DBeaver user guide

Table of contents

General User Guide

Database Management

Customizing DBeaver

Databases

Troubleshooting

Admin Guide

Other

Installation

The installation process depends on the distribution type and your Operational System - see https://dbeaver.io/download/

Windows / MacOS Installer

The installer distribution is the recommended way to install DBeaver on Windows and MacOS X. It is the most lightweight method due to the advanced archiving technology that is uses and that is not available in the case of installation using ZIP archive. Besides this, the installer automatically upgrades DBeaver to the new version, if a previous version is already installed. To install DBeaver, run the installer executable and follow the instructions in its screens.

NOTE:

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

ZIP Archive

When installing DBeaver manually, without using an installer:

1. Install Java 1.8.
2. 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.
3. 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.

MacOS DMG

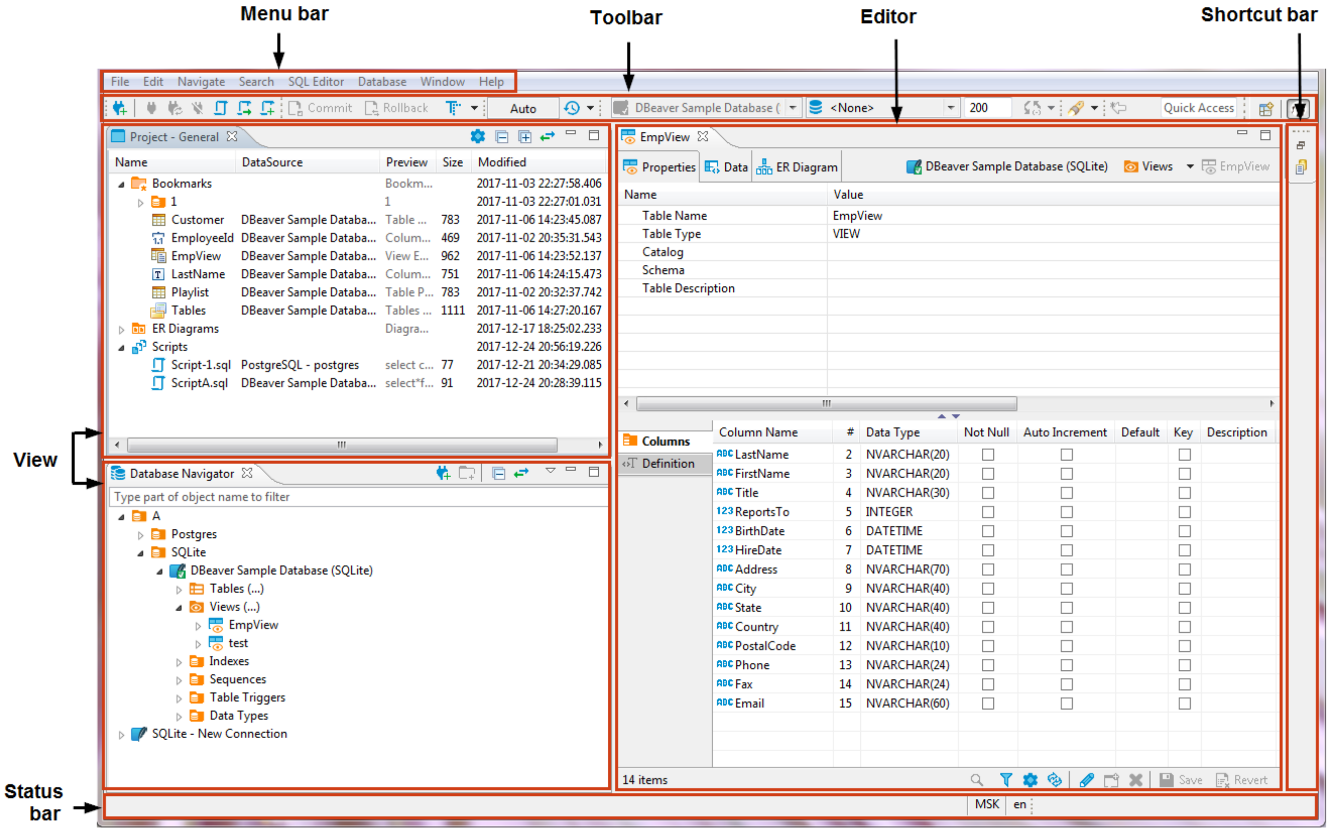
You can try DMG or ZIP archive if you cannot use PKG installer for some reason.
To install DBeaver on MacOS, just drag-and-drop the DMG archive to your disk.
NOTE: DBeaver requires Java 1.8 to be installed, so you need to install JDK 1.8+ prior to installing DBeaver.

You can also use brew:

```
brew install Caskroom/cask/java
```

Application Window Overview

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



Name	DataSource	Preview	Size	Modified
Bookmarks		Bookm...		2017-11-03 22:27:58.406
1		1		2017-11-03 22:27:01.031
Customer	DBeaver Sample Databa...	Table ...	783	2017-11-06 14:23:45.087
EmployeeId	DBeaver Sample Databa...	Colum...	469	2017-11-02 20:35:31.543
EmpView	DBeaver Sample Databa...	View E...	962	2017-11-06 14:23:52.137
LastName	DBeaver Sample Databa...	Colum...	751	2017-11-06 14:24:15.473
Playlist	DBeaver Sample Databa...	Table P...	783	2017-11-02 20:32:37.742
Tables	DBeaver Sample Databa...	Tables ...	1111	2017-11-06 14:27:20.167
ER Diagrams		Diagra...		2017-12-17 18:25:02.233
Scripts				2017-12-24 20:56:19.226
Script-1.sql	PostgreSQL - postgres	select c...	77	2017-12-21 20:34:29.085
ScriptA.sql	DBeaver Sample Databa...	select*f...	91	2017-12-24 20:28:39.115

Type part of object name to filter

- A
  - Postgres
  - SQLite
    - DBeaver Sample Database (SQLite)
      - Tables (...)
      - Views (...)
        - EmpView
        - test
      - Indexes
      - Sequences
      - Table Triggers
      - Data Types
- SQLite - New Connection

Name	Value
Table Name	EmpView
Table Type	VIEW
Catalog	
Schema	
Table Description	

Column Name	#	Data Type	Not Null	Auto Increment	Default	Key	Description
LastName	2	NVARCHAR(20)	□	□		□	
FirstName	3	NVARCHAR(20)	□	□		□	
Title	4	NVARCHAR(30)	□	□		□	
ReportsTo	5	INTEGER	□	□		□	
BirthDate	6	DATETIME	□	□		□	
HireDate	7	DATETIME	□	□		□	
Address	8	NVARCHAR(70)	□	□		□	
City	9	NVARCHAR(40)	□	□		□	
State	10	NVARCHAR(40)	□	□		□	
Country	11	NVARCHAR(40)	□	□		□	
PostalCode	12	NVARCHAR(10)	□	□		□	
Phone	13	NVARCHAR(24)	□	□		□	
Fax	14	NVARCHAR(24)	□	□		□	
Email	15	NVARCHAR(60)	□	□		□	

14 items

Menu Bar

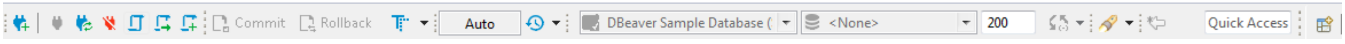
By default, the menu bar contains the following menus:

- **File** menu contains menu items for the creation of files, folders, projects, database connections, database projects, and ER diagrams as well as Import and Export items.
- **Edit** menu contains global commands like Cut, Copy, Paste, and Delete targeted at the active element.
- **Navigate** menu allows navigating through scripts and database objects.
- **Search** menu provides options to search among files, database objects and across data.
- **SQL Editor** menu is for opening SQL Editor and managing its appearance.
- **Database** menu allows managing database drivers, connections and transactions as well as reconnecting to and disconnecting from a database.
- **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.
- **Help** menu contains links to information and help resources, as well as 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 basic and most frequently used commands:



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

You can customize the toolbar, for this, go to **Window -> Customize Perspective -> Tool Bar Visibility** tab.

You can hide or show the toolbar in the application window. To do it, on the Window menu, click **Appearance -> Hide Toolbar / Show Toolbar**.

Shortcut Bar

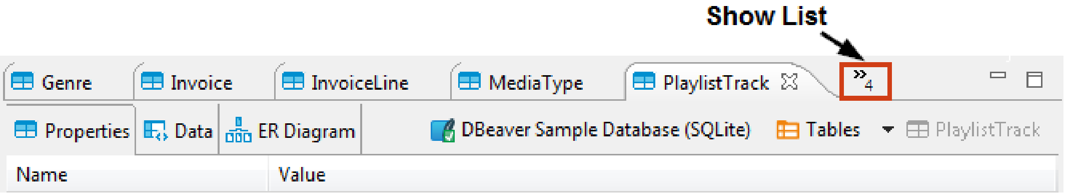
There are two shortcut bars - on the left and on the right side of the workspace zone. Shortcut bars host shortcuts of views and editors and appear if at least one view or editor is minimized, otherwise they are hidden.

Workspace: Views and Editors

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

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

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



There can be several views and editors simultaneously open in the workspace but only one of them 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 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 title bar of the view, then drag and drop it onto the desired position.

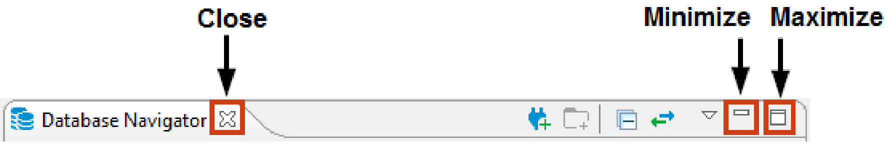
You can resize the view and editor windows. To resize, place the cursor to the border of the window to see it change to a double-ended arrow, then click and drag the border to the needed size.

To close a view or editor, click the Close button or right-click the title bar of the view / editor and 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)

Maximizing, Minimizing and Restoring 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.

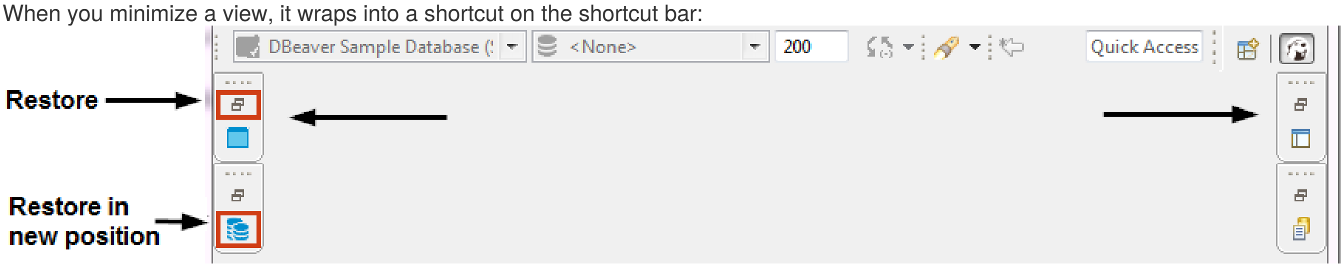
To maximize a view or editor to the size of the whole workspace, do one of the following:

- 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 click 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 on 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.

Views

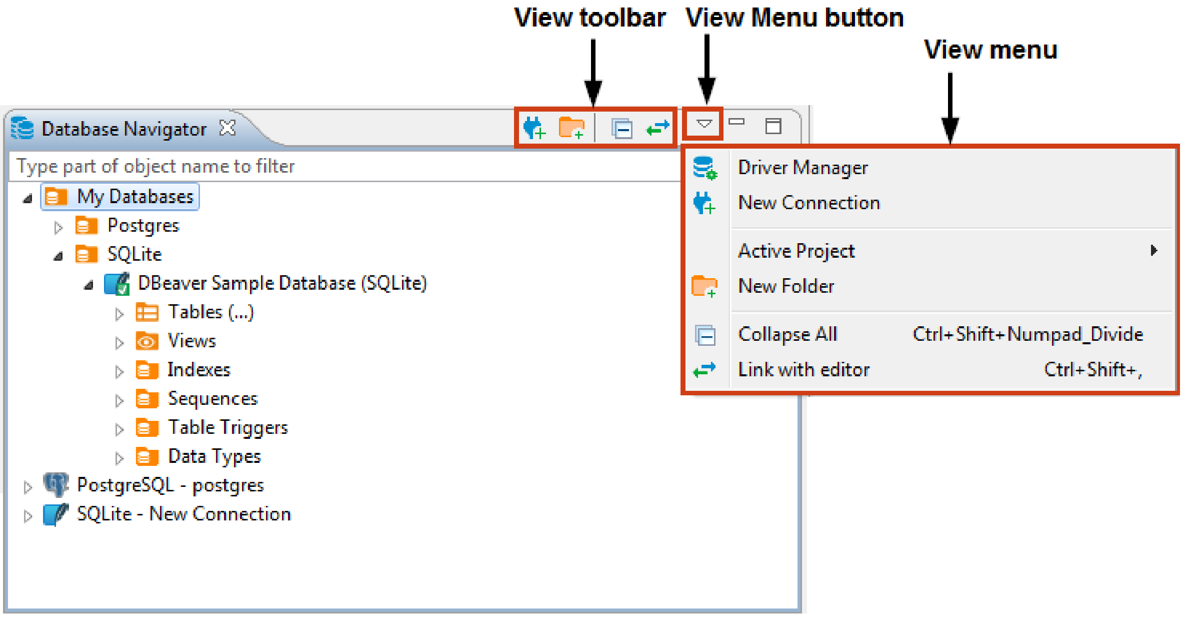
Views are windows within the workspace that provide presentations and ways to navigate the information. The main views in DBeaver are: Database Navigator, Projects and Project Explorer.

To open a view:

- On the Window menu, click **Show View** and then, on the submenu, click the name of the view. Click **Other** if the view is not visible on the submenu.
- For Database Navigator, Projects, and Project Explorer views, on the Window menu, just click the name of the view.

Some views open on demand, for example the Search view opens to show search results.

Views provide their own toolbar and menu:



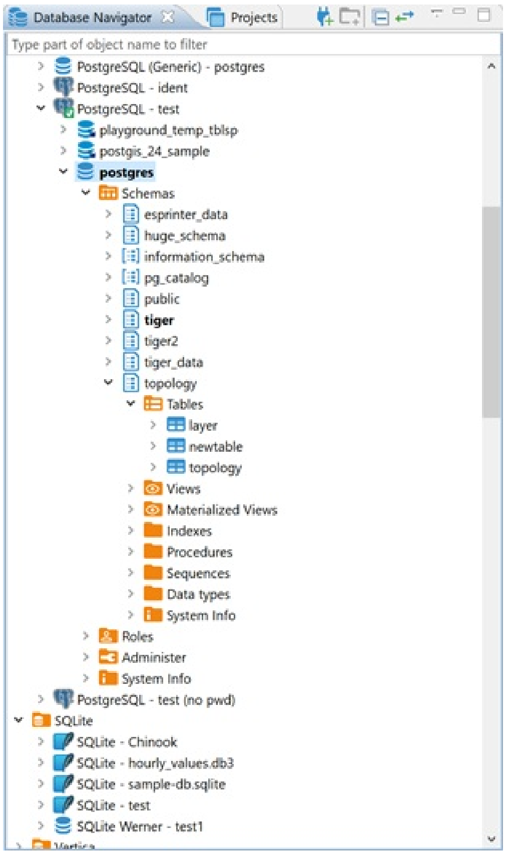
To open the view menu, click the View Menu button in the upper-right corner of the view`s title bar, next to the Minimize button.

The toolbar contains buttons applicable to the objects displayed in the view. The set of enabled and disabled buttons depends on the object in focus.

Views also provide context menus for objects they display. To open a context menu for an object, right-click the object.

Database Navigator

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.



- PostgreSQL (Generic) - postgres
- PostgreSQL - ident
- PostgreSQL - test
  - playground_temp_tblsp
  - postgis_24_sample
  - postgres
    - Schemas
      - esprinter_data
      - huge_schema
      - information_schema
      - pg_catalog
      - public
      - tiger
      - tiger2
      - tiger_data
      - topology
        - Tables
          - layer
          - newtable
          - topology
        - Views
        - Materialized Views
        - Indexes
        - Procedures
        - Sequences
        - Data types
        - System Info
    - Roles
    - Administer
    - System Info
- PostgreSQL - test (no pwd)
- SQLite
  - SQLite - Chinook
  - SQLite - hourly_values.db3
  - SQLite - sample-db.sqlite
  - SQLite - test
  - SQLite Werner - test1
- Vertica

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 creating, editing and deleting 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	Enabled when at least one editor is open, otherwise disabled - highlights the object in the tree that has its editor open

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 context menu items for all types of objects that may appear in the tree. Note that the presence or absence of 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
Paste	Inserts the copied object into a selected folder - most convenient for copy-pasting 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
View [objects]	- For objects that DBeaver can create and delete, opens the object in a separate editor - For objects that DBeaver cannot create and delete, opens the object in a separate viewer
Edit [object]	- For objects that DBeaver can create and delete, opens the object in a separate editor - For objects that DBeaver cannot create and delete, opens the object in a separate viewer
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): - 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	- 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	Opens a submenu on which you can select the type of SQL query to generate: - SELECT - INSERT - UPDATE - DELETE - MERGE - DDL Clicking one of the items (for example INSERT) generates a relevant query in a separate window.

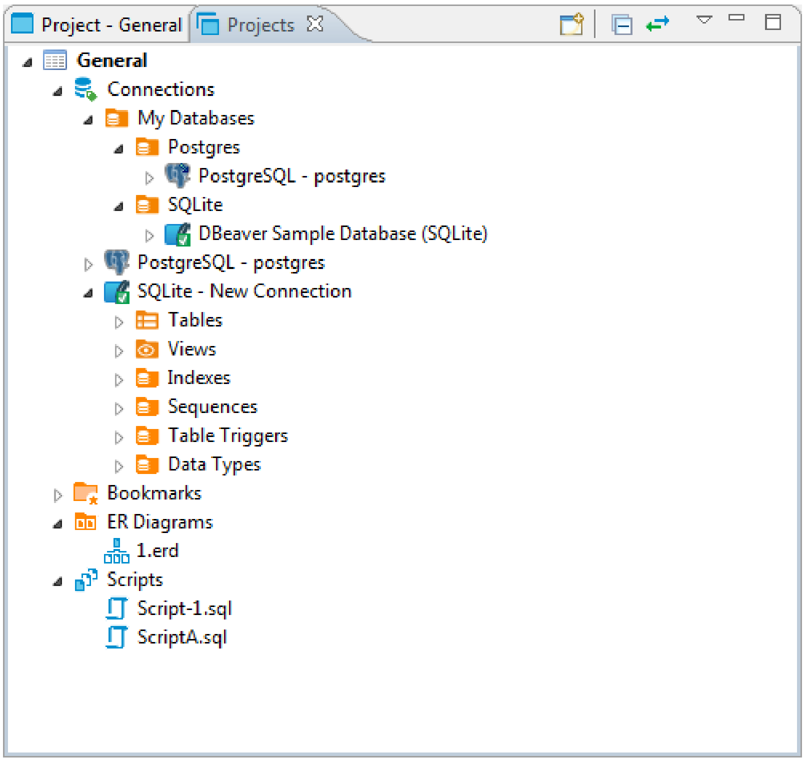
Menu item	Description
Export Table Data	Opens the Data Transfer wizard that helps you select a format and export table data
Import Table Data	Opens a window with existing database connections in which you can select a table to import data from
Tools	Opens a submenu that provides tools for database backup and restore, vacuum, etc.

For information on how to filter database objects in Database Navigator, please see the Filter Database Objects article.

Projects View

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** (or use `ALT+W+P` shortcut).



For information on how to change the view layout, please see the Application Window Overview article.

The projects are organized into a tree and all have the same high-level structure:

- **Connections** – repeat the content of the Database Navigator view for this project. You can perform the same actions over the objects of the databases as in Database Navigator.
- **Bookmarks** – contains bookmarks – shortcuts to database objects, see …
- **ER Diagrams** - contains ER diagrams that you can drag-and-drop here from other folders
- **Scripts** – contains scripts that you can drag-and-drop here from other folders

The Projects view provides a toolbar and View menu which contain generic items. Each object in the tree has its own context menu.

To open the view menu of the Projects view, click the View Menu button (▽) in the upper-right corner of the window. The view menu contains the following items:

Icon	Item	Description
□	Create Project	Opens the Create Project wizard
↻	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	- Enabled when at least one editor is open, otherwise disabled - Highlights the object in the tree that has its editor open

The toolbar is located in the title bar of the window, its buttons duplicate the view menu items except for the **Refresh Projects** one.

To open the context menu for an object in the tree, right-click the object. For information about context menu items of all objects under the **Connections** node of the tree, please see Database Navigator. The context menus of other nodes in the tree contain some basic items for copy-pasting, renaming, deleting objects, managing their properties, creating folders, etc.

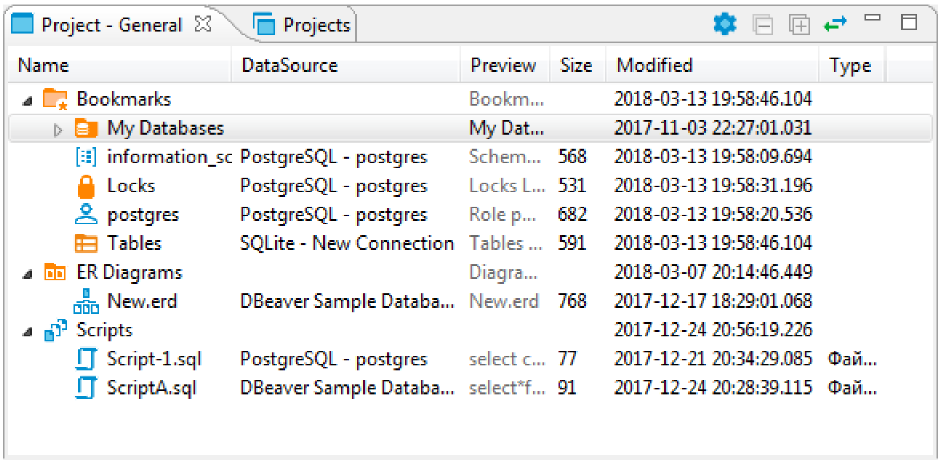
- The **Set Active Project** menu item (for a project root node) makes the project active, that is visible in the Database Navigator.

- The **Link File (SQL Script)** and **Link Folder** menu items allow creating links to files and folders in the file system.

For information about managing projects, please see 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**.



Name	DataSource	Preview	Size	Modified	Type
Bookmarks		Bookm...		2018-03-13 19:58:46.104	
My Databases		My Dat...		2017-11-03 22:27:01.031	
information_sc	PostgreSQL - postgres	Schem...	568	2018-03-13 19:58:09.694	
Locks	PostgreSQL - postgres	Locks L...	531	2018-03-13 19:58:31.196	
postgres	PostgreSQL - postgres	Role p...	682	2018-03-13 19:58:20.536	
Tables	SQLite - New Connection	Tables ...	591	2018-03-13 19:58:46.104	
ER Diagrams		Diagra...		2018-03-07 20:14:46.449	
New.erd	DBeaver Sample Databa...	New.erd	768	2017-12-17 18:29:01.068	
Scripts				2017-12-24 20:56:19.226	
Script-1.sql	PostgreSQL - postgres	select c...	77	2017-12-21 20:34:29.085	Фай...
ScriptA.sql	DBeaver Sample Databa...	select*f...	91	2017-12-24 20:28:39.115	Фай...

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**	- 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.

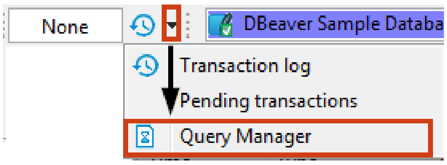
Query Manager

Query Manager is a view that shows the history of all SQL queries that DBeaver has executed during the current session.

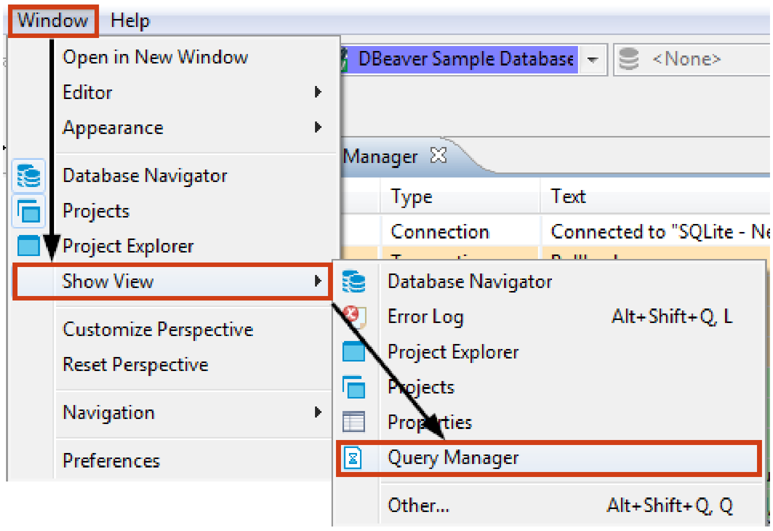
NOTE: DBeaver EE version persists all executed queries in the internal database so execution history is available after the program restart.

To open the Query Manager, do one of the following:

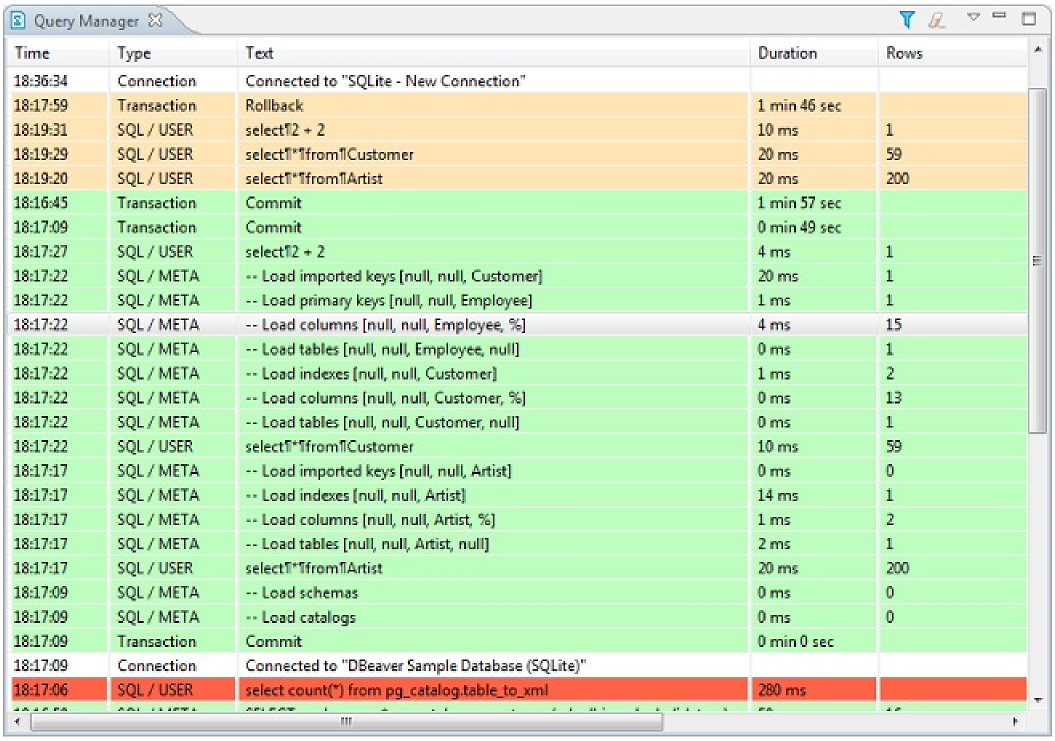
- Click the arrow next to the **Transaction Log** button in the toolbar and then click **Query Manager** on the dropdown menu:



- 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.):



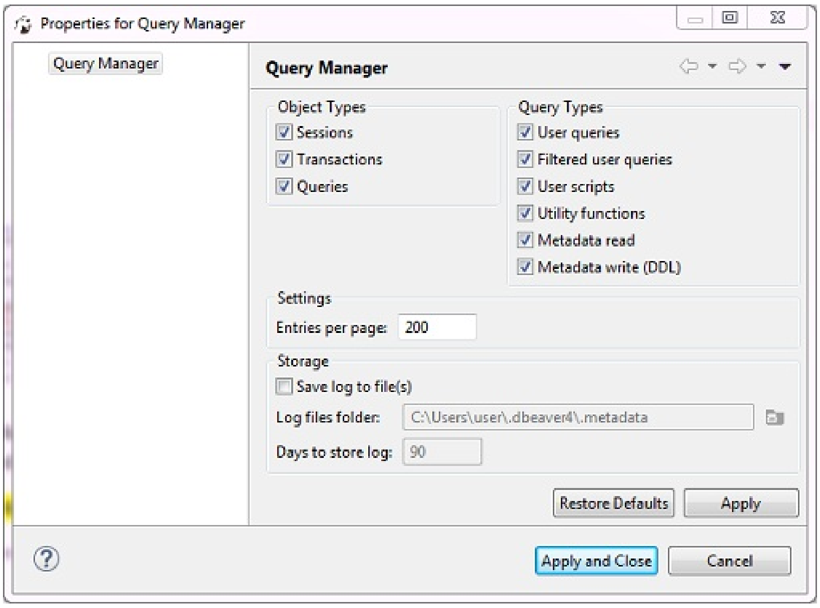
Time	Type	Text	Duration	Rows
18:36:34	Connection	Connected to "SQLite - New Connection"		
18:17:59	Transaction	Rollback	1 min 46 sec	
18:19:31	SQL / USER	select¶2 + 2	10 ms	1
18:19:29	SQL / USER	select¶*¶from¶Customer	20 ms	59
18:19:20	SQL / USER	select¶*¶from¶Artist	20 ms	200
18:16:45	Transaction	Commit	1 min 57 sec	
18:17:09	Transaction	Commit	0 min 49 sec	
18:17:27	SQL / USER	select¶2 + 2	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 you can specify 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.

