sessions in a table format.

Error Log

Table of contents

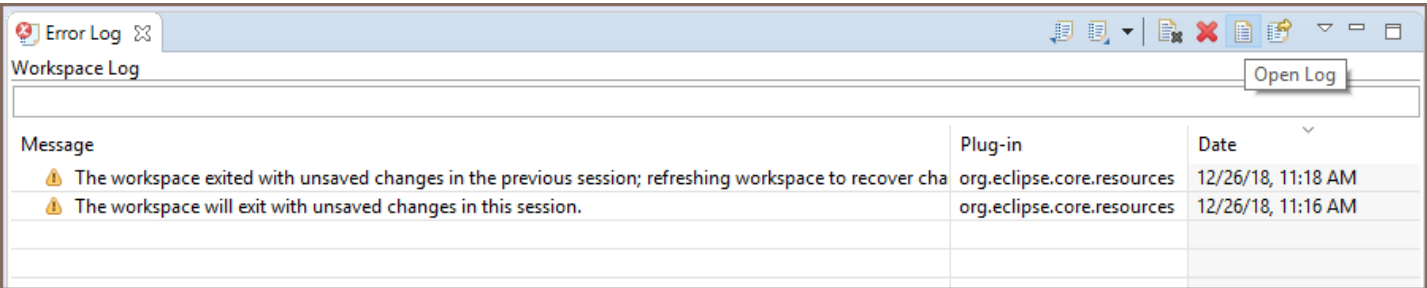
- [Error Log view](#)
- [Log files](#)
- [Java fatal logs](#)
- [Log files for Team Edition](#)
- [Gathering log files for error reporting](#)

Error Log view

There is an Error Log view (main menu Window->Show View->Error Log) which contains all errors which occur during the DBeaver runtime.

You can double click on the warning/error in the log viewer and see the error stacktrace. Please attach it to the bug report.

Also, you can open the full log (all error messages) if you need:



Log files

DBeaver writes different log files. Most of them are Eclipse logs.

Log files usually reside in the [workspace](#)/workspace6/.metadata .

- In Windows open Explorer and paste path `%APPDATA%\DBeaverData\workspace6\.metadata` .
- In Linux just type `cd $XDG_DATA_HOME/DBeaverData/workspace6/.metadata`
-

In MacOS open path `~/Library/DBeaverData/workspace6/.metadata` in Finder.

- To view hidden folders press `Cmd+Shift+.` in the folder view.

Two standard log files:

- `workspace/workspace6/.metadata/.log` - all warnings and errors which happen during normal work
- `workspace/workspace6/.metadata/dbeaver-debug.log` - the same as `.log` plus debug information

In special cases log files can be written in other directories. A special case is an emergency situation when DBeaver cannot start and there is no workspace. Two typical places to find emergency logs:

- `<install-path>/configuration`
- `${HOME}/.eclipse/org.jkiss.dbeaver.product_<dbeaver-version>`

Java fatal logs

On the rare occasion that the DBeaver process dies, it does not leave any valuable logs. This is caused by a Java VM crash.

JVM creates a fatal log file for each crash (log file `hs_err_PID.log`). This log usually resides in the same directory where the DBeaver launcher is (e.g. `dbeaver.exe`).

But in some cases it is a write-protected directory and the log file will be created in other folder.

Instructions on how to find the Java fatal log file: <https://docs.oracle.com/javase/9/troubleshoot/fatal-error-log.htm>

Log files for Team Edition

Log files usually reside in the `workspace/team-workspace/.metadata`.

- In Windows open Explorer and paste path `%APPDATA%\DBeaverData\team-workspace\.metadata`.
- In Linux just type `cd $XDG_DATA_HOME/DBeaverData/team-workspace/.metadata`
- In MacOS open path `~/Library/DBeaverData/team-workspace/.metadata` in Finder.
- To view hidden folders press `Cmd+Shift+.` in the folder view.

Gathering log files for error reporting

When encountering an issue with DBeaver, providing log files is essential for diagnosis. To gather the logs:

1. Navigate to **Help -> Collect Diagnostic info**.
2. Select the folder to save the collected log information.

Important: Diagnostic info may contain sensitive information. You can disable sensitive data in the logs by following the guidelines in [our guide](#). Be cautious when sharing logs with third parties.

Background Tasks

Table of contents

[Overview](#)

[Opening the Background Tasks View](#)

[Utilizing the Background Tasks View](#)

[Task management](#)

[Canceling a Task](#)

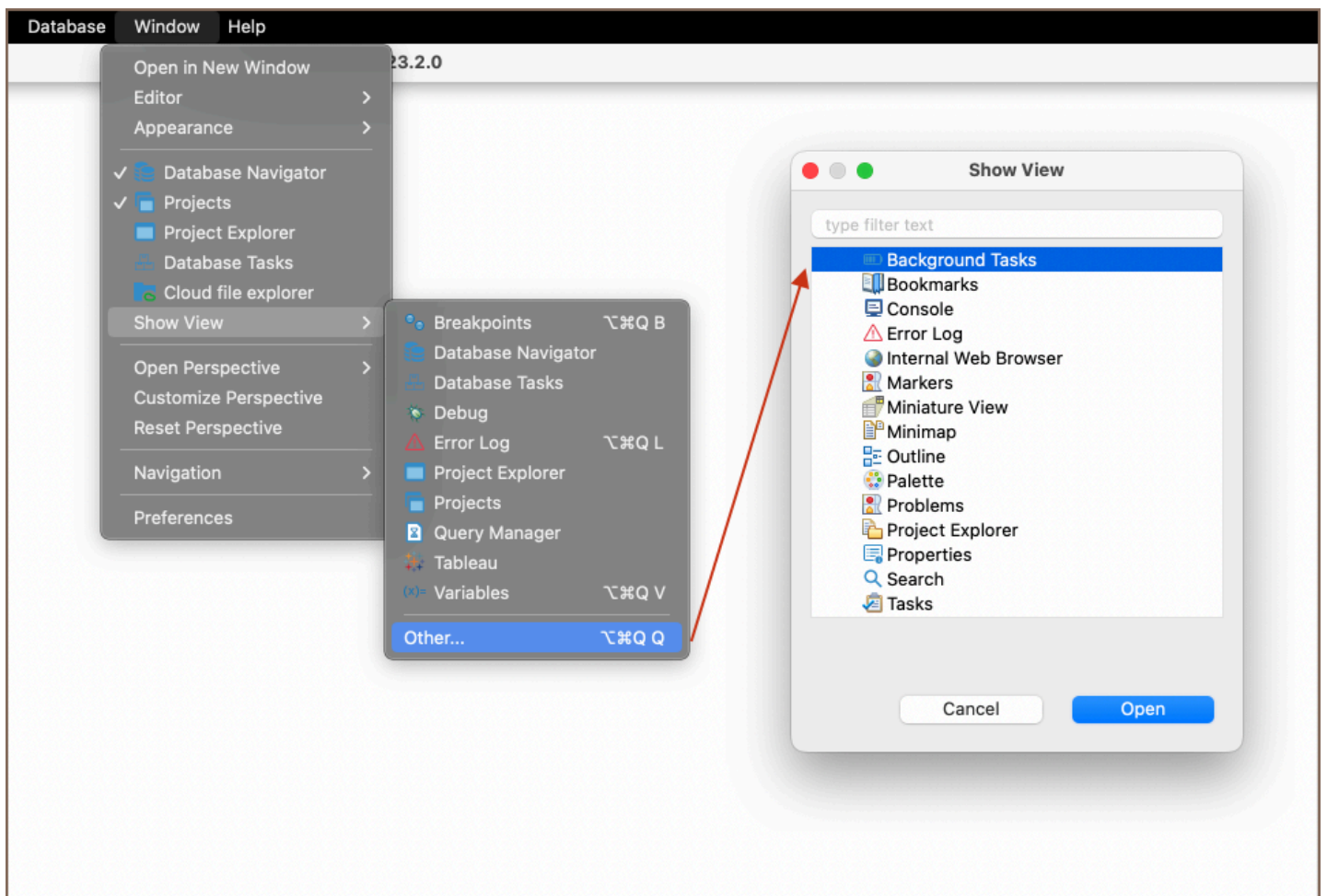
[Removing all finished operations](#)


Overview

Background tasks in DBeaver are operations that run behind the scenes, allowing you to continue working without disruption. These tasks can involve a range of operations, such as [data transfer](#) and [SQL script execution](#), amongst others.

Opening the Background Tasks View

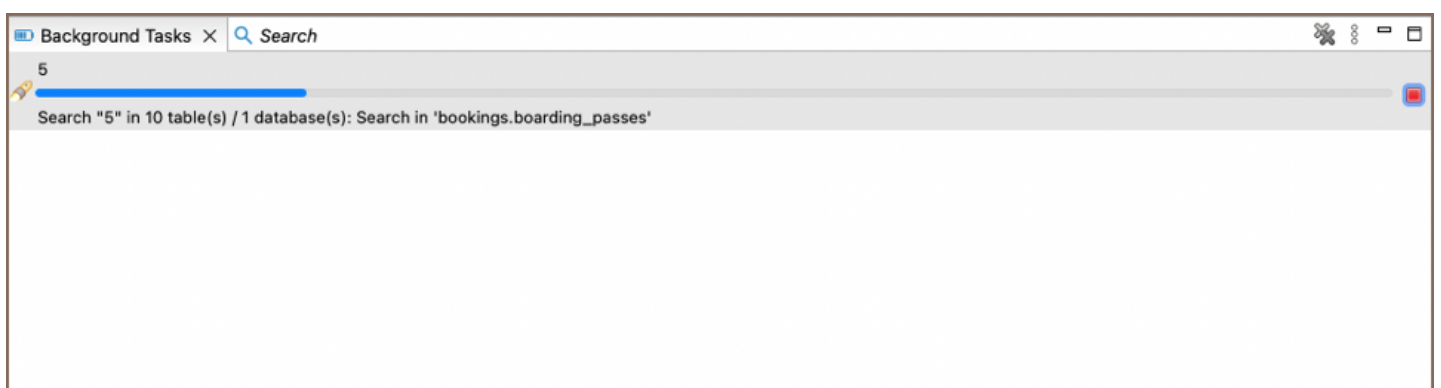
You can open the **Background Tasks** view from the main menu by clicking **Window** -> **Show View** -> **Other**. In the **Show View** window, click **Background Tasks**, and then click **Open**:



Tip: You can also open the Background Tasks view from other views or editors. For example, in the [Search](#) view, when a search takes time, a button  appears in the bottom-right corner. Click it to open the Background Tasks view.


Utilizing the Background Tasks View

The **Background Tasks** view comprehensively displays the progress of background tasks like **Search** and **SQL query execution**. It becomes beneficial when you want to monitor the progress of time-consuming operations. The view displays detailed data when background tasks consume a noticeable amount of time.




Task management

Canceling a Task

If you need to cancel a task in progress, it is possible to do so directly from the Background Tasks view. Click the **Cancel Operation** button  to stop the ongoing task immediately, freeing up resources for other operations.

Removing all finished operations

In the Background Tasks view, DBeaver also provides an option to remove all finished operations from the list to declutter the view and focus on ongoing tasks. To use this feature, click the **Remove all finished operations** button . It helps in maintaining a neat and organized task view, enabling you to manage your current tasks without distractions from completed operations.

Troubleshooting system issues

Table of contents

[Linux](#)

[GTK](#)

[Fixing screen flickering](#)

[Parts of DBeaver are white and dark on dark and white theme](#)

[No more handles because there is no underlying browser available](#)

[Theme parsing error](#)

[Gtk box gadget distribute issue](#)

Linux

GTK

DBeaver is an Eclipse RCP application. Therefore, it may have issues with GTK. Here are known workarounds regarding various issues which may arise.

Fixing screen flickering

- Add `export GTK_IM_MODULE=ibus` to `~/.profile`.

Parts of DBeaver are white and dark on dark and white theme

To fix the issue, we recommended having a graphical theme which is similar to system one in terms of the color palette.

No more handles because there is no underlying browser available

You need to install `libwebkit2gtk` library to your system, this will solve the issue and allow you to use DBeaver browser functionality

Theme parsing error

`GTK-WARNING xxx:Theme parsing error` issue requires you to change the GTK-program-style `system settings-appearance-program style` (the setting location may vary for different systems).

Gtk box gadget distribute issue

To fix `gtk_box_gadget_distribute: assertion size 'size >= 0 0' failed in GtkScrollbar` issue, you need to remove the overlay scrollbars in `~/.config/gtk-3.0/settings.ini`.

```
[Settings]
gtk-overlay-scrolling = false
```

Making a thread dump

Table of contents

[Overview](#)

[Finding jstack](#)

[For macOS and Linux](#)

[For Windows](#)

[Generating a thread dump on Windows](#)

[Generating a thread dump on Mac and Linux](#)

[Generating a thread dump without Java](#)

[For Linux users](#)

[For Windows users](#)

Overview

Sometimes (due to some bug) DBeaver UI hangs, freezes, or works incorrectly. It is usually impossible to find the reason for such an issue without a thread dump. A thread dump is information about the internal execution state of the Java program.

DBeaver incorporates the `jstack` tool for generating thread dumps. To use this pre-installed tool, it is essential to locate its directory in the DBeaver installation and execute it from there.

Important: The `jstack` tool is not included in the tar.gz without Java on the [DBeaver download page](#), and in Downstream packages.

Finding jstack

The location of `jstack` varies depending on your operating system:

For macOS and Linux



Find the directory where DBeaver is installed. On macOS, proceed to the `Contents/Eclipse` subdirectory.

Then, in both macOS and Linux systems, locate the `jre/bin/jstack` directory. This is the typical location for `jstack` in DBeaver installations.

For Windows

- Find the directory where DBeaver is installed.
- Within this directory, go to `jre/bin/jstack.exe`. This is the standard path to find `jstack` in DBeaver installations on Windows systems.

Note: If you did not install Java during the DBeaver installation (by unchecking the `Install Java` checkbox), `jstack.exe` will not be present. In this case, you need to install Java separately and locate `jstack.exe` within the Java installation directory.

Generating a thread dump on Windows

To create a thread dump in DBeaver on Windows, follow these steps:

1. Find DBeaver's Process ID:

- Open Task Manager by pressing `CTRL+SHIFT+ESC`.
- In Windows 8 and later, switch to the "Details" tab.
- Locate `DBeaver` (or `DBeaver[your version]`) in the process list and note its process ID (PID).

> **Tip:** For example, if you are using the DBeaver Lite version, you should look for `DBeaverLite`.

2. Open Command Prompt:

- Press `Win+R`, type `cmd`, and press `ENTER` to open the command prompt.

3. Run the Thread Dump command:

- Navigate to the `jre/bin` directory within the DBeaver installation path.

- Run the following command, replacing `PID` with the actual process ID of DBeaver: `cmd jstack.exe PID > "%HOMEPATH%\dbeaver-thread-dump.txt"`
- This command executes `jstack.exe`, generates a thread dump for the specified PID, and saves the output to `dbeaver-thread-dump.txt` in your home directory (e.g., `C:\Users\Username`). This file contains the thread dump information that can be attached to a GitHub issue for further analysis.

Generating a thread dump on Mac and Linux

To create a thread dump in DBeaver on macOS and Linux systems, you can use the included `jstack` tool via the terminal. The following script will generate a thread dump and save it as `dbeaver-thread-dump.txt` in your home directory:

```
/Applications/YOUR-VERSION-OF-DBEAVER.app/Contents/Eclipse/jre/Contents/Home/bin/jstack $(pgrep dbeaver) >
```

Tip: Replace `YOUR-VERSION-OF-DBEAVER.app` with the version of DBeaver you have installed. For example, if you have DBeaver Lite version, the path would be

```
/Applications/DBeaverLite.app/Contents/Eclipse/jre/Contents/Home/bin/jstack .
```

Generating a thread dump without Java

For Linux users

If you are running DBeaver on Linux using the tar.gz version without Java included, you will need to install the Java Development Kit (JDK) to generate a thread dump. Follow these steps:

1. Check for Existing JDK Installation:

- Open your terminal.
- Type `which jstack` and press `Enter`.
- If the JDK is installed, a version number will appear.

2. Install JDK:

- If JDK is not installed, you need to install it.
- Use your distribution's package manager to install JDK. For example, on Ubuntu, you can use

```
sudo apt install default-jdk .
```

After successfully installing JDK, proceed with the thread dump script:

```
jstack $(pgrep dbeaver) > ~/dbeaver-thread-dump.txt
```

After running this script, you will find the `dbeaver-thread-dump.txt` file in your home directory, containing the necessary thread information for diagnosing issues.

For Windows users

If you are using DBeaver on Windows without Java, consider installing [Adoptium OpenJDK](#).

Alternatively, you can reinstall DBeaver and ensure to check the `Install Java` checkbox during installation.

After installing Java through either method, you can follow the instructions in the section [Generating a Thread Dump on Windows](#) to successfully generate a thread dump.

Posting issues

A few tips.

- Check existing issues for your issue (including closed ones). Duplicating an issue is slower for both parties so search through the open and closed issues to see if what you are running into has already been addressed.
- Be clear about what your problem is: what was the expected outcome, what happened instead? Detail how someone else can recreate the problem.
- If you posting a bug report check "Error Log" view. If there are any errors related to your bug then post a complete stacktrace. Sometimes there are no errors in Error Log - if so, try to find them in [log files](#).
- If your issue is related to database data or metadata management - check the Query Manager view. It contains information about all queries DBeaver executes (explicitly or implicitly). To see more detailed information you can configure Query Manager in Preferences.

Time	Type	Text	Duration	Rows	Result	Data Source	Connection
17:43:39	SQL / META	SELECT c.*FROM SYS.ALL_TAB_COLS cWHERE c.OWNER='APEX_0402...	221 ms	32	Success	Oracle - orcl	Metadata
17:43:38	SQL / META	SELECT /*+RULE*/ t.c.TABLE_NAME, c.CONSTRAINT_NAME,c.CONSTR...	809 ms	18	Success	Oracle - orcl	Metadata
17:43:38	SQL / META	SELECT /*+RULE*/ t.OWNER,t.TABLE_NAME as TABLE_NAME,'TABLE' a...	198 ms	661	Success	Oracle - orcl	Metadata
17:43:36	SQL / META	SELECT /*+RULE*/ t.c.TABLE_NAME, c.CONSTRAINT_NAME,c.CONSTR...	2417 ms	1	Success	Oracle - orcl	Metadata
17:43:36	SQL / USER	SELECT x.*,x.ROWID FROM FLOWS_FILES.WWV_FLOW_FILE_OBJECTS\$ x	20 ms	0	Success	Oracle - orcl	Main
17:43:36	SQL / META	SELECT c.*FROM SYS.ALL_TAB_COLS cWHERE c.OWNER='FLOWS_FL...	257 ms	22	Success	Oracle - orcl	Metadata
17:43:33	SQL / META	SELECT COMMENTS FROM ALL_TAB_COMMENTS WHERE OWNER='FL...	67 ms	1	Success	Oracle - orcl	Metadata
17:43:33	SQL / META	SELECT * FROM SYS.DBA_TABLESPACES ORDER BY TABLESPACE_NAME	141 ms	5	Success	Oracle - orcl	Metadata
17:43:33	SQL / META	SELECT COUNT(*) FROM FLOWS_FILES.WWV_FLOW_FILE_OBJECTS\$	47 ms	1	Success	Oracle - orcl	Metadata
17:43:31	SQL / META	SELECT /*+RULE*/ t.OWNER,t.TABLE_NAME as TABLE_NAME,'TABLE' a...	51 ms	1	Success	Oracle - orcl	Metadata

- Depending on the nature of your bug report provide information about:
 - Operating system
 - Window manager (for Linux)
 - Database (name and version)
 - Database driver (name and version)
- Do not write issue type in the issue title (like Feature Request;, Bug; etc). We will review your issue and assign a corresponding label.

How to set a variable if dbeaver.ini is read only

Table of contents

[config path](#)

[For Linux](#)

[For Windows](#)

[config example](#)

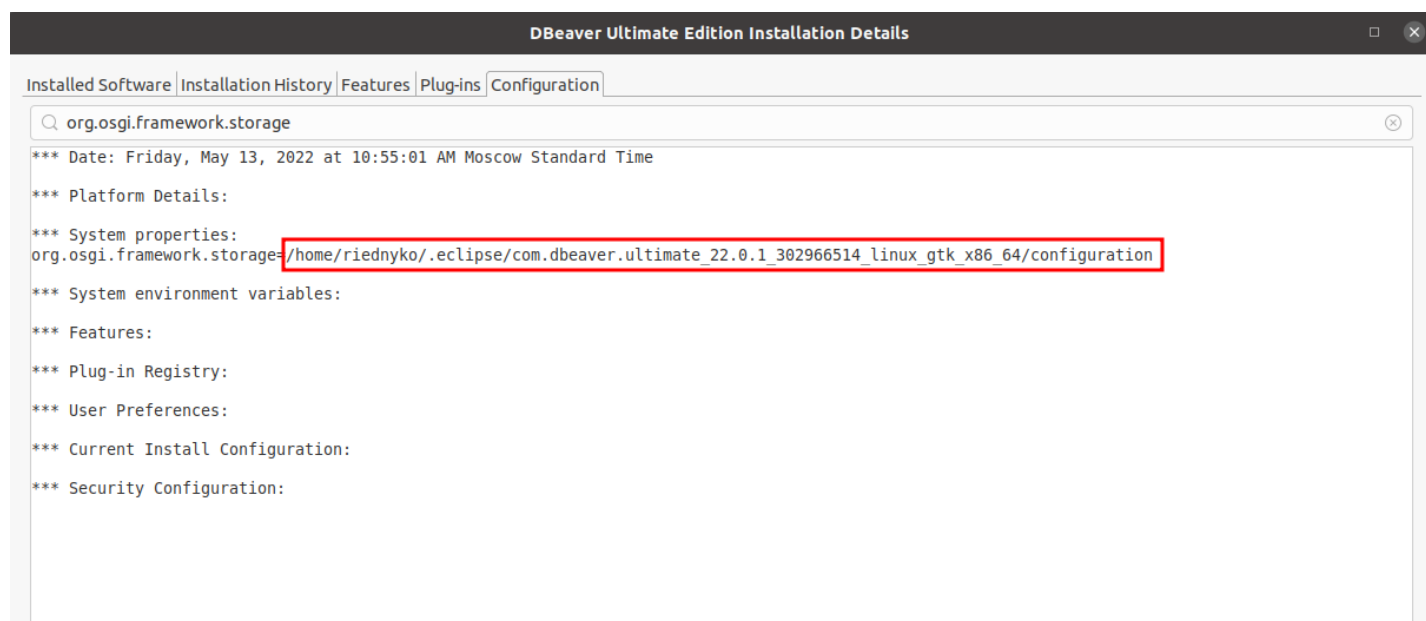
If you need to set some variable and `dbeaver.ini` have read-only permissions (this can happen, for example, in Flatpak or Snap), you can use the `config.ini`.

config path

For Linux

You can find correct path to your configuration directory with `config.ini` when open DBeaver

`Help -> Installation Information -> Configuration` then *type filter text* `org.osgi.framework.storage`



If `config.ini` does not exist, you can create it in the configuration directory `nano config.ini`

For Windows

You can find correct path to your configuration directory with **config.ini** when open DBeaver

Help -> Installation Information -> Configuration then *type filter text* `org.osgi.framework.storage`

Example output:

```
org.osgi.framework.storage=file:/C:/Users/user/.eclipse/org.jkiss.dbeaver.product_22.0.5_1535670467_win32_
```

If **config.ini** does not exist, you can create it in the configuration directory.

config example

For example, you can set system property with your custom path to point to the keystore you created.

```
javax.net.ssl.trustStore=/path/to/your/cert
```

Or, for example, you can change the language.

```
osgi.nl=fr
```

License administration overview

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

Table of contents

[Manual license import](#)

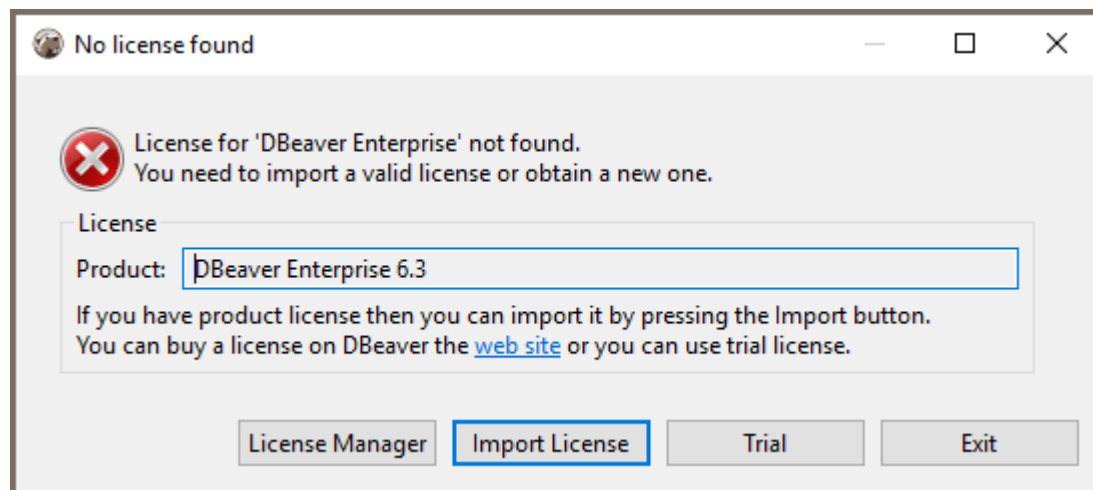
[License management automation](#)

[Put the license file to the predefined locations](#)

[Passing license file through command line](#)

Manual license import

Commercial versions of DBeaver ask the user to import the license file if they cannot find it locally. It is the most simple and convenient way to import the product license for individual users.



License management automation

There are several ways to automate the license management process. It makes sense for a multi-user environment.

Put the license file to the predefined locations

1. Install DBeaver

2

2. Save license text to a file. It should be named as `.dbeaver-%PRODUCT_PREFIX%-license.dat` or `.dbeaver-%PRODUCT_PREFIX%-license.txt`.

3. Add the license file to one of the following locations:

- Windows
 - `%HOMEPATH%`
 - `%APPDATA%\DBeaverData\workspace6\.metadata\`
 - `%AppData%/.jkiss-lm`
- MacOS X
 - `~/` (user home)
 - `~/Library/DBeaverData/workspace6/.metadata/`
 - `~/Library/.jkiss-lm`
- Linux
 - `~/` (user home)
 - `$XDG_DATA_HOME/DBeaverData/workspace6/.metadata/`
 - `$XDG_DATA_HOME/.jkiss-lm`

4. Launch DBEaver from Start Menu

Product prefixes:

Product name	Prefix
DBeaver Enterprise	ee
DBeaver Lite	le
DBeaver Ultimate	ue

Passing license file through command line

You can add the command line parameter `license <license-path>` to the DBEaver EE shortcut. Also, you can add this parameter to the `dbeaver.ini` config file.

Tip: Detailed instructions on finding `dbeaver.ini` are available in [our article](#).

[Command line](#) reference.

How to import license

Table of contents

[Import from email](#)

[Import from the personal account](#)

[Insert the License key into the License Manager](#)

[Import of Monthly Subscription license](#)

[License extension](#)

[License Manager](#)

To start using commercial versions of DBeaver you can

- request a [Trial license](#) for 2 weeks;
- request an [Academic license](#) if you are a student or a teacher;
- [buy](#) a monthly or annual subscription.

After purchasing the DBeaver license or getting the Trial/Academic license, you will receive a License text by email. It will also be available in your personal account on our [site](#). This License text will contain your License ID e. g. DB-841MRZHY-ZH54, the start date and license owner's name and company name. It is very important to import your License correctly.

Import from email

You can just copy-paste the License Key to import the license into the License Manager. Please note that you need to copy-paste the full license text (not just the license ID). The license text starts with "-" and ends with "==" characters.

Hello,

Thank you for your interest to DBeaver.

We are glad to see you among DBeaver users. Please, find your license below:

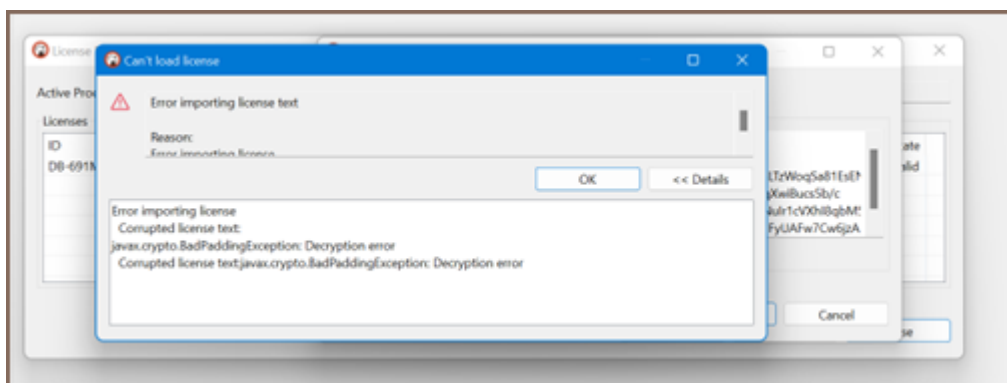
```
-- DBeaver EE LICENSE - DB-841MRZHJ-ZH53
-- Starts at Tue Oct 11 10:14:33 UTC 2022 to Louis Anderson //
PK2uJ9eCXU4Zmgd1wvMufQyC2Jk+cSTtePOXewaUoS2/jYvuaX1opFJ0G/H81sLTzWogSa81EsEN
BG3Z6TJ0VFPko5bvXkm2r0hRXjRusa18gM47snPK/NaH/7fg/CEzHhy3+NrsttRqXwiBuc55b/c
lRgIZm3Z4B5qg8G11V8K2gG99p8K0e690gTB6vWY1hECRfs3vdrz4x5QHnQrNuIr1cVXhI8qbM5
13L9/7e3QgBMQya+XF9yVeGiVRQgDQDP8GNwY1WMoHBzpGKuhm1twBj+jF7FyUAFw7Cw6jzAk0es
gb3Fg2REPkLwvIq2yxangUpnNGDUE1bH7VXoSw==
```

Enjoy your database research with DBeaver and good luck!

Regards,

DBeaver team

Sometimes an email client can corrupt the formatting of the License Key that can cause an error.



Therefore, you need to import your License Key from your personal account on our site <https://dbeaver.com/>.

Import from the personal account

Firstly, you need to [Sign in](#).

Secondly, you should open the **Licenses** tab, where you can find all your licenses.

Ben Johnson

ORDERS

LICENSES

ACCOUNT












TICKETS

Licenses

Assignment

Search licenses...

Rows Per Page: 10 | 20 | 50 | 100

<input type="checkbox"/>	License ID 	Users 	Product 	Type 	End Support Date 	
			All 	All 	All 	
<input type="checkbox"/>	DB-9Q1MSZGF-ZDAM	5	DBeaver Enterprise Edition	Yearly subscription	Dec 6, 2024	
<input type="checkbox"/>	DB-9Q1MSZGC-ZDAL	1	DBeaver Enterprise Edition	Trial	Dec 21, 2023	
<input type="checkbox"/>	DB-9Q1MSZGB-ZDAK	1	DBeaver Enterprise Edition	Monthly subscription	Jan 7, 2024	

Bulk actions:

EXTEND

GET LICENSE KEYS

EXPORT INFO

Export all licenses info as csv

To open License details and copy the license key text click the license ID link. Here you can find your license status, type, maintenance period, and end support date.

Ben Johnson

[ORDERS](#)[LICENSES](#)[ACCOUNT](#)[TICKETS](#)

License ID:	DB-9Q1MSZGF-ZDAM
License Type:	Yearly subscription
Maintenance Period:	365 days
Maximum users:	5
Issue Time:	Dec 7, 2023
Expiration Time:	Dec 6, 2024
License Status:	Valid
License customer:	Ben Johnson, Test

License Keys:

DBeaver EE

-- DBeaver EE LICENSE - DB-9Q1MSZGF-ZDAM
-- Starts at 2023-12-07 09:35:25.0 to Test / Ben Johnson //
DB-9Q1MSZGF-ZDAM-2023-12-07 09:35:25.0 to Test / Ben Johnson //
DB-9Q1MSZGF-ZDAM-2023-12-07 09:35:25.0 to Test / Ben Johnson //
DB-9Q1MSZGF-ZDAM-2023-12-07 09:35:25.0 to Test / Ben Johnson //
DB-9Q1MSZGF-ZDAM-2023-12-07 09:35:25.0 to Test / Ben Johnson //

DOWNLOAD LICENSE

COPY TO CLIPBOARD

At the bottom of the page you can find the License Key required to start using DBeaver. There are two options how to copy your License Key from the personal account:

1) Press the **COPY TO CLIPBOARD** button, then press OK. The license text will be copied to the clipboard.



2) Press the **DOWNLOAD LICENSE** button, then press OK.

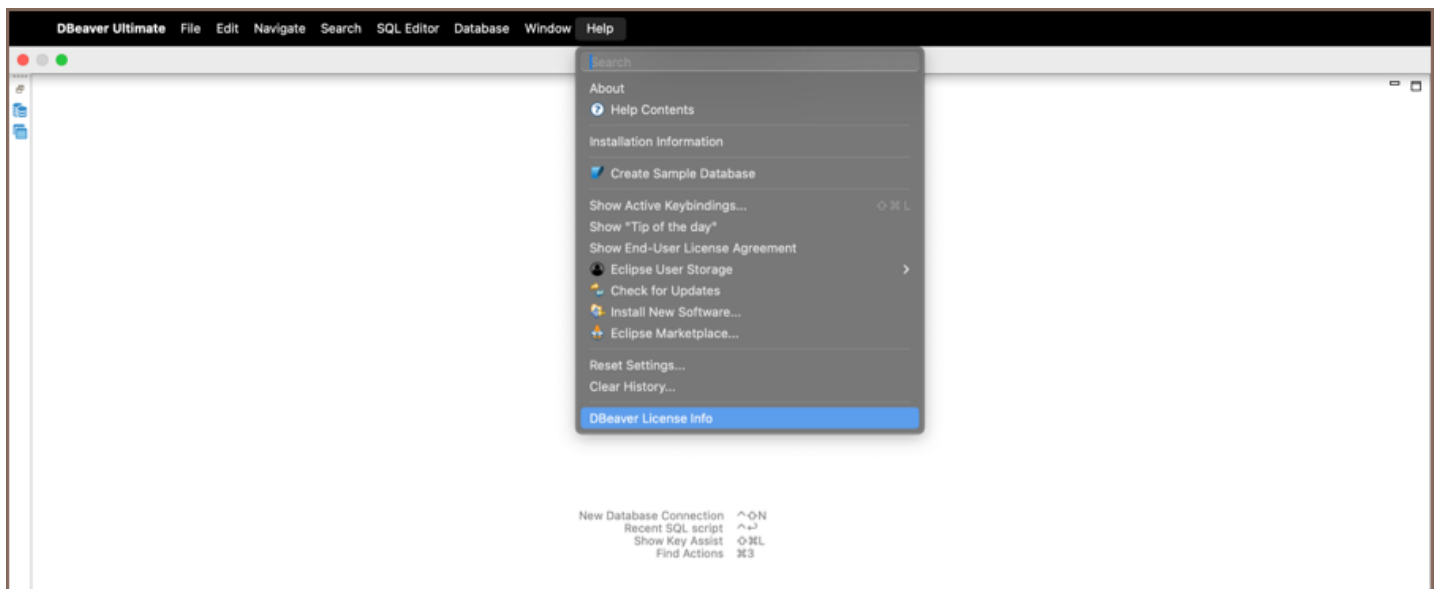
.txt file with your License Key will be downloaded to your download folder. The file name is License ID, e. g. DB-841MRZHY-ZH54.



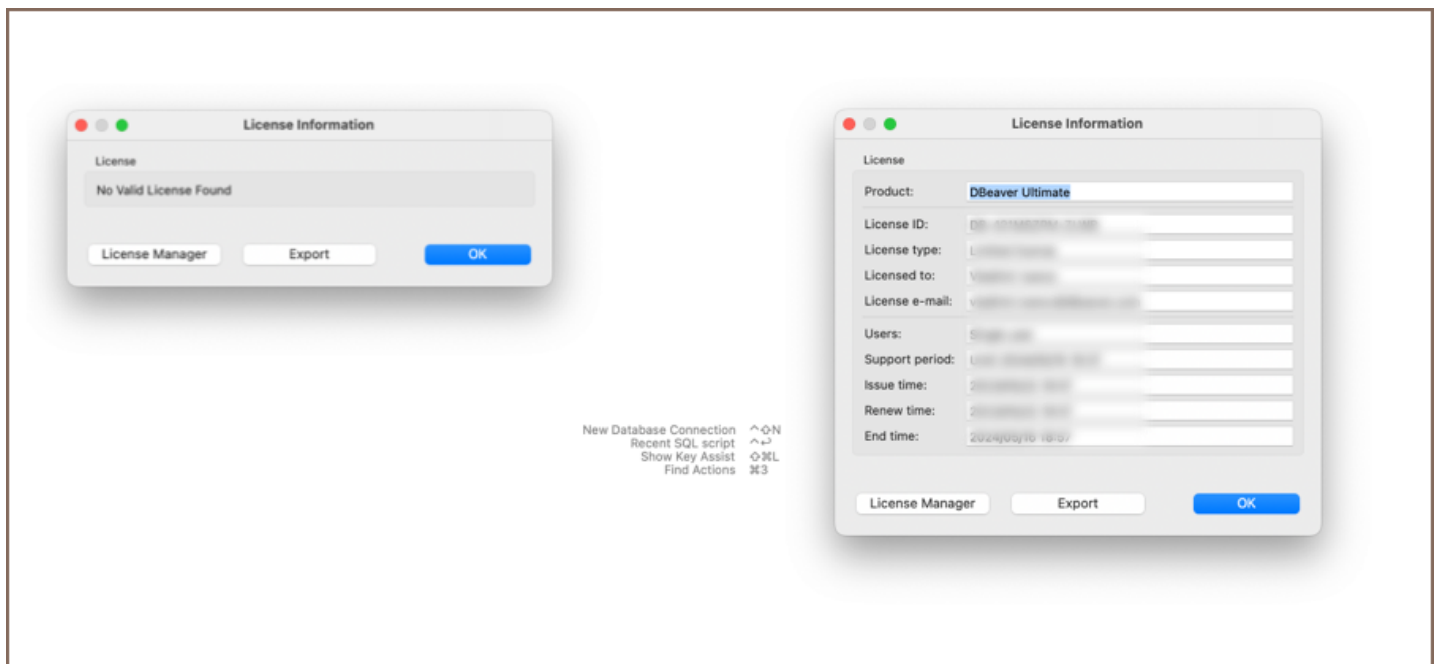
Then you need to insert the copied License Key to License Manager in DBBeaver.

Insert the License key into the License Manager

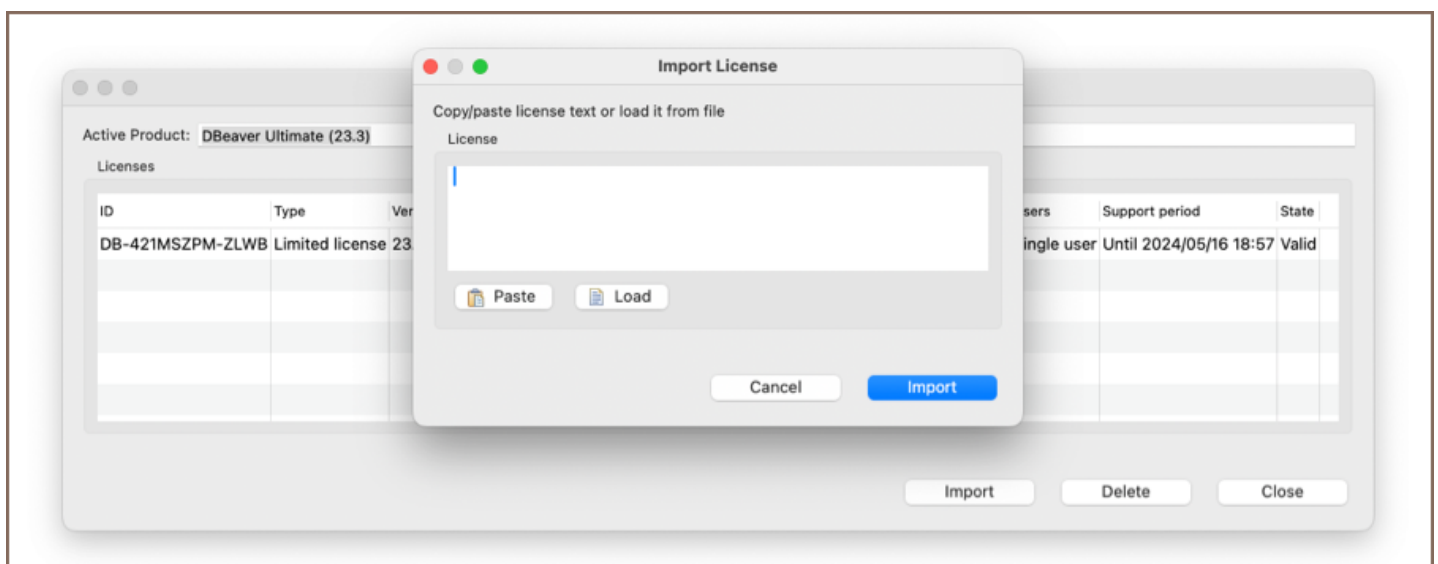
To start using commercial versions of DBBeaver with your License Key you need to open License Manager in DBBeaver: Help -> DBeaver License Info



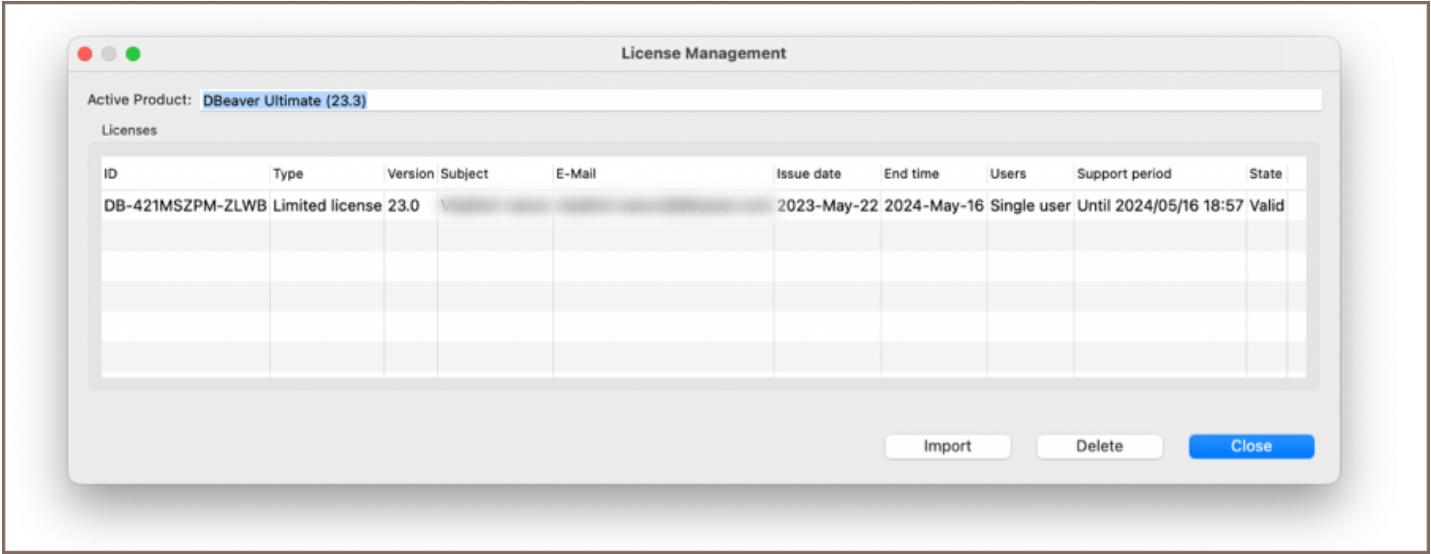
The License information window can look different depending on whether you already have a valid license or not.



Then you open the License Manager and press the Import button to paste your License Key.



If you copied the License Key to the clipboard, press the Paste button and then Import. If you downloaded a .txt file with the License Key, press the Load button and then select the file from Downloads. The License Key will be pasted automatically. Then press the Import button and your license will be added to the License Manager. You have successfully imported your license.



You have successfully imported your license. Now you can close the License Manager and start using DBeaver.

Import of Monthly Subscription license

A monthly subscription requires internet access on the workstation for the first activation and each prolongation.

If you do not have an active internet connection or work behind a corporate firewall while importing the Subscription license, the following error can appear:

```
Invalid subscription
Can't find the subscription information for license 'DB-841MRZHY-ZH54'.
Check your internet connection and/or firewall settings and restart the application.
```

In this case, you need to check that DBeaver has internet access, or you will need to configure your firewall.

License extension

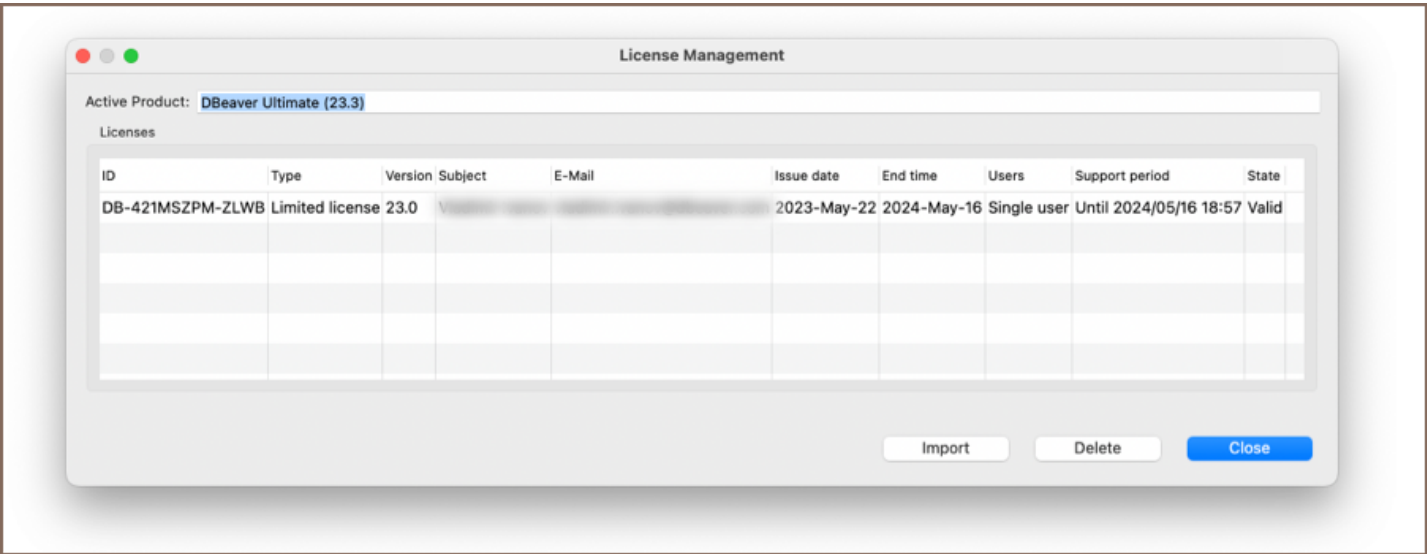
After the perpetual license expires, customers can purchase a license extension on the website.

If the customer buys the DBeaver license extension and DBeaver has internet access, the license in DBeaver will be updated automatically. Otherwise, the customer must again import the license key from the personal account.

License Manager

License Manager provides you with the following information about your licenses:

- License ID e. g. `DB-841MRZHY-ZH54` ;
- License type: Trial/Academic/Subscription/Standard;
- Version;
- License owner's name and company name;
- License owner's email;
- Issue time is the date of the initial license purchase;
- The end time is the date the license expires;
- Number of users: single user or multiuser for group licenses;
- The support period is the period you have access to the internal support system on the site and the possibility to download new product versions;
- State: valid or expired.