Query Manager Properties

To manage the look of the Query Manager, filter the entries, and modify storage settings, click the **Set query manager filter** button () in the view`s toolbar. The Properties for Query Manager window opens:



Properties for Query Manager

Query Manager

Query Manager

Object Types
- ☑ Sessions
- ☑ Transactions
- ☑ Queries

Query Types
- ☑ User queries
- ☑ Filtered user queries
- ☑ User scripts
- ☑ Utility functions
- ☑ Metadata read
- ☑ Metadata write (DDL)

Settings

Entries per page: 200

Storage
- □ Save log to file(s)

Log files folder: C:\Users\user\.dbeaver4\.metadata

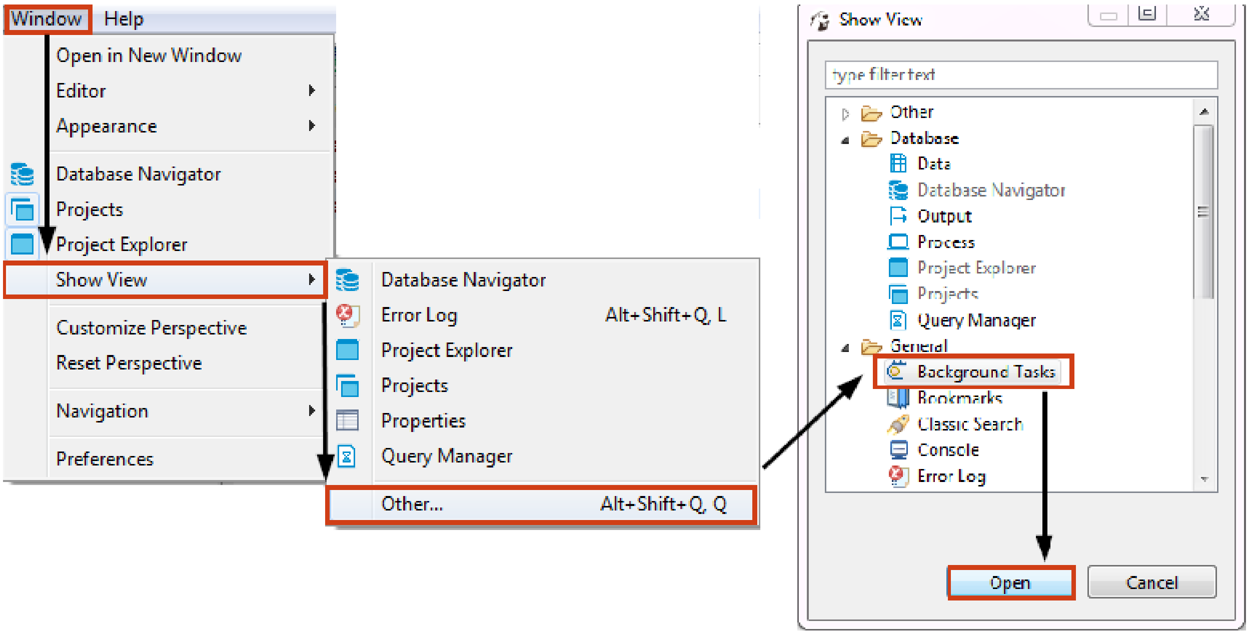
Days to store log: 90

- To filter entries by object type, select or clear the checkboxes in the **Object Types** section. To filter entries by query type, select clear the checkboxes in the **Query Types** section.
- To change the number of entries displayed per page, enter the new number in the **Entries per page** 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**.

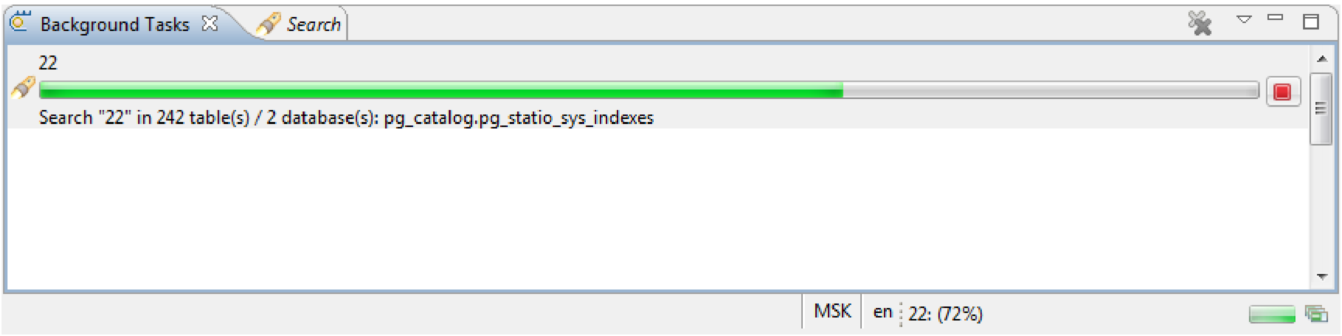
Background Tasks

You can open the Background view from the main menu - click **Window -> Show View -> Other**, then in the Show View window, expand the **General** folder, click **Background Tasks** and then click **Open**:



You can also open the Background Tasks view from some other views or editors using a special button, for example from the Search view.

The Background Tasks view shows the progress of such background tasks as search, SQL query execution, etc. The view shows data when background tasks take some noticeable time and is useful when you want to track the progress of lengthy operations. If you open this view at a short task, the view will be empty.



22

Search "22" in 242 table(s) / 2 database(s): pg_catalog.pg_statio_sys_indexes

You can cancel the task in progress - click the **Cancel Operation** button (🟥).

Database Object Editor

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`

The editor has three tabs:

- **Properties** tab appears for all objects, contains properties of the database object and its sub-entities, see further in this article
- **Data** tab appears for tables and views and represents the Data Editor
- **ER Diagram** tab appears for tables and schemas and displays ERD (Entity Relation Diagrams), see ER Diagrams and Database Structure Diagrams

The tabs have the following common parts:

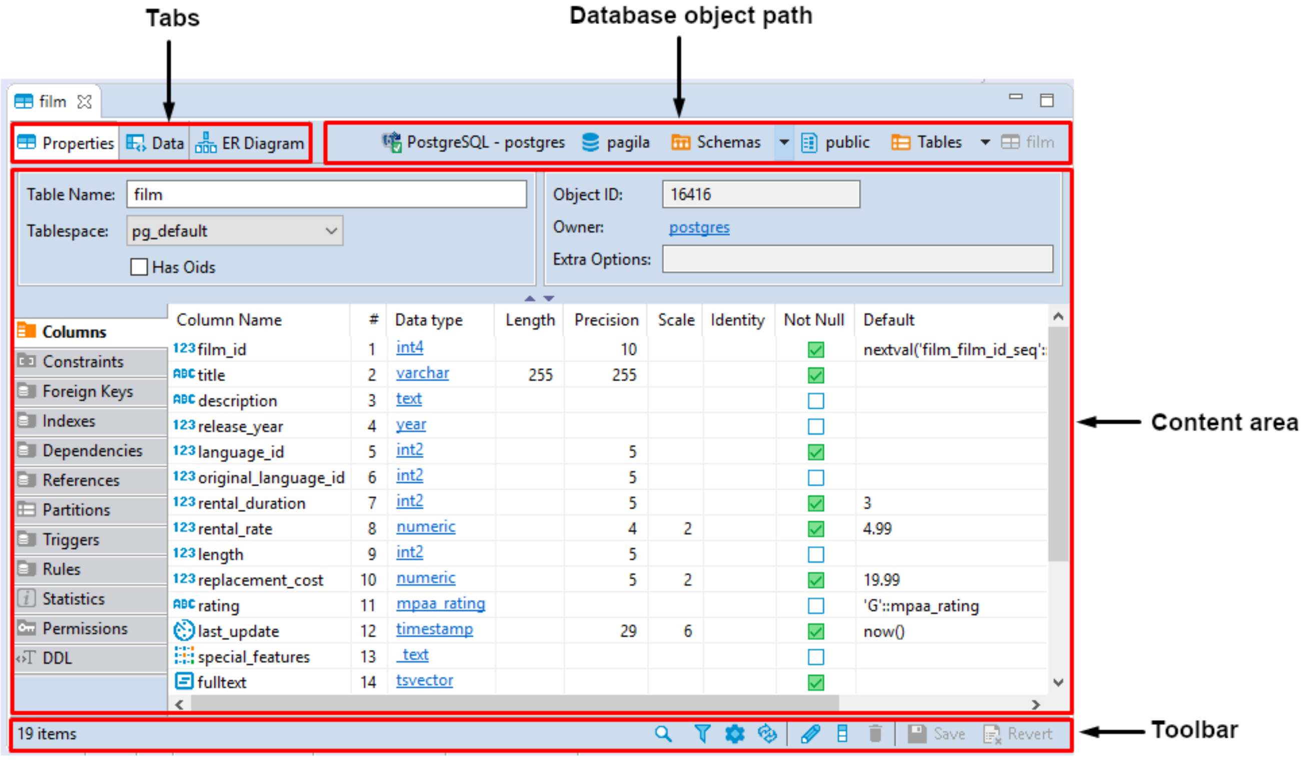


Table Name: film

Tablespace: pg_default

□ Has Oids

Object ID: 16416

Owner: postgres

Extra Options:

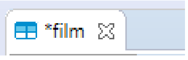
Column Name	#	Data type	Length	Precision	Scale	Identity	Not Null	Default
film_id	1	int4		10			☑	nextval('film_film_id_seq':
title	2	varchar	255	255			☑	
description	3	text					□	
release_year	4	year					□	
language_id	5	int2		5			☑	
original_language_id	6	int2		5			□	
rental_duration	7	int2		5			☑	3
rental_rate	8	numeric		4	2		☑	4.99
length	9	int2		5			□	
replacement_cost	10	numeric		5	2		☑	19.99
rating	11	mpaa_rating					□	'G'::mpaa_rating
last_update	12	timestamp		29	6		☑	now()
special_features	13	_text					□	
fulltext	14	tsvector					☑	

19 items

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.

The toolbar contains different tools on each of the three tabs.

An asterisk appears in the title of an editor if it contains unsaved changes:



The Database Object editor supports the `Ctrl+Z` (undo) function.

Properties Editor

The Properties tab of the Database Object Editor provides tools to view and edit the database object`s properties.

The content area of the Properties tab falls into two parts: the top pane displays properties of the current database object itself while the bottom pane contains properties of the object`s sub-entities or some complex properties like DDL (an SQL description of the current database object).

Properties of sub-entities appear in side tabbed editors – to open such an editor, click the tabs on the left side of the area:

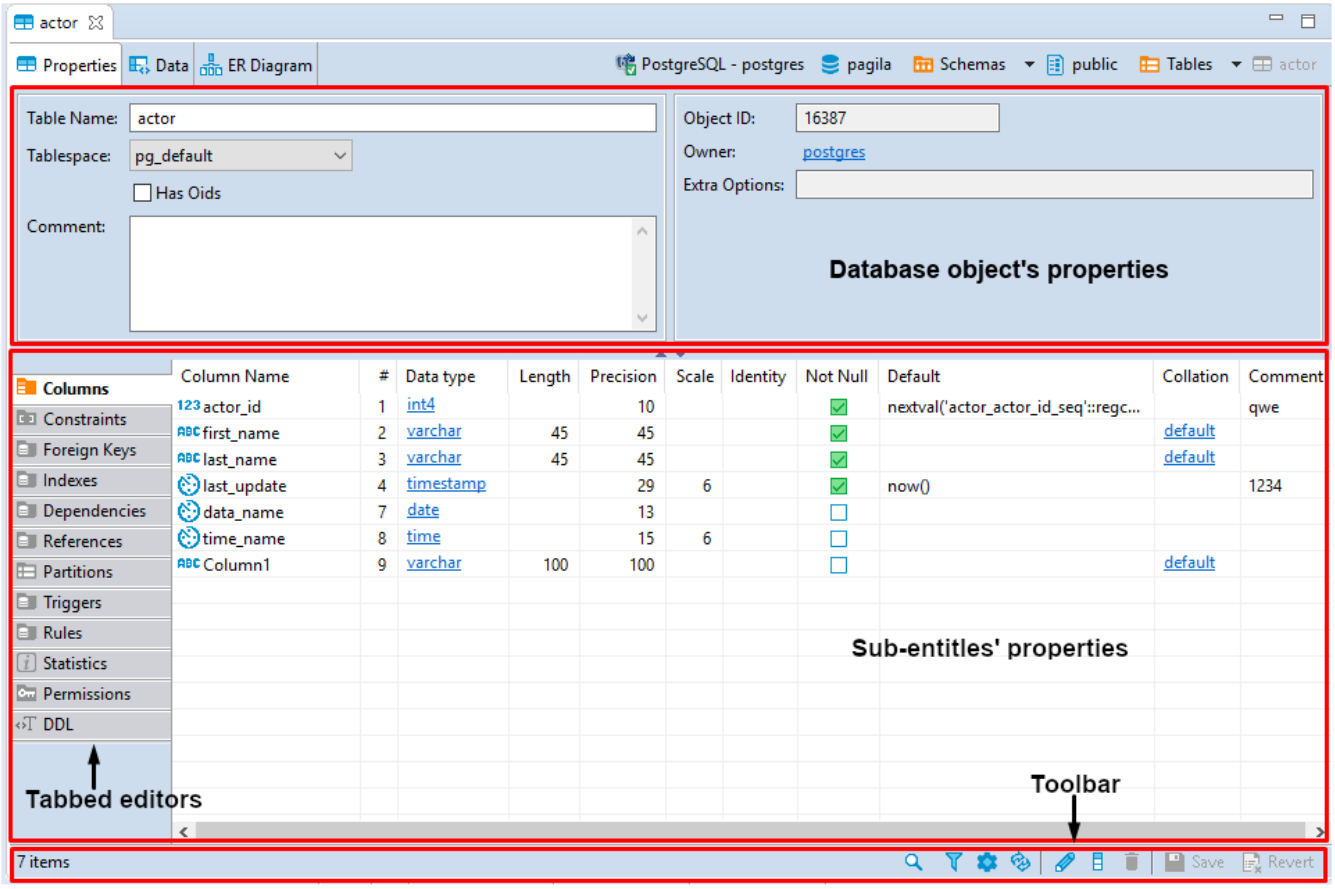


Table Name: actor

Tablespace: pg_default

☐ Has Oids

Comment:

Object ID: 16387

Owner: postgres

Extra Options:

Column Name	#	Data type	Length	Precision	Scale	Identity	Not Null	Default	Collation	Comment
123 actor_id	1	int4		10			☑	nextval('actor_actor_id_seq'::regc...		qwe
ABC first_name	2	varchar	45	45			☑		default	
ABC last_name	3	varchar	45	45			☑		default	
last_update	4	timestamp		29	6		☑	now()		1234
data_name	7	date		13			☐			
time_name	8	time		15	6		☐			
ABC Column1	9	varchar	100	100			☐		default	

7 items

The toolbar at the bottom of the editor provides the following tools for the majority of sub-entities except for some specific ones like Permissions (in PostreSQL) or SQL based views (DDL and Source):

Button	Name	Description
🔍	Search items	Displays a search field next to the button: - 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 in which you can 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
OBJ	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
Save	Save the current contents	- Same as the **Save** button on the application main toolbar - Same as `Ctrl+S` - Opens the **Persist Changes** window that allows saving changes in the currently open sub-entity NOTE: DBeaver recommends saving work after each change.

Button	Name	Description
Revert	Revert to the last saved state	Reverts all changes made to the whole database object to the last saved state

Items in the tabbed editors have context menus which provide the same commands as those in the Database Navigator. To open a context menu for an item, right-click the item.

SQL Script Editors

SQL script editors (**DDL** and **Source**) of the Properties tab contain SQL script that you can either view or modify. The toolbar of the DDL and Source tabs provides the following tools:

Button	Name	Description
📂	Load form file	- Allows selecting a file from the file system - Disabled if the SQL code is read-only
💾	Save to file	Allows saving the current SQL code to a file
>_	Open in SQL console	Opens the SQL code in an SQL Editor

You can select parts of the SQL code and apply generic commands such as copy-paste or SQL-specific commands like formatting – using the context menu. To open the context menu, right-click the SQL code. See SQL Editor for information about SQL-specific commands.

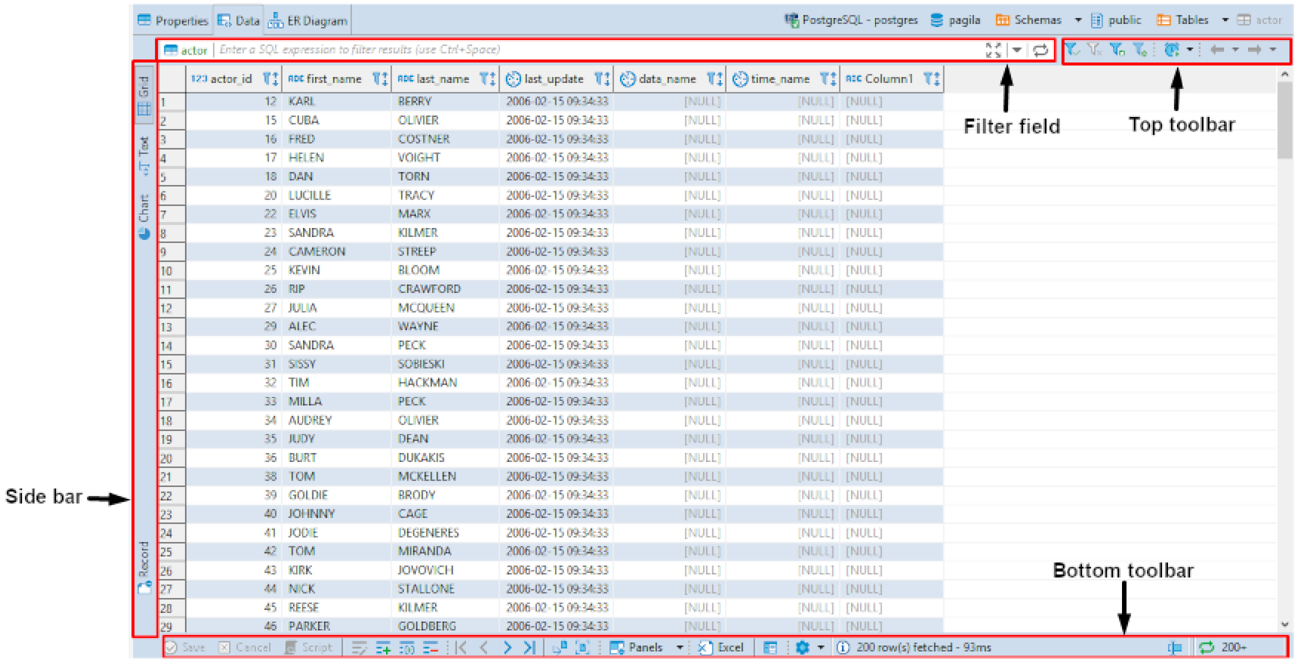
NOTE: **SQL Assist**, **SQL Template**, and **SQL Context Information** menu items on the context menu are disabled if the SQL script is read-only.

Data Editor

The Data editor appears:

- As the **Data** tab of the Database Object Editor available only for tables and views.
- As the **Results** tab when you run a custom SQL query in SQL Editor

The Data editor allows viewing and editing the data of a database table or view. The central part of the Data editor is the data table. The editor also provides two toolbars and a filter field:



	actor_id	first_name	last_name	last_update	data_name	time_name	Column1
1	12	KARL	BERRY	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
2	15	CUBA	OLIVIER	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
3	16	FRED	COSTNER	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
4	17	HELEN	VOIGHT	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
5	18	DAN	TORN	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
6	20	LUCILLE	TRACY	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
7	22	ELVIS	MARX	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
8	23	SANDRA	KILMER	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
9	24	CAMERON	STREEP	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
10	25	KEVIN	BLOOM	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
11	26	RIP	CRAWFORD	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
12	27	JULIA	MCQUEEN	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
13	29	ALEC	WAYNE	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
14	30	SANDRA	PECK	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
15	31	SISSY	SOBIESKI	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
16	32	TIM	HACKMAN	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
17	33	MILLA	PECK	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
18	34	AUDREY	OLIVIER	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
19	35	JUDY	DEAN	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
20	36	BURT	DUKAKIS	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
21	38	TOM	MCKELLEN	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
22	39	GOLDIE	BRODY	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
23	40	JOHNNY	CAGE	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
24	41	JODIE	DEGENERES	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
25	42	TOM	MIRANDA	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
26	43	KIRK	JOVOVICH	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
27	44	NICK	STALLONE	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
28	45	REESE	KILMER	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]
29	46	PARKER	GOLDBERG	2006-02-15 09:34:33	[NULL]	[NULL]	[NULL]

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: 8,715

To learn about ways to navigate data in the data table, see Navigation article.

The top toolbar contains the following buttons:

Button	Name	Description
	Apply filter criteria	Applies filter criteria entered in the filter field above the data table, see Data Filters article for more information
	Remove all filters/orderings	Removes all filters and orderings applied to the data
	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 Data Filters article
	Custom Filters	Opens the Result Set Order/Filter Settings window, see Data Appearance article for more information
	Configure auto-refresh	Allows configuring data auto-refresh settings, see Data Refresh article for details
← →	Forward and backward - history navigation buttons	Navigate forward and backward in the Data Editor history, see *History* section of Navigation article for more information. The buttons are equivalent to pressing the key combinations: `Alt+Left` (backward) and `Alt+right` (forward).

The side bar contains the following tabs:

Button	Name	Description Chart_button
Grid	**Grid**	Switches to grid view of data

Button	Name	Description Chart_button
Text	**Text**	Switches to plain text view of data
Chart	**Chart**	Switches to chart view. For more details on charts, see Managing Charts article.
Record	**Record**	- Same as pressing `Tab` - Switches the positions of rows and columns so that columns appear as rows, and rows hide in one **Value** column, see details in the *Table vs. Record Views* section of Data Appearance article.

The bottom toolbar provides the following buttons:

Button	Name	Description
✓ Save	**Save**	Saves all unsaved changes to the data such as adding, duplicating, deleting rows, inline editing of values, see Data Viewing and Editing article for information
☒ Cancel	**Cancel**	Discards all unsaved changes to the data
Script	**Script**	Opens the Preview Changes window in which you can see changes that you have made to the data, see details in Data Viewing and Editing article
	Edit cell value in separate dialog/editor	Opens the cell in focus for editing in a separate editor or dialog box, see details in *Cell Editor* section of Data Viewing and Editing article
	Add new row	Adds a new empty row below the current row, see details in *Adding, Copying and Deleting Rows* section of Data Viewing and Editing 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 Data Viewing and Editing article
	Delete current row	Colors the rows in focus in red to mark them for deletion, see details in *Adding, Copying and Deleting Rows* section of Data Viewing and Editing 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 next page of results	Fetches the next portion of data (next N rows) making it ready for display, see *Scrolling Results Page* section of Navigation article for more information
	Fetch all rows	Fetches the whole result set making it ready for display, see *Scrolling Results Page* section of Navigation article for more information
Panels	**Panels**	Opens panels on the right side of the Data Editor, see Panels for information
⚙	**Configure**	Opens a dropdown menu with settings
JSON	**JSON**	- Available in EE version only for MongoDB documents and JSON tables - Switches to JSON view of data
XML	**XML**	- Available in EE version only for XML tables - Switches to XML view of data
	Generate Mock Data	Opens the Mock Data Generator window, see TBD for details
	Rows count details	Opens the Status details dialog box showing the timing details of fetching table rows
	Calculate total row count	Calculates the total number of rows in the table

Every cell in the data table has a context menu – right-click the cell to open the menu. The context menu provides the following items:

Menu Item	Description
Cut	Cuts the content of the current cell or column to the clipboard

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
Delete	Deletes the row that has the cell in focus NOTE: In fact, when users click **Delete**, the system only highlights the row red while the actual deletion happens when users click **Save**.
Edit cell	- For CLOB/BLOB data format, opens the contents of the cell in a new tab - For all formats except CLOB/BLOB, opens a properties window for the cell
Inline edit	- Same as double-click on a cell - Makes the cell editable
Set to NULL	Sets the value of selected cells to NULL
Hide column	Hides the column currently in focus, see the Managing Display of Columns in Data Table section further in this article
Save to file…	- Appears only for columns with BLOB/CLOB data - Opens the standard Save As window that allows saving data contained in the cell to a file
Load from file…	- Appears only for columns with BLOB/CLOB data - Opens a standard window for opening files
Order/Filter	Displays a submenu that allows selecting 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 tools for formatting and modifying the view of data, see Data View and Format
Navigate	Opens a submenu that helps users navigate throughout the data table, see Navigation
Layout	Changes the layout of data, see the *Table vs. Record Views* section of the Data View and Format article
Export Resultset	Opens the Data Transfer wizard that guides you through the steps to select a format and export data NOTE: The system exports the whole result set including records that are not visible in the screen and preserves all applied data filters and ordering.
Generate SQL	Opens a submenu on which you can select the type of SQL query to generate
Refresh	Refreshes the whole results set including all items that are not visible in 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.

Navigation

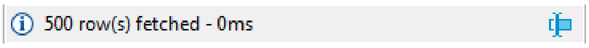
Scrolling Results Page

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: 8,715 . 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.

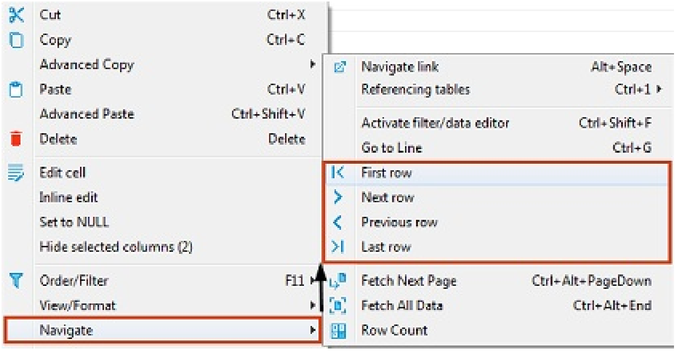
To fetch the whole result set, click the **Fetch all rows** button ( ) in the bottom toolbar or right-click the table and click **Navigate -> Fetch All Data** on the context menu.

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`, `Ctrl+End`.