How to reassign license

Note: This feature is available in [Lite](#), [Enterprise](#), [Ultimate](#) and [Team](#) editions only.

After purchasing a set of DBeaver licenses, you have to assign each license to an end user.

If an employee subsequently is leaving the company or the team that is using DBeaver, the license admin may need to reassign the license to another employee.

You can reassign the license to another user in your personal account.

Firstly, you need to [Sign in](#).

Secondly, you should open the **Licenses tab**, where you can find all your licenses.

Ben Johnson

ORDERS

LICENSES

ACCOUNT

TICKETS

Licenses

Assignment

Search licenses...

Rows Per Page: 10 | 20 | 50 | 100

<input type="checkbox"/>	License ID	Users	Product	Type	End Support Date	
			All	All	All	
<input type="checkbox"/>	DB-9Q1MSZGF-ZDAM	5	DBeaver Enterprise Edition	Yearly subscription	Dec 6, 2024	⋮
<input type="checkbox"/>	DB-9Q1MSZGC-ZDAL	1	DBeaver Enterprise Edition	Trial	Dec 21, 2023	⋮
<input type="checkbox"/>	DB-9Q1MSZGB-ZDAK	1	DBeaver Enterprise Edition	Monthly subscription	Jan 7, 2024	⋮

Bulk actions:

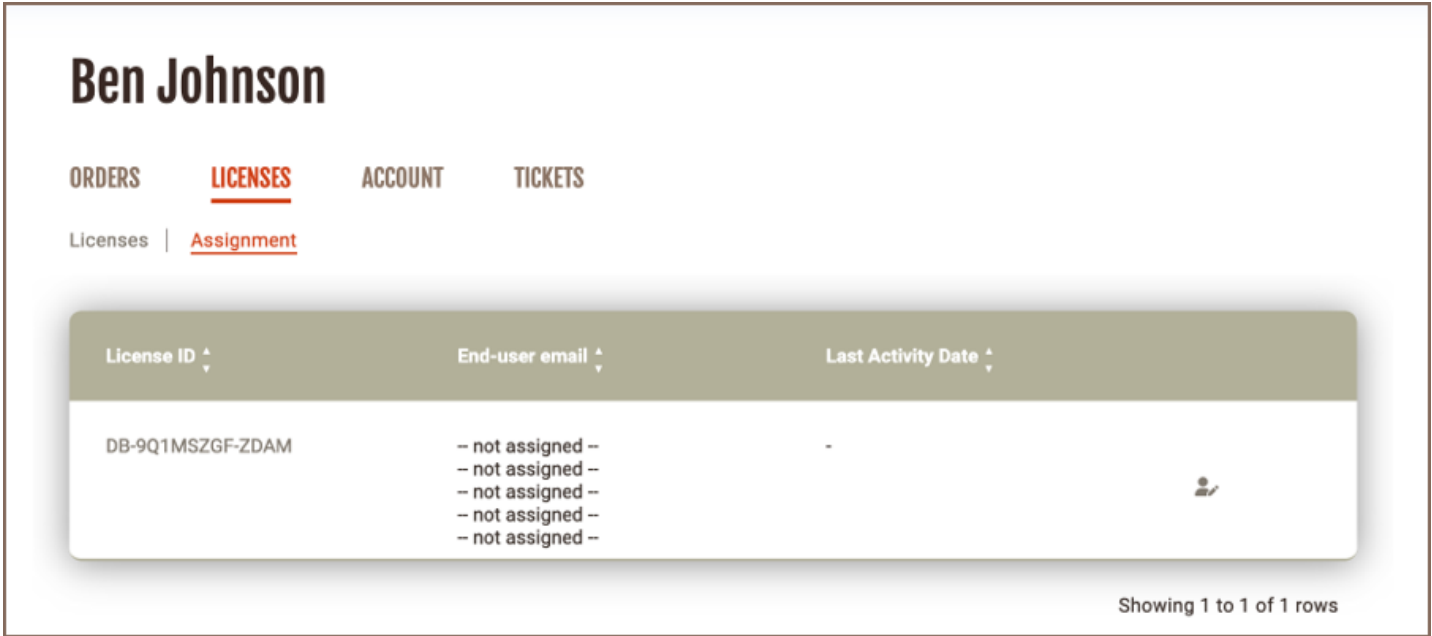
EXTEND

GET LICENSE KEYS

EXPORT INFO

Export all licenses info as csv

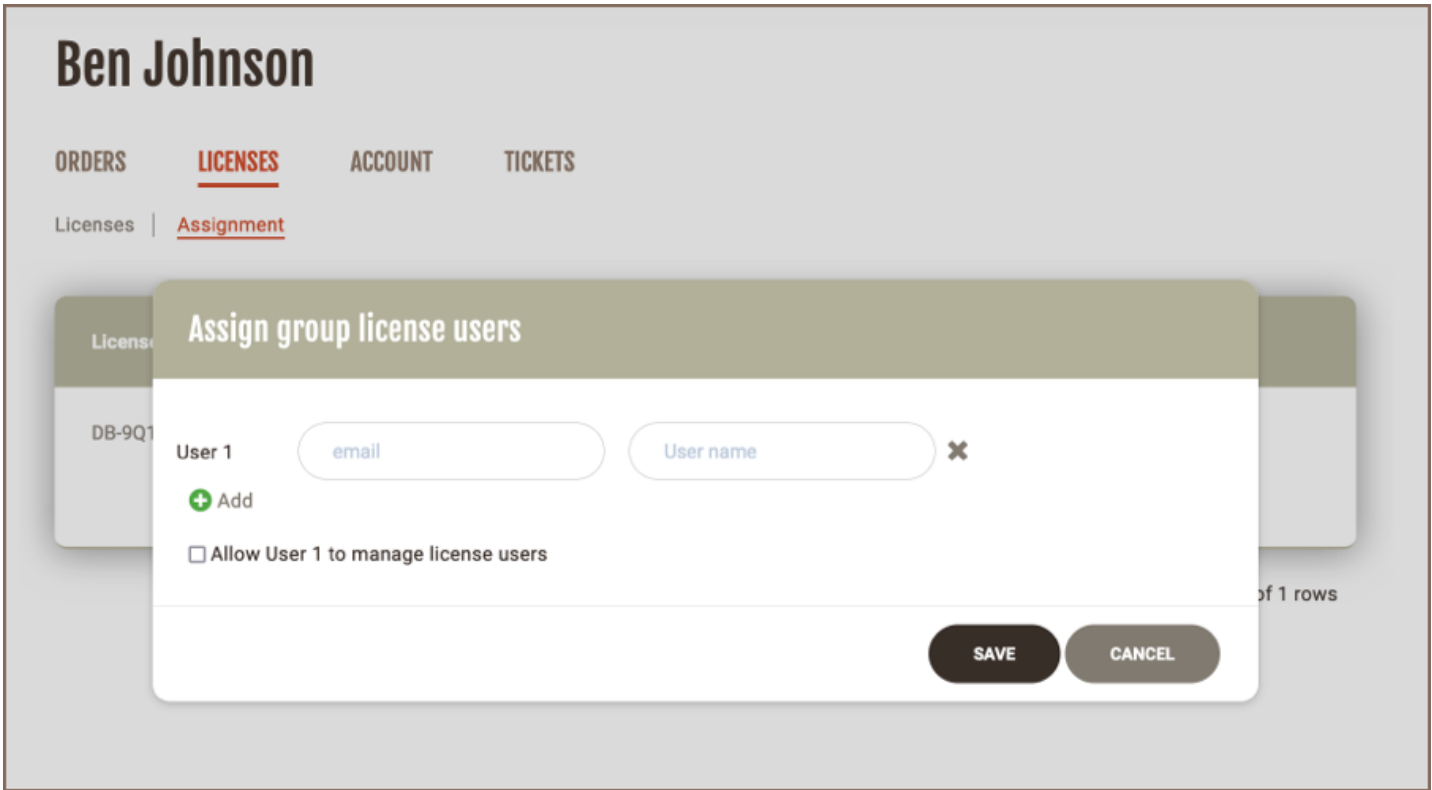
Secondly, you click the Assignment subtab. Here you can manage the license end-users and track the license usage.



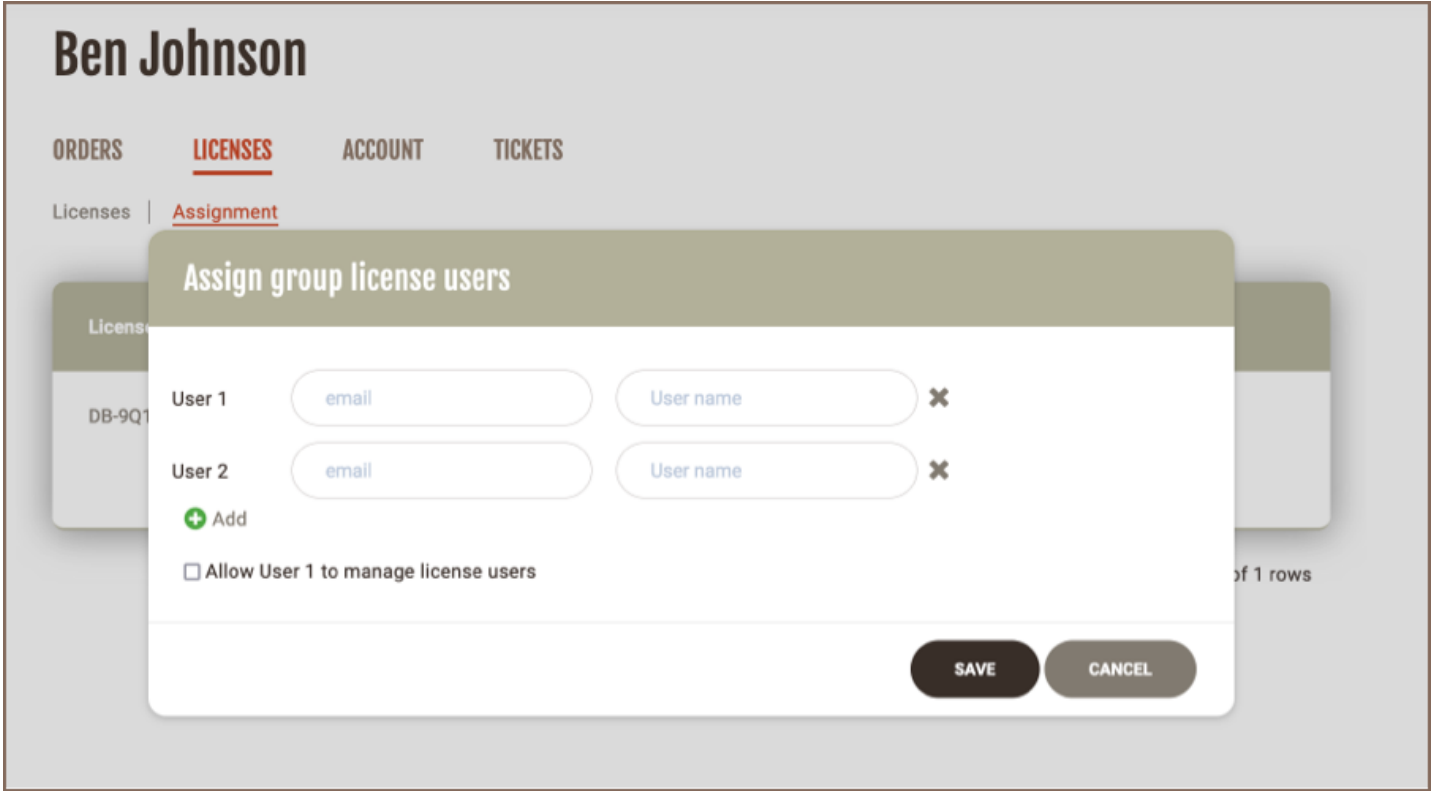
You select which license you need to reassign and press the icon in the right column.

A pop-up opens, and you can enter the new license end user's data: email; first and last name; and company. After filling in the form, you need to press the **SAVE** button and the license owner will be changed. The license key that contains the license end user's name and company name will be changed too.

Note: When the license manager reassigns a license to another user, the license is completely recreated. The new user receives a new license, and the old license becomes invalid and is no longer visible in the customer's account.



If you have a group or unlimited license, it is also possible to assign users. In addition, if you click the checkbox 'Allow User 1 to manage license users', this user will become a license manager and will be able to assign users.



The license will remain in your personal account. The end user's email will be in the end-user column on the Assignment tab.

Also, the new license end user will be able to find the license in the personal account and [import it](#) but the end user cannot change the license data. If the new license end-user has not been signed up on our website, the account will be created automatically. The new user will receive a welcome email with a link to set up a password while the previous user will receive an email that the license has been withdrawn.

Differences between license types

The following comparison table details the specific differences between **Annual** and **Perpetual** subscription. The license type is viewable at [your DBeaver profile](#).

	Annual subscription	Perpetual subscription (Discontinued after version 23.3)
Version updates	Access to all updates and support during the active license period, including Regular and EA releases.	Access to all updates and support during the active license period, including Regular and EA releases.
Duration of access	After the subscription expiration, access to new updates and support ceases.	Continuous use for the last accessible version is permitted indefinitely post-license expiration.
Extension options	Annual renewal of the subscription is required for ongoing access to updates and support.	Optional extensions are available for continued access to updates and support.

Note: The primary distinctions between Perpetual and Annual subscriptions are tied to the **Duration of access**. While Perpetual license holders can use DBeaver indefinitely with the last update received, Annual subscribers need to renew their subscription to access DBeaver.

Windows Silent Install

Table of contents

[Overview](#)

[Parameters](#)

[For install](#)

[For uninstall](#)

[Installer return codes](#)

Overview

It is possible to install/uninstall DBeaver in silent mode using the Windows Installer command line parameters. This might be beneficial for mass install automation (SSCM and other similar systems).

Parameters

Command line parameters supported by DBeaver installer/uninstaller:

Parameter	Description
<code>/S</code>	Enables silent mode. Requires either <code>/allusers</code> or <code>/currentuser</code> . Case-sensitive.
<code>/D=path</code>	Sets the installation directory. This parameter must be the last one used and is case-sensitive. No quotes needed.
<code>/allusers</code>	Installs or uninstalls the software for all users. Case-insensitive.
<code>/currentuser</code>	Installs or uninstalls the software for the current user only. Case-insensitive.

For install

To install DBeaver, navigate to the directory where the installer is located and append the desired parameters to the command:

For instance:

```
C:\Program Files\DBEaver> [parameters]
```

Note: To install with the `/allusers` parameter the current user must have the administrator's permission.

For uninstall

To uninstall DBEaver, use the `Uninstaller.exe` from the DBEaver directory with the required parameters:

For instance:

```
C:\Program Files\DBEaver>Uninstaller.exe [parameters]
```

Installer return codes

Code	Meaning
0	normal execution (no error)
1	(un)installation aborted by user (Cancel button)
2	(un)installation aborted by script
666660	invalid command-line parameters
666661	elevation is not allowed by defines
666662	uninstaller detected there is no installed version
666663	executing uninstaller from the installer failed
666666	cannot start elevated instance
other	Windows error code when trying to start elevated instance"

Installer was improved in DBEaver 5.3.3, special thanks to the [NsisMultiUser](#) team.

Snap Installation

Table of contents

[Installation](#)

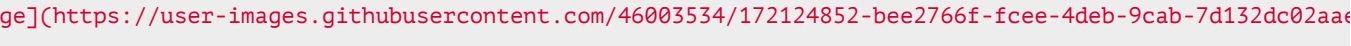
Installation

For install *dbeaver-ce* snap package with stable release (stable by default):

```
## Connect interfaces:

You can find list of all _dbeaver-ce_ snap connections:
```snap connections dbeaver-ce```

And connect what you need. For example:
```snap connect dbeaver-ce:ssh-keys``` for access to private keys

Or you can do it in Ubuntu Software (dbeaver-ce - Permissions):


### Problems with dbeaver-ce snap package:

At the moment there is a well-known problem with opening the browser and gis maps via _dbeaver-ce_ snap package.

`SWT WebKitGDBus: error creating DBus server Error binding to address (GUnixSocketAddress): Permission denied`
`SWT WebKit: error initializing DBus server, dBusServer == 0`

Snapcraft forum topic: https://forum.snapcraft.io/t/classic-confinement-for-dbeaver-ce/27502

### Workaround

While the problem is being fixed, you can use the workaround:

Run _dbeaver-ce_ snap package from: ```/snap/dbeaver-ce/current/usr/share/dbeaver-ce/dbeaver```
```

Statistics collection

Table of contents

[Overview](#)

[What data is shared](#)

[How is the data sent](#)

[User Control](#)

Overview

DBeaver collects usage statistics to enhance the user experience and improve the application's features. This article explains the purpose behind the data collection, the type of data collected, and how it aligns with privacy policies.

What data is shared

The data shared includes:

- Anonymized information about the operating system and [locale](#). A locale is a set of parameters that defines user-specific settings such as language, country, and formatting preferences for dates, times, and numbers.
- A list of actions performed in the user interface to analyze workflow efficiency, such as opening an editor, executing a query, using the data view panel, etc. These insights help us understand how users interact with DBeaver features.
- Types of databases utilized to focus on popular database support.

DBeaver does not collect:

- Personal information.
- Sensitive details such as database connection configurations, executed queries, or database information.

How is the data sent

The data is transmitted securely at the application's shutdown or during startup. To ensure transparency, you may review the collected data types in the DBeaver source code, available at [DBeaver's GitHub repository](#).

User Control

You have the authority to manage your data sharing preferences. To modify these settings:

1. Navigate to **Window -> Preferences -> General -> Usage Statistics**.
2. Use the checkbox to enable or disable the sending of usage statistics.
3. Click on **Apply** to save your preferences.

Note: While you can opt out of data sharing, participating in the statistics collection helps us enhance DBeaver's features and user experience. We value your contribution to our continuous improvement and encourage you to join this collective effort. All data is collected using the [DBeaver Corporation Privacy Policy](#).

Release cycles

Table of contents

[Release cycles](#)

[Release schedule](#)

Release cycles

DBeaver releases updates on a structured schedule, providing users with regular enhancements, new features, and fixes.

Release schedule

- **Regular releases:** DBeaver releases a new stable version every three months, resulting in four major updates annually. For detailed information, please refer to [DBeaver's official release schedule](#).
- **Early Access (EA) versions:** Additionally, DBeaver provides Early Access (EA) versions every two weeks. These versions allow users to engage with upcoming features and changes, offering feedback to help refine these enhancements before their official release.

Important: The EA version may not be as stable as the regular releases. Please use EA versions with caution.

For release notes and to download the current release, EA version, or archive releases, please visit [DBeaver's Download page](#). The GitHub repository also hosts all release notes, accessible at [GitHub](#).

Technical support information

Table of contents

[Accessing technical support](#)

[End of support](#)

Accessing technical support

DBeaver ensures all users receive technical support for the duration of their paid license. Tickets for technical support can be submitted via the [website](#).

End of support

Please be aware that technical support ends alongside the expiration of the license period. This occurs even though DBeaver remains usable for users with a Perpetual license type.

Tip: For detailed information on the differences between Perpetual and Annual license types, please refer to the [relevant article](#).

Debug options

Table of contents

[Command line parameters](#)

Command line parameters

Option	Meaning
<code>-Ddbeaver.debug.skip-version-check=true</code>	Disable new version release check. Always shows new version update dialog
<code>-Ddbeaver.debug.override-product-version=major.minor.micro</code>	Override current product version. Used only for version update dialog
<code>-Ddbeaver.debug.disable-data-transfer-settings-save=true</code>	Cancels export data transfer format specific settings saving. Like <code>Header position</code> for CSV or <code>Keyword case</code> for SQL export. Can be used for UI tests.
<code>-Ddbeaver.trace.enabled=true</code>	Enable additional log level for printing <i>finest</i> messages

Build from sources

Table of contents

[Prerequisites](#)

[Build steps](#)

Prerequisites

Before proceeding with building DBeaver, ensure that your system meets the following prerequisites:

- **JDK 17 or higher:** DBeaver currently uses [OpenJDK 17](#) as the default Java Development Kit. Make sure you have JDK 17 or a newer version installed.
- **Apache Maven 2.9.5 or higher:** This is required for managing the project's build. You can download it from the [Apache Maven website](#).
- **Git:** Needed for cloning the DBeaver repository. If you do not have Git installed, visit the [official Git website](#) for download and installation instructions.
- **Internet Access:** Required for downloading dependencies and other necessary components during the build process.

Build steps

1. Clone the DBeaver repository:

```
git clone https://github.com/dbeaver/dbeaver.git dbeaver
cd dbeaver
```

2. To build the project, use the appropriate script for your operating system:

On Windows, run: `sh build.cmd`

On Linux or macOS, run:

`sh ./build.sh`

After the build process completes, the binaries can be found in the `product/community/target/products` directory.