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

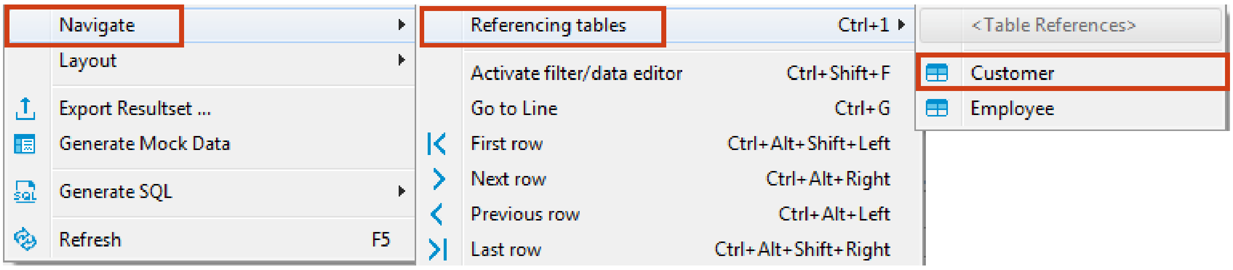
DBeaver remembers the history of such actions as applying filters to data, opening reference tables or other tables via links. You can navigate among such tables and filtered views:

- 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 / Referencing Tables

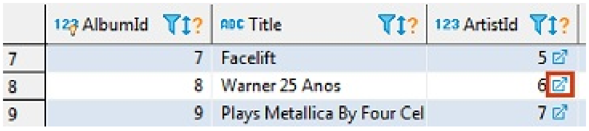
You can navigate by foreign keys or to referencing tables – those that reference the current table. To open a referencing table, press `Ctrl+1` or right-click the cell and click **Navigate->Referencing tables->[table name]**:


Navigate
Layout
Export Resultset ...
Generate Mock Data
Generate SQL
Refresh F5
Referencing tables Ctrl+1
Activate filter/data editor Ctrl+Shift+F
Go to Line Ctrl+G
First row Ctrl+Alt+Shift+Left
Next row Ctrl+Alt+Right
Previous row Ctrl+Alt+Left
Last row Ctrl+Alt+Shift+Right
<Table References>
Customer
Employee

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 referencing table in a new window use `Ctrl+Shift+1` shortcut to show menu.

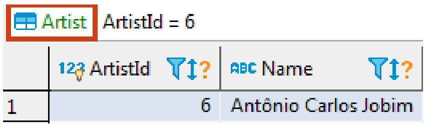
Navigation Links

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:



	AlbumId	Title	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

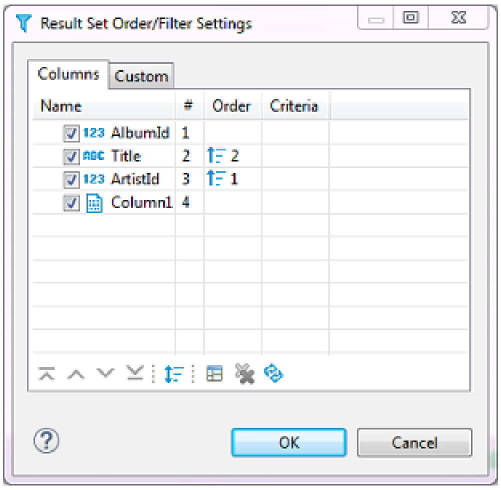
	ArtistId	Name
1	6	Antônio Carlos Jobim

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 and click the Navigate link icon (↗) in the cell.

Data View and Format

The main tool for managing the appearance of the data table is the Result Set Order/Filter Settings window.



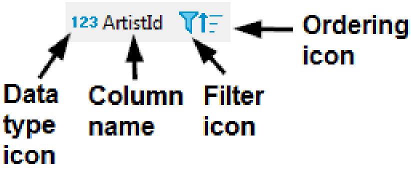
Name	#	Order	Criteria
☑ 123 AlbumId	1		
☑ ABC Title	2	↑ 2	
☑ 123 ArtistId	3	↑ 1	
☑ Column1	4		

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 Settingswindow 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, see … below

Another tool for managing data appearance are column headers. In the data table, every column header contains three elements each having its own function: Data (column) type icon, column name, filter icon, and ordering icon.

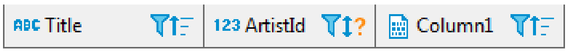


- Simply clicking the column name or column type icon highlights the whole column.
- You can click the column type icon and then drag and drop the column to a different position in the table.
- You can click the column name and then drag the cursor right or left to highlight multiple columns.
- Clicking the ordering icon allows ordering the data in the column in ascending or descending order - see 'Ordering Data in Columns' section further in this article
- Clicking the filter icon allows filtering the data by a cell value, see [TBA]

Ordering Data in Columns

You can order data in columns in one of the ways:

1. Click the ordering icon (↕?) in the header of the column.

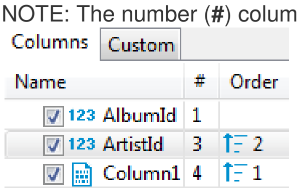
   The icon has three states:

   - 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 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
☑ 123 AlbumId	1	
☑ 123 ArtistId	3	2
☑ 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.

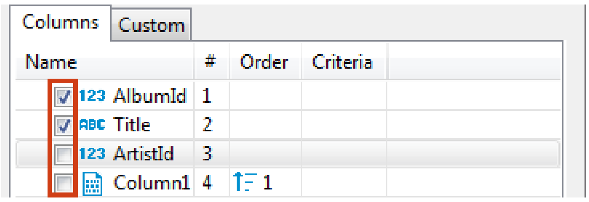
Also, to remove all ordering settings, click the Remove All Filters/Orderings button ( ) in the top toolbar of the Data Editor.

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 in 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.



Columns | Custom

Name	#	Order	Criteria
☑ 123 AlbumId	1		
☑ ABC Title	2		
☐ 123 ArtistId	3		
☐ Column1	4	1	

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 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 focus to it and then move it using the navigation buttons: (⤒ ∧ ∨ ⤓)

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 data in a grid view, similar to Excel spreadsheet, click the **Grid** button (Grid) in the bottom toolbar of the editor.

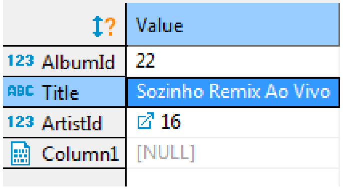
- To switch to the plain text view, click **Text** (Text) in the bottom toolbar.

- To switch to JSON view (available in EE version only for MongoDB documents and JSON tables), click **JSON** in the toolbar.

- To switch to XML view (available in EE version only for XML tables), click **XML** in 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 – columns appear as rows, and rows hide in one **Value** column which now shows only one row of data, and column headers shift from the top of the table to its left side:



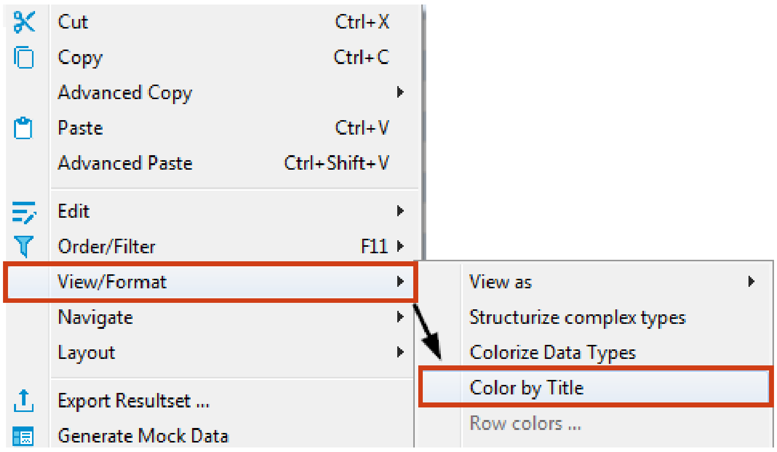
	Value
123 AlbumId	22
ABC Title	Sozinho Remix Ao Vivo
123 ArtistId	16
Column1	[NULL]

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 in the bottom toolbar of the editor: |< < > >|

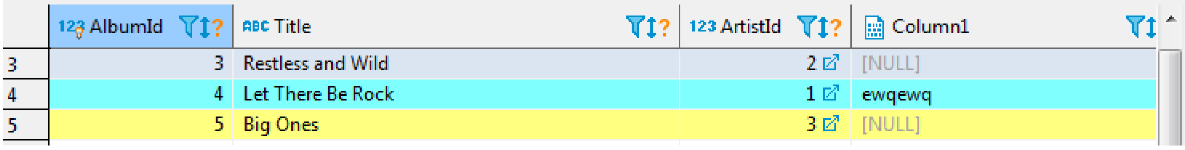
To return back to the standard table view, click the **Record** button again.

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 -> Color by {column name}** 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 selected:

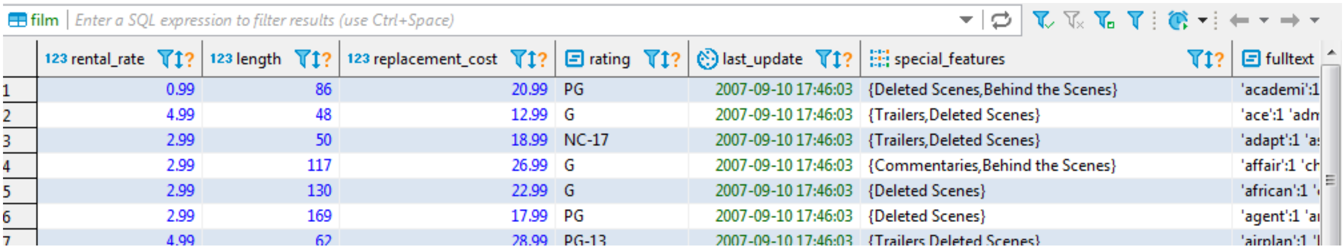


	123 AlbumId	ABC Title	123 ArtistId	Column1
3	3	Restless and Wild	2	[NULL]
4	4	Let There Be Rock	1	ewqewq
5	5	Big Ones	3	[NULL]

To remove the coloring by a particular column, right-click the cell again and click **View/Format -> Reset color by [column name]** on the context menu. To remove coloring from all rows, right-click anywhere in the table and, on the context menu, click **View/Format -> Reset all colors**.

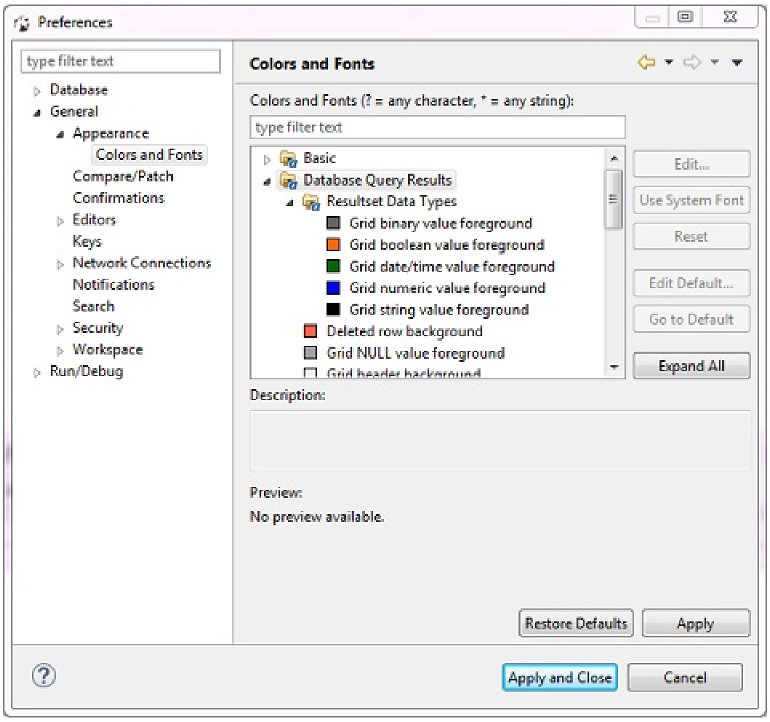
Coloring by Data Types

Besides coloring rows by a value, you can colorize values in 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**. Values in cells are colored in different colors according to preferences currently set:



	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 '
6	2.99	169	17.99	PG	2007-09-10 17:46:03	{Deleted Scenes}	'agent':1 'a
7	4.99	62	28.99	PG-13	2007-09-10 17:46:03	{Trailers,Deleted Scenes}	'airplan':1 '

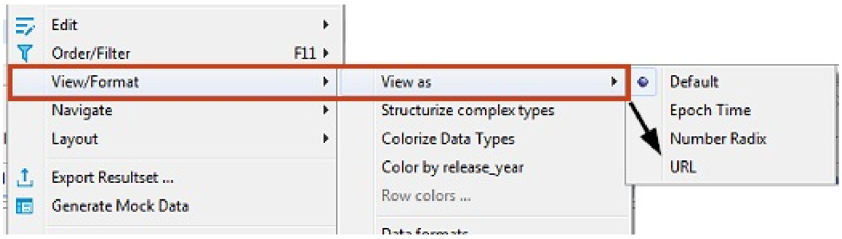
You can change the color preferences in the Preferences window: click **Window -> Preferences** on the main menu. Then, in the window, in the navigation pane on the left, expand **General** and then **Appearance**, and then click **Colors and Fonts**:



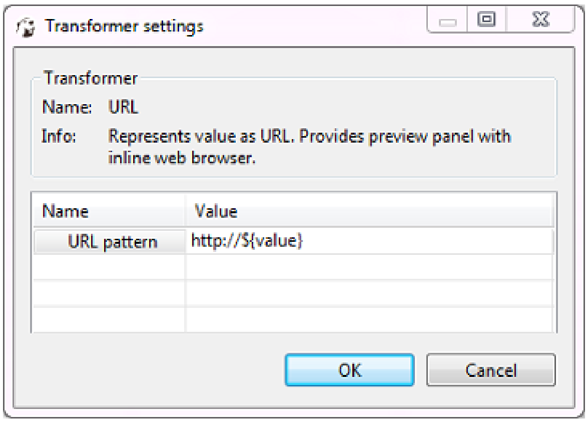
To remove coloring by data types, on the context menu, click **View/Format -> Colorize Data Types** again.

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 -> View as** and then click the presentation type name:



The Transformer settings window opens showing the value in the chosen format. Click **OK** to apply the change:



Name: URL
Info: Represents value as URL. Provides preview panel with inline web browser.

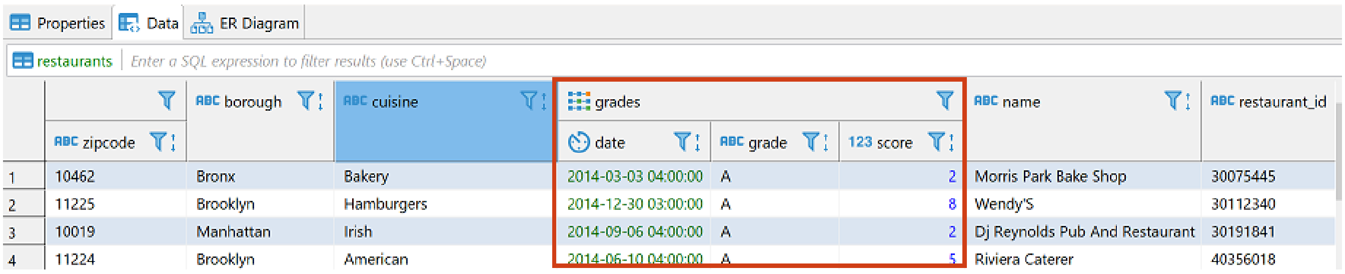
Name	Value
URL pattern	http://${value}

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**.

Structurizing 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:

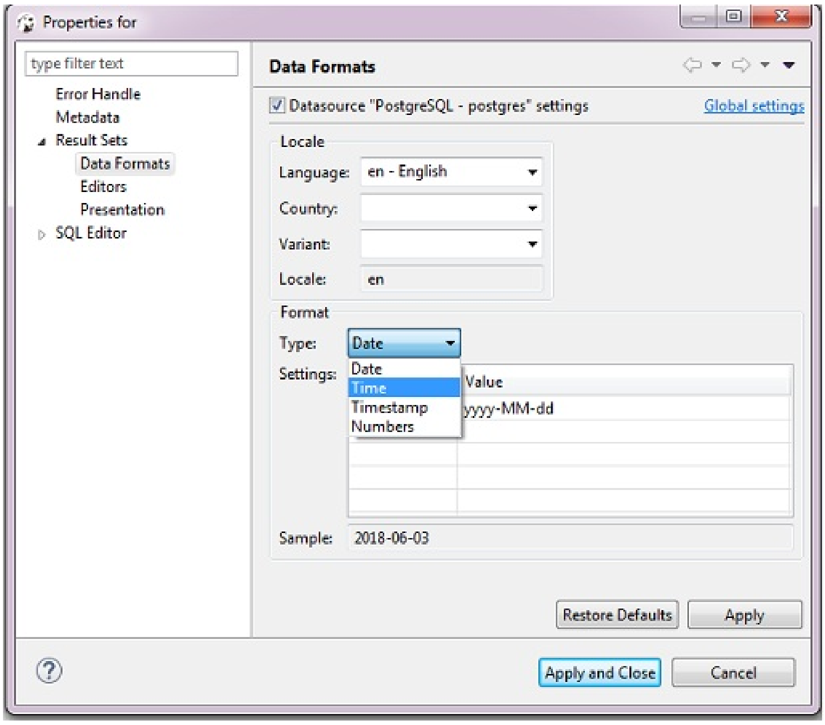


		borough	cuisine	grades			name	restaurant_id
	zipcode			date	grade	score		
1	10462	Bronx	Bakery	2014-03-03 04:00:00	A	2	Morris Park Bake Shop	30075445
2	11225	Brooklyn	Hamburgers	2014-12-30 03:00:00	A	8	Wendy'S	30112340
3	10019	Manhattan	Irish	2014-09-06 04:00:00	A	2	Dj Reynolds Pub And Restaurant	30191841
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:



☑ Datasource "PostgreSQL - postgres" settings

Language: en - English

Country:

Variant:

Locale: en

Type: Date

Settings: yyyy-MM-dd

Sample: 2018-06-03

To configure the format for the current database only, select the **Datasource "[Connection name]" settings** checkbox. To configure the settings globally, to all databases that you have in DBeaver, click **Global settings**.

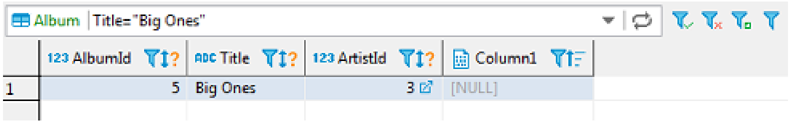
You can specify the locale for the data format in the **Locale** area, then, in the **Type** dropdown list, click the name of the data type and then, in the **Settings** table, click the required format.

To apply the changes and make them visible in the table, click **Apply and Close** and then refresh the window (`F5`).

Data Filters

You can apply custom filters to any table contents or query results. There are several ways in which you can filter data in the table.

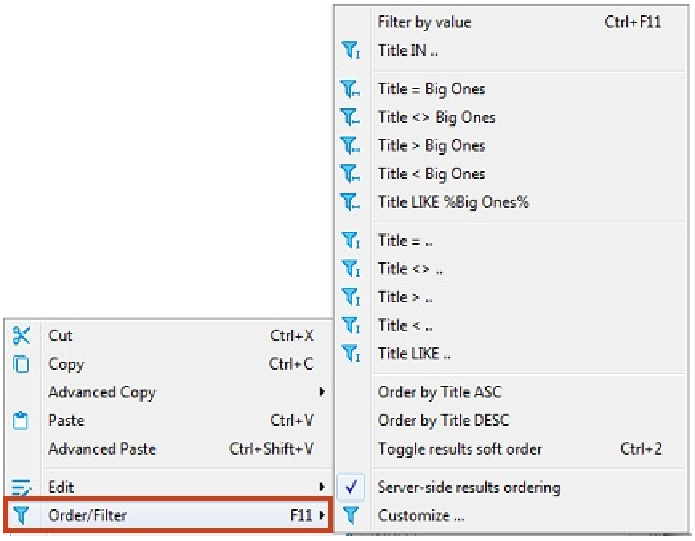
One of the ways is to use the filter field above the table next to the top toolbar. To filter data, enter an SQL expression into the field and click the Apply filter criteria button (▽) next to the field or press `Enter`.



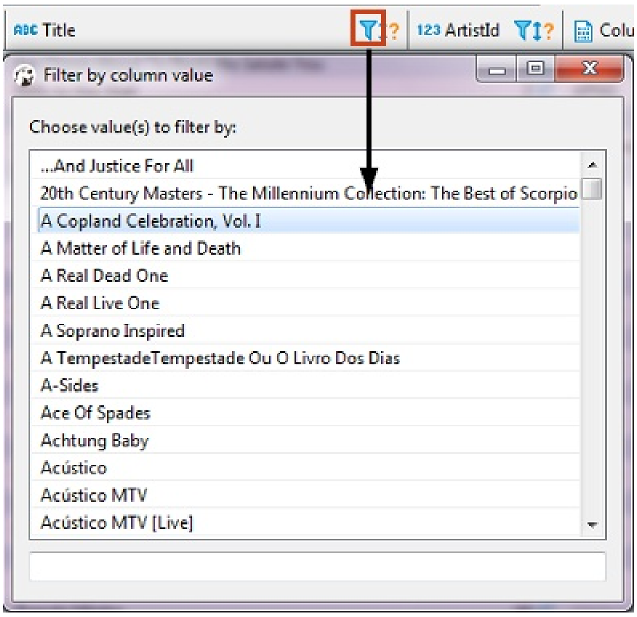
Album | Title="Big Ones"

	123 AlbumId	ABC Title	123 ArtistId	Column1
1	5	Big Ones	3	[NULL]

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 column value dialog box:


ABC Title
123 ArtistId
Colu
Filter by column value
Choose value(s) to filter by:
...And Justice For All
20th Century Masters - The Millennium Collection: The Best of Scorpio
A Copland Celebration, Vol. I
A Matter of Life and Death
A Real Dead One
A Real Live One
A Soprano Inspired
A TempestadeTempestade Ou O Livro Dos Dias
A-Sides
Ace Of Spades
Achtung Baby
Acústico
Acústico MTV
Acústico MTV [Live]

The data updates dynamically. To remove a filter, click the Remove All Filters/Orderings button (▽) in the top toolbar of the editor.

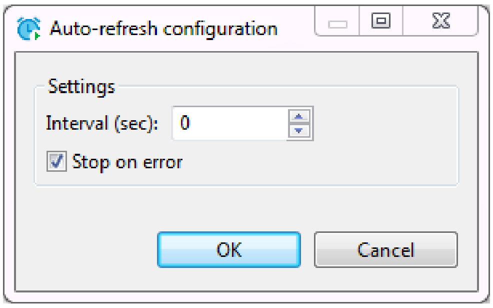
You can save the current filter settings for the database object to apply next time when you reopen it in the editor. To save the current filter settings, click the Save filter settings for current object button (▽) in the top toolbar.

Data Refresh

Refresh 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`.

You can also schedule auto-refresh to happen on a regular basis. To auto-refresh the database on schedule:

1. Click the Configure auto-refresh button ( ) on the top toolbar of the editor. The Auto-refresh configuration dialog box opens:



Auto-refresh configuration

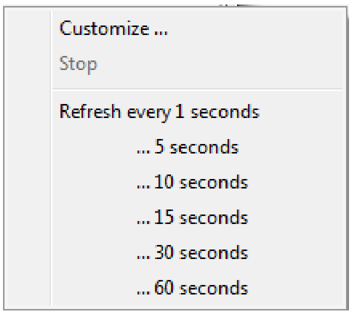
Settings

Interval (sec): 0

☑ Stop on error

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 errors.
c) Click **OK**.

2. Alternatively, click the arrow next to the Configure auto-refresh button ( ) to open the auto-refresh menu:



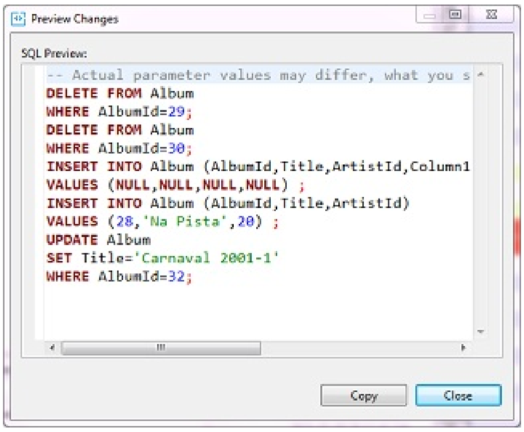
On the menu, you can click one of the preset options or click **Customize** to open the Auto-refresh configuration dialog box, see option 1.

When you perform either of the two alternative options above, the system starts refreshing the data as scheduled and the Configure auto-refresh button changes to **Stop auto-refresh** button ( ). To stop the auto-refresh, click the **Stop auto-refresh** button or click the arrow next to it and click **Stop** on the auto-refresh menu.

Data Viewing and Editing

You can do inline editing (see the *Inline Editing* section below) as well as open the content of a cell in a separate editor (see the *Cell Editor* section below).

When you make any changes to the data and save them using steps described in this section, the changes apply to the database itself. Prior to saving the changes, you can preview the SQL script that the system sends to the database to apply the changes there. To see the SQL script, after making changes and before saving or discarding them, click the **Script** button (Script) in the bottom toolbar. The Preview Changes window opens, in which you can only view the SQL script and copy it, if necessary:



```
-- Actual parameter values may differ, what you s
DELETE FROM Album
WHERE AlbumId=29;
DELETE FROM Album
WHERE AlbumId=30;
INSERT INTO Album (AlbumId,Title,ArtistId,Column1
VALUES (NULL,NULL,NULL,NULL) ;
INSERT INTO Album (AlbumId,Title,ArtistId)
VALUES (28,'Na Pista',20) ;
UPDATE Album
SET Title='Carnaval 2001-1'
WHERE AlbumId=32;
```

Inline Editing

Inline editing is when you modify the content right in the cell. To edit a cell inline, in the table do one of the following:

- Double-click the cell.
- Click the cell to set focus to it and press `Enter`.
- Right-click the cell and click **Inline edit** on the context menu.

The cell becomes editable, now you can change its value.

To set the cell value to NULL, right-click the cell and click **Set to NULL** on the context menu.

To save the changes, click the **Save** button (✓ Save) in the bottom toolbar. To discard the changes, click the **Cancel** button (☒ Cancel) in the bottom toolbar.

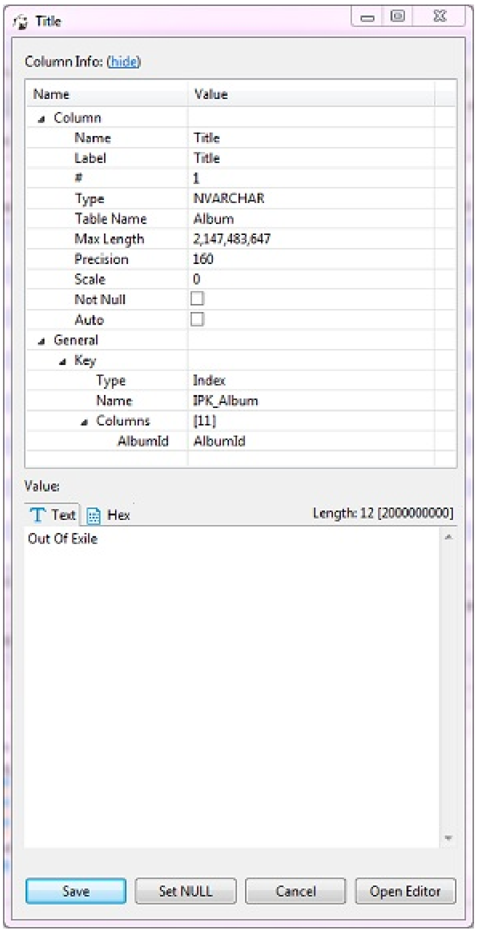
NOTE: Both the **Save** and **Cance**l buttons become editable only when you make changes in a cell and then jump to another cell.

Cell Editor

To edit data in a cell using a separate editor, do one of the following:

- Right-click the cell and click **Edit cell** on the context menu.
- Click the cell to set focus to it and press `Shift+Enter` or click the **Edit cell value in separate dialog/editor** button ( ) in the bottom toolbar.

For cells of CLOB/BLOB data format, this action opens the contents of the cell in a new tab. For all other formats except CLOB/BLOB, this action opens a properties window for the cell:



Name	Value
Column	
Name	Title
Label	Title
#	1
Type	NVARCHAR
Table Name	Album
Max Length	2,147,483,647
Precision	160
Scale	0
Not Null	□
Auto	□
General	
Key	
Type	Index
Name	IPK_Album
Columns	[11]
AlbumId	AlbumId

Length: 12 [2000000000]

Out Of Exile

The window displays properties of the column in the **Column Info** section and provides the **Value** section where you can modify the value of the cell. Edit the value as required and click **Save**. To set the value to NULL, click **Set NULL**. To continue editing the cell in a separate editor (tab), click **Open Editor**.