How to manage DBeaver snap package

Table of contents

[Install and Update](#)

Install and Update

To download the dbeaver-ce snap package, run the following command:

```
Or update it to the latest version:
```

```
```sudo snap refresh dbeaver-ce```
```

```
Managing snap interfaces
```

```
In order to have access to the private and public keys located in ./ ssh, you need to manually connect the
```

```
```sudo snap connect dbeaver-ce:ssh-keys``` for access to private keys
```

```
and
```

```
```sudo snap connect dbeaver-ce:ssh-public-keys``` for access to public keys
```

# Developing in Eclipse

---

## Table of contents

[Opening workspace in Eclipse](#)

[Running and Debugging in Eclipse](#)

[Sources structure](#)

[Notes](#)

[Include Plugins in the Debug Info](#)

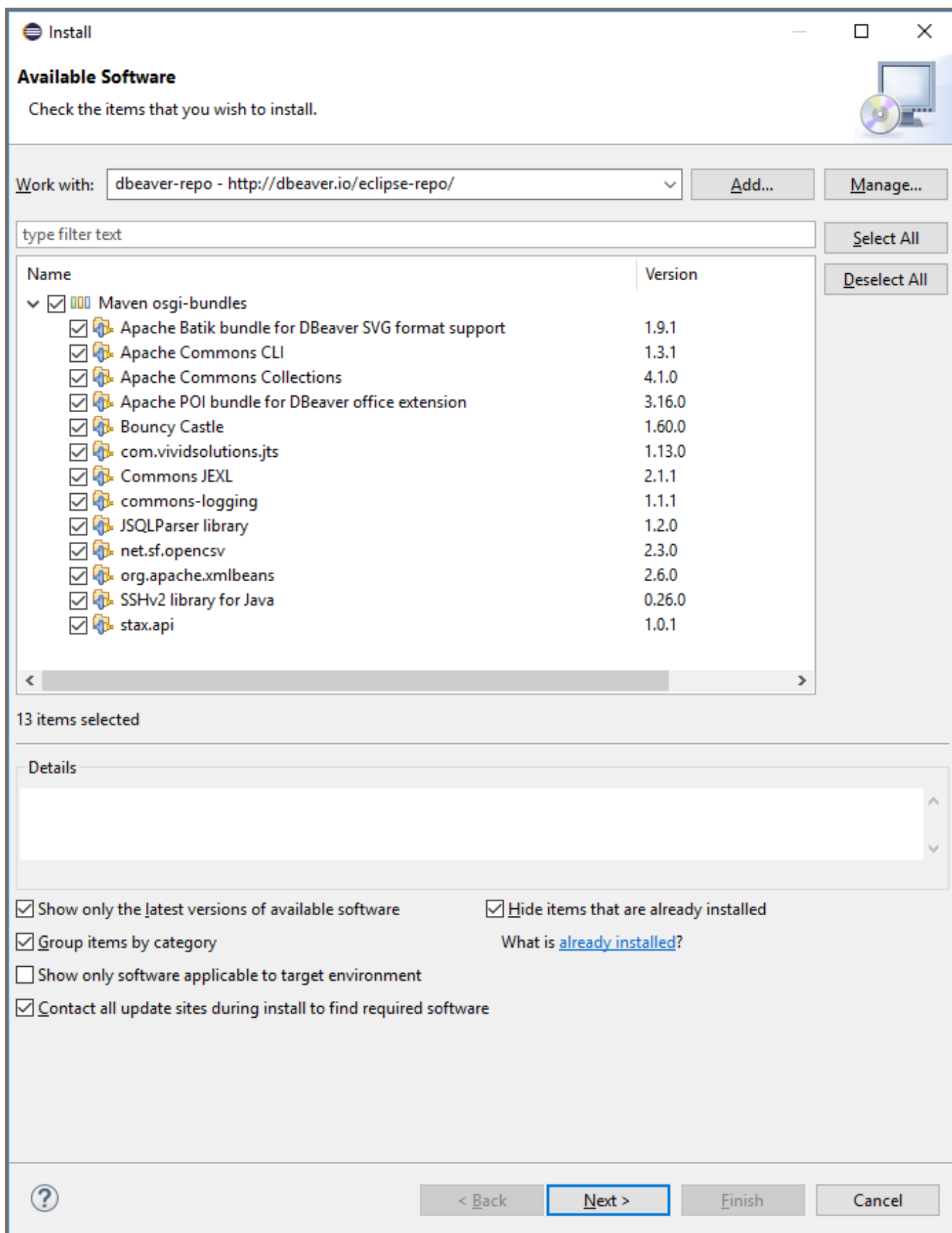
DBeaver is an [Eclipse RCP](#) application. It consists of a set of Eclipse plugins, features, and products.

## Opening workspace in Eclipse

---

You will need to set up Eclipse IDE for RCP and RAP Developers. The current default Eclipse version is **2023-12**. As the build may fail on older and newer versions, you will need to install a few additional Eclipse plugins which DBeaver depends on (and which are not included in the standard Eclipse distribution):

1. Open the menu **Help -> Install New Software**.
2. Copy <https://p2.dev.dbeaver.com/eclipse-repo/> into the "Work with" field and press Enter.
3. Check all plugins and finish the installation process:



4. Do the same for <https://eclipse-color-theme.github.io/update> (optional).

5. Restart Eclipse.

6. Check out the DBeaver repository somewhere with the command

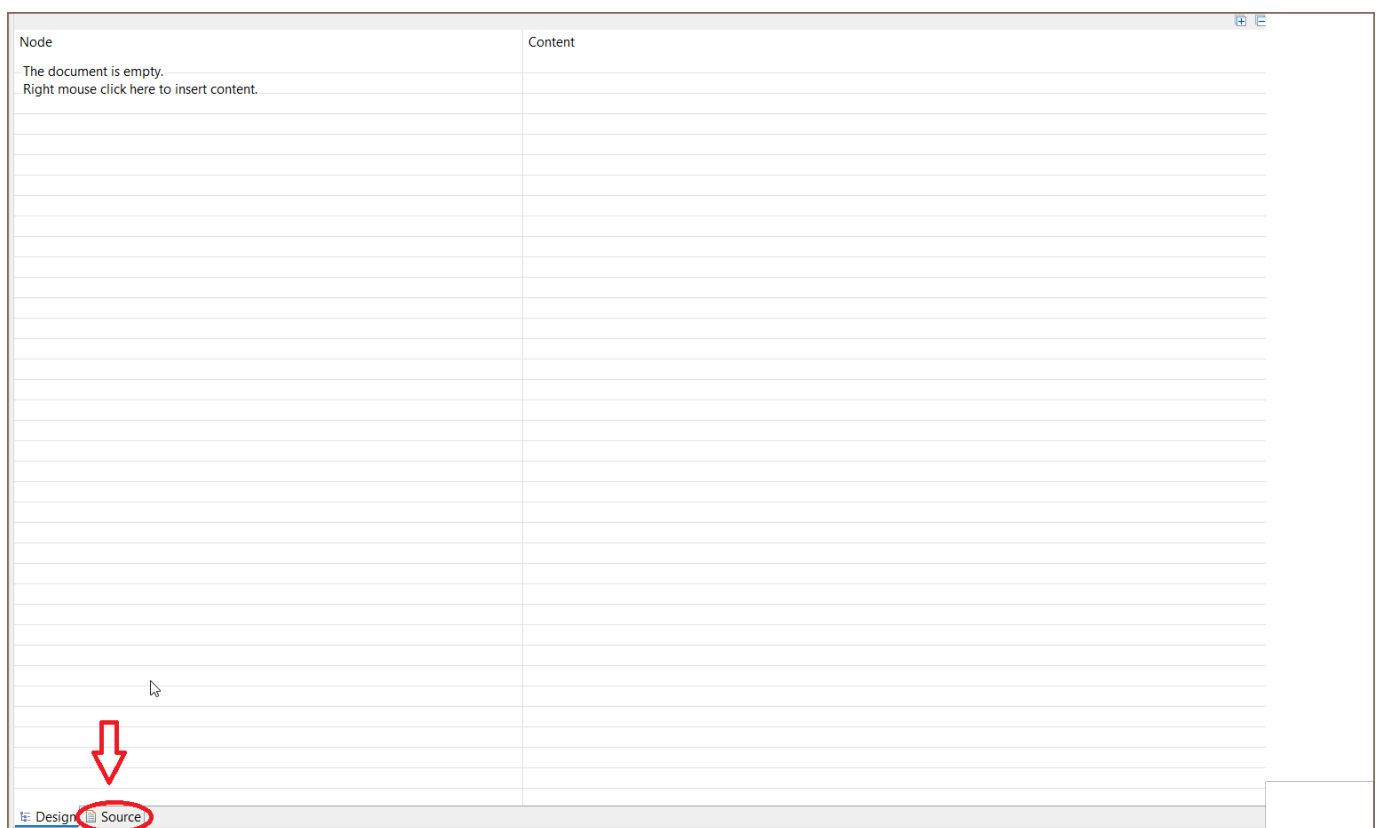
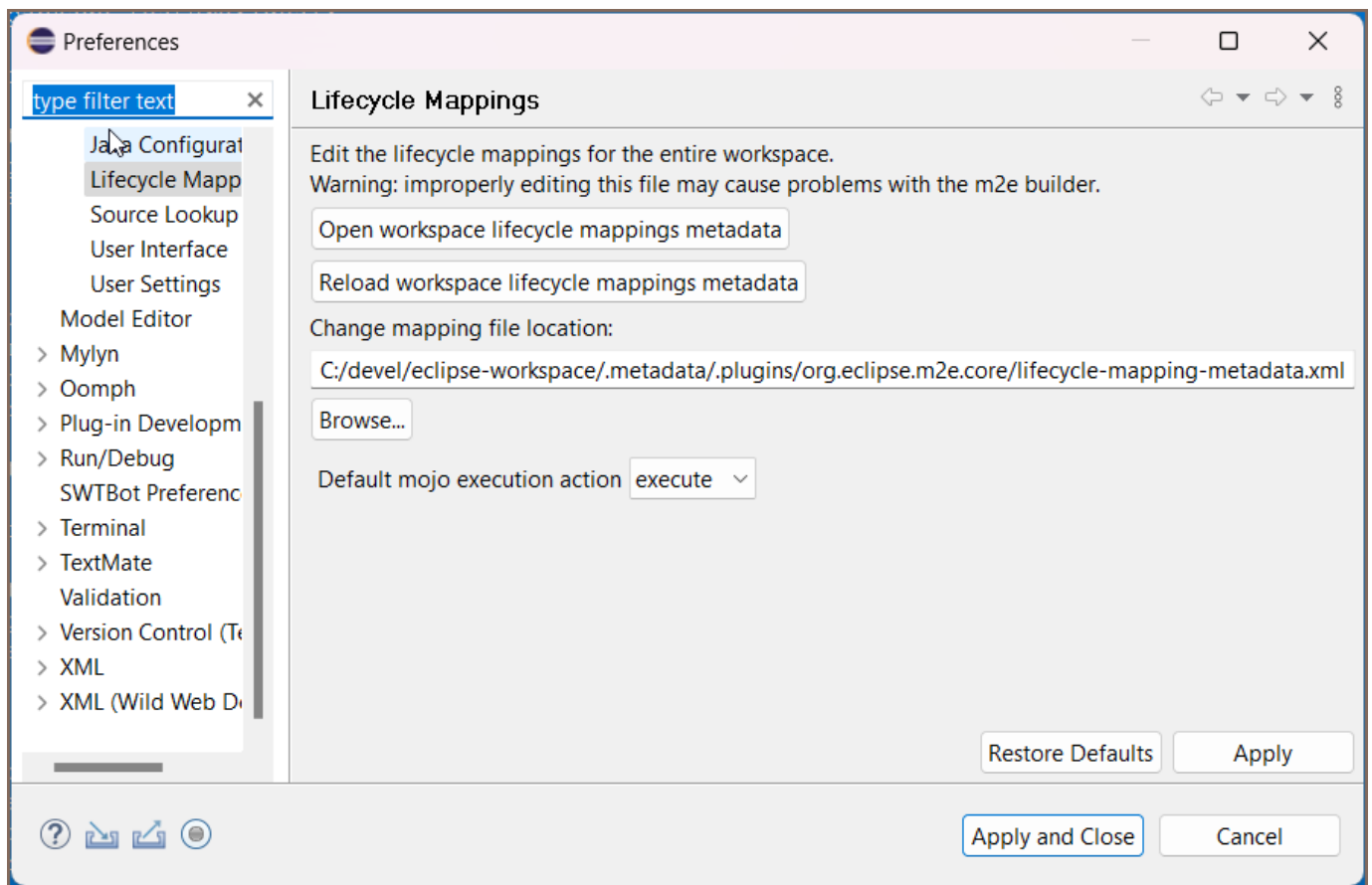
```
git clone https://github.com/dbeaver/dbeaver.git .
```

7. Check out additional dependencies with the command

```
git clone https://github.com/dbeaver/dbeaver-common.git .
```

8.

Before importing the projects, proceed to **Window -> Preferences -> Maven -> Lifecycle mapping** , press **Open workspace lifecycle mappings metadata** , then switch to text by pressing **Source** :



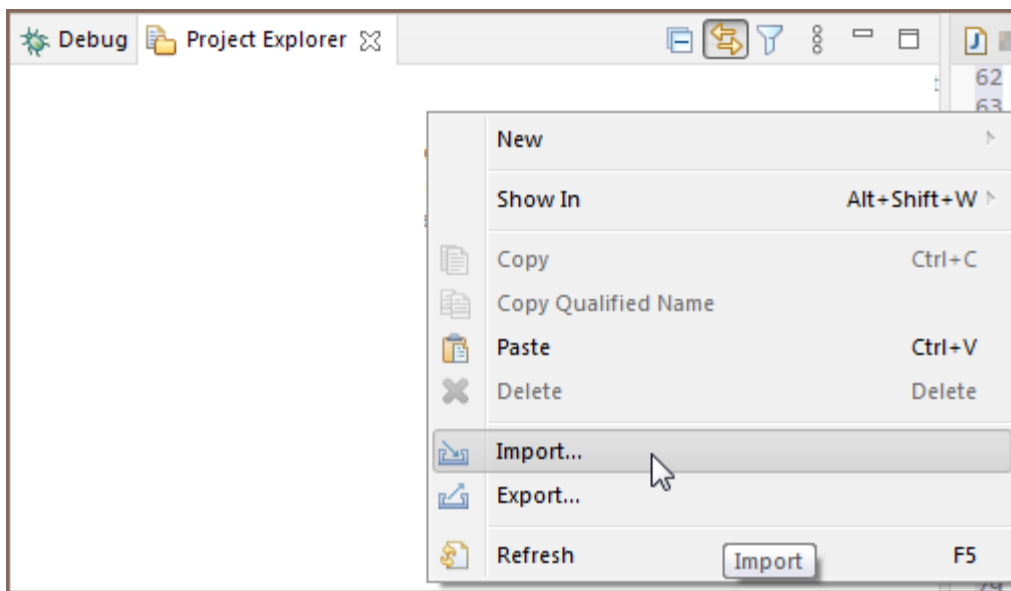
9. Insert the following configuration and save it by pressing **CTRL + S** :