NOTE: DBeaver has full support of CLOB/BLOB data types. You can view values, edit them and save back to the database. You can open CLOB/BLOB value in a separate editor (press `Shift+Enter` on a selected cell). You can save/load LOB value to/from regular files. DBeaver can recognize that some BLOB column keeps images (gif, png, jpeg, bmp). In this case DBeaver shows LOB contents as image. It is convenient to open value view panel (press `F7`) and browse images.

Adding, Copying and Deleting Rows

You can add an empty row below the row in focus. To add an empty row, click the **Add new row** button () in the bottom toolbar. Use inline editing or open cell values in a separate editor to populate them with data (see the sections above).

You can copy any row or several rows currently in focus. To copy rows, highlight one or more rows and click the **Duplicate current row** button () in the bottom toolbar. The duplicate rows appear below the rows in focus.

To delete a row or rows, set focus to the rows and click the **Delete current row** button () in the bottom toolbar. The rows are colored red, which means that they are marked for deletion and will be deleted when you save the changes.

To save any of such changes, click the **Save** button (✓ Save) in the bottom toolbar. To discard the changes, click the **Cancel** button (☒ Cancel) in the bottom toolbar.

Copying/Pasting Cells

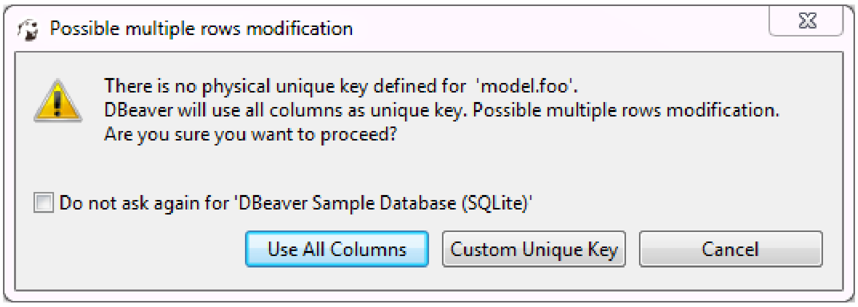
To copy the content of one or several cells to the clipboard in TAB-delimited format, press `Ctrl+C` or right-click the cell or cell selection and click **Copy** on the context menu. Then you can paste the copied selection into some spreadsheet editor (similar to Excel).

DBeaver provides the advanced copy option that allows configuring additional copy settings (copy with column names/row numbers, configure delimiter and choose value format). To copy cells with additional settings, press `Ctrl+Shift+C` or right click the cell(s) and click **Advanced Copy** on the context menu.

Pressing `Ctrl+V` on a cell pastes the copied content into the cell applying appropriate data type conversion. The **Advanced Paste** option on the context menu or pressing `Ctrl+Shift+V` pastes several cells.

Defining Virtual Keys

To be able to persist 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:



There is no physical unique key defined for 'model.foo'.
DBeaver will use all columns as unique key. Possible multiple rows modification.
Are you sure you want to proceed?

□ Do not ask again for 'DBeaver Sample Database (SQLite)'

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 custom unique key, you can click the **Configure** (⚙) button in the bottom toolbar and then click **Define virtual unique key** on the Configure menu. The Define virtual unique identifier window opens:

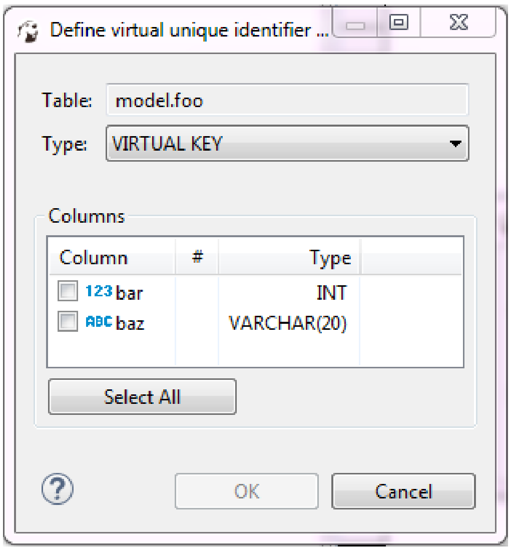


Table: model.foo

Type: VIRTUAL KEY

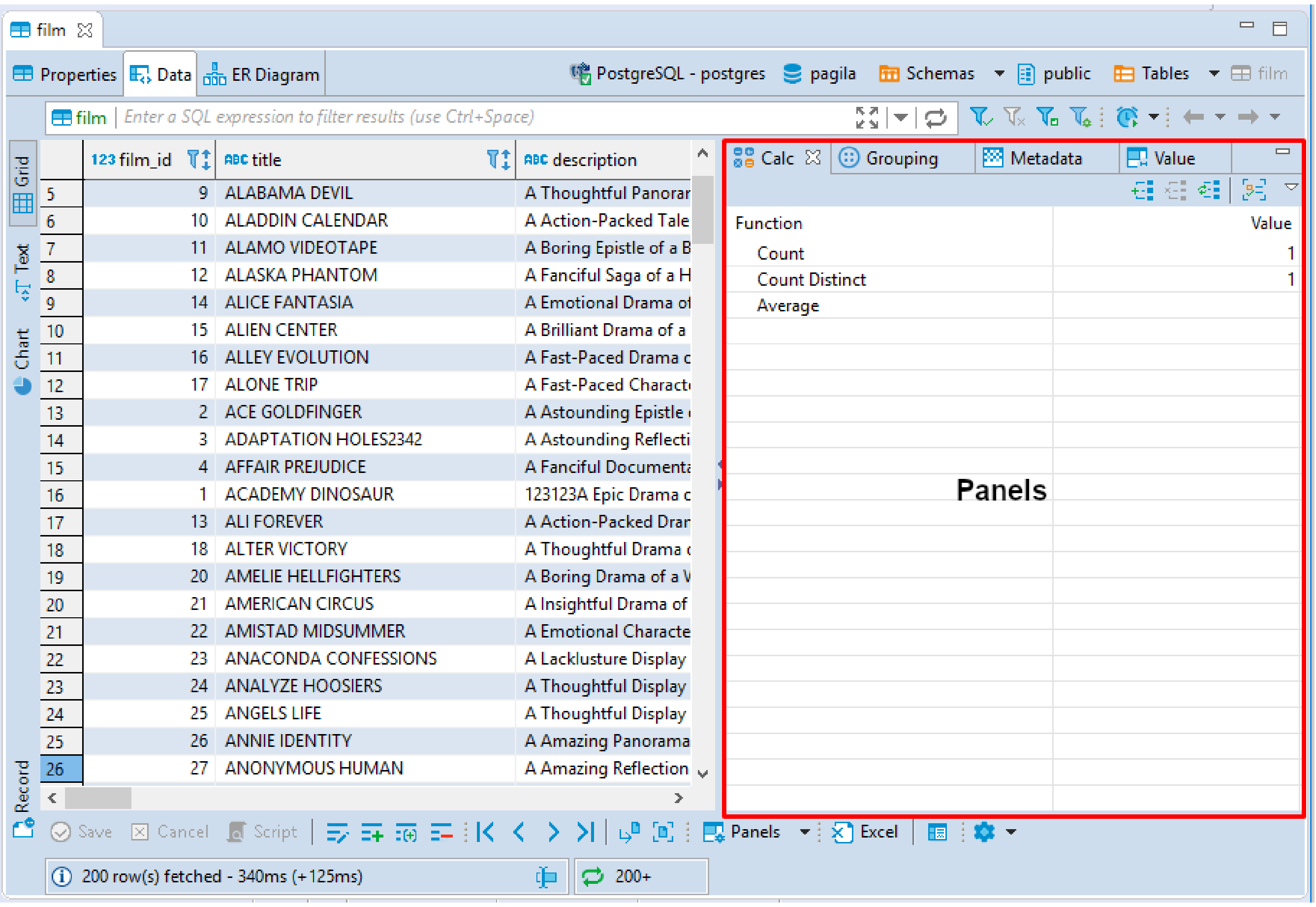
Columns

Column	#	Type
□ 123 bar		INT
□ ABC baz		VARCHAR(20)

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**.

Panels

Panels provide additional space in the Data editor in which you can manipulate with data. The panels are handy if you work with complex types (structures, arrays), or long text data, or BLOBs. Panels appear as tabs in an additional pane in the right part of the Data tab:



	123 film_id	ABC title	ABC description
5	9	ALABAMA DEVIL	A Thoughtful Panorar
6	10	ALADDIN CALENDAR	A Action-Packed Tale
7	11	ALAMO VIDEOTAPE	A Boring Epistle of a B
8	12	ALASKA PHANTOM	A Fanciful Saga of a H
9	14	ALICE FANTASIA	A Emotional Drama of
10	15	ALIEN CENTER	A Brilliant Drama of a
11	16	ALLEY EVOLUTION	A Fast-Paced Drama c
12	17	ALONE TRIP	A Fast-Paced Characte
13	2	ACE GOLDFINGER	A Astounding Epistle
14	3	ADAPTATION HOLES2342	A Astounding Reflecti
15	4	AFFAIR PREJUDICE	A Fanciful Documenta
16	1	ACADEMY DINOSAUR	123123A Epic Drama c
17	13	ALI FOREVER	A Action-Packed Drar
18	18	ALTER VICTORY	A Thoughtful Drama c
19	20	AMELIE HELLFIGHTERS	A Boring Drama of a V
20	21	AMERICAN CIRCUS	A Insightful Drama of
21	22	AMISTAD MIDSUMMER	A Emotional Characte
22	23	ANACONDA CONFESSIONS	A Lacklusture Display
23	24	ANALYZE HOOSIERS	A Thoughtful Display
24	25	ANGELS LIFE	A Thoughtful Display
25	26	ANNIE IDENTITY	A Amazing Panorama
26	27	ANONYMOUS HUMAN	A Amazing Reflection

Function	Value
Count	1
Count Distinct	1
Average	

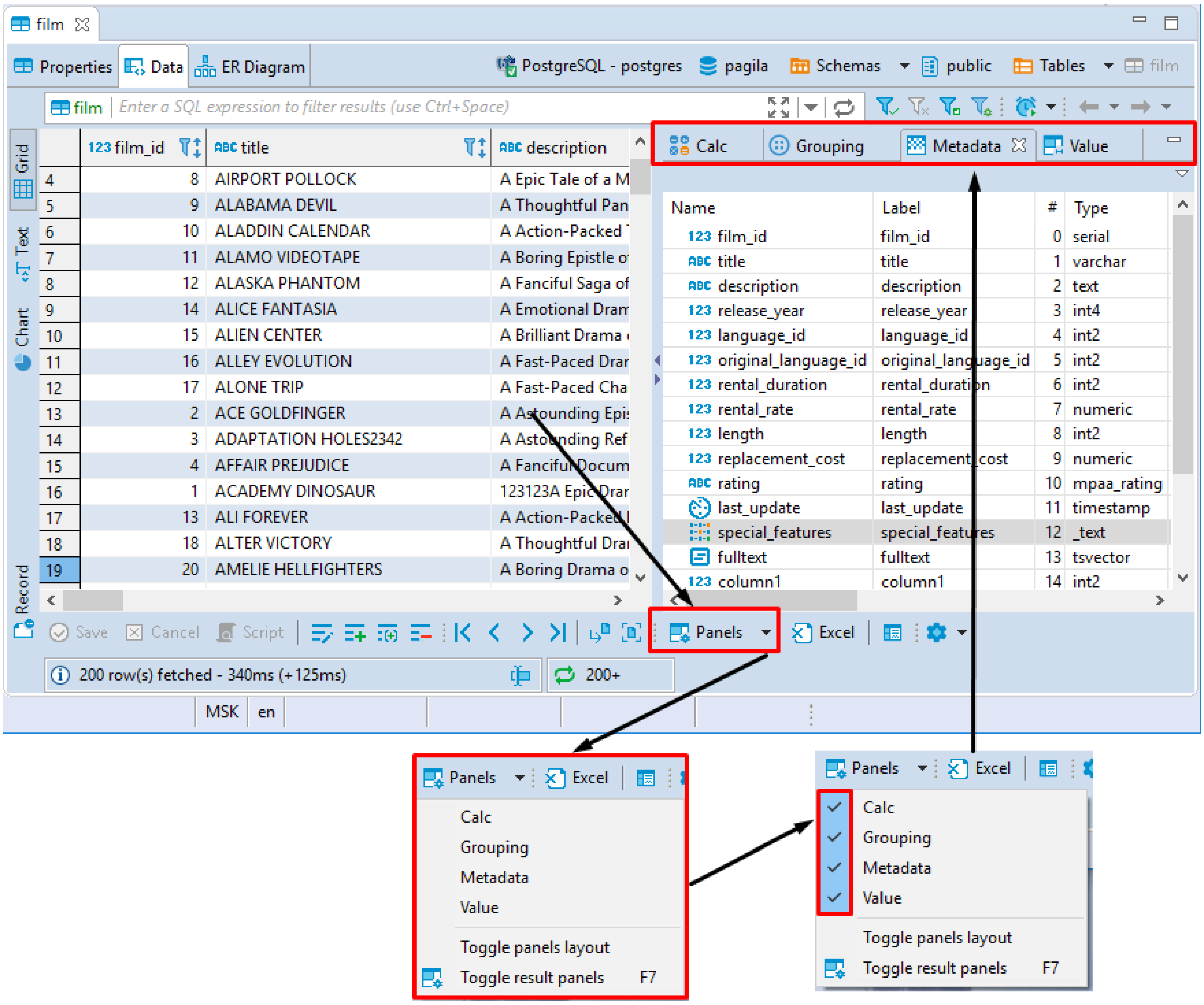
Panels

200 row(s) fetched - 340ms (+125ms)

This additional pane appears only when you open one of the four panels:

- Calc
- Grouping
- Metadata
- Value viewer (default)

To open panels, click **Panels** in 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:



	film_id	title	description
4	8	AIRPORT POLLOCK	A Epic Tale of a M
5	9	ALABAMA DEVIL	A Thoughtful Pan
6	10	ALADDIN CALENDAR	A Action-Packed
7	11	ALAMO VIDEOTAPE	A Boring Epistle o
8	12	ALASKA PHANTOM	A Fanciful Saga of
9	14	ALICE FANTASIA	A Emotional Dram
10	15	ALIEN CENTER	A Brilliant Drama
11	16	ALLEY EVOLUTION	A Fast-Paced Drar
12	17	ALONE TRIP	A Fast-Paced Cha
13	2	ACE GOLDFINGER	A Astounding Epi
14	3	ADAPTATION HOLES2342	A Astounding Ref
15	4	AFFAIR PREJUDICE	A Fanciful Docum
16	1	ACADEMY DINOSAUR	123123A Epic Drar
17	13	ALI FOREVER	A Action-Packed
18	18	ALTER VICTORY	A Thoughtful Dra
19	20	AMELIE HELLFIGHTERS	A Boring Drama o

Name	Label	#	Type
film_id	film_id	0	serial
title	title	1	varchar
description	description	2	text
release_year	release_year	3	int4
language_id	language_id	4	int2
original_language_id	original_language_id	5	int2
rental_duration	rental_duration	6	int2
rental_rate	rental_rate	7	numeric
length	length	8	int2
replacement_cost	replacement_cost	9	numeric
rating	rating	10	mpaa_rating
last_update	last_update	11	timestamp
special_features	special_features	12	_text
fulltext	fulltext	13	tsvector
column1	column1	14	int2

200 row(s) fetched - 340ms (+125ms)

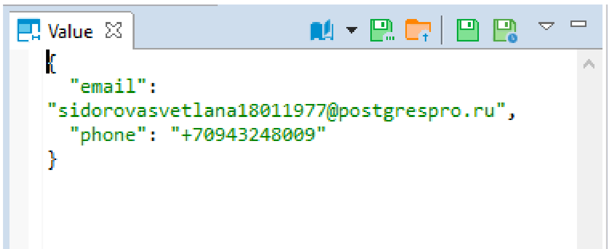
200+

Panels also open if you try to inline-edit a cell with a complex data type.

To close 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 (⚙) in the bottom toolbar and then **Toggle result panels** on the Configure dropdown menu.

Value Viewer

The Value viewer panel displays just one value that is currently in focus and allows editing it.



```
{
  "email":
"sidorovasvetlana18011977@postgrespro.ru",
  "phone": "+70943248009"
}
```

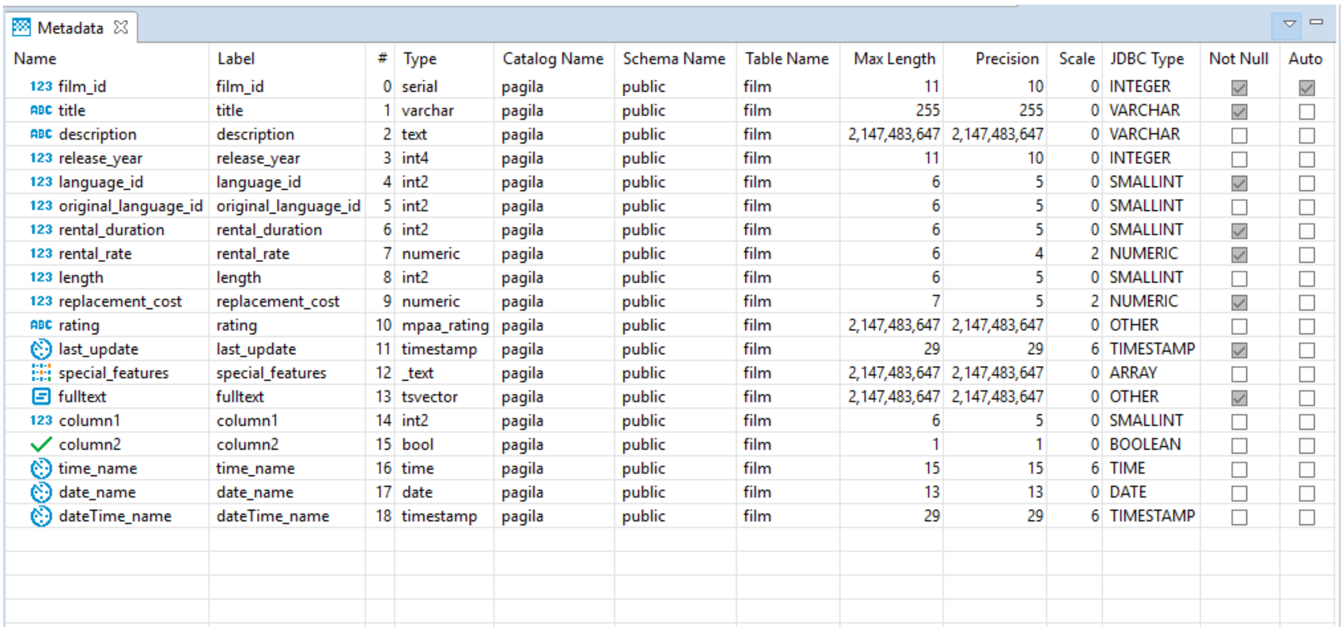
The toolbar of Value panel contains the following buttons:

Button	Name	Description
	Content viewer settings	Opens a menu with a set of options for content view change.
	Save to file...	Allows saving the content to a local file. **NOTE**: This button is only available for XML, JSON and Binary content.

Button	Name	Description
📂	Load from file...	Allows uploading data from a local file. **NOTE**: This button is only available for XML,JSON and Binary content.
💾	Apply cell value	Displays in the data table the changes made in the Value viewer. **NOTE**: This button does not save changes made to the database. To save the changes in the database, you need to use the **Save** button in the bottom toolbar of the Data Editor..
💾	Auto-apply value	Enables 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 button does not save changes made to the database. To save the changes in the database, you need to use the **Save** button in the bottom toolbar of the Data Editor.

Metadata Panel

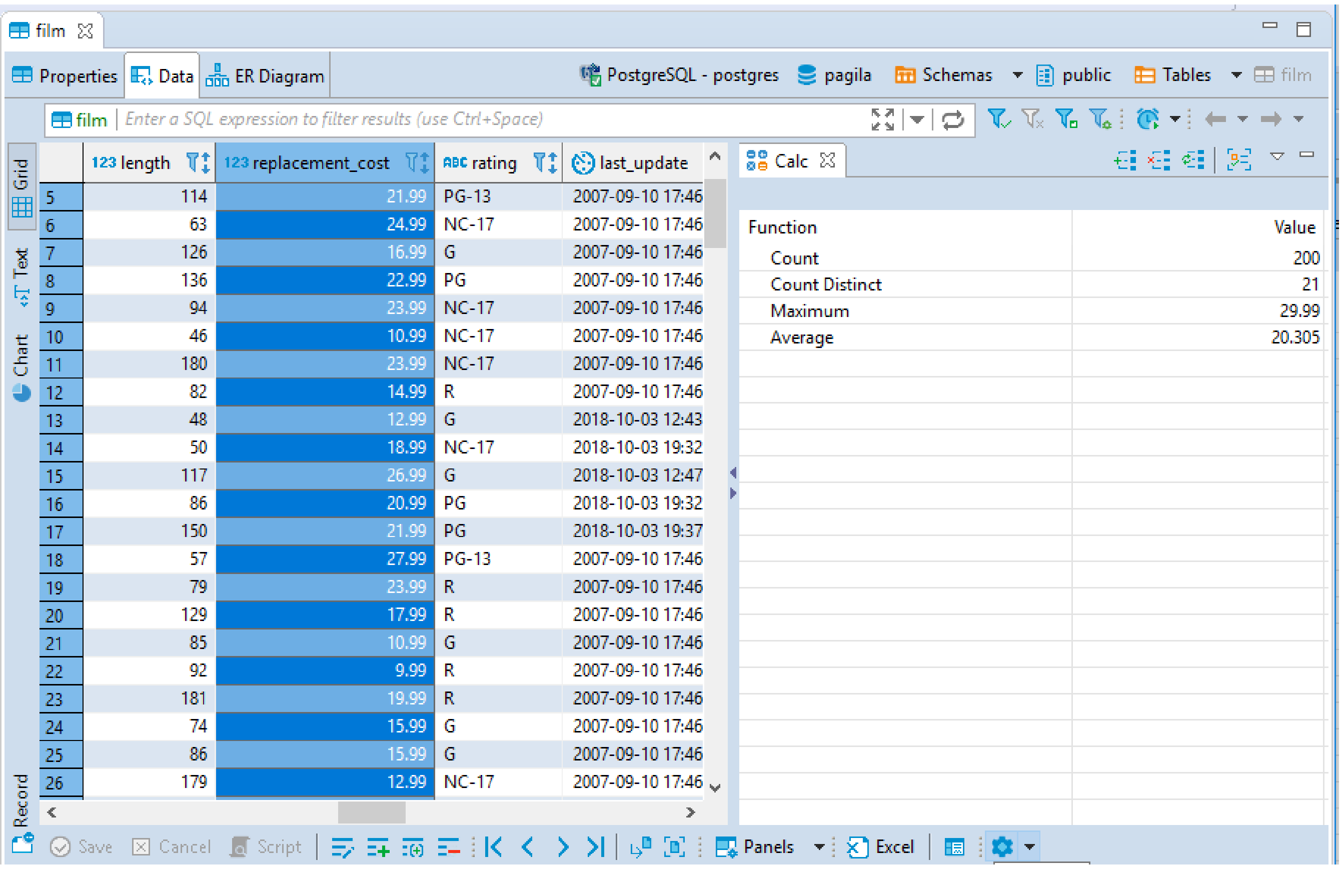
The Metadata panel displays metadata for each cell in the row containing the cell currently in focus. You can just view the metadata.



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	☑	☑
ABC title	title	1	varchar	pagila	public	film	255	255	0	VARCHAR	☑	☐
ABC description	description	2	text	pagila	public	film	2,147,483,647	2,147,483,647	0	VARCHAR	☐	☐
123 release_year	release_year	3	int4	pagila	public	film	11	10	0	INTEGER	☐	☐
123 language_id	language_id	4	int2	pagila	public	film	6	5	0	SMALLINT	☑	☐
123 original_language_id	original_language_id	5	int2	pagila	public	film	6	5	0	SMALLINT	☐	☐
123 rental_duration	rental_duration	6	int2	pagila	public	film	6	5	0	SMALLINT	☑	☐
123 rental_rate	rental_rate	7	numeric	pagila	public	film	6	4	2	NUMERIC	☑	☐
123 length	length	8	int2	pagila	public	film	6	5	0	SMALLINT	☐	☐
123 replacement_cost	replacement_cost	9	numeric	pagila	public	film	7	5	2	NUMERIC	☑	☐
ABC rating	rating	10	mpaa_rating	pagila	public	film	2,147,483,647	2,147,483,647	0	OTHER	☐	☐
last_update	last_update	11	timestamp	pagila	public	film	29	29	6	TIMESTAMP	☑	☐
special_features	special_features	12	_text	pagila	public	film	2,147,483,647	2,147,483,647	0	ARRAY	☐	☐
fulltext	fulltext	13	tsvector	pagila	public	film	2,147,483,647	2,147,483,647	0	OTHER	☑	☐
123 column1	column1	14	int2	pagila	public	film	6	5	0	SMALLINT	☐	☐
✓ column2	column2	15	bool	pagila	public	film	1	1	0	BOOLEAN	☐	☐
time_name	time_name	16	time	pagila	public	film	15	15	6	TIME	☐	☐
date_name	date_name	17	date	pagila	public	film	13	13	0	DATE	☐	☐
dateTime_name	dateTime_name	18	timestamp	pagila	public	film	29	29	6	TIMESTAMP	☐	☐

Calc Panel

The Calc panel is useful for getting basic statistics across data in several columns and rows:



	123 length	123 replacement_cost	ABC rating	last_update
5	114	21.99	PG-13	2007-09-10 17:46
6	63	24.99	NC-17	2007-09-10 17:46
7	126	16.99	G	2007-09-10 17:46
8	136	22.99	PG	2007-09-10 17:46
9	94	23.99	NC-17	2007-09-10 17:46
10	46	10.99	NC-17	2007-09-10 17:46
11	180	23.99	NC-17	2007-09-10 17:46
12	82	14.99	R	2007-09-10 17:46
13	48	12.99	G	2018-10-03 12:43
14	50	18.99	NC-17	2018-10-03 19:32
15	117	26.99	G	2018-10-03 12:47
16	86	20.99	PG	2018-10-03 19:32
17	150	21.99	PG	2018-10-03 19:37
18	57	27.99	PG-13	2007-09-10 17:46
19	79	23.99	R	2007-09-10 17:46
20	129	17.99	R	2007-09-10 17:46
21	85	10.99	G	2007-09-10 17:46
22	92	9.99	R	2007-09-10 17:46
23	181	19.99	R	2007-09-10 17:46
24	74	15.99	G	2007-09-10 17:46
25	86	15.99	G	2007-09-10 17:46
26	179	12.99	NC-17	2007-09-10 17:46

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 shows statistics for the selected data.

To see data grouped by columns, click the Group by columns button ( ). To remove the grouping by columns and see 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 in 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:

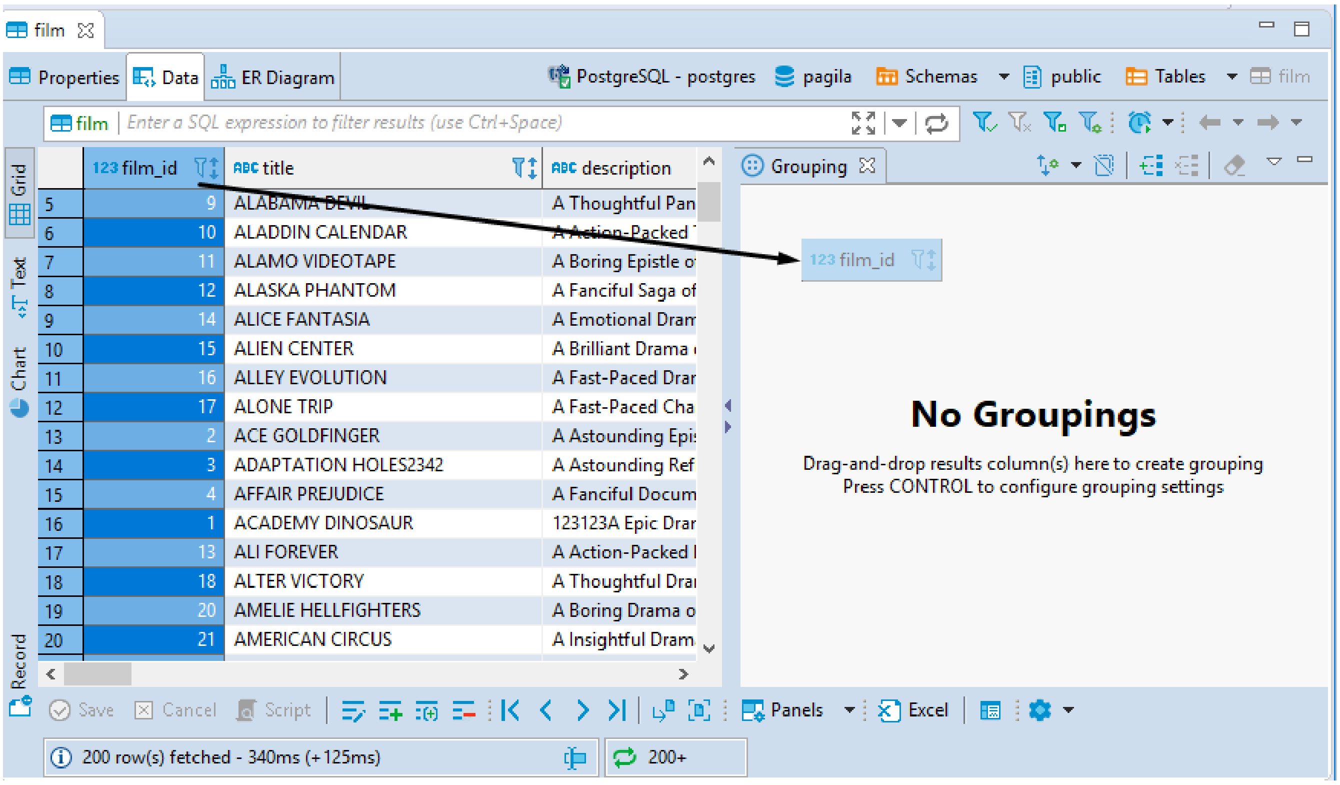
- Sum
- Average
- Minimum
- Maximum
- Median
- Mode

To remove an individual function, click the function and then click **Remove function** ( ) in the toolbar of the panel, or right-click the function and click **Remove function** on the context menu. To remove all functions, click **Reset** ( ) in 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 so that the cursor turns into a hand pointer ( ), and drag-n-drop the column(s) onto the panel:



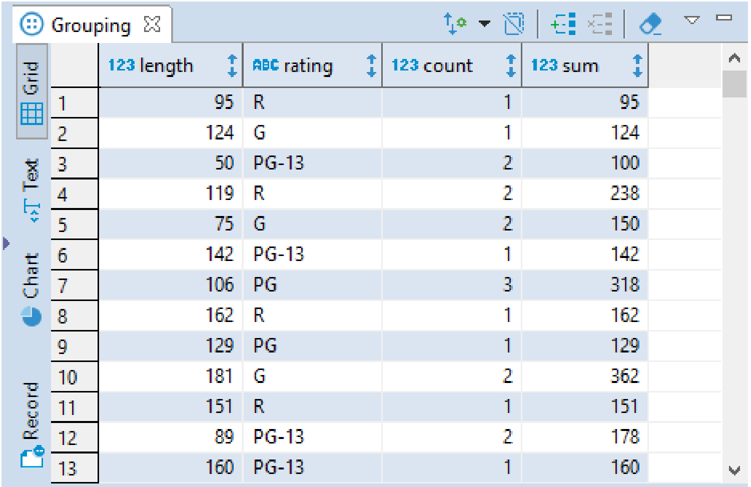
	123 film_id	ABC title	ABC description
5	9	ALABAMA DEVIL	A Thoughtful Pan
6	10	ALADDIN CALENDAR	A Action-Packed
7	11	ALAMO VIDEOTAPE	A Boring Epistle o
8	12	ALASKA PHANTOM	A Fanciful Saga of
9	14	ALICE FANTASIA	A Emotional Dram
10	15	ALIEN CENTER	A Brilliant Drama
11	16	ALLEY EVOLUTION	A Fast-Paced Drar
12	17	ALONE TRIP	A Fast-Paced Cha
13	2	ACE GOLDFINGER	A Astounding Epi:
14	3	ADAPTATION HOLES2342	A Astounding Ref
15	4	AFFAIR PREJUDICE	A Fanciful Docum
16	1	ACADEMY DINOSAUR	123123A Epic Drar
17	13	ALI FOREVER	A Action-Packed I
18	18	ALTER VICTORY	A Thoughtful Drai
19	20	AMELIE HELLFIGHTERS	A Boring Drama o
20	21	AMERICAN CIRCUS	A Insightful Dram

No Groupings

Drag-and-drop results column(s) here to create grouping
Press CONTROL to configure grouping settings

200 row(s) fetched - 340ms (+125ms)

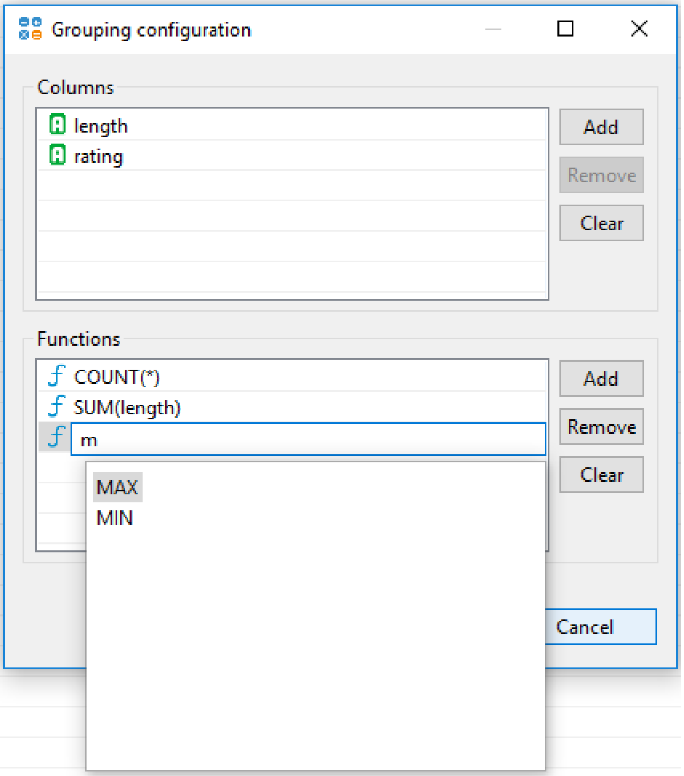
If you add several columns to the panel, DBeaver groups data in the order in which the columns go and calculates statistics based on the grouping.



	123 length	ABC rating	123 count	123 sum
1	95	R	1	95
2	124	G	1	124
3	50	PG-13	2	100
4	119	R	2	238
5	75	G	2	150
6	142	PG-13	1	142
7	106	PG	3	318
8	162	R	1	162
9	129	PG	1	129
10	181	G	2	362
11	151	R	1	151
12	89	PG-13	2	178
13	160	PG-13	1	160

By default, the COUNT function is used. You can add other functions as well. To add a function:

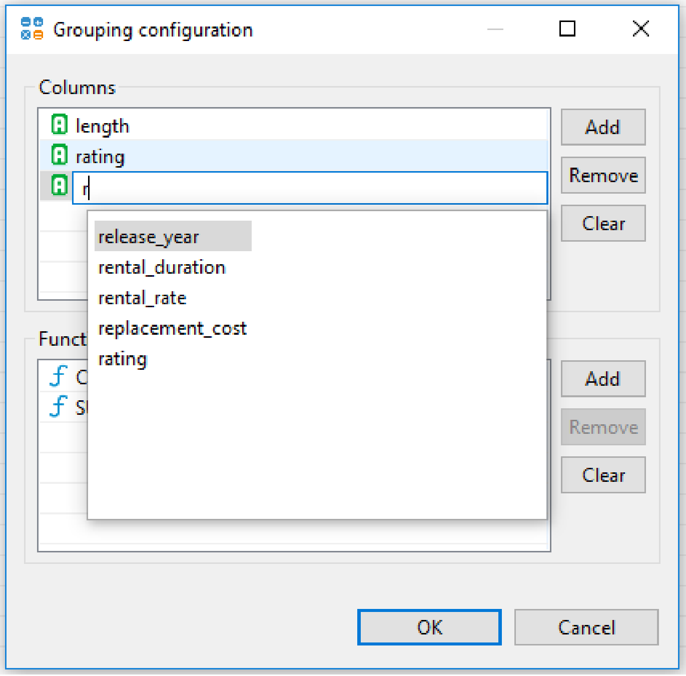
1. Click the **Edit grouping columns** button in the panel`s toolbar.
2. In the Grouping Configuration window, in the **Functions** area, click **Add**, then type the function into the new row:
   - You can use auto-complete options DBeaver provides.
   - You need to indicate the column name in brackets. COUNT is the only function that supports `*` instead of column name.
3. Click **OK**:



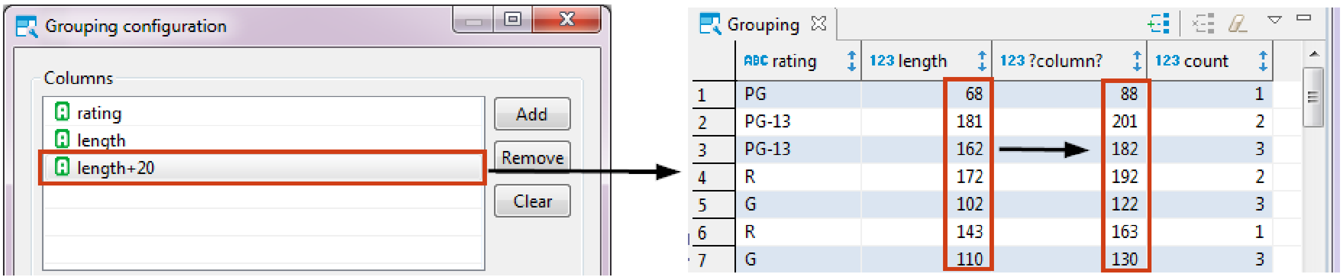
To remove a function, in the same Grouping Configuration window, click the function and click **Remove** and then **OK**. To remove all functions, click **Clear** and then **OK**.

You can also add or remove columns using the same Grouping Configuration window. To add a column:

1. Click the **Edit grouping columns** button in the panel`s toolbar.
2. In the Grouping Configuration window, in the **Columns** area, click **Add**, then type the column name into the new row (you can use auto-complete options DBeaver provides), and then click **OK**:



For MySQL/MariaDB databases you can also add a column with an expression - the expression will be calculated in the resulting column:



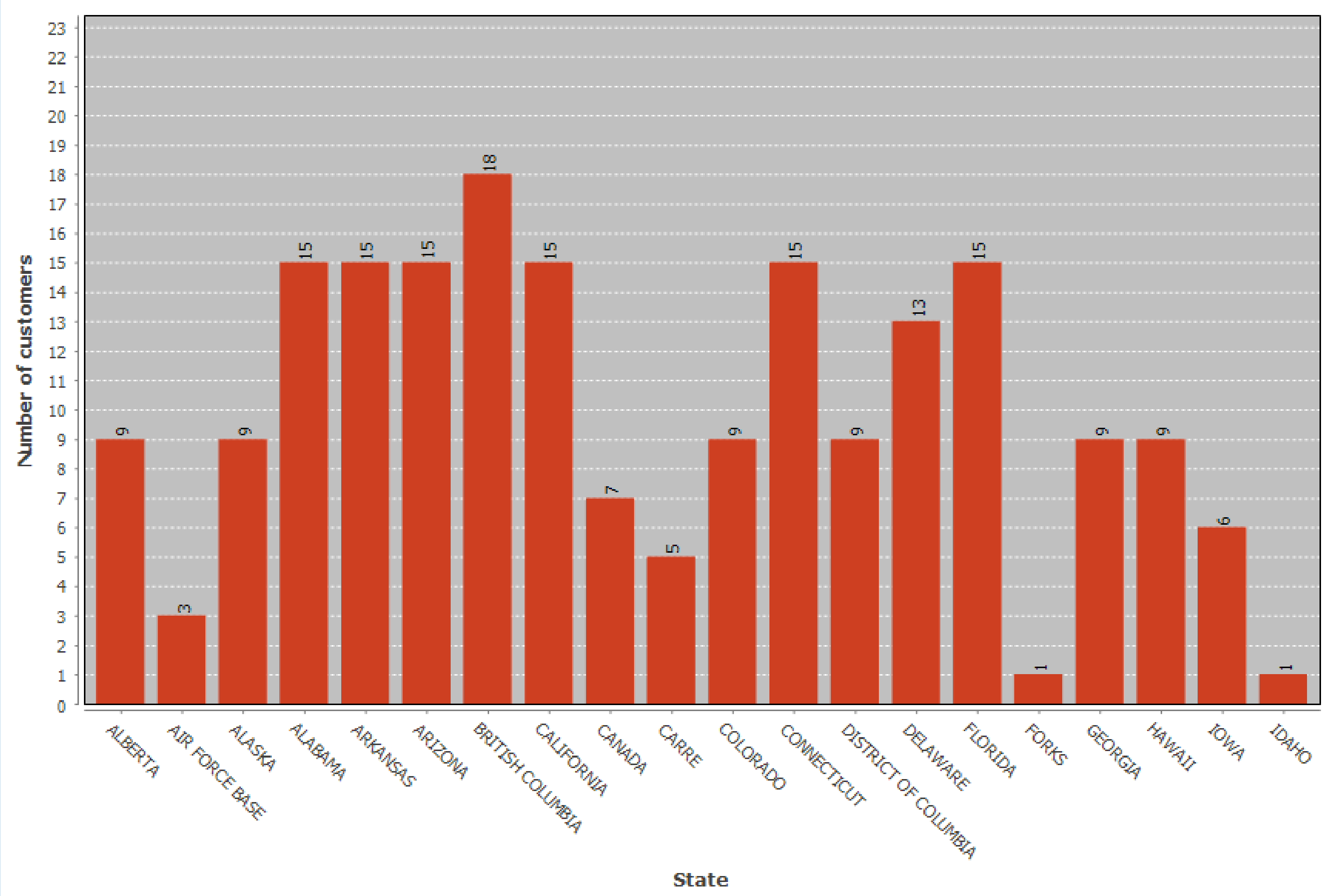
	ABC rating	123 length	123 ?column?	123 count
1	PG	68	88	1
2	PG-13	181	201	2
3	PG-13	162	182	3
4	R	172	192	2
5	G	102	122	3
6	R	143	163	1
7	G	110	130	3

To remove a column, in the Grouping Configuration window, in the **Columns** area, click the column name, then **Remove** and **OK**. To remove all columns, click **Clear** and **OK**.
Another way to remove a column is to click the column and then the **Remove grouping column** button ( ) in the panel`s toolbar.
Clicking the **Clear grouping** button ( ) removes all results from the Grouping panel.

Managing Charts

The default grid view of query resulting data is not very much impressive, especially to business analysts and other end users. The **Charts** feature lets you quickly and easily turn your SELECT queries' output into a colorized bar chart.

Note: Analytical Charts present only in DBeaver Enterprise Edition

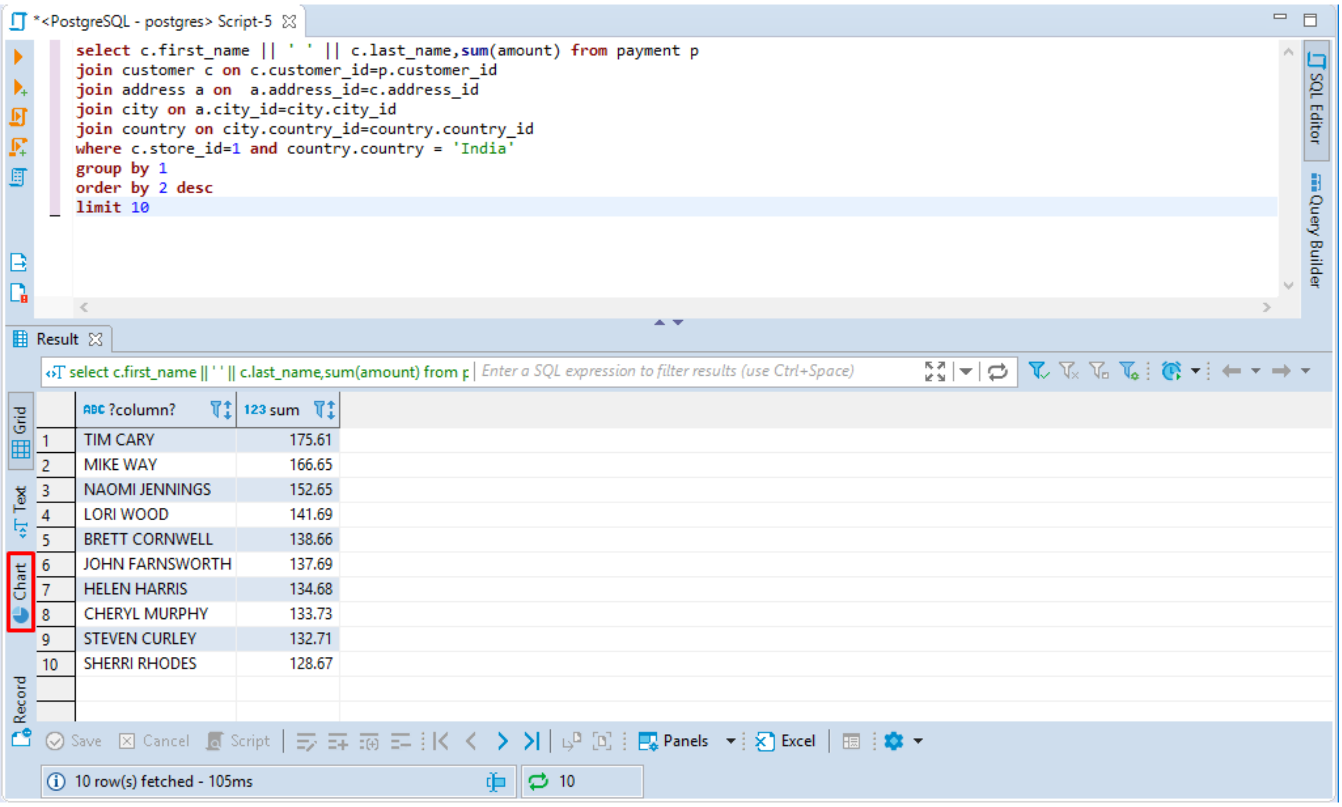


You can easily visualize your data by creating a chart bar both in **SQL Editor** and **Data Editor**.

Creating Charts In SQL Editor

Visual representation of vast data permits the analytical reasoning process to become faster and more focused. Charts make it easy for analysts to perceive salient aspects of their data quickly.

To build a bar chart in the **SQL Editor**, press the **Charts** button ◔ in the left vertical toolbar of the query results area.



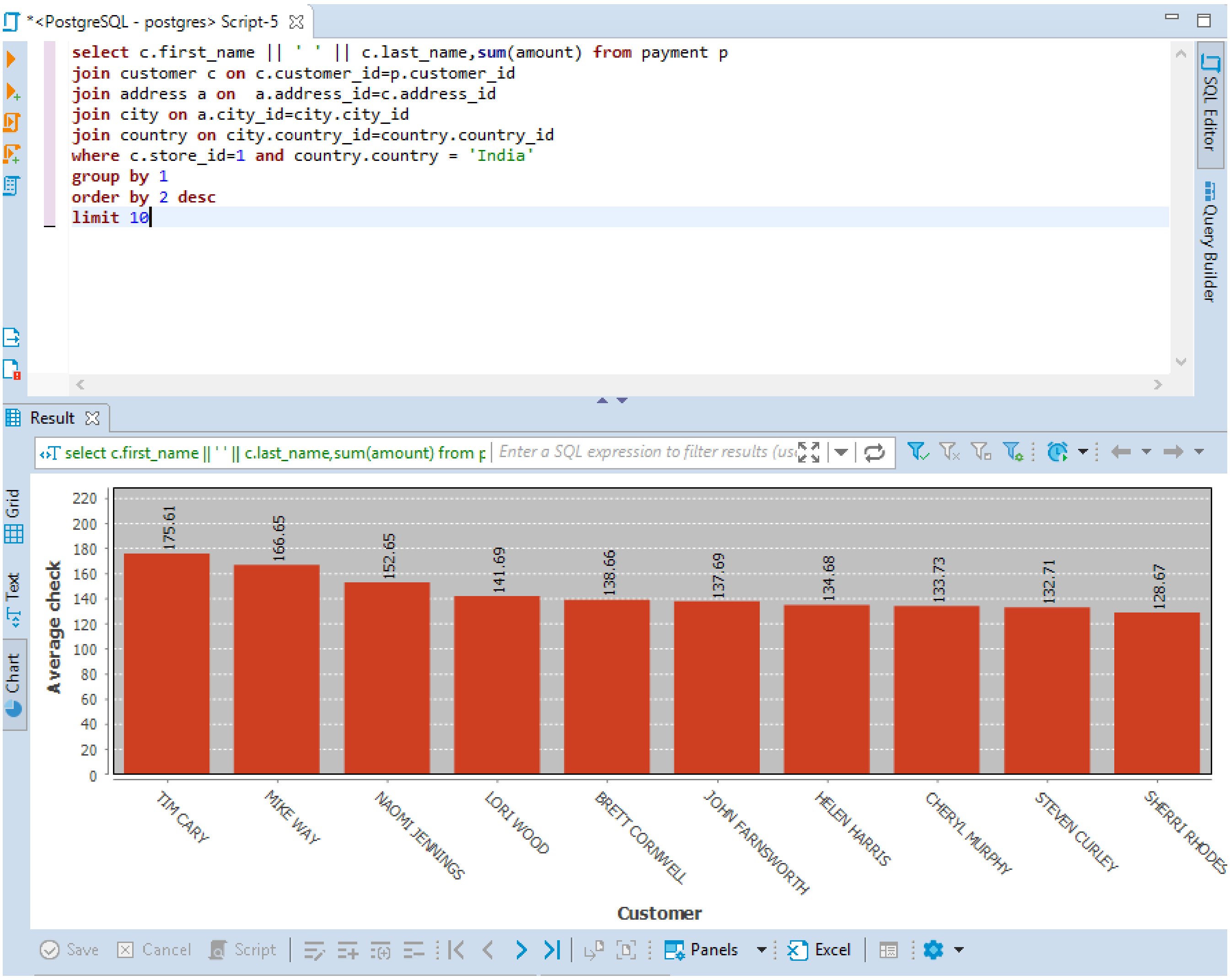
*<PostgreSQL - postgres> Script-5
SQL Editor
Query Builder
Result
Grid
Text
Chart
Record
Enter a SQL expression to filter results (use Ctrl+Space)
Save
Cancel
Script
Panels
Excel

```
select c.first_name || ' ' || c.last_name,sum(amount) from payment p
join customer c on c.customer_id=p.customer_id
join address a on  a.address_id=c.address_id
join city on a.city_id=city.city_id
join country on city.country_id=country.country_id
where c.store_id=1 and country.country = 'India'
group by 1
order by 2 desc
limit 10
```

	?column?	sum
1	TIM CARY	175.61
2	MIKE WAY	166.65
3	NAOMI JENNINGS	152.65
4	LORI WOOD	141.69
5	BRETT CORNWELL	138.66
6	JOHN FARNSWORTH	137.69
7	HELEN HARRIS	134.68
8	CHERYL MURPHY	133.73
9	STEVEN CURLEY	132.71
10	SHERRI RHODES	128.67

10 row(s) fetched - 105ms

A bar chart will be created.



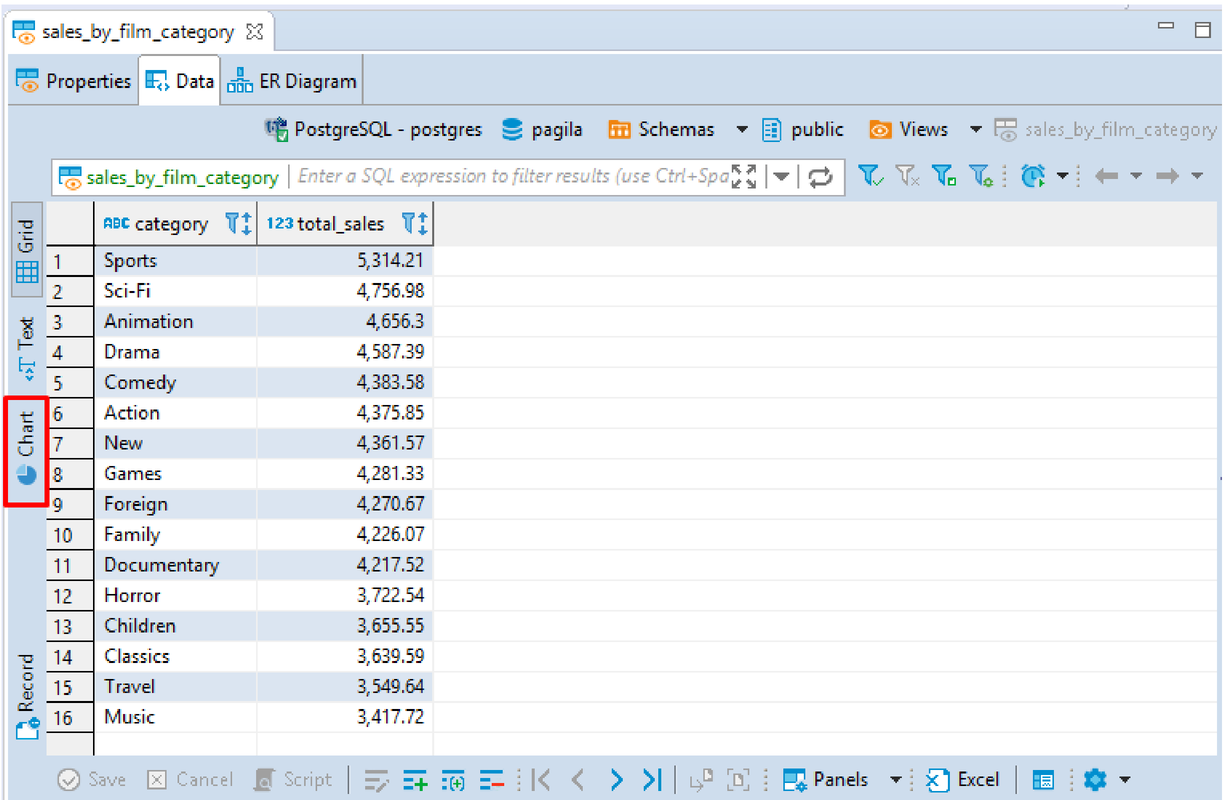
```
select c.first_name || ' ' || c.last_name,sum(amount) from payment p
join customer c on c.customer_id=p.customer_id
join address a on  a.address_id=c.address_id
join city on a.city_id=city.city_id
join country on city.country_id=country.country_id
where c.store_id=1 and country.country = 'India'
group by 1
order by 2 desc
limit 10
```

Creating Charts In Data Editor

Charts can be very helpful for visualizing structured analytical data stored as **Views**, for example.