```

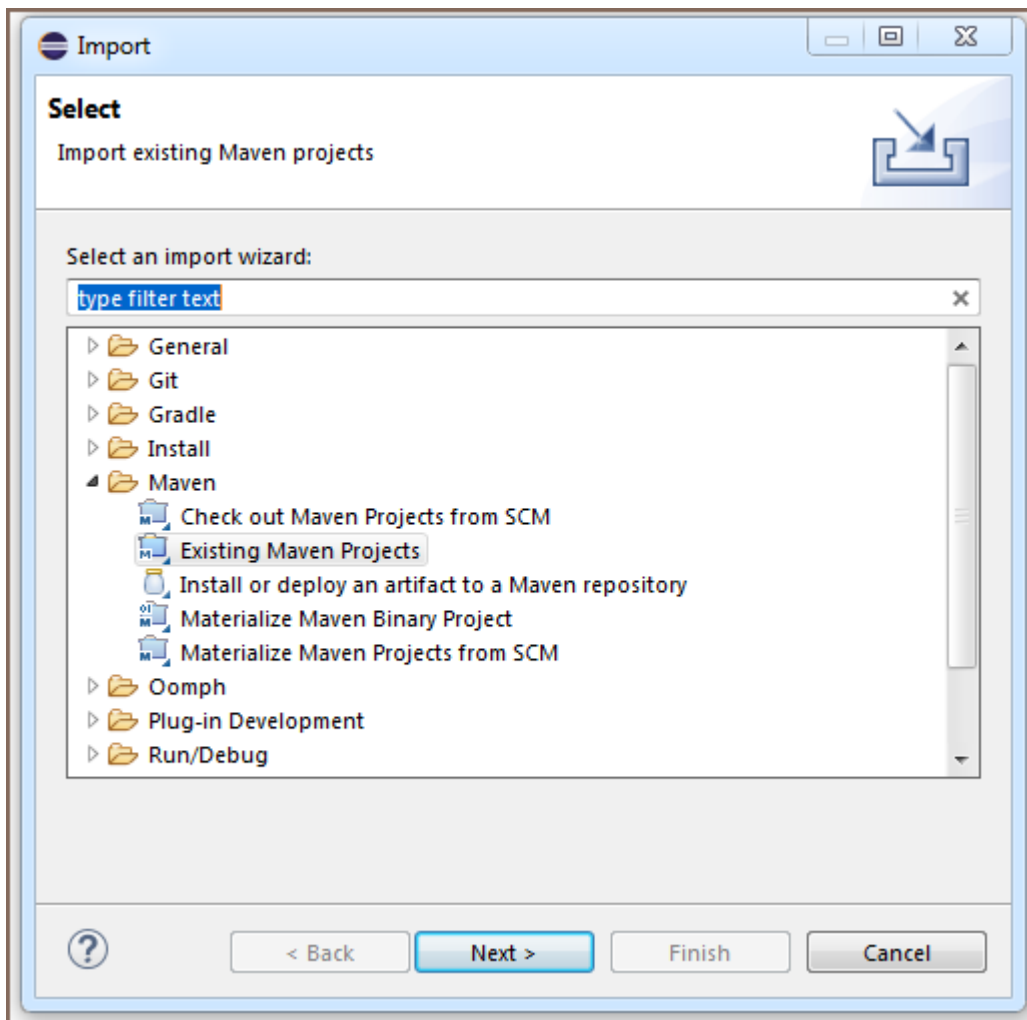
<lifecycleMappingMetadata>
 <pluginExecutions>
 <pluginExecution>
 <pluginExecutionFilter>
 <groupId>org.eclipse.tycho</groupId>
 <artifactId>tycho-packaging-plugin</artifactId>
 <versionRange>[1.0.0,)</versionRange>
 <goals>
 <goal>package-plugin</goal>
 <goal>package-feature</goal>
 </goals>
 </pluginExecutionFilter>
 <action>
 <configurator>
 <id>org.apache.m2e.pde.connector.configurator.tycho</id>
 </configurator>
 </action>
 </pluginExecution>
 <pluginExecution>
 <pluginExecutionFilter>
 <groupId>org.eclipse.tycho</groupId>
 <artifactId>tycho-surefire-plugin</artifactId>
 <versionRange>[1.0.0,)</versionRange>
 <goals>
 <goal>test</goal>
 </goals>
 </pluginExecutionFilter>
 <action>
 <configurator>
 <id>org.apache.m2e.pde.connector.configurator.tycho</id>
 </configurator>
 </action>
 </pluginExecution>
 <pluginExecution>
 <pluginExecutionFilter>
 <groupId>dev.aspectj</groupId>
 <artifactId>aspectj-maven-plugin</artifactId>
 <versionRange>1.13.1</versionRange>
 <goals>
 <goal>compile</goal>
 </goals>
 </pluginExecutionFilter>
 <action>
 <ignore />
 </action>
 </pluginExecution>
 <pluginExecution>
 <pluginExecutionFilter>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-assembly-plugin</artifactId>
 <versionRange>3.3.0</versionRange>
 <goals>
 <goal>single</goal>
 </goals>
 </pluginExecutionFilter>
 <action>
 <ignore />
 </action>
 </pluginExecution>
 </pluginExecutions>
</lifecycleMappingFilters>
<lifecycleMappingFilter>
 <symbolicName>org.eclipse.m2e.pde.connector</symbolicName>
 <versionRange>[2.1.2,)</versionRange>
 <packagingTypes>
 <packagingType>eclipse-test-plugin</packagingType>
 <packagingType>eclipse-plugin</packagingType>
 <packagingType>eclipse-feature</packagingType>
 </packagingTypes>
</lifecycleMappingFilter>
</lifecycleMappingFilters>
</lifecycleMappingMetadata>

```

10. In Eclipse, run the Import wizard:

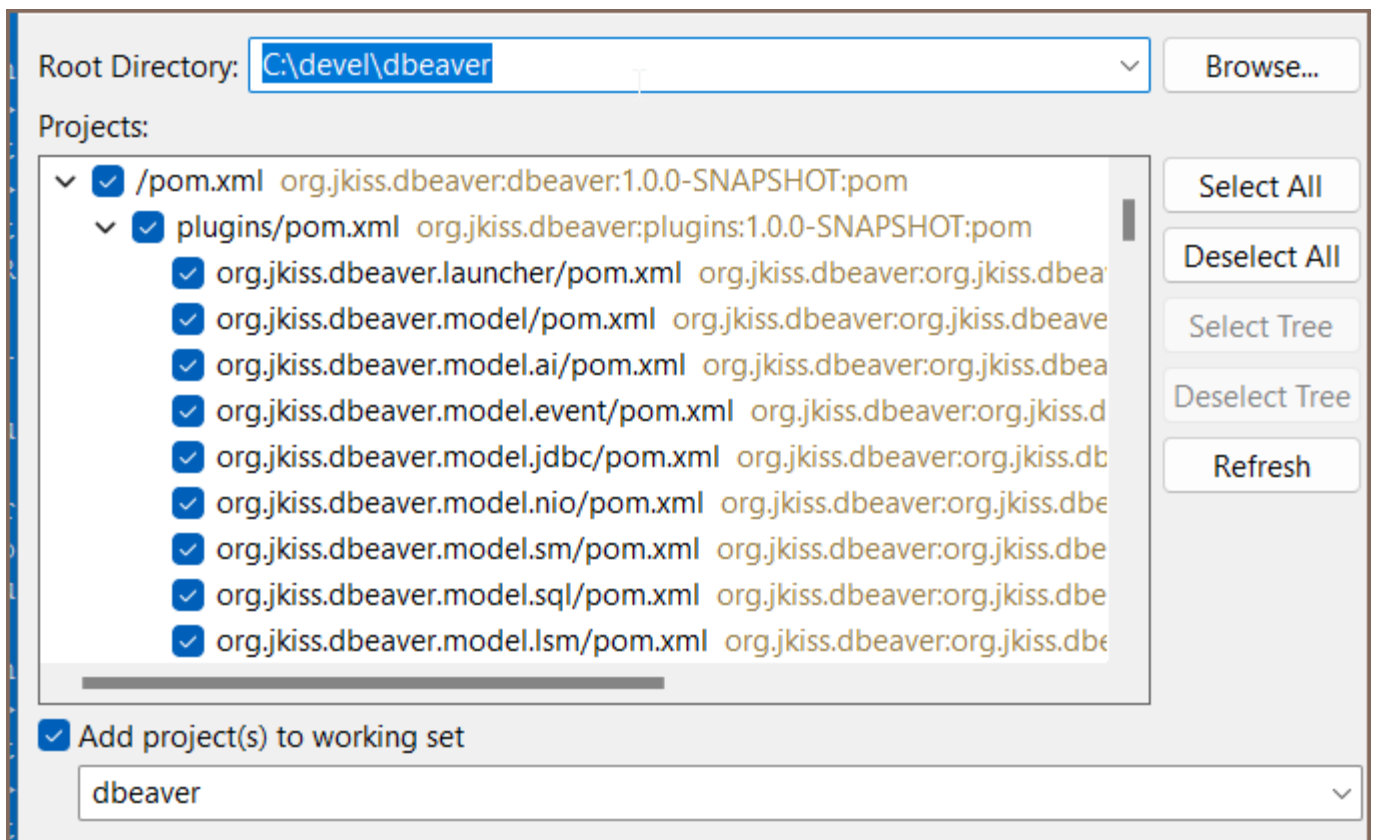


11. Choose **Maven -> Existing Maven Projects**:

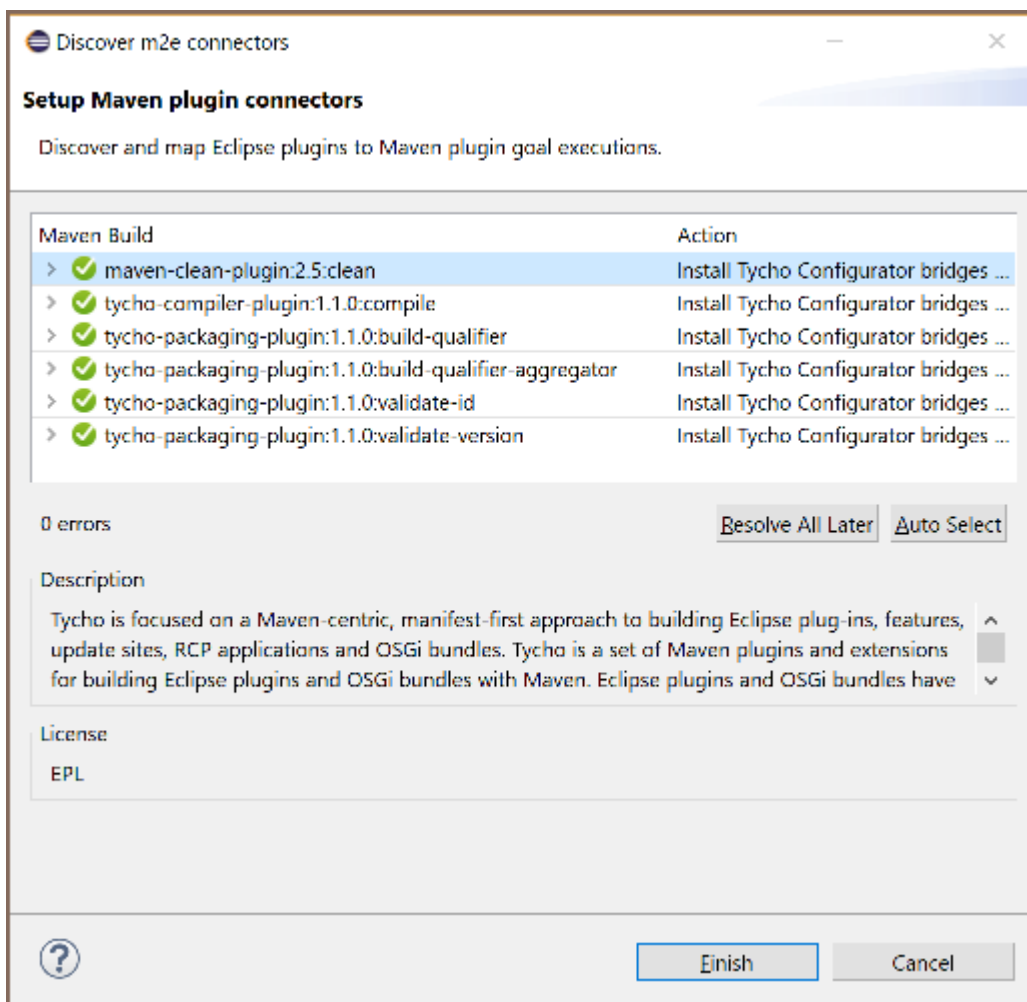


12. Select the DBeaver checkout directory. Check all plugins and features Eclipse will find and finish the wizard:

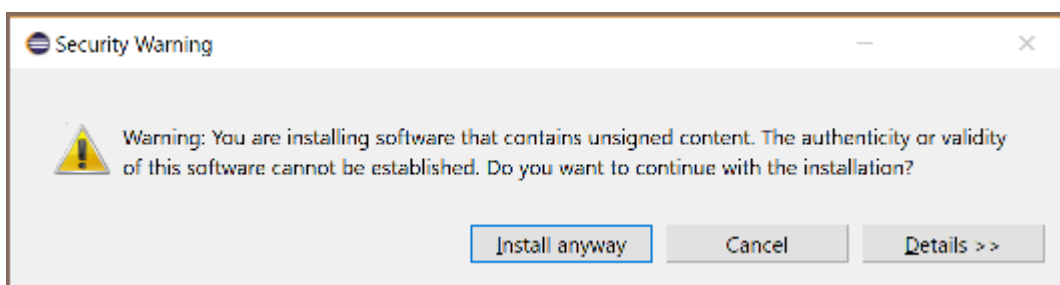
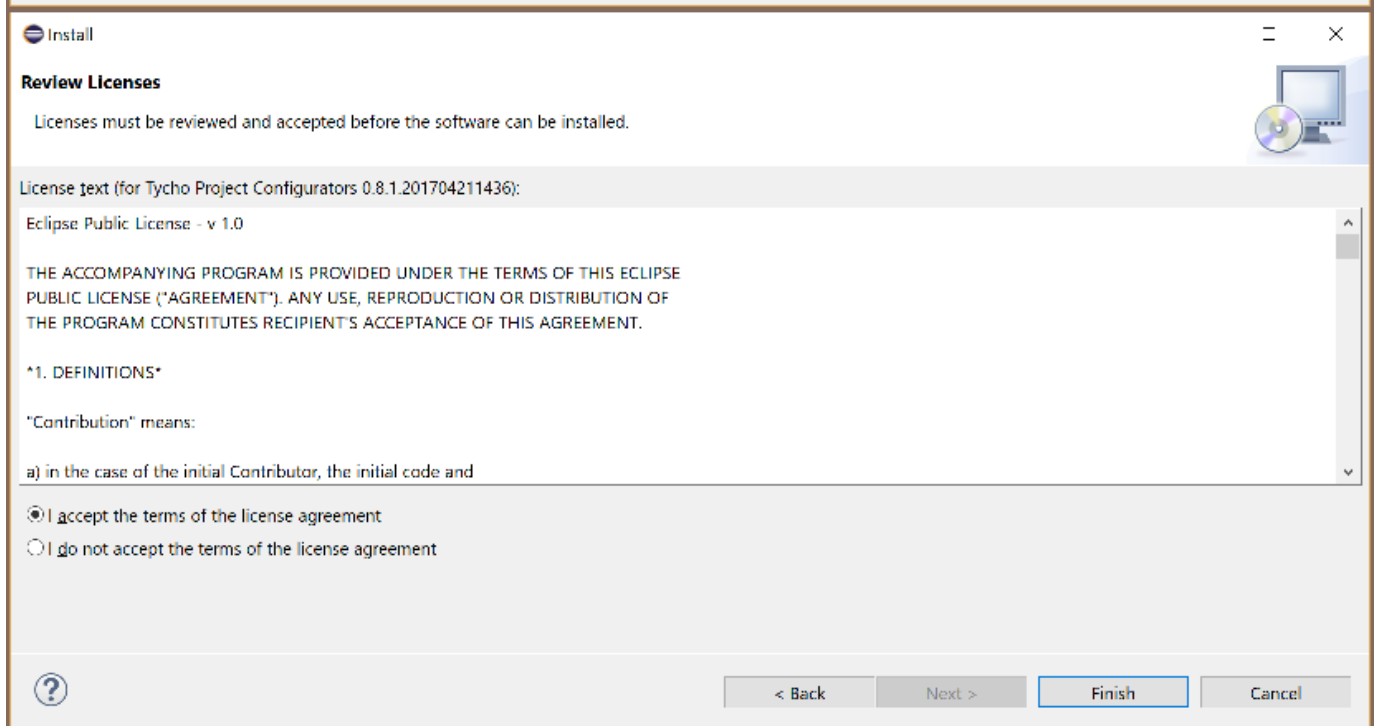
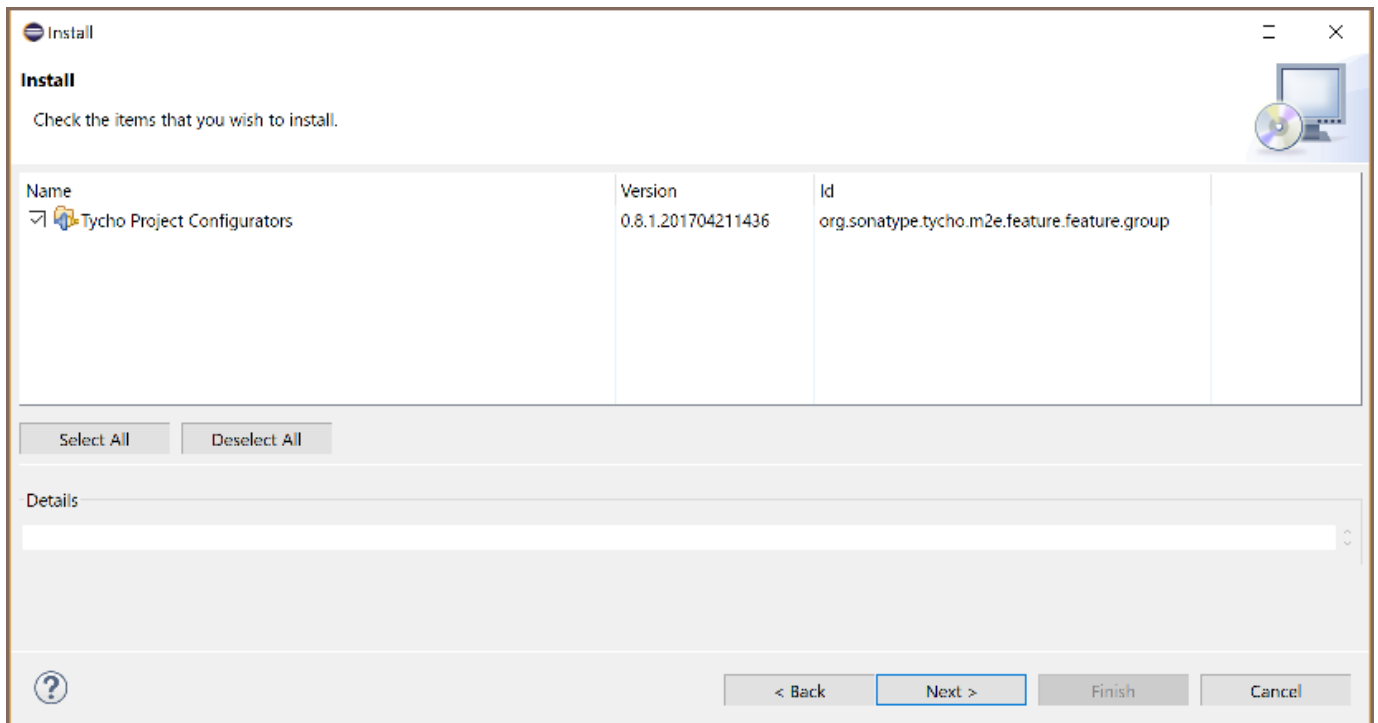




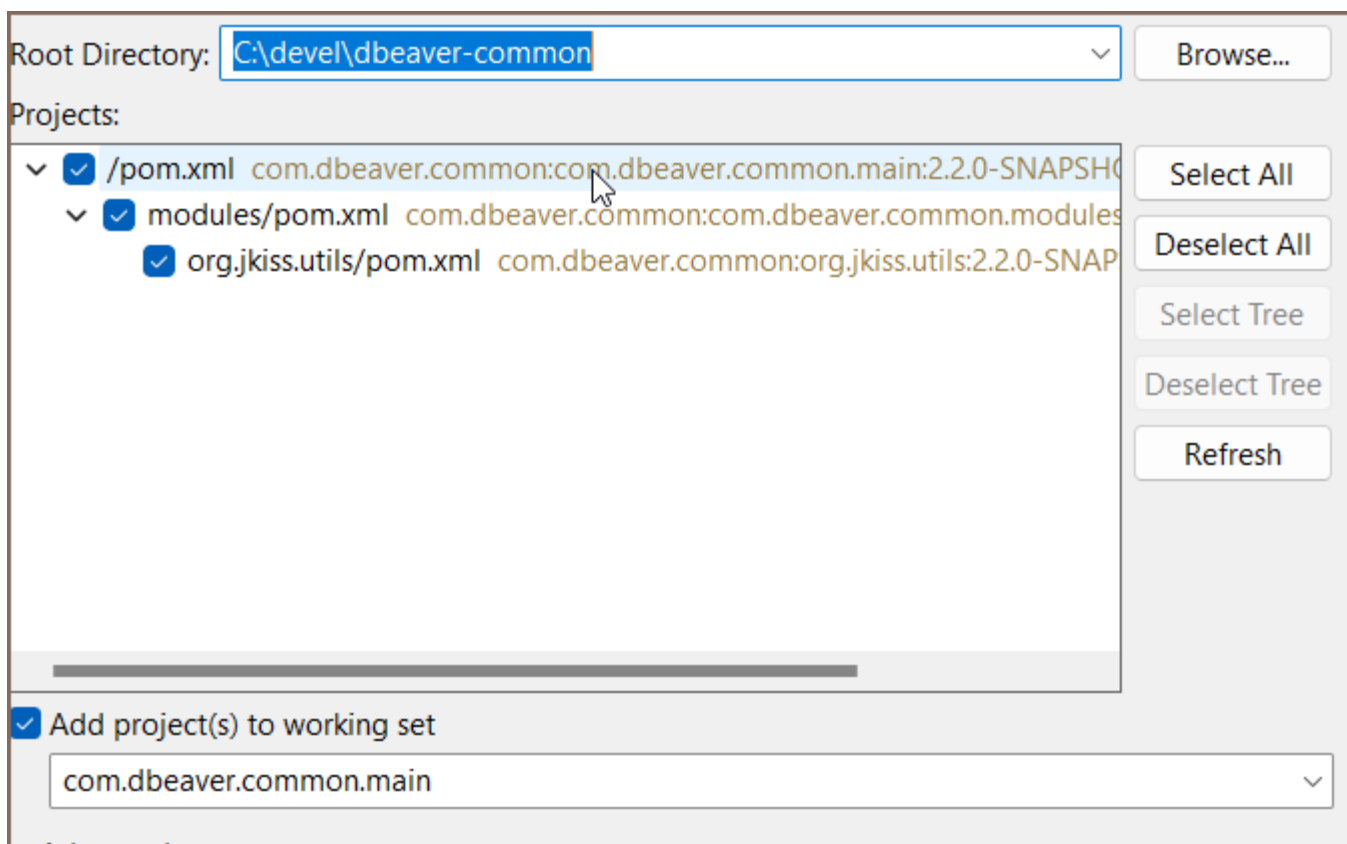
13. Eclipse will perform the project configuration process. This may take a few minutes:



14. If Eclipse asks about additional plugin installation, just agree:

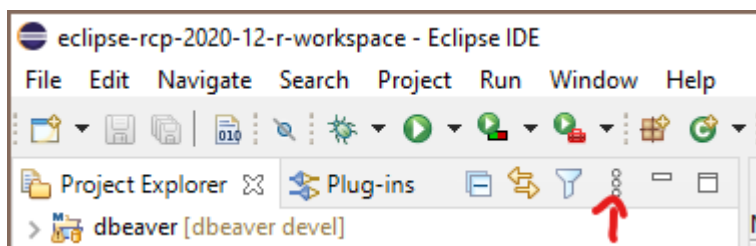


15. Repeat steps 10–14 for the folder with the `dbeaver-common` repository:

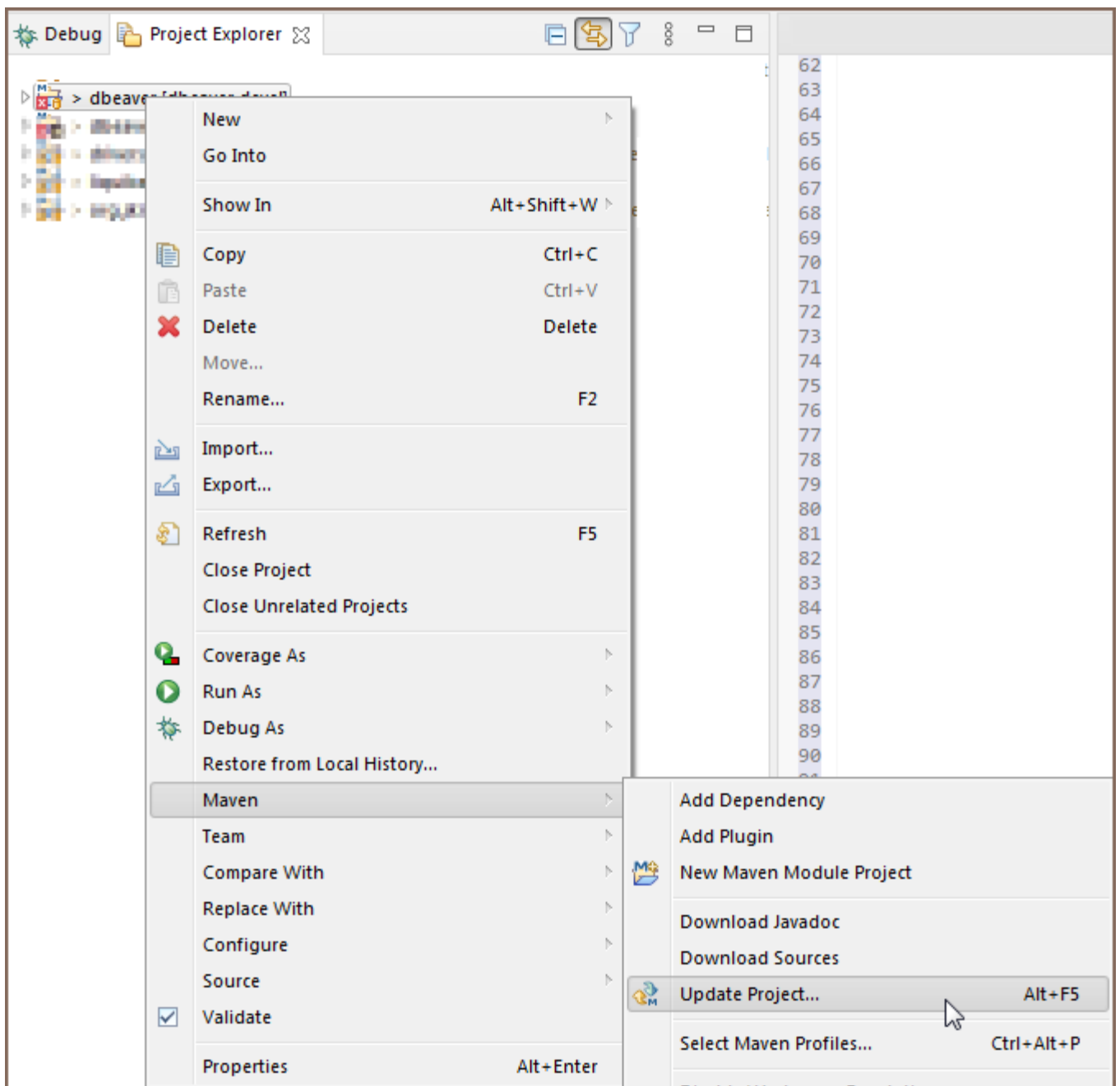


16. Restart Eclipse. Eclipse will build the project automatically, or you can press `Ctrl+B`.

17. Switch to Hierarchical project presentations:



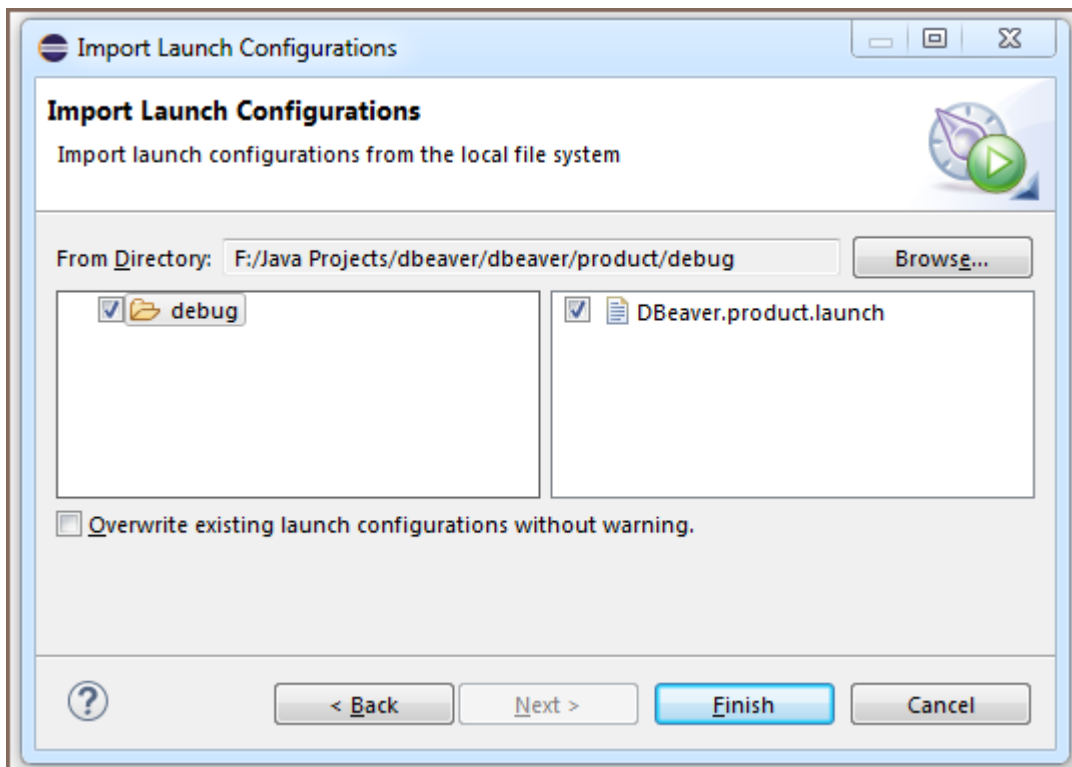
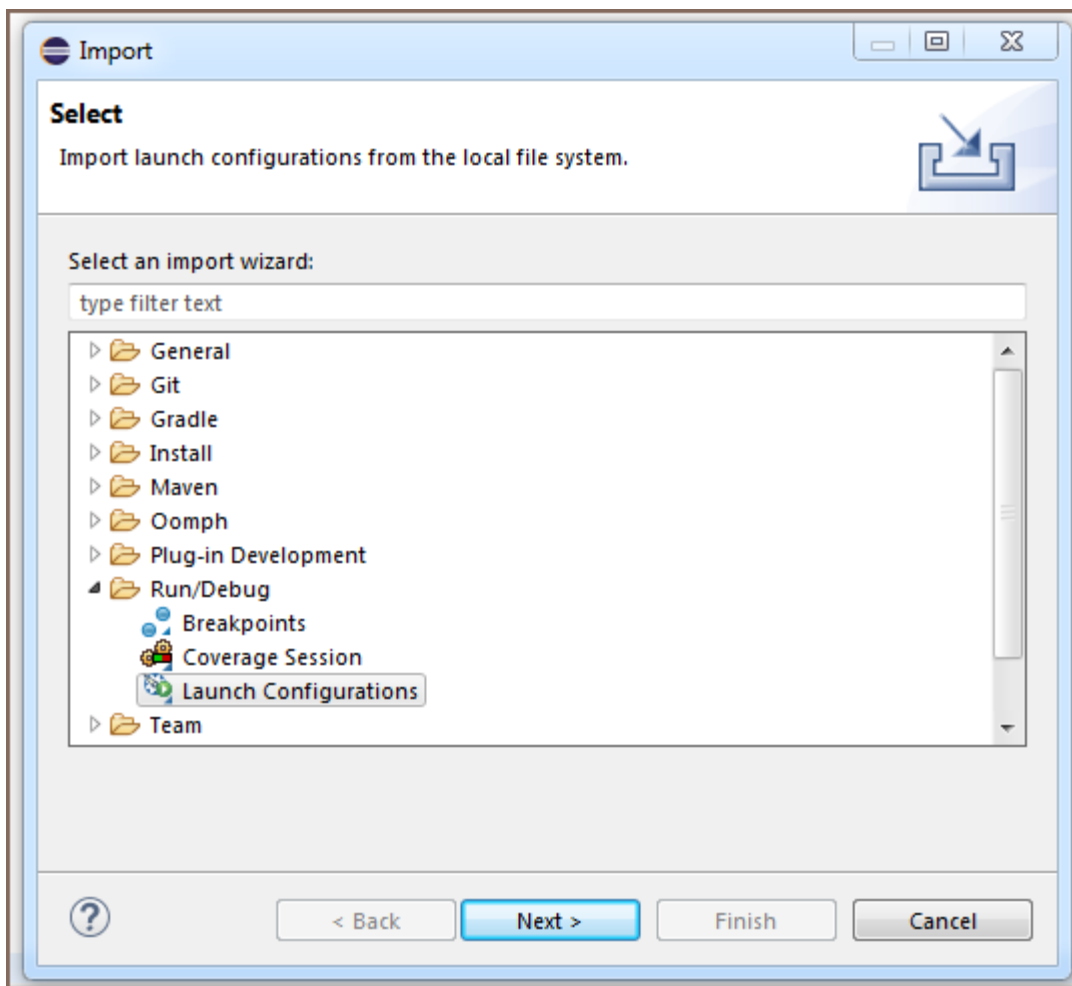
18. If some modules fail to build (they will be marked with a red cross) - update the `dbeaver` and `dbeaver-common` Maven projects. If this does not help, try restarting Eclipse:



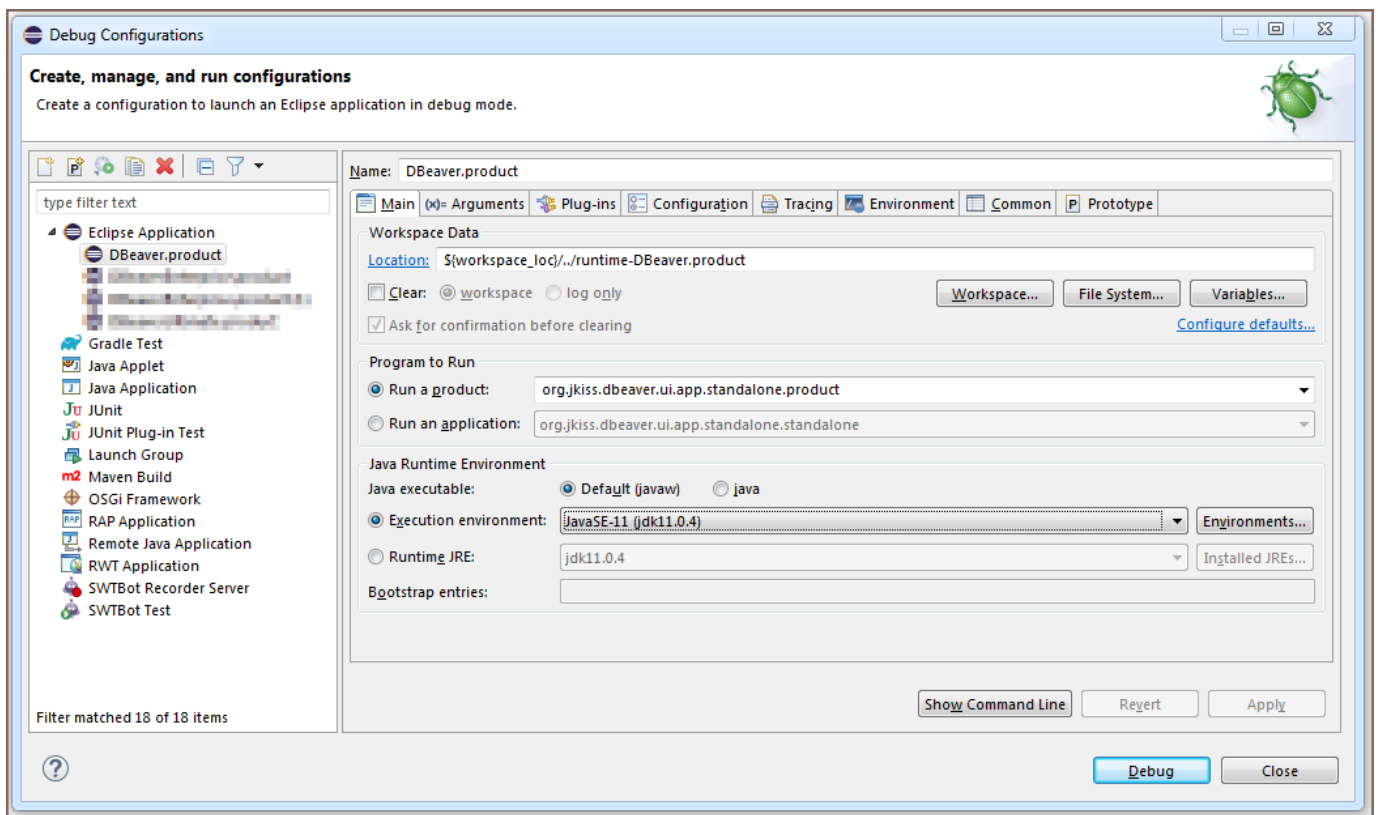
19. If you cannot compile and see **Access Restriction** errors in the Error log, then you can turn these errors into warnings. Open the context menu on the **Access Restriction** error line -> **Quick Fix** -> **Configure problem severity**.

## Running and Debugging in Eclipse

1. Import the DBeaver launch configuration using the Eclipse Import wizard:



2. Click on the **Debug** icon -> **Debug configurations**.
3. Choose **DBEaver.product** and click the **Debug** button:



If the product does not start with the `No application id has been found` error, try this workaround:

<https://stackoverflow.com/questions/13575695/eclipse-rcp-no-application-id-has-been-found>

**Tip:** What you need to do is go to your Run Configurations (**Run -> Run Configurations**). Select the Plug-ins tab and then hit **Add Required Plug-ins**. **Apply**, then **Run**, and it should work now.

That is it.

## Sources structure

- `docs` - some documentations (mostly outdated)
- `features` - feature descriptors. Does not contain any source code. Used to structurize the product plugins /dependencies.
- `bundles` - very base plugins (like utils)
- `plugins` - main sources location
- `org.jkiss.dbeaver.model` - model API and base classes. Does not contain any UI dependencies, just a pure data model.

- `org.jkiss.dbeaver.core` - main DBeaver module. Most of the base UI classes are here.
- `org.jkiss.dbeaver.ui.app.standalone` - a relatively small module which configures standalone DBeaver application.
- `org.jkiss.dbeaver.ui.app.eclipse` - the main Eclipse plugin. Adds some extra menus/views to the standard Eclipse IDE.
- `org.jkiss.dbeaver.ext.*` - DBeaver extensions
- `product` - the final products (standalone and Eclipse plugin) configuration.

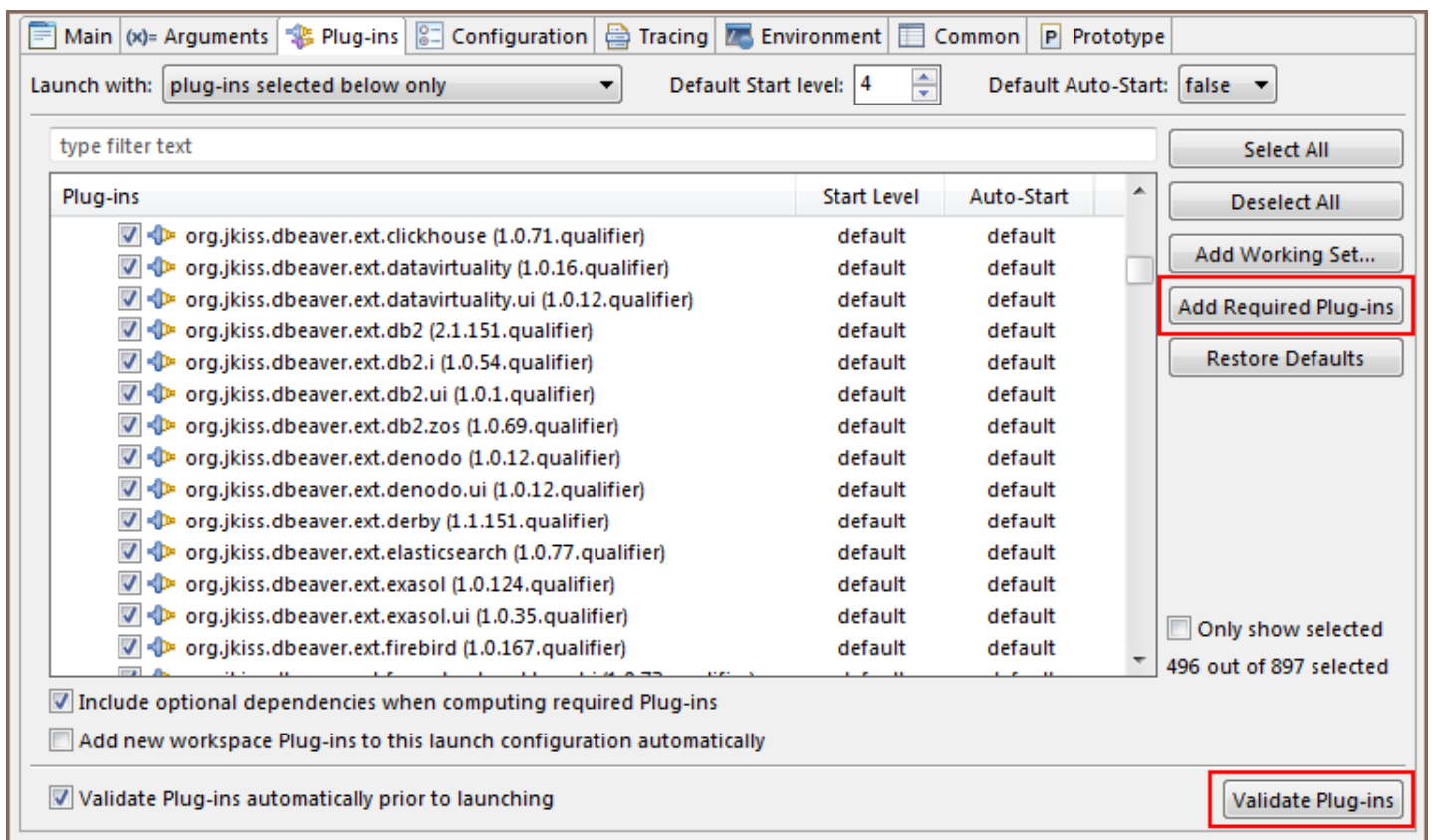
## Notes

---

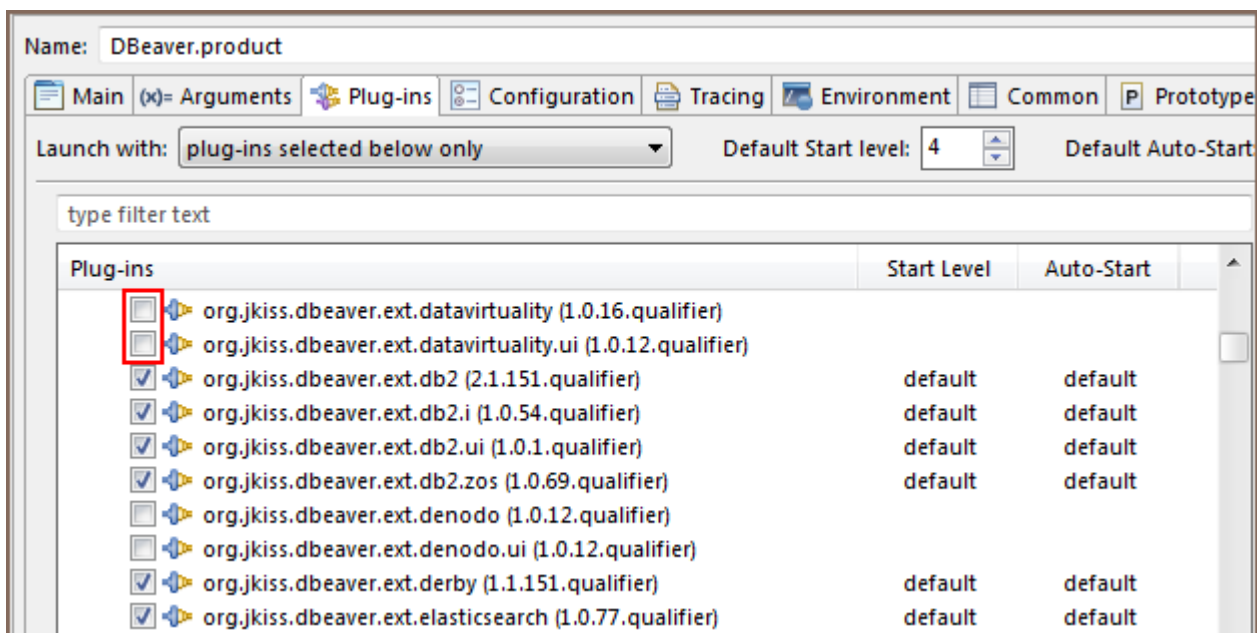
The instructions above cover the Community Edition version.

### Include Plugins in the Debug Info

Sometimes, your plugins may depend on other plugins that are not included in the `DBeaver.product.launch` file. You can verify this in **Debug Configurations** -> **Plug-ins** tab by clicking the **Validate Plug-ins** button. If dependency problems appear in the dialogue, you can use the **Select Required** button. This action sometimes resolves all dependency issues.



Alternatively, you can manually add the necessary dependencies. A small tip: If you have added a new plugin and are confident it should work, but do not see the changes after the project build, it might be because the plugin is not marked in the **Plug-ins** tab.





## 3rd party dependencies

---

Besides the base [Eclipse](#) platform DBeaver uses the following components:

- [JSch](#) – Java Secure Channel library (SSH tunnels support)
- [JEXL](#) – expression language libraries
- [JSqlParser](#) – SQL parser for SQL injections
- [H2](#) – embedded database engine for Query Manager (EE version)
- [ASM](#) – Java byte-code manipulations
- [OpenCSV](#) – simple CSV reading/writing library
- [Silk](#) icon set

Windows installer created by [NSIS](#). Debian package created by [ant-deb-task](#). RPM package created by [redline-rpm](#).

# Developing in IDEA

---

## Table of contents

[Overview](#)

[Instructions](#)

[Opening DBeaver in IDEA](#)