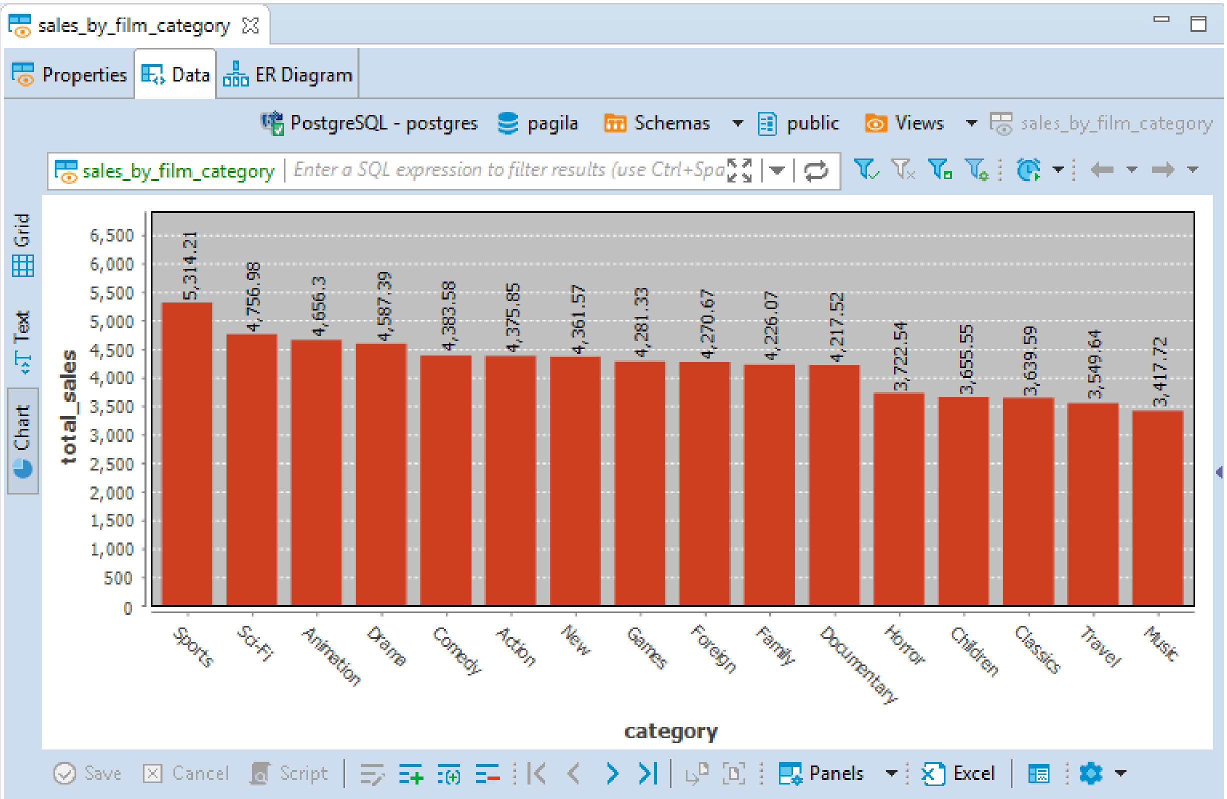
You can also create a chart for any table but you have to structure its data by sorting and applying various filters to its columns first. All the structural changes you make will then affect the chart you build, this way you can adjust the chart representation to the desired one.

To build a chart in the **Data Editor**, press the **Charts** button in the left vertical toolbar.



	category	total_sales
1	Sports	5,314.21
2	Sci-Fi	4,756.98
3	Animation	4,656.3
4	Drama	4,587.39
5	Comedy	4,383.58
6	Action	4,375.85
7	New	4,361.57
8	Games	4,281.33
9	Foreign	4,270.67
10	Family	4,226.07
11	Documentary	4,217.52
12	Horror	3,722.54
13	Children	3,655.55
14	Classics	3,639.59
15	Travel	3,549.64
16	Music	3,417.72

A bar chart will be created.

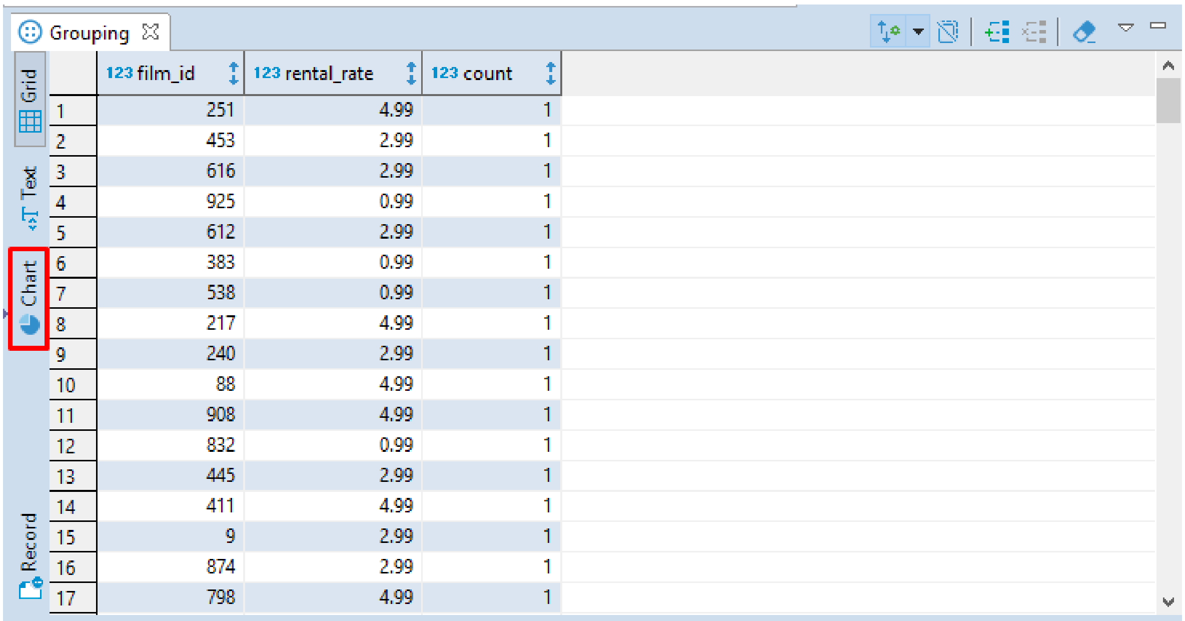


Note, that by default, the data for axis Y is taken from the first column of the table containing numeric values.

Creating Charts In Grouping Panel

Such analytical tool as the Grouping Panel also supports the **Charts** feature. In a chart built for a table containing the grouping results for one or more columns of a data table, you can easily change axes X and axes Y source data by switching the columns in the **Charts Editor**.

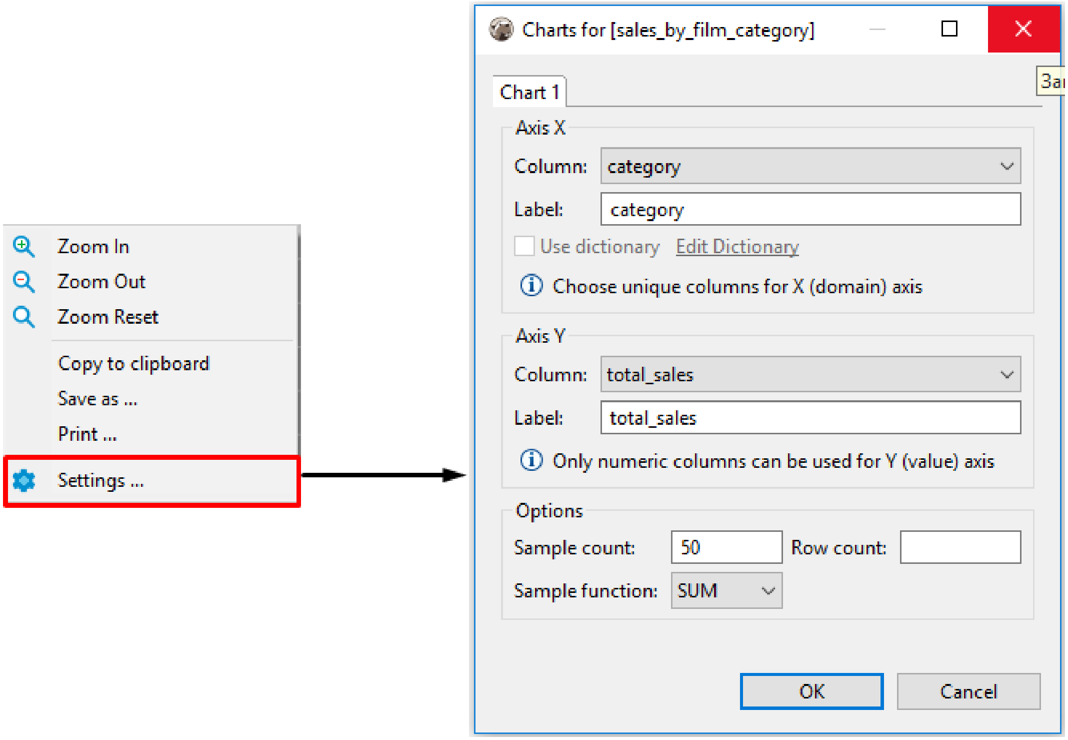
To build a chart in **Grouping Panel**, press the **Charts** button in the left vertical toolbar.



	123 film_id	123 rental_rate	123 count
1	251	4.99	1
2	453	2.99	1
3	616	2.99	1
4	925	0.99	1
5	612	2.99	1
6	383	0.99	1
7	538	0.99	1
8	217	4.99	1
9	240	2.99	1
10	88	4.99	1
11	908	4.99	1
12	832	0.99	1
13	445	2.99	1
14	411	4.99	1
15	9	2.99	1
16	874	2.99	1
17	798	4.99	1

Editing Chart Settings

To edit chart settings select the **Charts...** option in the chart's context menu and the **Chart Editor** will appear.



Axis X

Column: category

Label: category

□ Use dictionary Edit Dictionary

ⓘ Choose unique columns for X (domain) axis

Axis Y

Column: total_sales

Label: total_sales

ⓘ Only numeric columns can be used for Y (value) axis

Options

Sample count: 50 Row count:

Sample function: SUM

The following chart settings can be adjusted:

Setting Axis X

1. In the **Column** drop down list of available columns select a column whose data will be used on axis X of the bar chart. Make sure you choose unique columns for X axis.

2. Define a user-friendly axis name in the **Label** text field.

Setting Axis Y

1. In the **Column** drop down list of available columns select a column whose data will be used on axis Y of the bar chart. **Note**, that only columns containing numeric data can be used for axis Y.

2. Define a user-friendly axis name in the **Label** text field.

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;
- Sample function - an aggregate function where the values of multiple rows are grouped together to form a single summary value displayed on axis Y.

The following sample functions are suppoerted:

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

Copying to clipboard

You can copy a chart to clipboard by selecting the **Copy to clipboard** option in the chart's context menu.

Exporting Charts

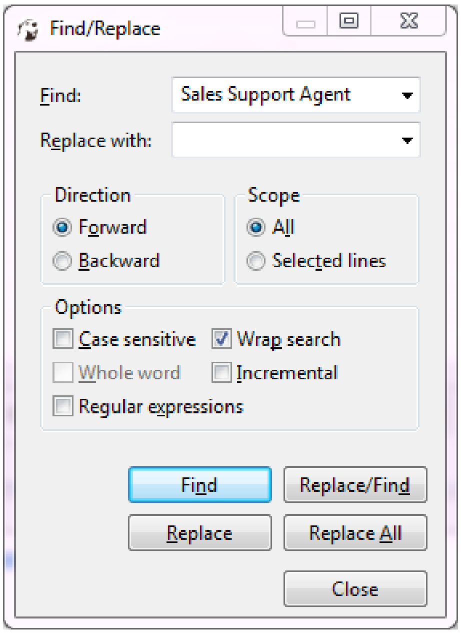
You can export a chart into PNG format by selecting the **Save as...** option in the chart's context menu.

Printing Charts

You can print a chart by selecting the **Print...** option in the chart's context menu.

Data Search

To search for data in the result set, press `CTRL+F`. The standard Find/Replace search dialog box opens:



Find: Sales Support Agent

Replace with:

Direction
● Forward
○ Backward

Scope
● All
○ Selected lines

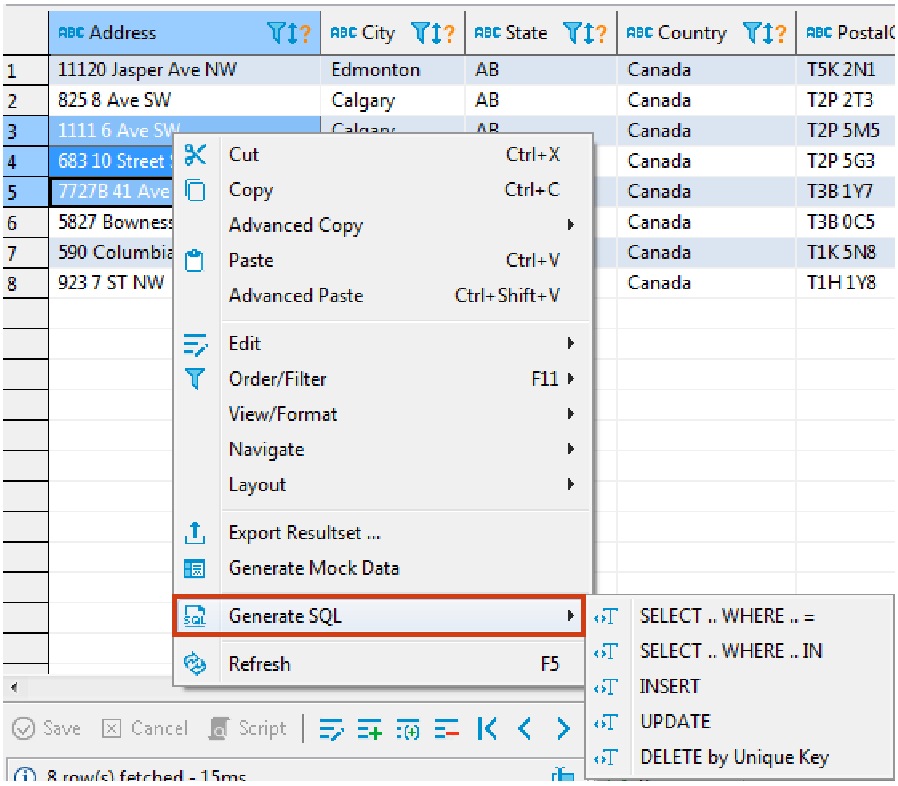
Options
☐ Case sensitive ☑ Wrap search
☐ Whole word ☐ Incremental
☐ Regular expressions

You can also use the Find and Replace feature.

NOTE: The system searches only in already fetched rows.

SQL Generation

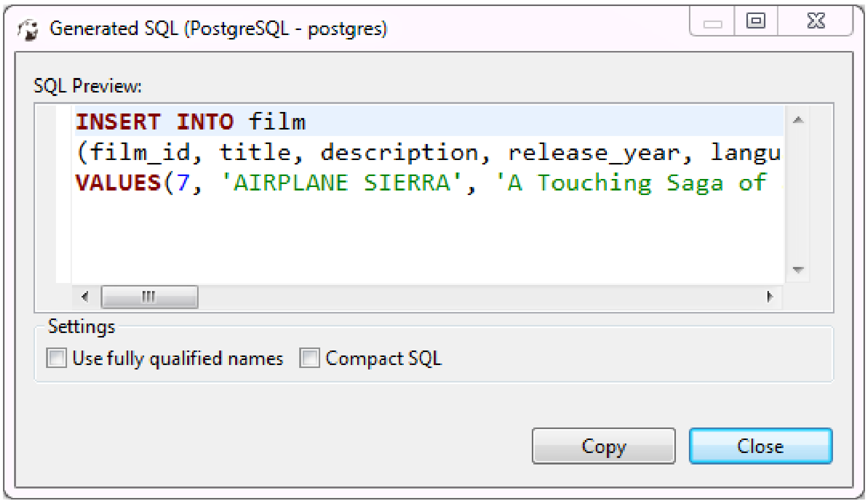
You can generate SQL statements (SELECT/INSERT/UPDATE/DELETE) based on selected rows. To generate SQL, right-click the selected rows and click **Generate SQL** and then one of the SQL commands on the context menu:



	Address	City	State	Country	PostalC
1	11120 Jasper Ave NW	Edmonton	AB	Canada	T5K 2N1
2	825 8 Ave SW	Calgary	AB	Canada	T2P 2T3
3	1111 6 Ave SW	Calgary	AB	Canada	T2P 5M5
4	683 10 Street			Canada	T2P 5G3
5	7727B 41 Ave			Canada	T3B 1Y7
6	5827 Bowness			Canada	T3B 0C5
7	590 Columbia			Canada	T1K 5N8
8	923 7 ST NW			Canada	T1H 1Y8

8 row(s) fetched - 15ms

The SQL result opens in a separate window where you can view and copy it:



Generated SQL (PostgreSQL - postgres)

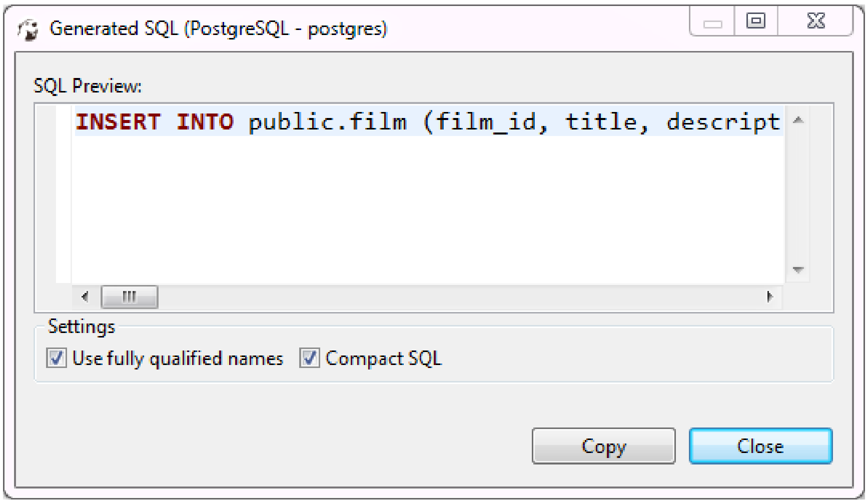
SQL Preview:

```
INSERT INTO film
(film_id, title, description, release_year, langu
VALUES(7, 'AIRPLANE SIERRA', 'A Touching Saga of
```

Settings

☐ Use fully qualified names ☐ Compact SQL

To use table names in the format '[schema name].[table name]', select the **Use fully qualified names** checkbox.
To wrap the SQL query into one line, select the **Compact SQL** checkbox:



```
INSERT INTO public.film (film_id, title, descript
```

☑ Use fully qualified names ☑ Compact SQL

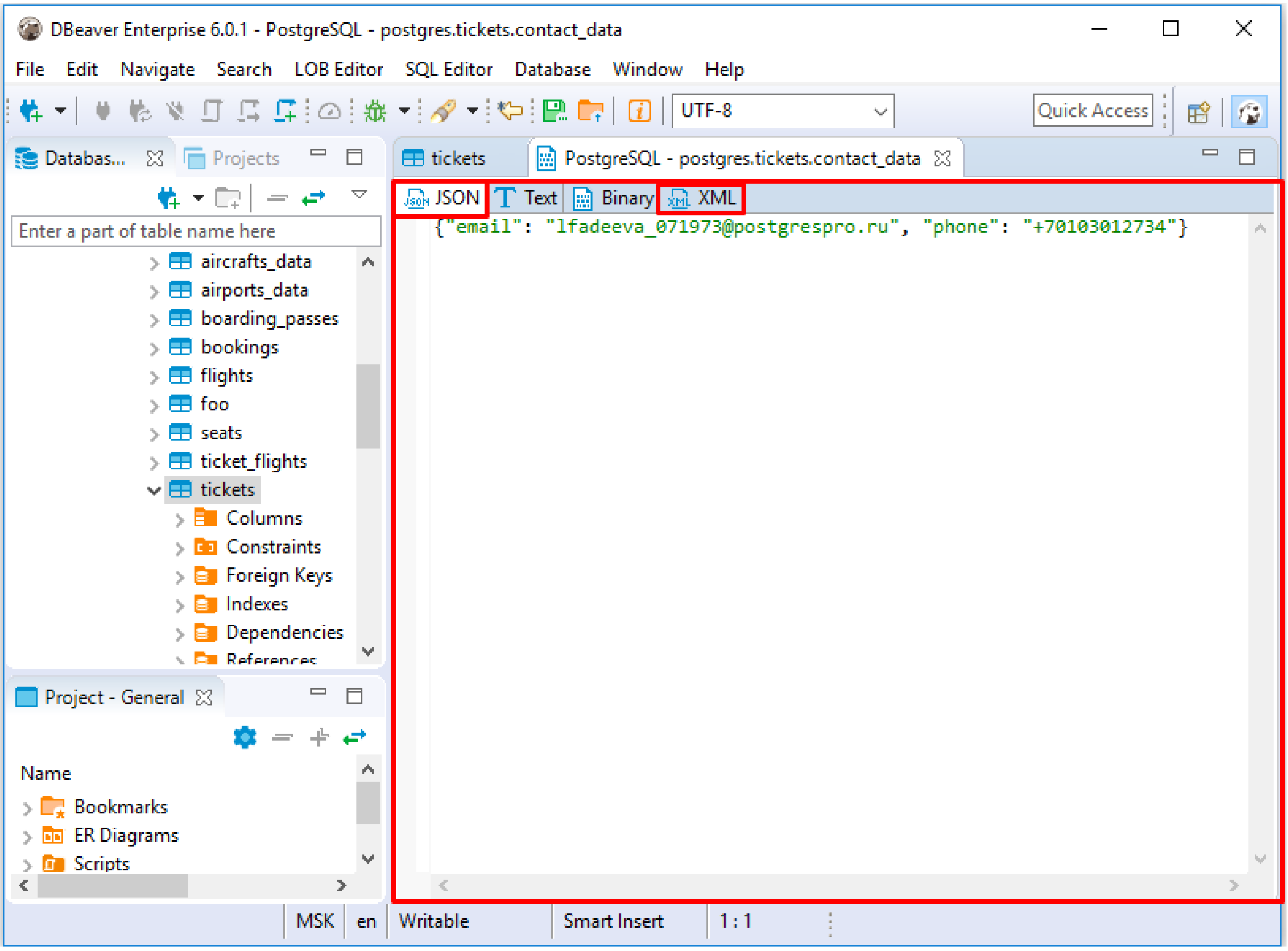
Working with XML and JSON

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 big amount of data may require a larger editor, you may want to save XML/JSON scripts to a local file or upload this type of data from a local file.

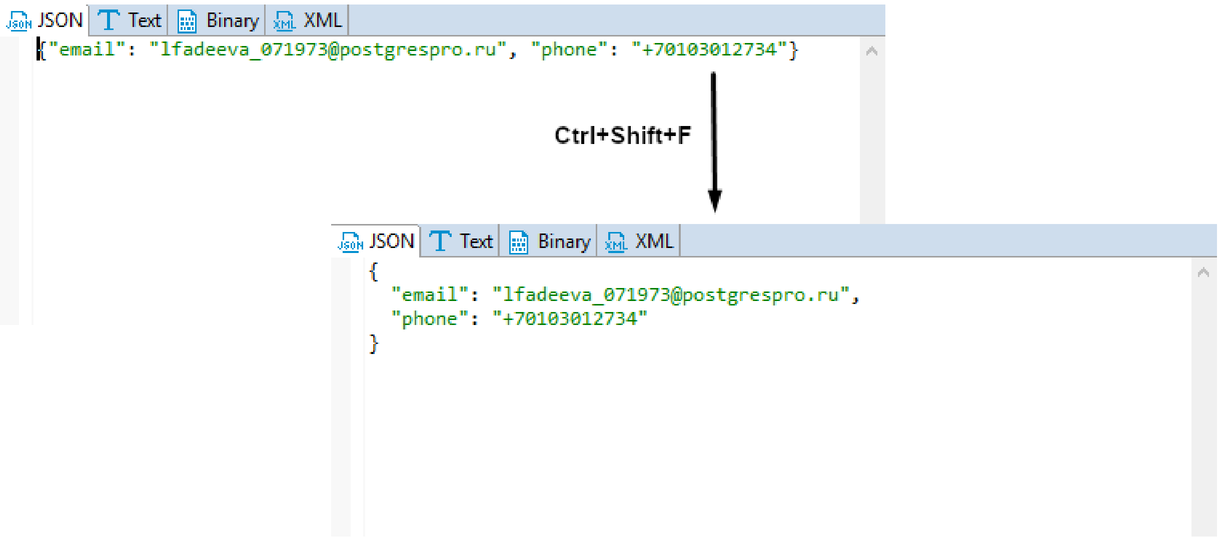
To open the full-size XML/JSON editor click the cell containing data in XML/JSON format and press `Shift`+`Enter`.

By default the editor opens on JSON tab, open XML tab to modify XML data.



```
{"email": "lfadeeva_071973@postgrespro.ru", "phone": "+70103012734"}
```

To auto-format XML/JSON script press `Ctrl`+`Shift`+`F` keyboard buttons.

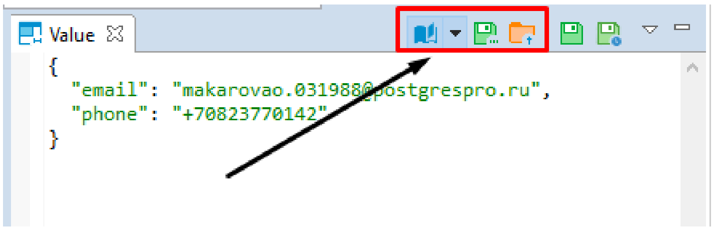


```
{"email": "lfadeeva_071973@postgrespro.ru", "phone": "+70103012734"}
```

```
{
  "email": "lfadeeva_071973@postgrespro.ru",
  "phone": "+70103012734"
}
```

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.

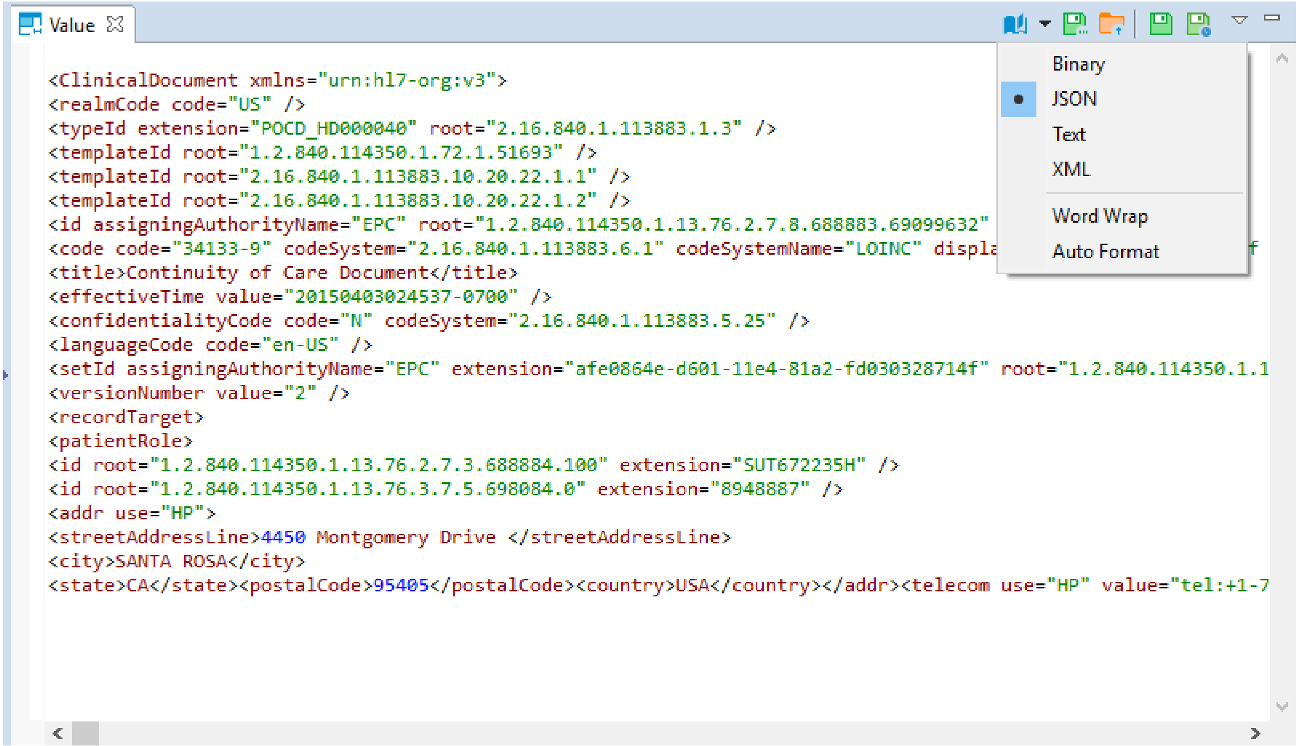


```
{
  "email": "makarovao.031988@postgrespro.ru",
  "phone": "+70823770142"
}
```

To upload data from a local file, press the **Load from file...** button .

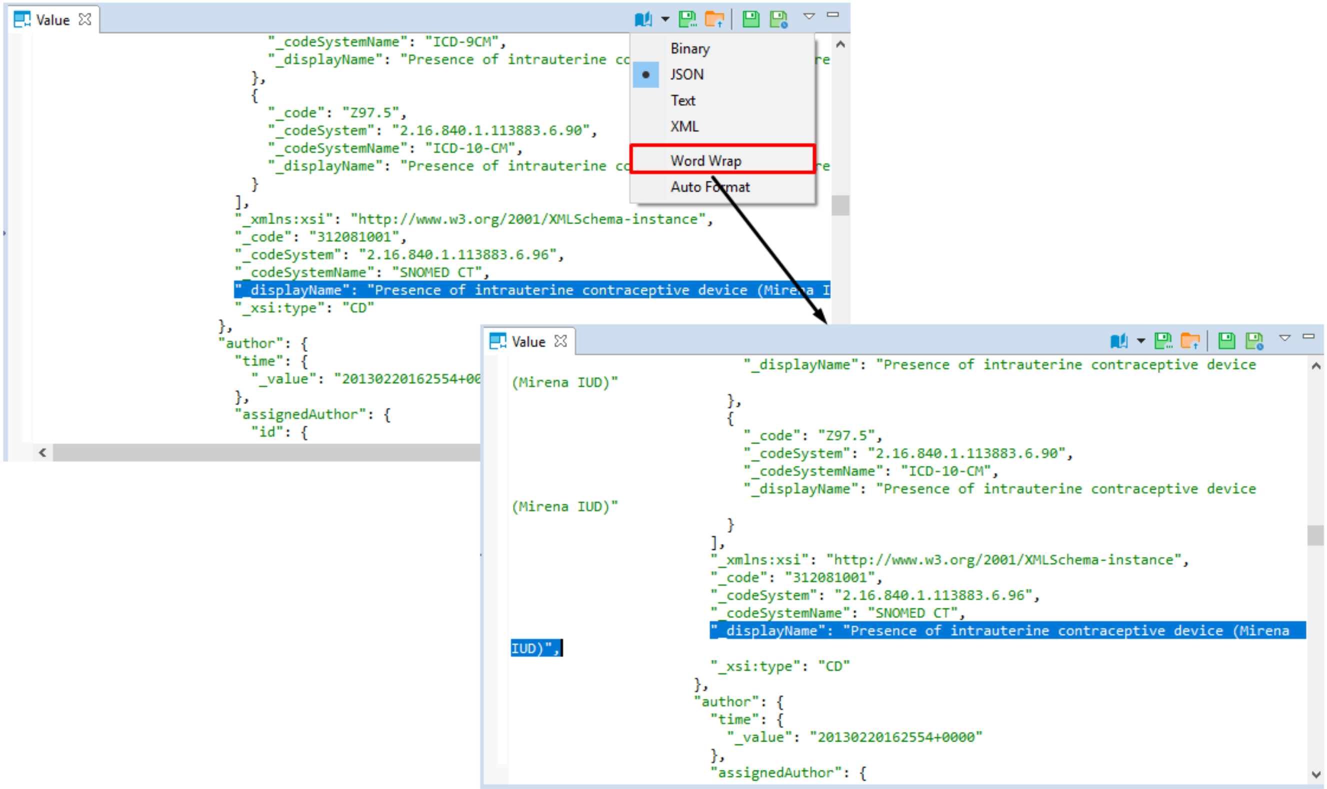
To save the content to a local file, press the **Save to file...** button .

To switch between the formats, press the **Content viewer settings** button and select the format.



```
<ClinicalDocument xmlns="urn:hl7-org:v3">
<realmCode code="US" />
<typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3" />
<templateId root="1.2.840.114350.1.72.1.51693" />
<templateId root="2.16.840.1.113883.10.20.22.1.1" />
<templateId root="2.16.840.1.113883.10.20.22.1.2" />
<id assigningAuthorityName="EPC" root="1.2.840.114350.1.13.76.2.7.8.688883.69099632"
<code code="34133-9" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC" displa
<title>Continuity of Care Document</title>
<effectiveTime value="20150403024537-0700" />
<confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25" />
<languageCode code="en-US" />
<setId assigningAuthorityName="EPC" extension="afe0864e-d601-11e4-81a2-fd030328714f" root="1.2.840.114350.1.1
<versionNumber value="2" />
<recordTarget>
<patientRole>
<id root="1.2.840.114350.1.13.76.2.7.3.688884.100" extension="SUT672235H" />
<id root="1.2.840.114350.1.13.76.3.7.5.698084.0" extension="8948887" />
<addr use="HP">
<streetAddressLine>4450 Montgomery Drive </streetAddressLine>
<city>SANTA ROSA</city>
<state>CA</state><postalCode>95405</postalCode><country>USA</country></addr><telecom use="HP" value="tel:+1-7
```

Use **Word Wrap** feature that wraps the text within a screen.

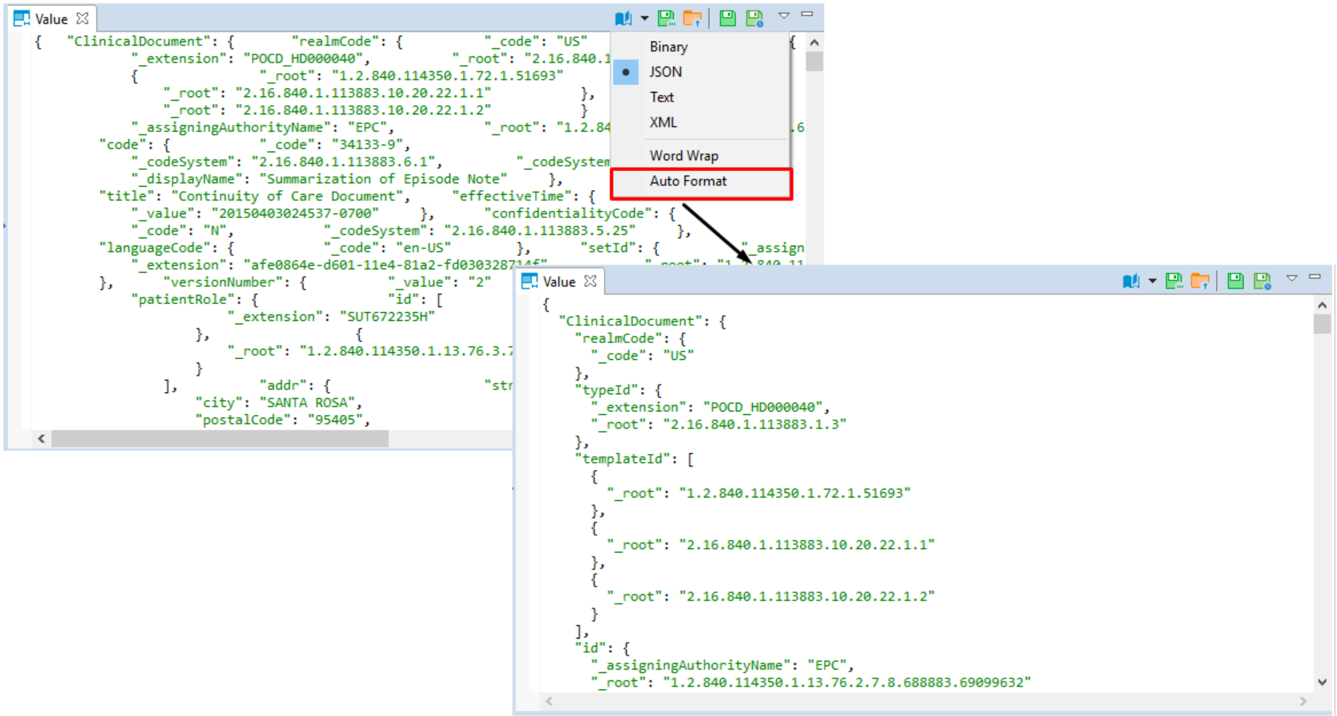


```
        "_codeSystemName": "ICD-9CM",
        "_displayName": "Presence of intrauterine cc
      },
      {
        "_code": "Z97.5",
        "_codeSystem": "2.16.840.1.113883.6.90",
        "_codeSystemName": "ICD-10-CM",
        "_displayName": "Presence of intrauterine cc
      }
    ],
    "_xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
    "_code": "312081001",
    "_codeSystem": "2.16.840.1.113883.6.96",
    "_codeSystemName": "SNOMED CT",
    "_displayName": "Presence of intrauterine contraceptive device (Mirena I
    "_xsi:type": "CD"
  },
  "author": {
    "time": {
      "_value": "20130220162554+00
    },
    "assignedAuthor": {
      "id": {
```

```
                                  "_displayName": "Presence of intrauterine contraceptive device
(Mirena IUD)"
                                },
                                {
                                  "_code": "Z97.5",
                                  "_codeSystem": "2.16.840.1.113883.6.90",
                                  "_codeSystemName": "ICD-10-CM",
                                  "_displayName": "Presence of intrauterine contraceptive device
(Mirena IUD)"
                                }
                              ],
                              "_xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
                              "_code": "312081001",
                              "_codeSystem": "2.16.840.1.113883.6.96",
                              "_codeSystemName": "SNOMED CT",
                              "_displayName": "Presence of intrauterine contraceptive device (Mirena
IUD)",

                              "_xsi:type": "CD"
                            },
                            "author": {
                              "time": {
                                "_value": "20130220162554+0000"
                              },
                              "assignedAuthor": {
```

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.



```
{   "ClinicalDocument": {       "realmCode": {          "_code": "US"
        "_extension": "POCD_HD000040",          "_root": "2.16.840.1
        {               "_root": "1.2.840.114350.1.72.1.51693"
            "_root": "2.16.840.1.113883.10.20.22.1.1"           },
            "_root": "2.16.840.1.113883.10.20.22.1.2"           }
        "_assigningAuthorityName": "EPC",           "_root": "1.2.84
    "code": {           "_code": "34133-9",
        "_codeSystem": "2.16.840.1.113883.6.1",         "_codeSystem
        "_displayName": "Summarization of Episode Note"     },
    "title": "Continuity of Care Document",     "effectiveTime": {
        "_value": "20150403024537-0700"     },      "confidentialityCode": {
        "_code": "N",           "_codeSystem": "2.16.840.1.113883.5.25"     },
    "languageCode": {           "_code": "en-US"        },      "setId": {
        "_extension": "afe0864e-d601-11e4-81a2-fd0303287
    },      "versionNumber": {          "_value": "2"
        "patientRole": {            "id": [
                "_extension": "SUT672235H"
            },          {
                "_root": "1.2.840.114350.1.13.76.3.7
            }
        ],          "addr": {                   "str
            "city": "SANTA ROSA",
            "postalCode": "95405",
```

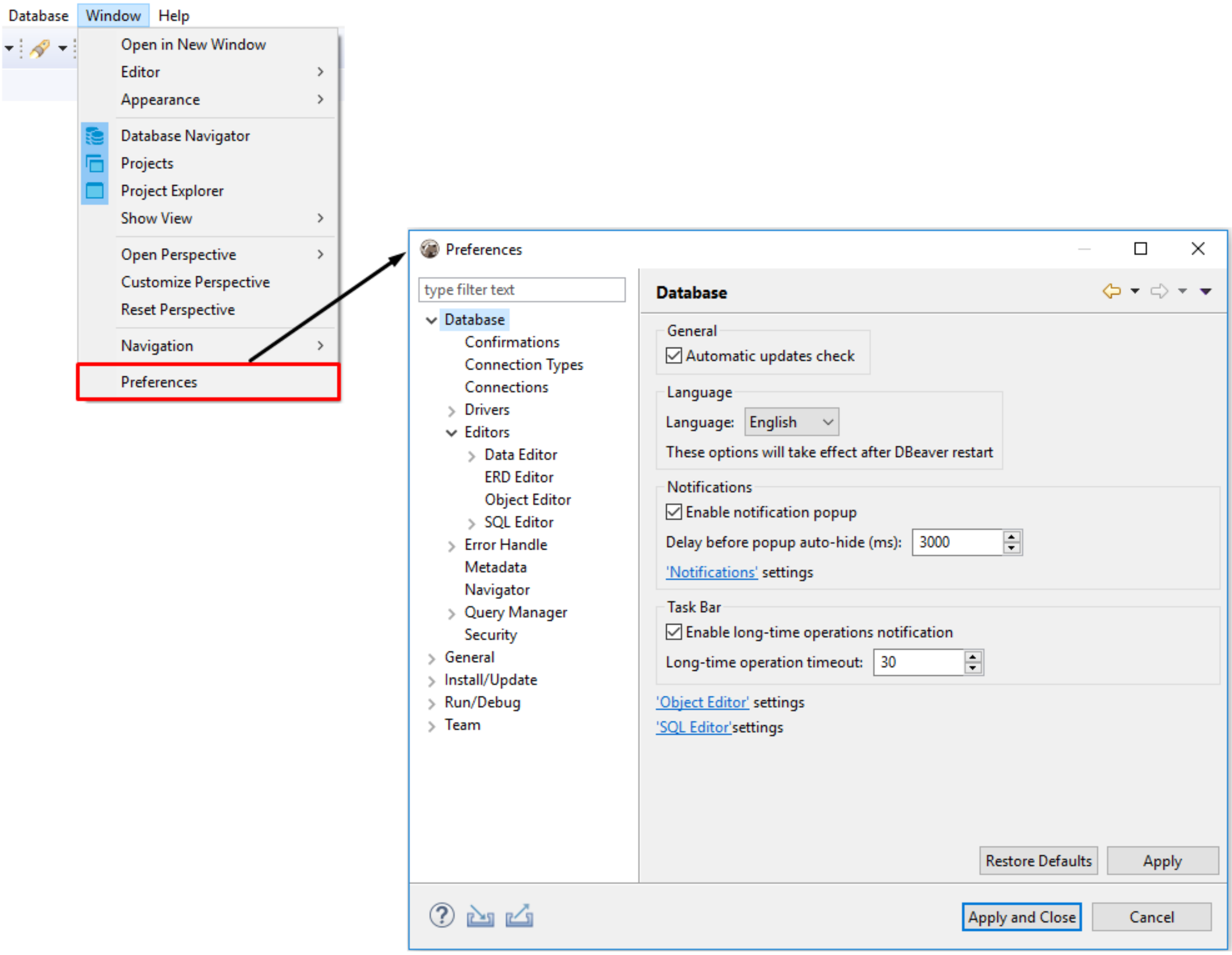
```
{
  "ClinicalDocument": {
    "realmCode": {
      "_code": "US"
    },
    "typeId": {
      "_extension": "POCD_HD000040",
      "_root": "2.16.840.1.113883.1.3"
    },
    "templateId": [
      {
        "_root": "1.2.840.114350.1.72.1.51693"
      },
      {
        "_root": "2.16.840.1.113883.10.20.22.1.1"
      },
      {
        "_root": "2.16.840.1.113883.10.20.22.1.2"
      }
    ],
    "id": {
      "_assigningAuthorityName": "EPC",
      "_root": "1.2.840.114350.1.13.76.2.7.8.688883.69099632"
```

To learn more about **Value** panel, see Panels.

Managing Data Formats

The DBeaver formatting functions allow you to set up database locale and change datasource format settings. This feature can be very useful on database migration, for example.

To change data format settings use option Window -> Preferences in main menu.



Database

General

☑ Automatic updates check

Language

Language: English

These options will take effect after DBeaver restart

Notifications

☑ Enable notification popup

Delay before popup auto-hide (ms): 3000

'Notifications' settings

Task Bar

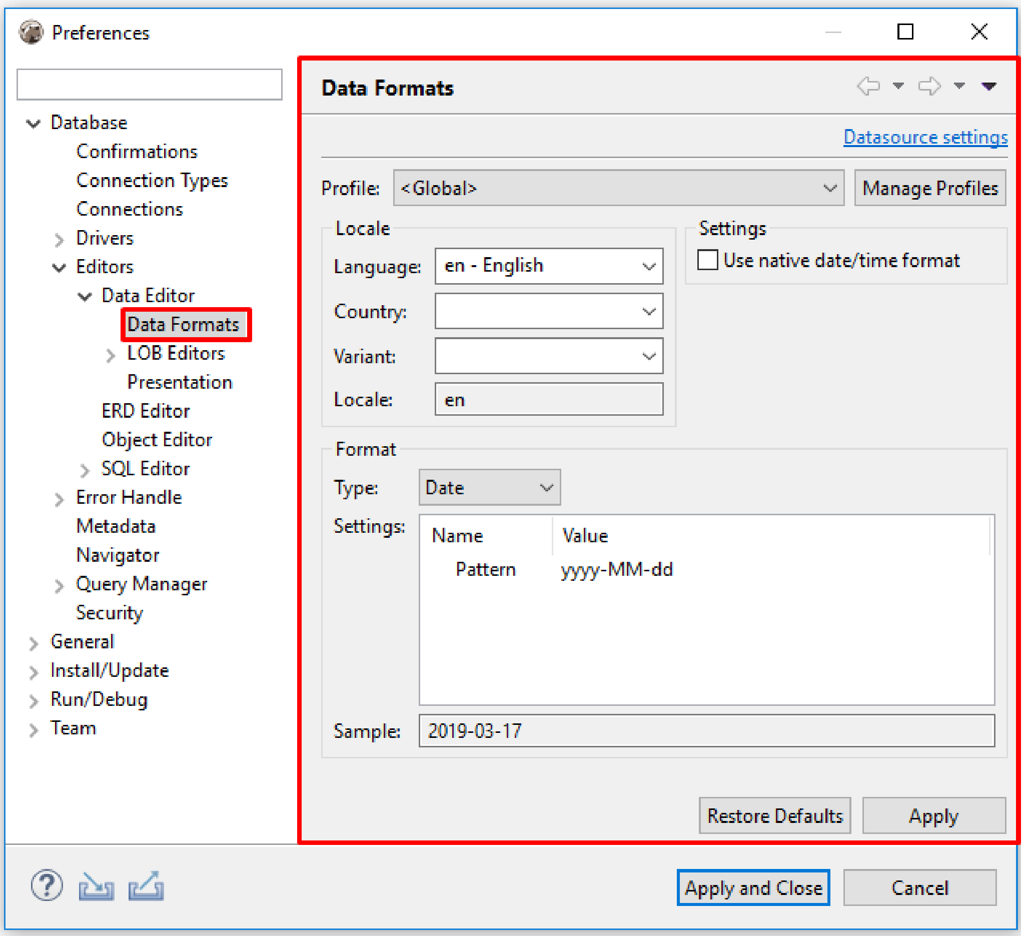
☑ Enable long-time operations notification

Long-time operation timeout: 30

'Object Editor' settings

'SQL Editor'settings

In the Preferences dialog box go to Database -> Editors -> DataEditor -> Data Formats.



Profile: <Global>

Locale

Language: en - English

Country:

Variant:

Locale: en

Settings

□ Use native date/time format

Format

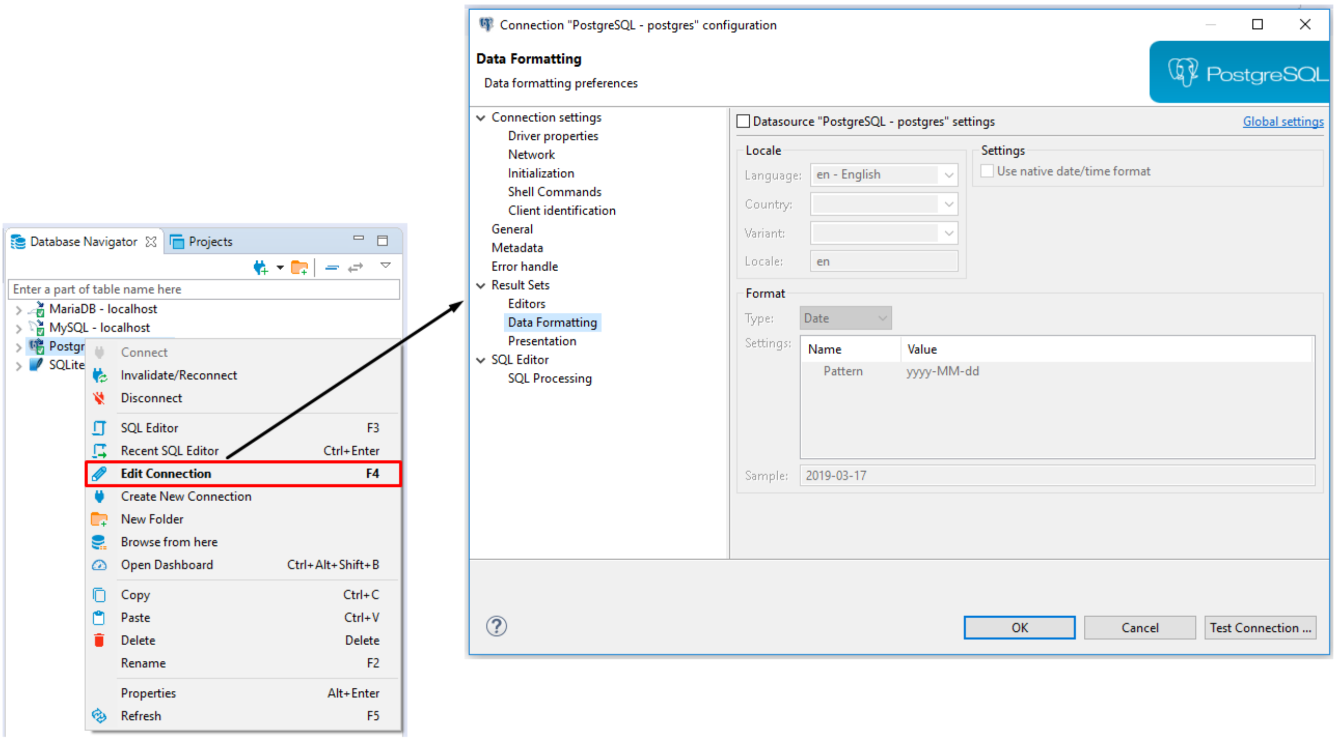
Type: Date

Settings:

Name	Value
Pattern	yyyy-MM-dd

Sample: 2019-03-17

Or, in the Database Navigator right-click a connection and select **Edit Connection** menu option.



□ Datasource "PostgreSQL - postgres" settings

Locale

Language: en - English

Country:

Variant:

Locale: en

Settings

□ Use native date/time format

Format

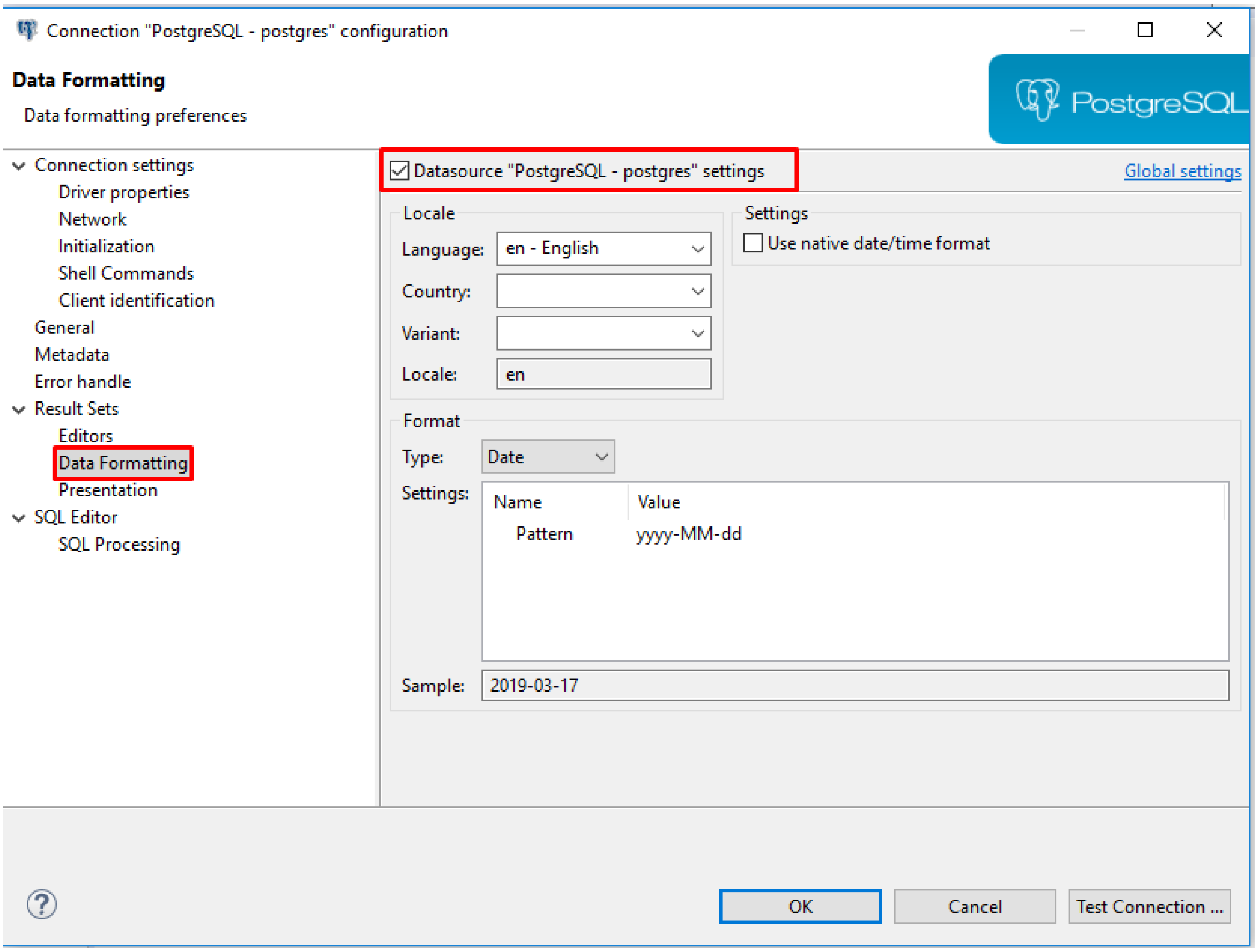
Type: Date

Settings:

Name	Value
Pattern	yyyy-MM-dd

Sample: 2019-03-17

In the right area of the opened **Data formatting preferences** dialog window go to Result Sets -> Data Formatting and select the **Datasource settings** check box in the left area to customize the data format settings.


Connection "PostgreSQL - postgres" configuration

Data Formatting

Data formatting preferences

☑ Datasource "PostgreSQL - postgres" settings

Locale

Language: en - English

Country:

Variant:

Locale: en

Settings

□ Use native date/time format

Format

Type: Date

Settings:

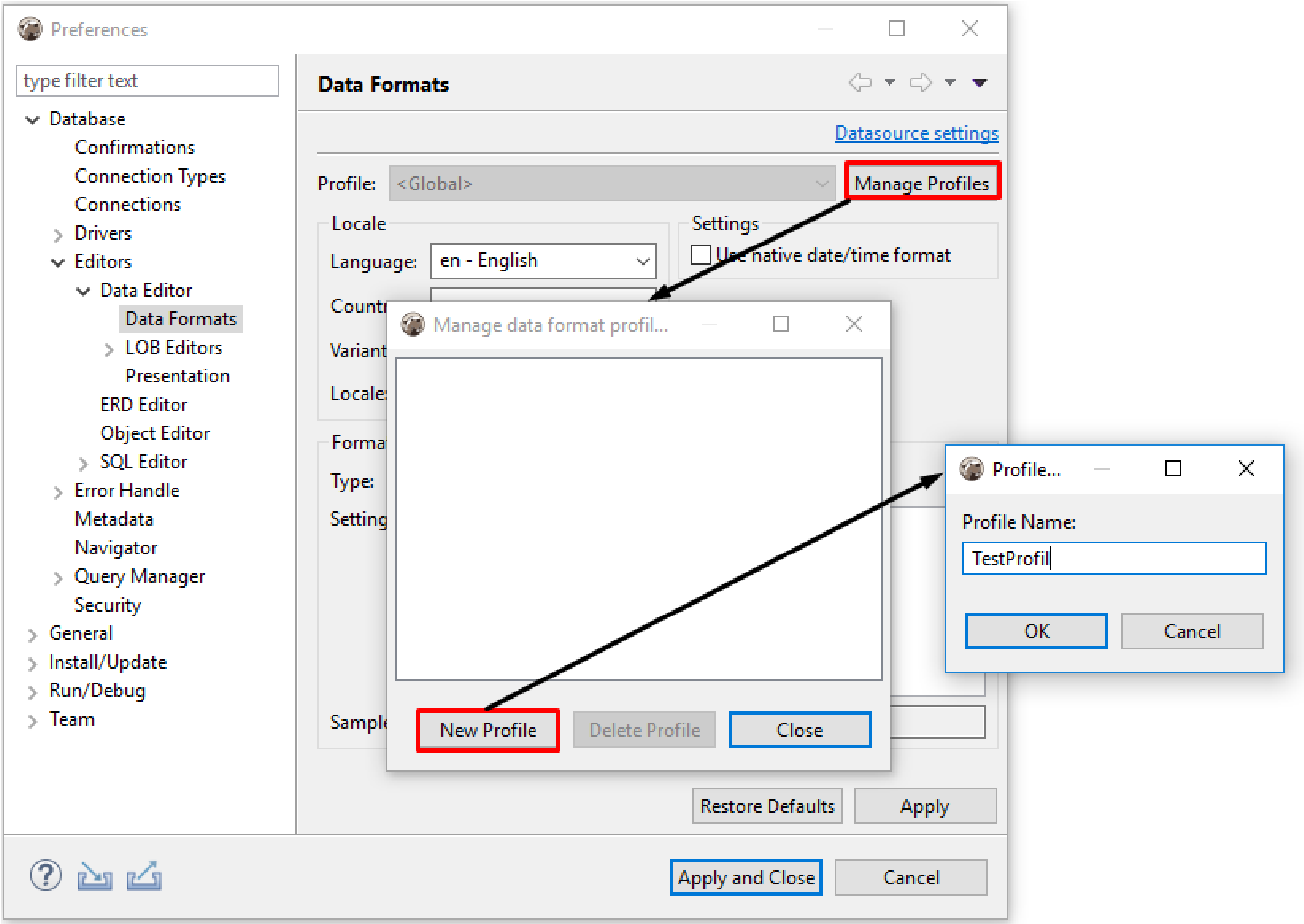
Name	Value
Pattern	yyyy-MM-dd

Sample: 2019-03-17

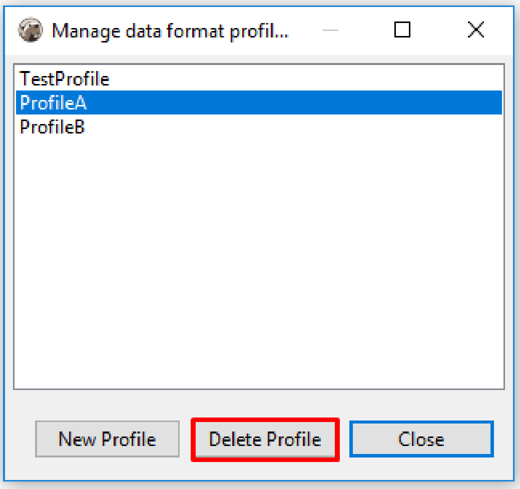
Data Format Profiles

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**.