[Prepare Eclipse IDE and workspace](#)

[Configuring dependencies and compiling](#)

[Running and debugging](#)

[Additional information](#)

[Cannot launch DBeaver debug on macOS or Linux](#)

[Troubleshooting](#)

## Overview

DBeaver is built on the [Eclipse Platform](#), making the [Eclipse IDE](#) a logical choice for its development. However, many developers prefer the [IntelliJ IDEA](#).

In this guide, we aim to demonstrate how to open the DBeaver project in IDEA, write and compile code, and, most importantly, run and debug the application, leveraging the excellent Java debugger available in IDEA.

## Instructions

Opening the project and understanding its module structure in IDEA is relatively straightforward. The complexity arises when attempting to run or debug the project, as this process requires an Eclipse runtime workspace, a feature exclusive to the Eclipse IDE that IntelliJ IDEA does not support. Therefore, Eclipse is still necessary to configure the workspace for debugging purposes. Once configured, however, Eclipse will not be needed unless there are changes to the plugin.

## Opening DBeaver in IDEA

1. Create a folder where you will clone DBeaver repositories.

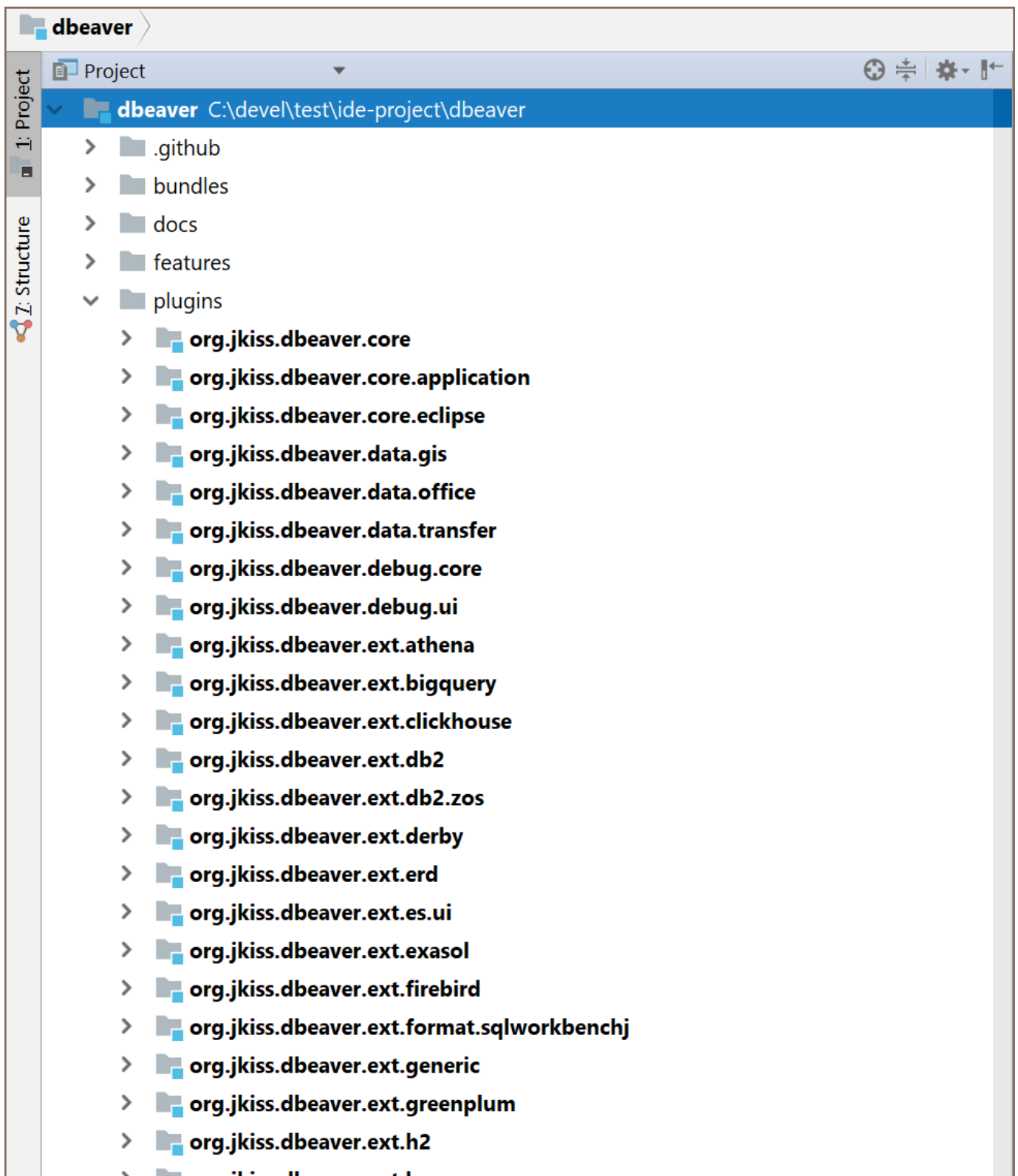
**Note:** all repositories must be located in the same folder. For example, name it `dbeaver-ce`.

2. Navigate to the created folder using the command: `cd dbeaver-ce`.
3. Clone the DBeaver repository with: `git clone https://github.com/dbeaver/dbeaver.git`.
4. Clone the IDEA project settings with: `git clone https://github.com/dbeaver/dbeaver-idea-project.git`.
5. Clone the common components with: `git clone https://github.com/dbeaver/dbeaver-common.git`.

**Tip:** Optionally, for documentation purposes, clone the WIKI repository with:

`git clone https://github.com/dbeaver/dbeaver.wiki.git` (skip this step if you do not need the WIKI).

6. Launch IDEA, choose **Open**, and select the folder `dbeaver-ce/dbeaver-idea-project`.
7. Now, you should see the DBeaver modules structure:



**Note:** You can edit the code, but you cannot compile it yet due to missing Eclipse dependencies.

## Prepare Eclipse IDE and workspace

1. Follow all the steps outlined in the [Develop in Eclipse](#) article, using the just cloned `dbeaver-ce/dbeaver` as the source root.

2. After completing these steps, you should be able to compile and run DBeaver from within Eclipse.
3. Make a note of your Eclipse IDE's location, referred to as `eclipse-ide-path`.

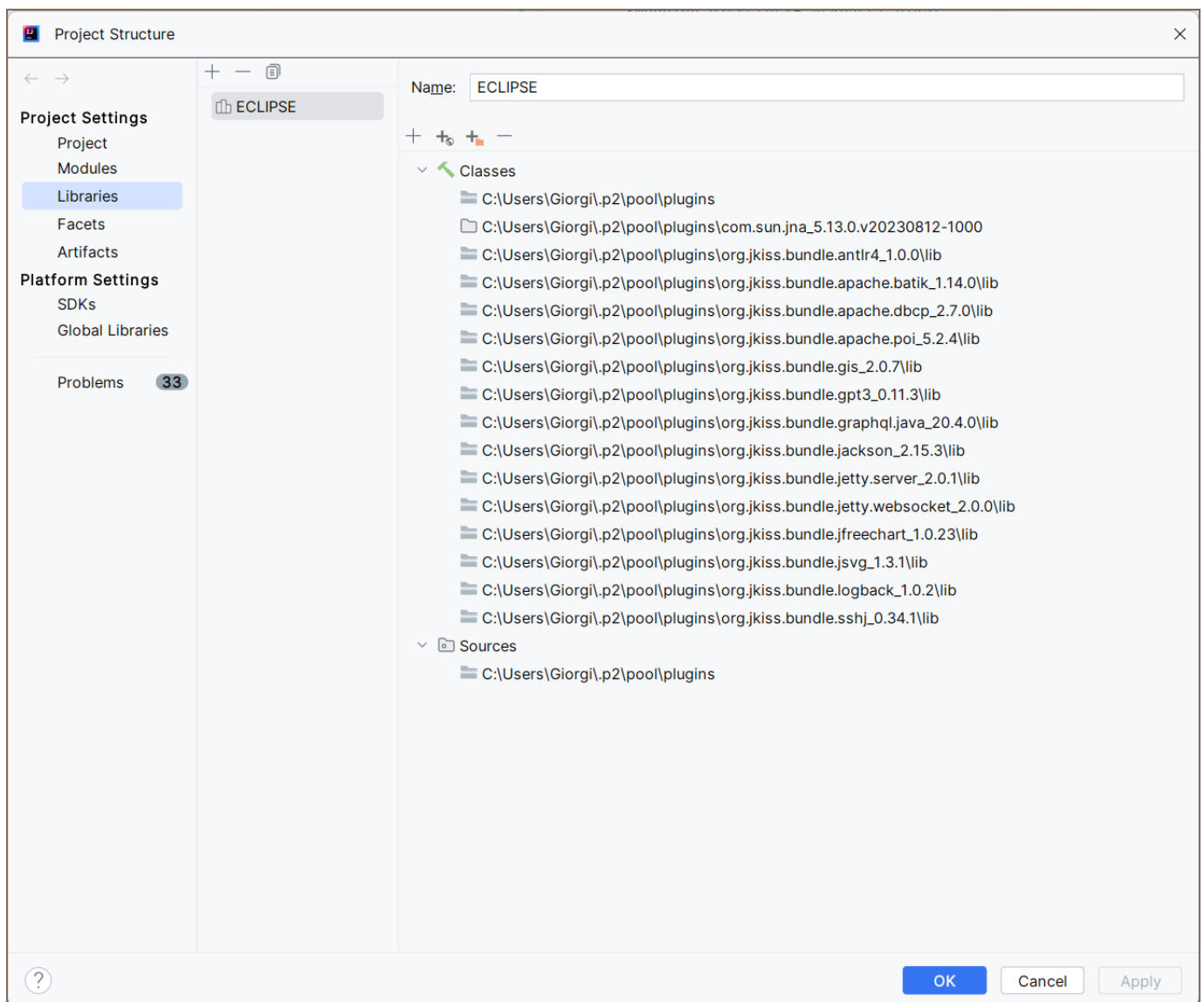
## Configuring dependencies and compiling

1. First, we need to add Eclipse dependencies to the IDEA project configuration.
2. Navigate to **File -> Project Structure....**
3. Ensure the project's Java SDK is correctly configured (Java 17 or newer is required).
4. Proceed to the **Libraries** section.
5. Click the **Add** button, choose Java, then select the `eclipse-ide-path/plugins` folder. After clicking the OK button, a dialog will appear to add the library to modules. In this dialog, click **Cancel**.
6. Rename the library to ECLIPSE (both the name and case are significant).
7. Recent versions of IDEA should automatically include all Eclipse JARs as dependencies. However, you may need to add some libraries manually.

- `eclipse-ide-path/plugins/com.sun.jna*`.
- `eclipse-ide-path/plugins/org.jkiss.bundleantlr*/lib`.
- `eclipse-ide-path/plugins/org.jkiss.bundle.apache.batik*/lib`.
- `eclipse-ide-path/plugins/org.jkiss.bundle.apache.dbcp*/lib`.
- `eclipse-ide-path/plugins/org.jkiss.bundle.apache.poi*/lib`.
- `eclipse-ide-path/plugins/org.jkiss.bundle.gis*/lib`.
- `eclipse-ide-path/plugins/org.jkiss.bundle.gpt3*/lib`.
- `eclipse-ide-path/plugins/org.jkiss.bundle.graphql.java*/lib`.
- `eclipse-ide-path/plugins/org.jkiss.bundle.jackson*/lib`.
- `eclipse-ide-path/plugins/org.jkiss.bundle.jetty.server*/lib`.

- `eclipse-ide-path/plugins/org.jkiss.bundle.jetty.websocket*/lib` .
- `eclipse-ide-path/plugins/org.jkiss.bundle.jfreechart_*/lib` .
- `eclipse-ide-path/plugins/org.jkiss.bundle.svg_*/lib` .
- `eclipse-ide-path/plugins/org.jkiss.bundle.logback_*/lib` .
- `eclipse-ide-path/plugins/org.jkiss.bundle.sshj_*/lib` .

8. You should see the following list:



9. If there are any other missing dependencies, add all folders in `eclipse-ide-path/plugins/org.jkiss*` .

**Tip:** If you require tests, manually download and add Mockito and JUnit.

10. Before compiling, you will need to build parser grammar files. To do this, locate

`org.jkiss.dbeaver.model.lsm` in your Project. Right-click on the module and select **Open In -> Terminal**.

Then, execute `mvn clean install` in the terminal.

11. If the module does not compile, first build the `dbeaver-common` components, and then, in the `dbeaver` folder, run the build process again.

**Tip:** If your processor has more than one core, execute the command `mvn clean install -T 1C` to speed up compilation. This will not be effective if there is insufficient RAM.

12. To avoid version conflict when compiling **JSCH** module open in project explorer

**External Libraries -> ECLIPSE** and locate two libraries called `com.jcraft.jsch_*.jar` remove the older version before proceeding.

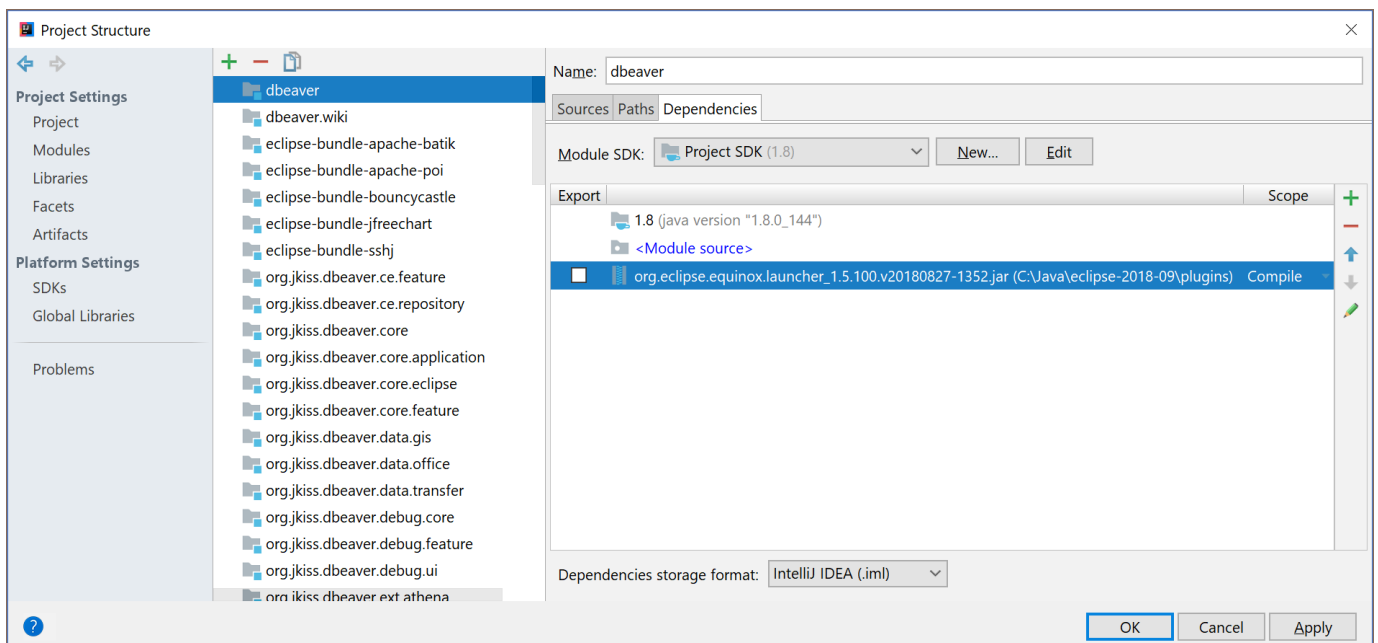
13. Now, you should be able to compile the project. Hit **CTRL+F9** and wait.

## Running and debugging

DBeaver is an Eclipse RCP application and must be started using the Eclipse Equinox Launcher bundle. This involves a special JAR file found in the Eclipse plugins folder, which we need to add directly to dependencies to enable debugging.

1. Add the launcher configuration as a JAR dependency for the `dbeaver` module. Locate the JAR

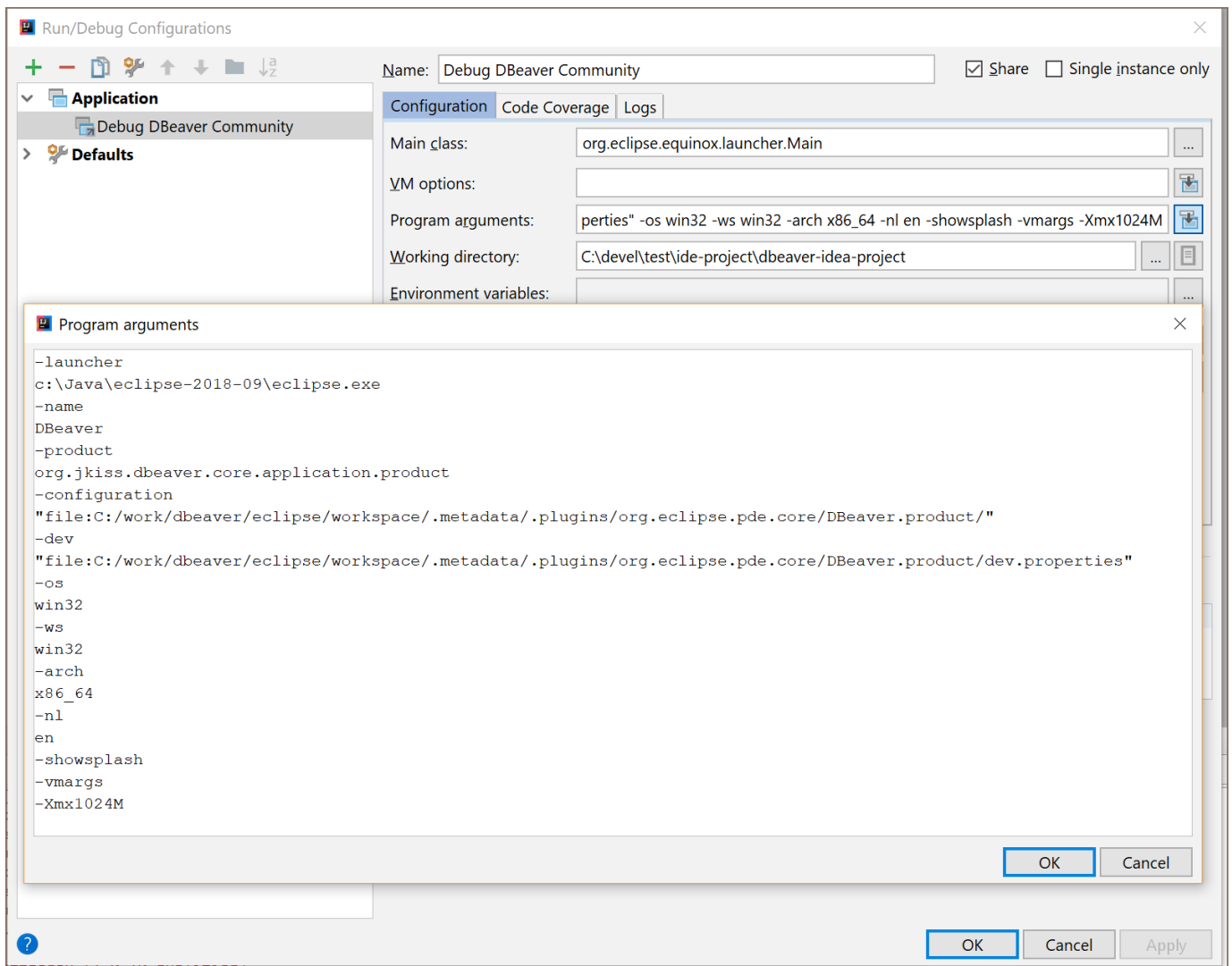
`org.eclipse.equinox.launcher_*.jar` in the `eclipse-ide-path/plugins/` folder, as shown here:



2. Open the debug configuration "Debug DBeaver Community" and modify the path in the parameters

`-configuration` and `-dev`.

Replace the path preceding `.metadata` with the path to your Eclipse workspace. It should eventually look like this:



3. Click OK and start the debugger. DBeaver should launch, allowing you to debug using IDEA.

## Additional information

### Cannot launch DBeaver debug on macOS or Linux

The initial launch arguments are set for Windows. You will need to adjust these manually for macOS or Linux. Open the debug configuration "Debug DBeaver Community" and locate these parameters:



```
-os
win32
-ws
win32
-arch
x86_64
```

You may need to replace them. Eclipse debug configurations automatically use the correct arguments. Open the Eclipse debug configuration you created while following [Develop in Eclipse](#), and press **Show Command Line**. This action opens a dialog window containing the necessary arguments. Copy these arguments and paste them into the IDEA debug configuration.

## Troubleshooting

If you encounter problems, do not hesitate to double-check your configuration and try again. If you're still experiencing issues, feel free to ask for help at [tickets](#).

# Eclipse extensions

---

## Table of contents

[DBeaver specific extensions](#)

[Popular 3rd party extensions for Eclipse and DBeaver](#)

[Install Process](#)

[Extension installation in CE version](#)

You can install a lot of optional extensions (plugins) in DBeaver. Most of the extensions can be found on the [Eclipse Marketplace](#) website.

## DBeaver specific extensions

---

- [Office formats support \(XLSX\)](#)
- [Vector graphics support \(SVG\)](#)
- SSHJ and advanced cryptography (since version 21 it is included in the base distribution)
- [Git support](#) - Git version control integration
- [SQL debugger](#)

## Popular 3rd party extensions for Eclipse and DBeaver

---

- [Darkest Dark theme](#) - the best Dark theme for DBeaver
- [Eclipse Color Theme](#) - if for some reason you do not like the Darkest Dark theme, you can use this one
- [Subversion support](#) - Subversion integration
- [Embedded Shell](#) - Allows you to run shell commands directly from DBeaver
- [Editor vertical indents](#) - Adds vertical indents to all text editors

- [CodeTogether](#) - Allows you to share the IDE and collaborate

## Install Process

In DBeaver EE you can use drag-n-drop from the Marketplace web site (see button **Install** ) in the DBeaver main window. This will launch the Marketplace installation wizard automatically. In the DBeaver Community or other DBeaver-based products which do not include marketplace clients, you can use the following instructions:

### Extension installation in CE version

---

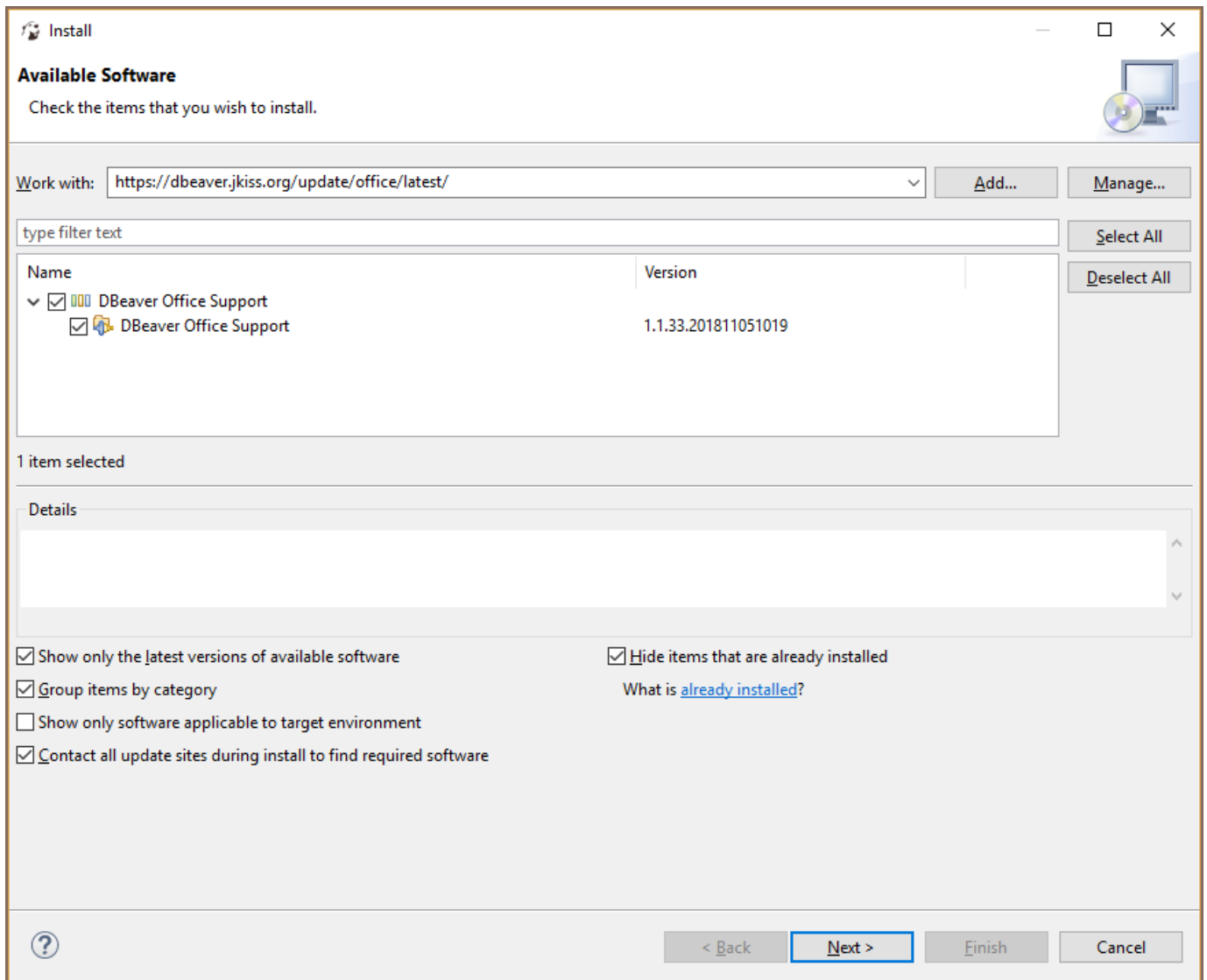
1. Copy URL of extension update site:

The screenshot shows the Eclipse Marketplace interface. The top navigation bar includes the Eclipse Marketplace logo and a breadcrumb trail: HOME / MARKETPLACE / TOOLS (1555) / DBEAVER - OFFICE INTEGRATION. The main content area is titled 'DBeaver - Office integration'. On the left, there are sections for 'MARKETS', 'SEARCH' (with a search bar and 'ADVANCED SEARCH' button), and 'FAVORITED BY'. The main listing for 'DBeaver - Office integration' shows a star rating of 13, an 'Install' button, and a description: 'Office integration with DBeaver (https://marketplace.eclipse.org/updates/office/latest)'. A dialog box titled 'Update site url' is open, displaying a list of Eclipse versions (Oxygen (4.7), Neon (4.6), Mars (4.5), Luna (4.4), Kepler (4.3), Photon (4.8), 2018-09 (4.9)) and a text input field containing the URL 'https://dbeaver.jkiss.org/update/office/latest/'. The dialog also includes icons for Linux, Windows, and Mac OS.

2. In the DBeaver main menu open **Help -> Install New Software**

3. Paste update site URL into **Work with** field and press **Enter**

4. Check items you wish to install (in most cases just all items)



The screenshot shows the 'Install' window with the title bar 'Install'. The main heading is 'Available Software' with the instruction 'Check the items that you wish to install.' Below this is a 'Work with:' field containing the URL 'https://dbeaver.jkiss.org/update/office/latest/' and buttons for 'Add...' and 'Manage...'. A search bar labeled 'type filter text' is present. A table lists available software with columns 'Name' and 'Version'. The table contains one entry: 'DBEaver Office Support' with version '1.1.33.201811051019'. To the right of the table are 'Select All' and 'Deselect All' buttons. Below the table, it says '1 item selected'. A 'Details' section is empty. At the bottom, there are several checkboxes: 'Show only the latest versions of available software' (checked), 'Group items by category' (checked), 'Show only software applicable to target environment' (unchecked), 'Contact all update sites during install to find required software' (checked), 'Hide items that are already installed' (checked), and a link 'What is already installed?'. The bottom right has buttons for '< Back', 'Next >', 'Finish', and 'Cancel'.

Install

**Available Software**  
Check the items that you wish to install.

Work with:   

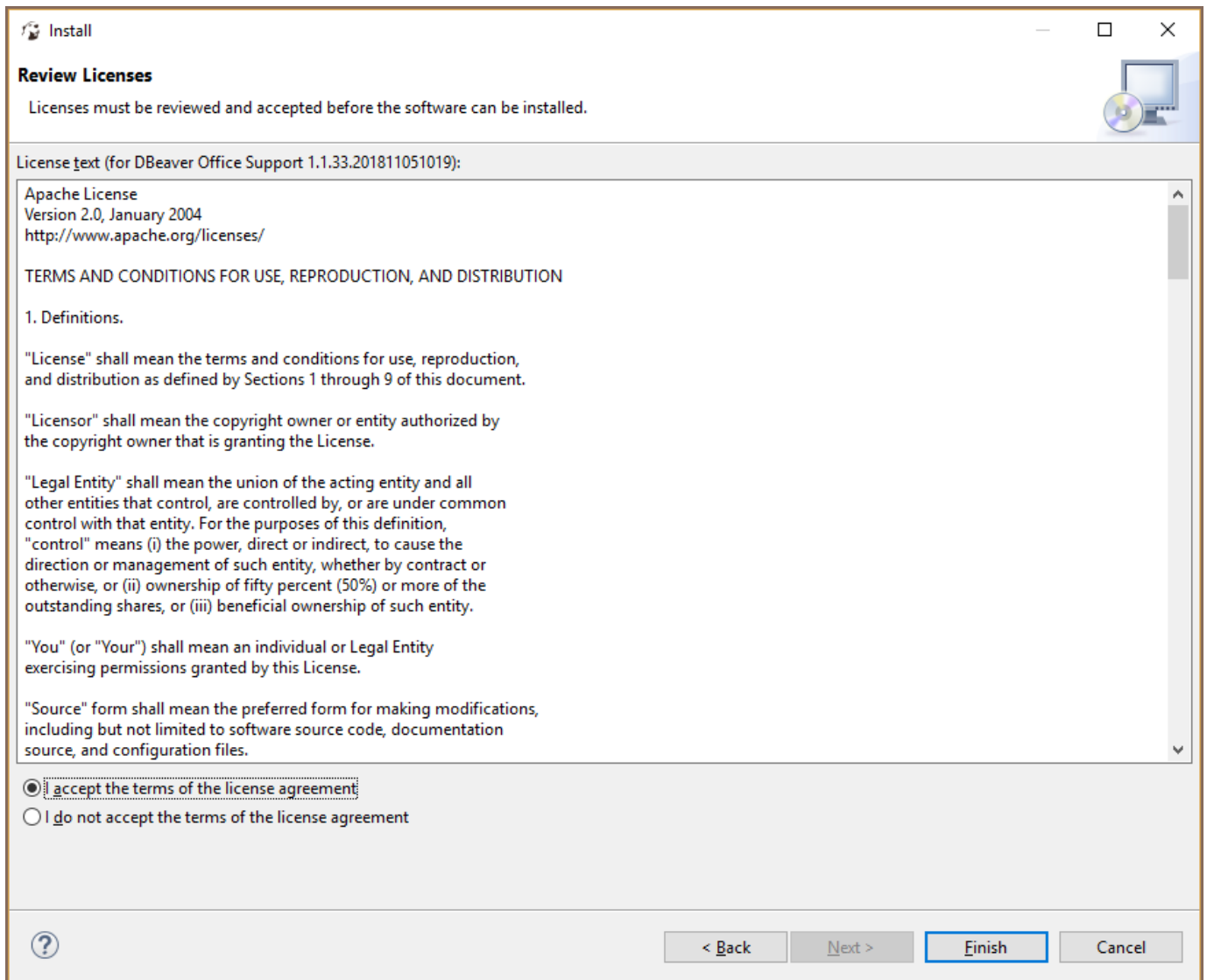
Name	Version
✓  DBEaver Office Support	
✓  DBEaver Office Support	1.1.33.201811051019

1 item selected

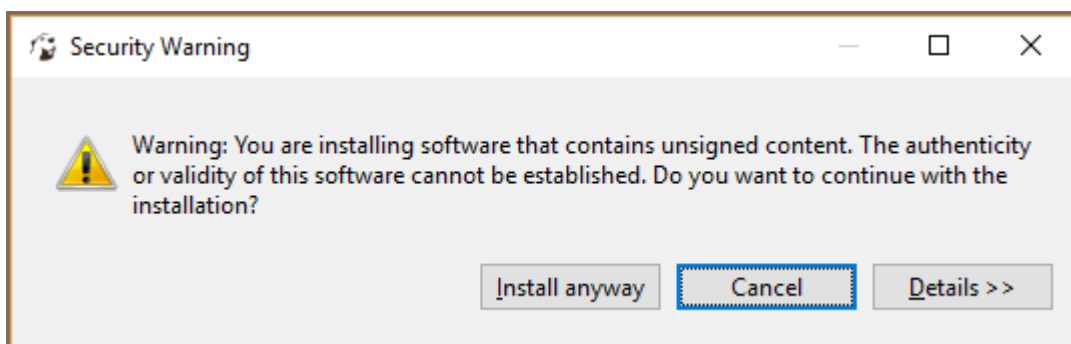
Details

☒ Show only the latest versions of available software ☒ Hide items that are already installed  
☒ Group items by category What is [already installed?](#)  
☐ Show only software applicable to target environment  
☒ Contact all update sites during install to find required software

5. Click Next. You may need to accept the extension license before installing



6. Some extensions may contain unsigned bundles. Only install such extensions if you really trust the author.



7. Click Next->Finish. The installation will take some time. Restart DBBeaver.

# Working with extension SVG format

---

## Table of contents

[Installation in CE version](#)

This extension supports data export in Office formats (XLS).

It is included in the EE version.

## Installation in CE version

1. Main menu `Help -> Install New Software`
2. Install SVG extension. You can do this from the online Eclipse Marketplace (<https://marketplace.eclipse.org/content/dbeaver-svg-format-support>). Or you can do this manually:
  - Paste `https://dbeaver.io/update/svg/latest/` into `Work with` field and press `Enter`
  - Check `DBeaver Extension->DBeaver Office Extension` item
3. Click Next->Finish. Restart DBeaver.

# Extension office for Data Transfer

---

## Table of contents

### [Installation in the CE version](#)

This extension supports data export in Office formats (XLS).

It is included in the EE version.

## Installation in the CE version

1. Main menu `Help -> Install New Software`
2. Install Office extension. You can do this from the online Eclipse Marketplace (<https://marketplace.eclipse.org/content/dbeaver-office-integration>). Or you can do this manually:
  - Paste `https://dbeaver.io/update/office/latest/` into `Work with` field and press `Enter`
  - Check `DBeaver Extension->DBeaver Office Extension` item
3. Click Next->Finish. Restart DBeaver.

Note: this works only if the DBeaver version matches the Office extension version (e.g. latest).



# Importing CA certificates from your local Java into DBeaver

---

**Note:** This article is relevant for you if you work behind the corporate firewall with a different root certificates set. In other cases, your issues can be caused by the incorrect [SSL Configuration](#).

## Table of contents

[Step 1 - locate your Java](#)

[Windows](#)

[macOS](#)

[Linux](#)

[Step 2 - Find the JRE in DBeaver installation](#)

[Step 3 - Copy the cacerts](#)

It's possible that your system administrator has installed a local Java and imported the required certificates to its keystore. We can use them to fix the issue.

## Step 1 - locate your Java

### Windows

Press Windows + R to open the Run window. Type `cmd` in the prompt and press OK. It will open the command prompt. In the command prompt, type the following and press Enter:

```
where java
```

### macOS

Open the Terminal and execute the following command:

```
/usr/libexec/java_home -V
```

### Linux

Open a terminal and execute the following command:

```
readlink -e /usr/bin/java
```

## Step 2 - Find the JRE in DBeaver installation

It's pretty easy. Just find the path where you installed DBeaver and open the `jre` folder there.

## Step 3 - Copy the cacerts

Open the folder with the Java you found in step 1. Locate the `cacerts` files under `/lib/security`, then copy-paste it into `<PATH_FROM_STEP_2>/lib/security`, replacing the old file. Restart DBeaver and you are ready to go.

# Contribute your code

---

## Table of contents

[Create pull requests](#)

[Pull request guidelines](#)

[Can I fix a bug by myself](#)

[What types of new features are acceptable](#)

[Code guidelines](#)

[Code style](#)

## Create pull requests

---

Before creating a pull request, you should create a ticket on our issue tracker. You can leave a comment saying that you are going to implement this feature (or fix a bug) by yourself. After the DBeaver dev/qa team replies to your comment, you can then start working on it.

You can use the [standard GitHub instruction](#). Generally, you need to create a fork of the `dbeaver/dbeaver` repository, create a new branch, commit your changes, and then create a pull request in the upstream repository.

Important: When you commit changes in your branch, add the ticket number in the commit comment.

Like this: `dbeaver/dbeaver#issue-number Initial commit for my super-duper feature`.

You can make any number of commits. We usually perform squash before merging changes in the main repository.

## Pull request guidelines

- Please avoid creating multiple pull requests for one issue.
- Please provide a detailed description of the affected functionality.
- Avoid force-pushing the commits in the PR branch, this makes reading commit history more difficult.
- If UI is affected, it is recommended to provide screenshots or video with a demonstration of the made changes

# Can I fix a bug by myself

---

If you think that a bug on our [board](#) has to be fixed and, for some reason, it is not in the nearest milestone, then you could try fixing it by yourself. It makes sense to ask about it in the ticket because we sometimes do not resolve tickets for a reason.

If you decide that you want to fix the bug and make a pull request (see above), the team will review it and write a PR review if necessary. If the PR is merged, then the fix will appear in the next version of the DBeaver community (which is released every two weeks).

## What types of new features are acceptable

---

Generally, you could suggest anything you think is useful in the database management tool. However, we usually do not implement features too specific for your development process or specific to your internal company processes.

Good options:

- Add new database or driver support
- Extend database metadata read/modify (e.g., add triggers read it a specific database)
- Add a new SQL generator
- Add new data export format
- Localize DBeaver interface (extend existing localization or add a new language. [Instruction](#))
- Add a new database-specific tool (e.g., table analyze for a specific database)

There are samples of all these features in our codebase.

TBD

## Code guidelines

---

The main rule: use the same codestyle, which is already used in a particular source file.

- Historically there are several slightly different code styles in our codebase. However if you modify an old file, it is better to use the same codestyle, which is used in this file.
- Do not reformat code or optimize all imports in a file you are changing, otherwise it is very hard to review your commits otherwise (due to a lot of changes)
- Use our automatic codestyle checks in PRs (every PR you make in the `dbeaver/dbeaver` repository will be checked, and you can see the report of the check list)

IntelliJ IDEA code style can be found at: <https://github.com/dbeaver/dbeaver-idea-project>

## Code style

---

# Localization

---

## Table of contents

[Localization](#)

[Environment setup](#)

[Localizing tools](#)

[Eclipse IDE](#)

[IntelliJ IDEA Community](#)

[Contribution](#)

[Push your changes](#)

[Properties](#)

[Important notice](#)

[Testing your changes](#)

[Pull request guidelines](#)

[Localization proposals](#)

[Changing user interface language](#)

# Localization

---

DBeaver uses the standard properties-based i18n model. All translatable resources reside in the \*.properties file. Each plugin (bundle) has its own set of resources. Almost all plugins have at least a `bundles.properties` resource. More extensive plugins have additional resources in the src folder. See the complete list of property files below.

bundle.properties contains an original string in the English language. All translated resources are placed in bundle\_XX.properties files where XX is a two-letter language code.

## Environment setup

- Clone DBeaver repository

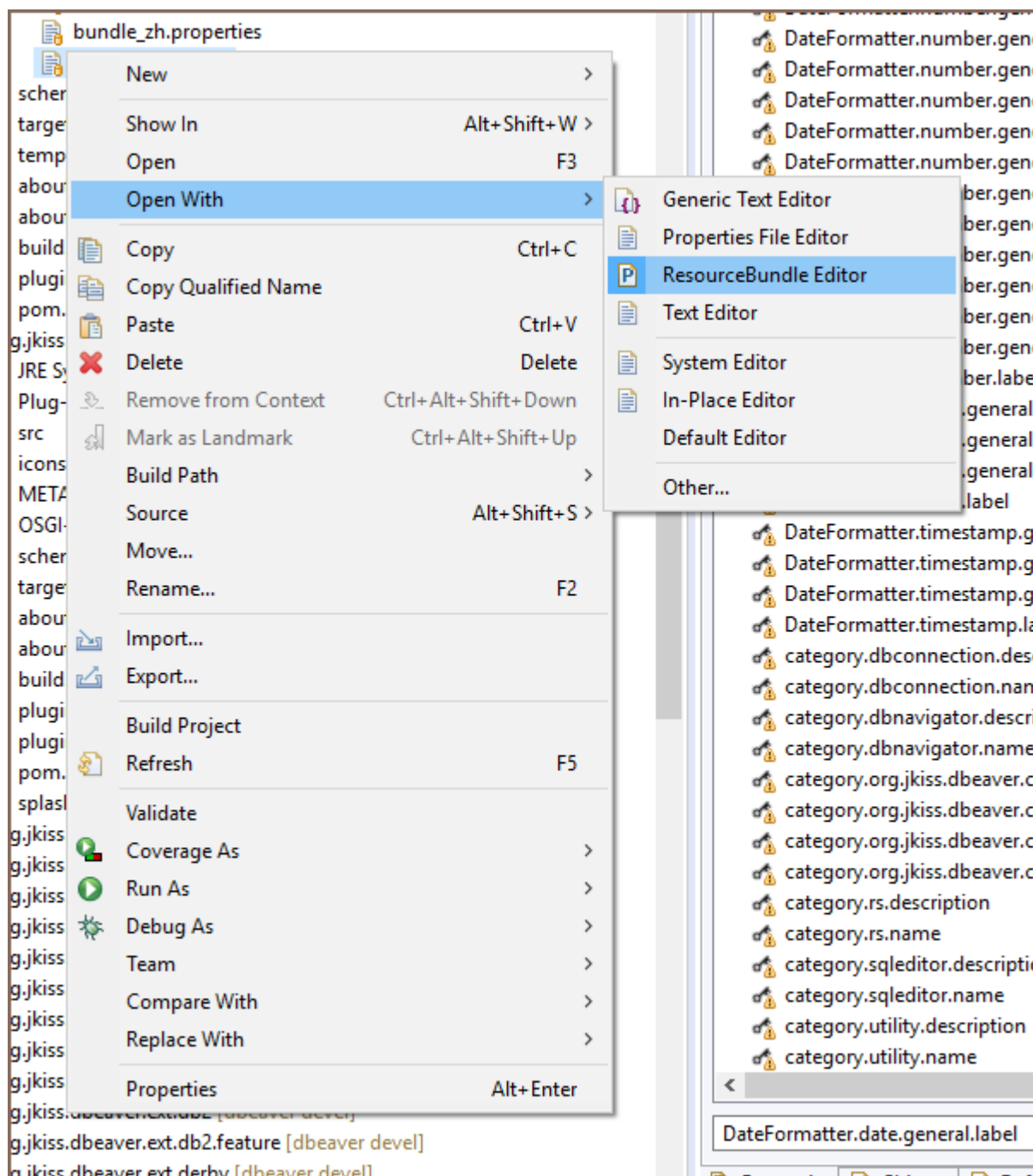
- Install [GitHub Desktop](#)
- In the top menu, select File -> Clone Repository
- In the appeared popup window, select the URL tab and paste DBeaver URL:  
`https://github.com/dbeaver/dbeaver.git`
- Press Clone

## Localizing tools

### Eclipse IDE

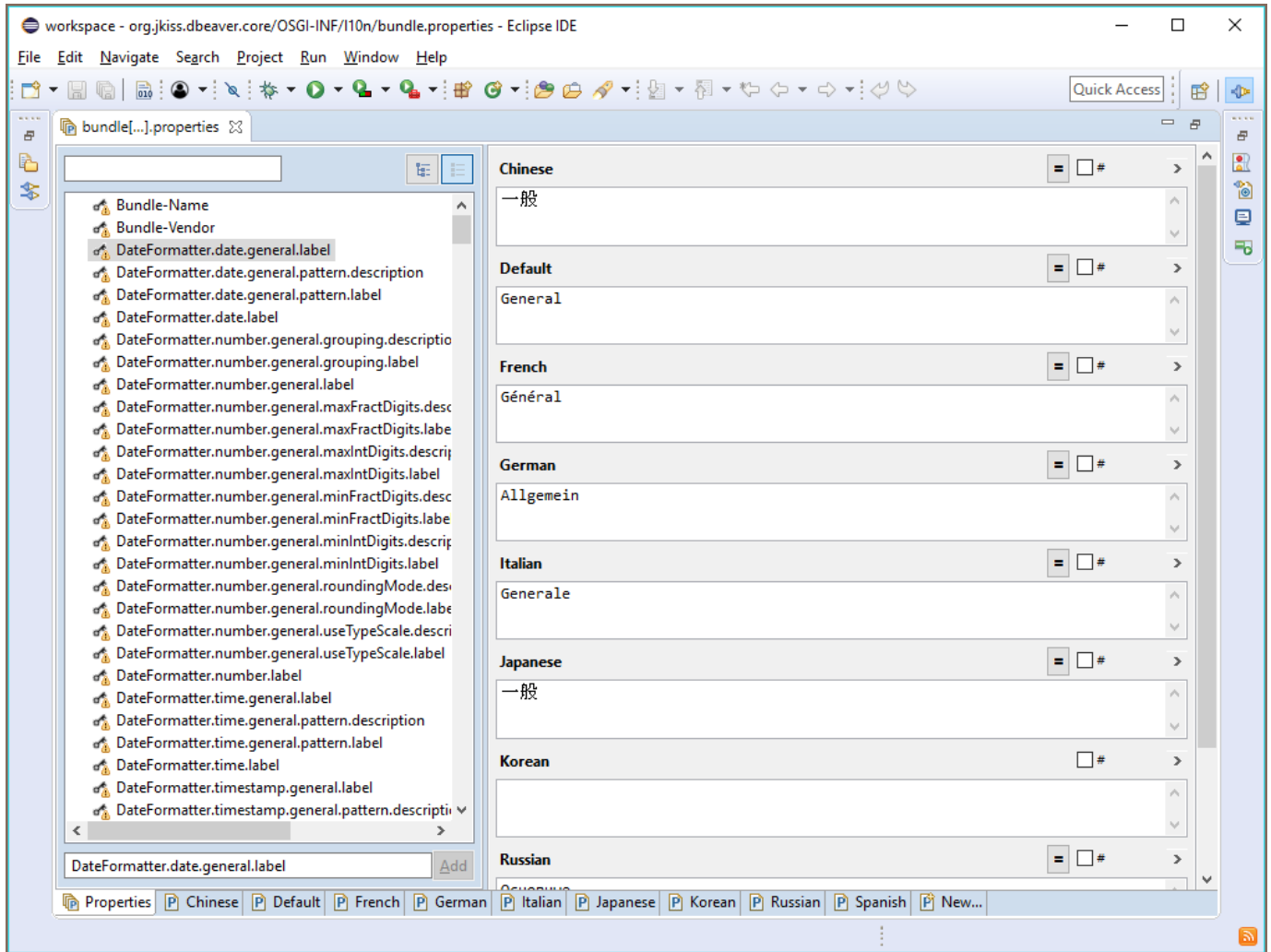
- Install Eclipse (any version, any package)
- Install [ResourceBundle Editor](#) plugin.
  - Main menu Help -> `Install New Software` ->  
`https://raw.githubusercontent.com/essiembre/eclipse-rbe/master/eclipse-rbe-update-site/site.xml`
- Main menu -> File -> Import... -> General -> Existing Projects into Workspace -> Browse
- Choose your DBeaver clone directory and import all projects

- Open some properties file (e.g. bundle.properties - [see below](#) ) in ResourceBundle editor:

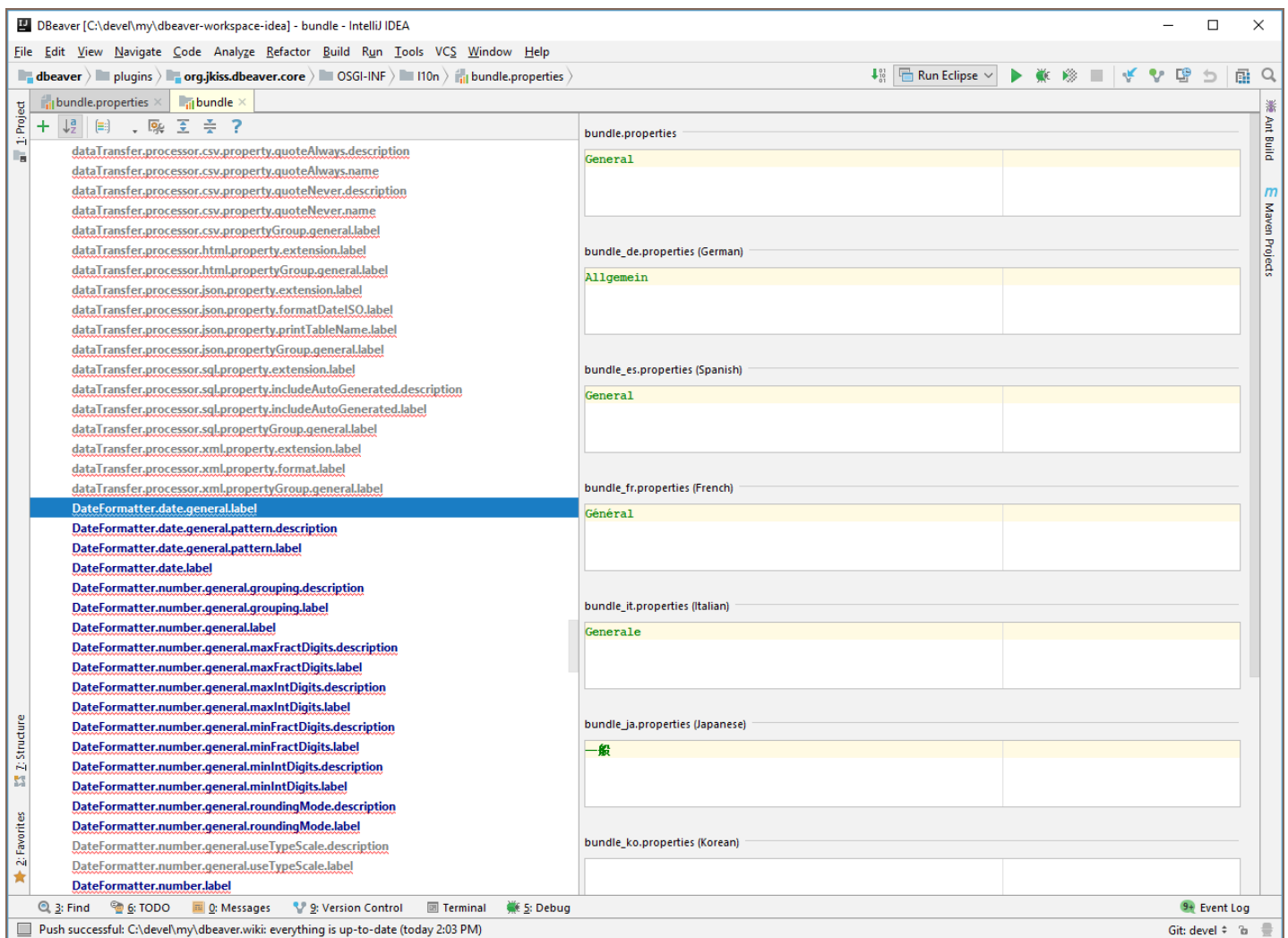




● Edit properties:



IntelliJ IDEA Community



Non-ASCII symbols are encoded to uppercase and lowercase hex sequences (for example, `\u00E3` and `\u00e3`). By default, only uppercase sequences are enabled in IntelliJ IDEA. Before starting your work, Please set the `idea.native2ascii.lowercase` property in the `idea.properties` file to `true` to avoid unintended changes.

## Contribution

Please note that according to the [Contribution guide](#), it's required to add the issue number to the commit message. If there is no such issue, then you should create one.

## Push your changes

- Open GitHub Desktop
- At the top, select Current branch and press the New Branch button
- Create a new branch with a name related to I10n

- Commit to the new branch
- Publish brunch/push (right button at the top)
- Fetch origin (right button at the top)
- In the main menu, select Branch -> Create Pull Request
- On the opened GitHub page, add some meaningful text for your pull request name and press the Create pull request button
- Return to GitHub Desktop and switch to devel branch

or

Create a Pull Request with your changes (in branch `devel` ) <https://help.github.com/articles/creating-a-pull-request-from-a-fork/>

## Properties

Module	Purpose	File
Core	Commands, properties	plugins/org.jkiss.dbeaver.core/OSGI-INF/l10n/bundle.properties
Core	Messages, UI strings	plugins/org.jkiss.dbeaver.core/src/org/jkiss/dbeaver/core/CoreResources.properties
API	Properties	plugins/org.jkiss.dbeaver.model/OSGI-INF/l10n/bundle.properties
API	Messages	plugins/org.jkiss.dbeaver.model/src/org/jkiss/dbeaver/model/messages/ModelResources.properties
UI	Properties	plugins/org.jkiss.dbeaver.ui/OSGI-INF/l10n/bundle.properties
UI	Messages	plugins/org.jkiss.dbeaver.ui/src/org/jkiss/dbeaver/bundle/UITMessages.properties
Application (Standalone)	Commands, properties	plugins/org.jkiss.dbeaver.ui.app.standalone/OSGI-INF/l10n/bundle.properties
Application (Eclipse)	Commands, properties	plugins/org.jkiss.dbeaver.ui.app.eclipse/OSGI-INF/l10n/bundle.properties

Data Transfer	Commands, properties	plugins/org.jkiss.dbeaver.ext.data.transfer/OSGI-INF/l10n/bundle.properties
Data Transfer	Messages	plugins/org.jkiss.dbeaver.ext.data.transfer/src/org/jkiss/dbeaver/tools/transfer/internal/DTMessages.properties
ERD	Commands, properties	plugins/org.jkiss.dbeaver.ext.erd/OSGI-INF/l10n/bundle.properties
ERD	Messages	plugins/org.jkiss.dbeaver.ext.erd/src/org/jkiss/dbeaver/ext/erd/ERDResources.properties
SSH	Commands, properties	plugins/org.jkiss.dbeaver.net.ssh/OSGI-INF/l10n/bundle.properties
SSH	Messages	plugins/org.jkiss.dbeaver.net.ssh.ui/src/org/jkiss/dbeaver/ui/net/ssh/SSHUIMessages.properties
Generic driver	Properties	plugins/org.jkiss.dbeaver.ext.generic/OSGI-INF/l10n/bundle.properties
Generic driver	Messages	plugins/org.jkiss.dbeaver.ext.generic/src/org/jkiss/dbeaver/ext/generic/GenericResources.properties
MySQL	Properties	plugins/org.jkiss.dbeaver.ext.mysql/OSGI-INF/l10n/bundle.properties
MySQL	Messages	plugins/org.jkiss.dbeaver.ext.mysql/src/org/jkiss/dbeaver/ext/mysql/MySQLResources.properties
..	..	The same for Oracle (ext.oracle), DB2 (ext.db2), Exasol (ext.exasol), PostgreSQL (ext.postgresql), and SQL Server (ext.mssql)

## Important notice

### 1. Using placeholders in localization files

When localizing messages that include segments from the running application, you should use text placeholders instead of manually concatenating strings in the code.

A placeholder is a special marker represented as `{N}`, where `N` is a number starting from `0`. These placeholders are later replaced with provided values.

```
` java // Will result in "This is my text: hello from DBeaver" NLS.bind("This is my text: {0} from {1}", "hello",
"DBeaver"); `
```

## 2. Dealing with single quotes

If you're editing the text manually and your message contains single quotes, you must double them:

- Wrong: `Can't delete '{0}'`
- Right: `Can''t delete ''{0}''`

## Testing your changes

Once you have changed one of these resources, you might want to test your changes. You can:

- Run DBeaver directly from [Eclipse workspace](#)
- [Build DBeaver from source](#) and run executable

## Pull request guidelines

- Please avoid creating multiple pull requests for one issue.
- Avoid force-pushing the commits in the PR branch. This makes reading commit history more complex.

## Localization proposals

[Brazilian Portuguese Standardization proposals by @brlarini](#)

## Changing user interface language

See [Changing interface language](#)

# Brazilian Portuguese standardization proposals

---

## Table of contents

[dbeaver Portuguese and Brazilian guidelines](#)

## dbeaver Portuguese and Brazilian guidelines

### Sugestões de padronização para tradução do DBeaver para o português brasileiro.

O intuito deste explicativo de é manter consistência na tradução do DBeaver para o português brasileiro. Nada aqui é imposição, são apenas sugestões. Se tiver ideias melhores de como expressar uma tradução, sinta-se à vontade para alterar o que for necessário. Apenas atualize as definições abaixo conforme sua nova tradução.

Em várias ocasiões a tradução de uma frase ou termo não soa natural quando passada do inglês para o português, de forma a se encaixar no espaço disponível para exibição do texto. Isso inclui os termos técnicos usados por profissionais da área de TI. Parece haver um consenso de que alguns termos devem ser mantidos em inglês por soarem mais natural, mesmo no meio de uma frase em português. Por isso preferi manter alguns termos em inglês, pois pelo que pude perceber é o que os usuários ou profissionais preferem. **Não faz sentido traduzir para termos em português que ninguém usa.**

Obviamente, opiniões sobre o que é mais adequado podem variar, portanto se acharem que algo deve ser mudado (e que há um consentimento da maioria), mude e inclua abaixo para manter a consistência e manter futuros tradutores informados.

Tudo isso vale para a tradução para o português brasileiro. Se você for de outro país que usa o português e que acha que essas regras não lhe fazem sentido, por favor inclua sua tradução em uma linha separada.

**Termos técnicos (incluindo palavras reservadas) mantidos em inglês, apesar de existirem softwares que usam em português. Atente-se que as mesmas palavras podem ser usadas no meio de uma frase mas que não têm ligação com o termo técnico. Nesses casos use a tradução para o português.**

- Commit - Enviar/Aplicar/Comprometer

- Rollback - Reverter/Retroceder
- Procedure - Procedimento
- Constraint - Restrição
- Package - Pacote
- Trigger - Gatilho
- View - Visão
- Unique - Único
- Instead - Em vez
- When - Quando
- Hold - Segurar/manter
- Tuple - Énuplo
- Freeze - Congelar
- Dump - Descarga
- Stored procedure - Procedimentos armazenados
- Role - Papel/Função
- Job - Trabalho (diferente de Task/Tarefa)
- String - Cadeia de caracteres
- Schema - Esquema
- Grant - Concessão
- Timeout - Tempo-limite
- Host - Hospedeiro
- Grid - Grade

- Bucket - Balde
- Breakpoint - Ponto de interrupção
- Collation - Agrupamento/colação/verificação
- Add-in - Extensão

**Termos que às vezes soam melhor em inglês, às vezes melhor em português, dependendo do contexto.**

- Template - Modelo
- Link - Vínculo
- Download - Baixar (verbo), download (substantivo)
- Upload - Enviar (verbo), upload (substantivo)
- Scan - Escanear, varrer
- Index - Indexar (verbo), índice (substantivo)
- Array - Matriz
- Inline - Integrado, alinhado
- Verbose - Prolixo
- Case-sensitive - Sensível à caixa (diferente de Capitalization/Capitalização) - mas eventualmente pode ser usado a versão em português quando encaixado melhor no contexto

**Abreviações mantidas na versão em inglês.**

- PK - CP (chave primária)
- FK - CE (chave estrangeira)
- SQL - LCE (linguagem de consulta estruturada)
- DDL - LDD (linguagem de definição de dados)

**Termos usados exclusivamente em inglês**



- Script
- Log
- Driver
- Bootstrap
- Shell
- Keep-alive
- Pop-up
- Deadlock
- Cluster
- Namespace
- Tablespace
- Cache
- Back-end
- Buffer
- Backup

**Traduções ambíguas. Sugere-se que mantenha o padrão em português como descrito abaixo.**

- Database - Banco de dados (evitar base de dados; se precisar encurtar use apenas banco)
- Tab - Aba (evitar guia; se refere às abas de pastas, não à tecla Tab)
- Background - Segundo-plano (evitar plano-de-fundo)
- Reset - Redefinir (evitar resetar)
- Completion - Completamento ou completação (compleção aparentemente não é uma palavra do português)
- Lock - Trava (em vez de bloqueio)

- Block - Bloqueio
- Boolean - Booleano (em vez de boleano)

#### **Observações.**

- Default - Padrão
- Pattern - Padrão
- Standard - Padrão

# Unit tests

---

## Table of contents

### [Using UNIX shell](#)

DBeaver uses tycho-surefire to run JUnit tests in the OSGi environment. This fact may slightly complicate the process of writing and running unit tests for DBeaver. Let's take a look at possible ways to do that.

## Using UNIX shell

Suppose we introduced some changes to our code and want to run the tests. It is actually quite easy:

```
$ cd /path/to/project/
$ mvn clean install
```

This compiles the project and runs unit tests.

If we only want to run unit tests:

```
$ cd test
$ mvn clean verify
```

The unit tests are separated into bundles. If you only want to run tests from a specific bundle, navigate to that bundle and run `mvn clean verify` there.

It is also worth pointing out that you need to recompile the whole project with `mvn clean install` after every change in the project's codebase before running unit tests, otherwise, you will be running tests against the previous version and it may lead to all sorts of confusion.