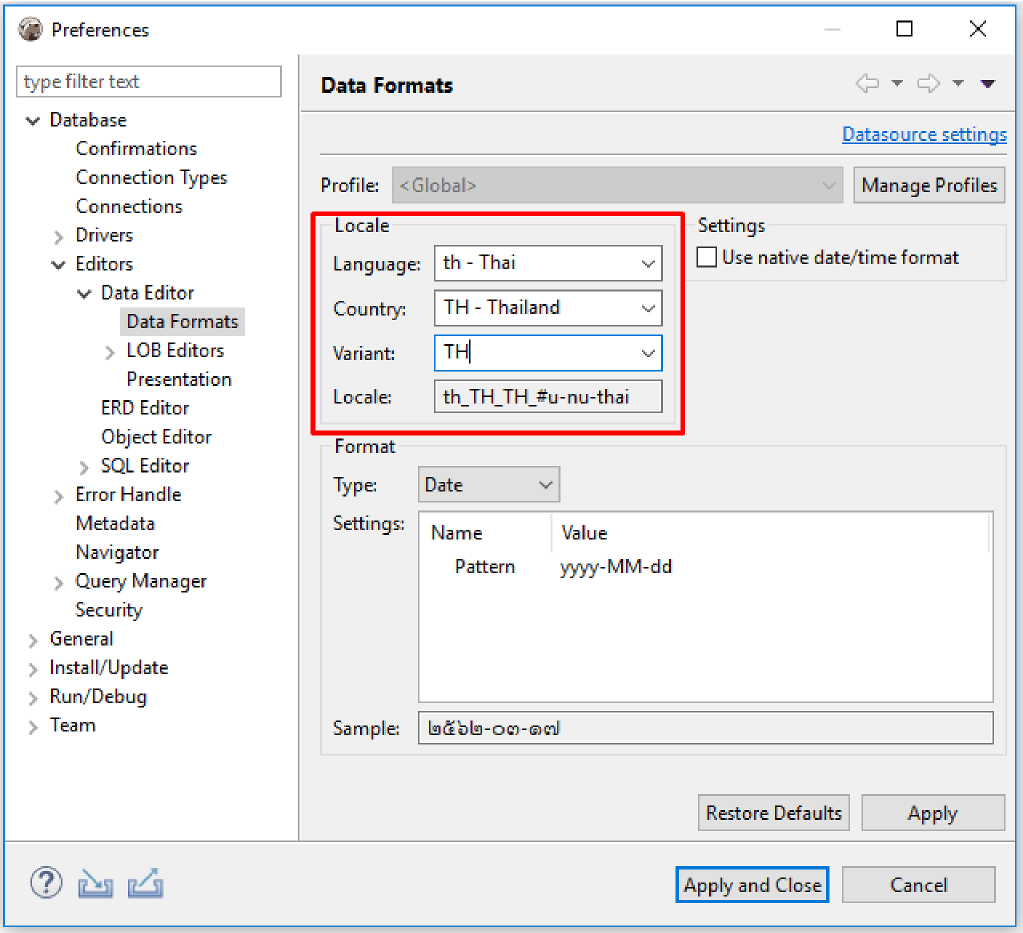
To delete a data format profile press the **Manage Profiles** button, then in the opened dialog window select the profile you want to delete and press the button **Delete Profile**.



Changing Data Formats

The following groups of data format settings can be adjusted:

Locale



Profile: <Global>

Locale

Language: th - Thai

Country: TH - Thailand

Variant: TH

Locale: th_TH_TH_#u-nu-thai

Settings

☐ Use native date/time format

Format

Type: Date

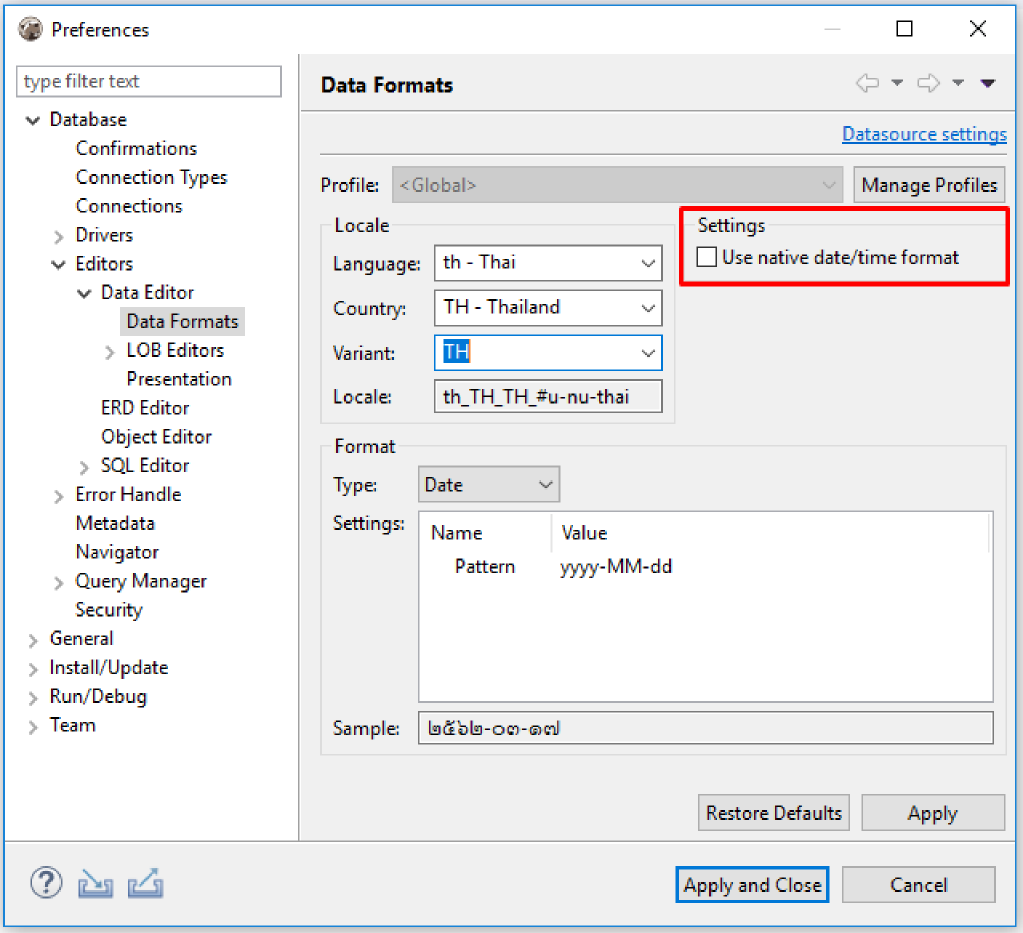
Settings:

Name	Value
Pattern	yyyy-MM-dd

Sample: ๒๕๖๒-๐๓-๑๗

To define this setting select a language, country and variant if available.

Native Date/Time Mode



Profile: <Global>

Locale

Language: th - Thai

Country: TH - Thailand

Variant: TH

Locale: th_TH_TH_#u-nu-thai

Settings

☐ Use native date/time format

Format

Type: Date

Settings:

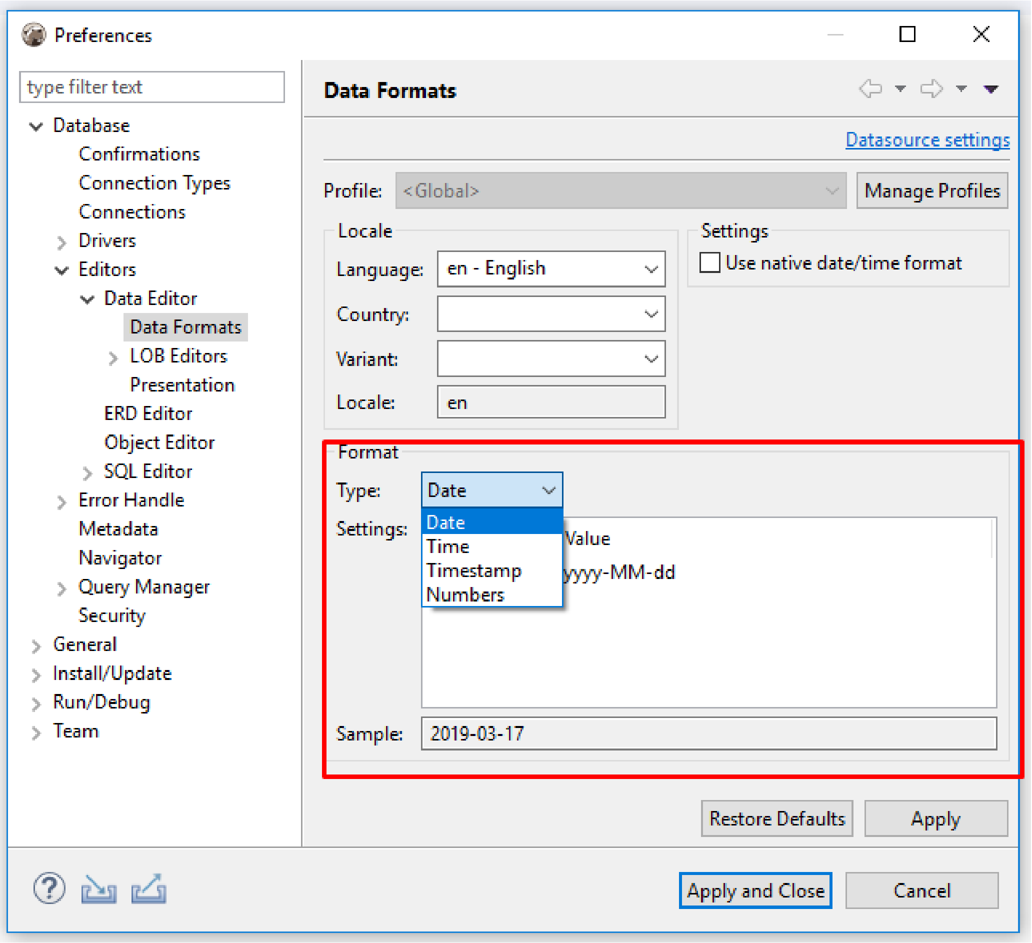
Name	Value
Pattern	yyyy-MM-dd

Sample: ๒๕๖๒-๐๓-๑๗

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



Profile: <Global>

Locale

Language: en - English

Country:

Variant:

Locale: en

Settings

□ Use native date/time format

Format

Type: Date

Settings: Value yyyy-MM-dd

Sample: 2019-03-17

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*.

Time

The default value for this data type is *HH:mm:ss*

Timestamp

The default value for this data type is *yyyy-MM-dd HH:mm:ss*

Numbers

The following parameters can be configured from thisy type of data:

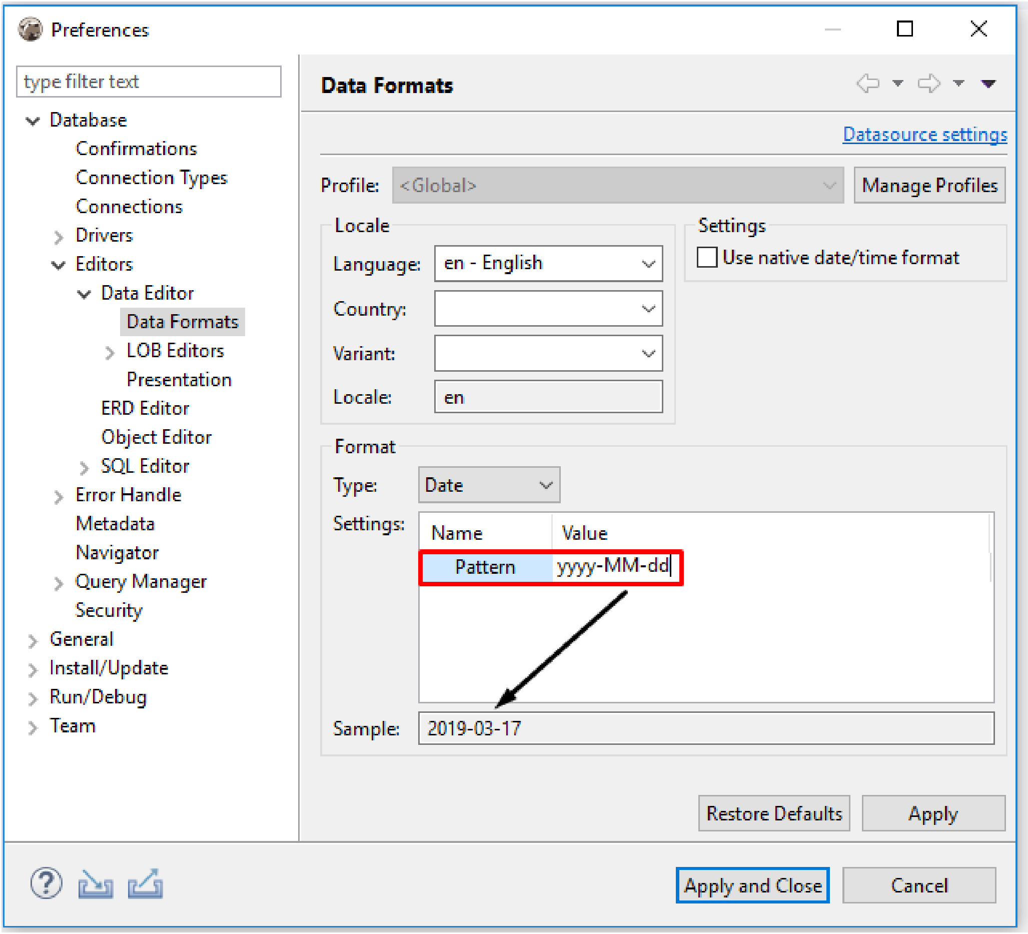
- **Use Grouping** - Long numbers can be hard to read if they have too many digits. For example, the factorial of 30 is 33 digits long! Select this check-box to enable Grouping mode, in which digits are displayed in clumps of 3 or 4 (depending on the current radix) separated by commas.
- **Maximum integer 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.



Profile: <Global>

Language: en - English

Country:

Variant:

Locale: en

□ Use native date/time format

Type: Date

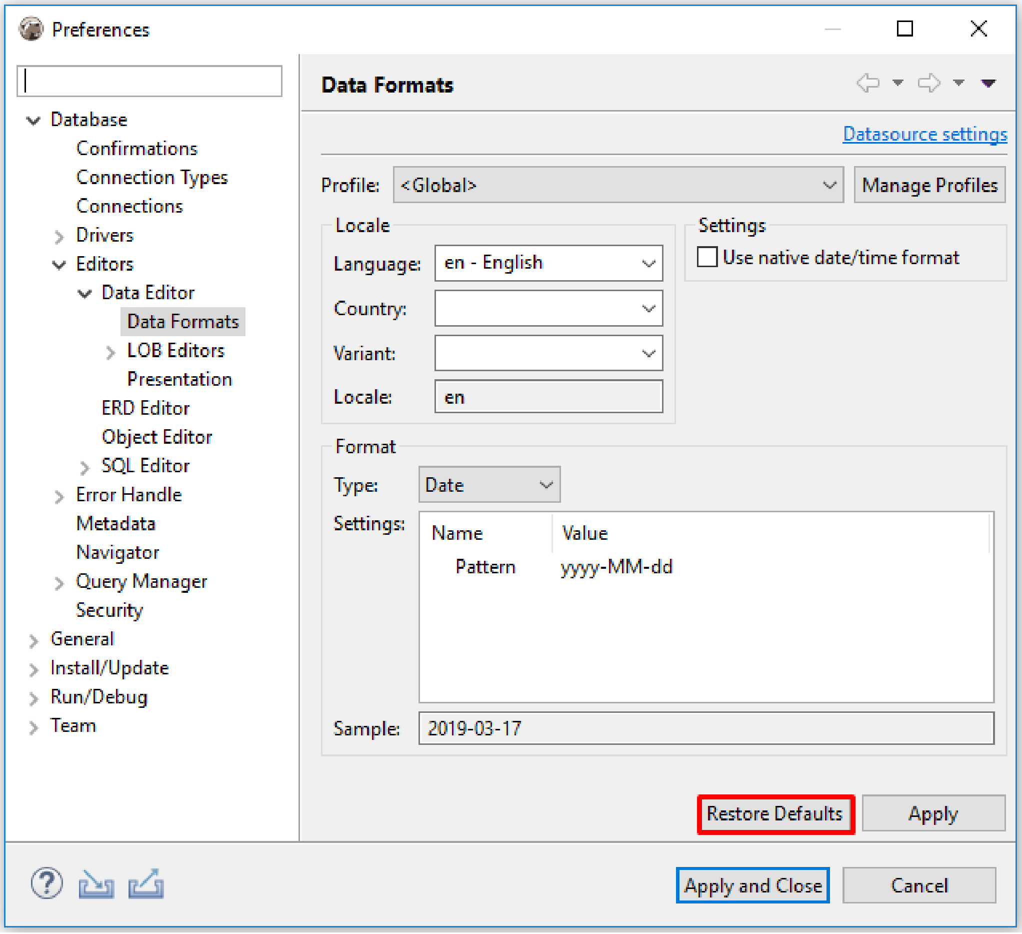
Settings:

Name	Value
Pattern	yyyy-MM-dd

Sample: 2019-03-17

Restoring Default Data Formats

To restore the default data format settings, press the **Restore Defaults** button.



Preferences

- Database
  - Confirmations
  - Connection Types
  - Connections
  - Drivers
  - Editors
    - Data Editor
      - Data Formats
      - LOB Editors
      - Presentation
    - ERD Editor
    - Object Editor
    - SQL Editor
  - Error Handle
  - Metadata
  - Navigator
  - Query Manager
  - Security
- General
- Install/Update
- Run/Debug
- Team

Data Formats

Datasource settings

Profile: <Global>

Manage Profiles

Locale

Language: en - English

Country:

Variant:

Locale: en

Settings

□ Use native date/time format

Format

Type: Date

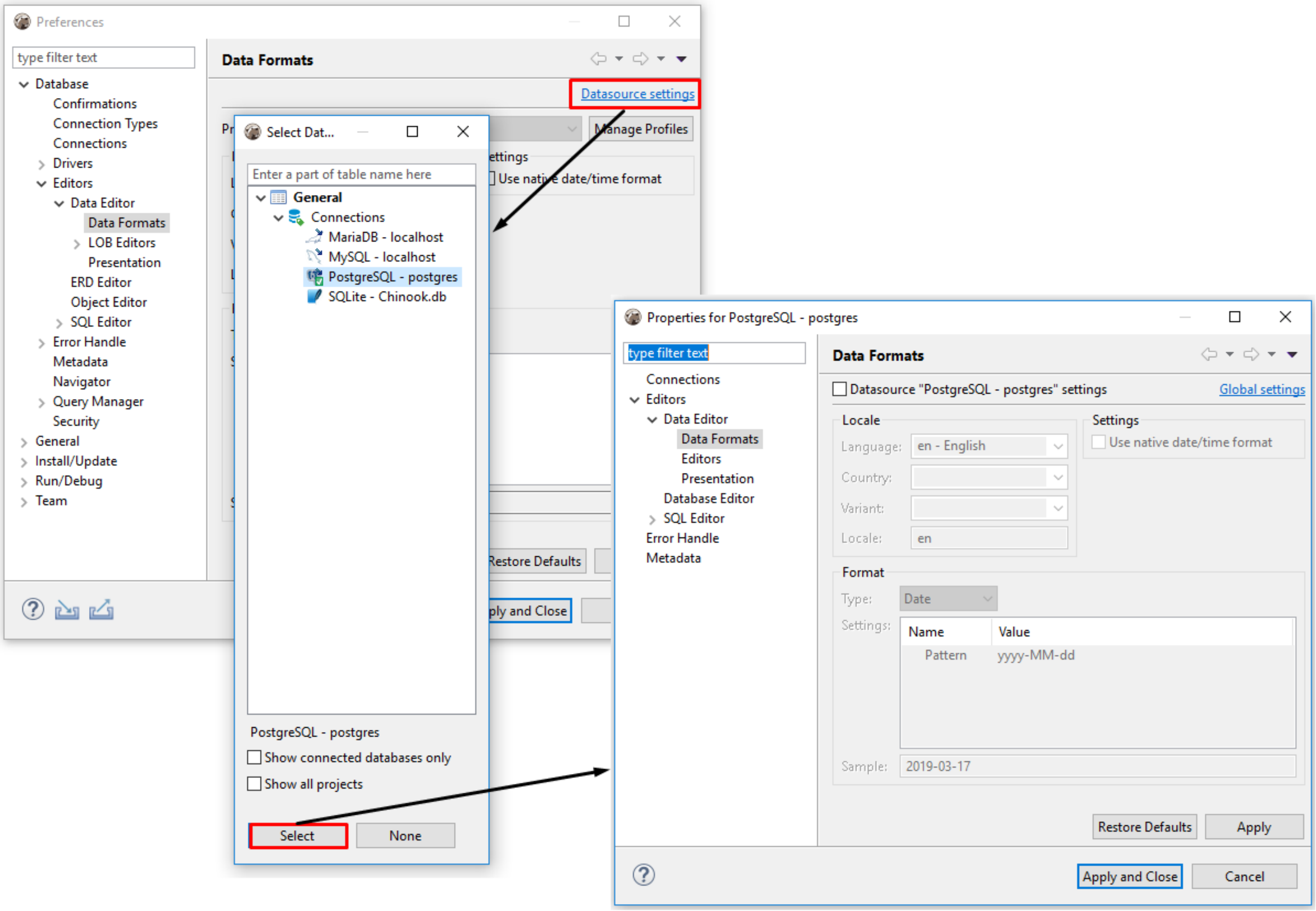
Settings:

Name	Value
Pattern	yyyy-MM-dd

Sample: 2019-03-17

Datasource Settings

Press **Datasource settings** link to change data format settings for a particular datasource, then adjust the settings in the opened dialog box.



Preferences

Data Formats

Datasource settings

Select Dat...

Enter a part of table name here

- General
  - Connections
    - MariaDB - localhost
    - MySQL - localhost
    - PostgreSQL - postgres
    - SQLite - Chinook.db

PostgreSQL - postgres

□ Show connected databases only

□ Show all projects

Properties for PostgreSQL - postgres

- Connections
- Editors
  - Data Editor
    - Data Formats
    - Editors
    - Presentation
  - Database Editor
  - SQL Editor
- Error Handle
- Metadata

Data Formats

□ Datasource "PostgreSQL - postgres" settings

Global settings

Locale

Language: en - English

Country:

Variant:

Locale: en

Settings

□ Use native date/time format

Format

Type: Date

Settings:

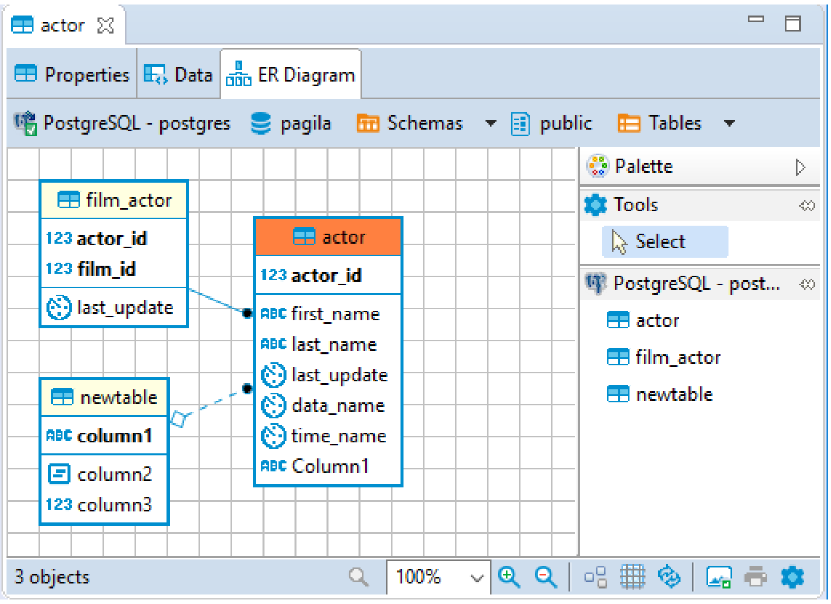
Name	Value
Pattern	yyyy-MM-dd

Sample: 2019-03-17

To save changes made press **Apply**.

ER Diagrams

ER diagrams appear on the rightmost tab of the Database Object Editor:



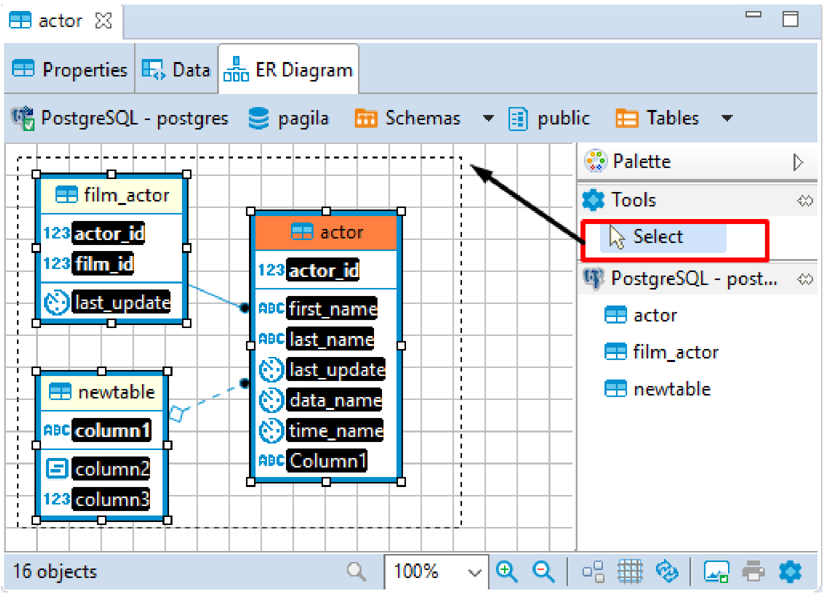
Entity Relation Diagrams (ERD) are graphic presentations of database entities and relations between them. DBeaver allows viewing diagrams of existing tables and whole database schemas, see Database Structure Diagrams, as well as create custom diagrams, see Custom Diagrams.
By default DBeaver uses IDEF1X notation.

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:



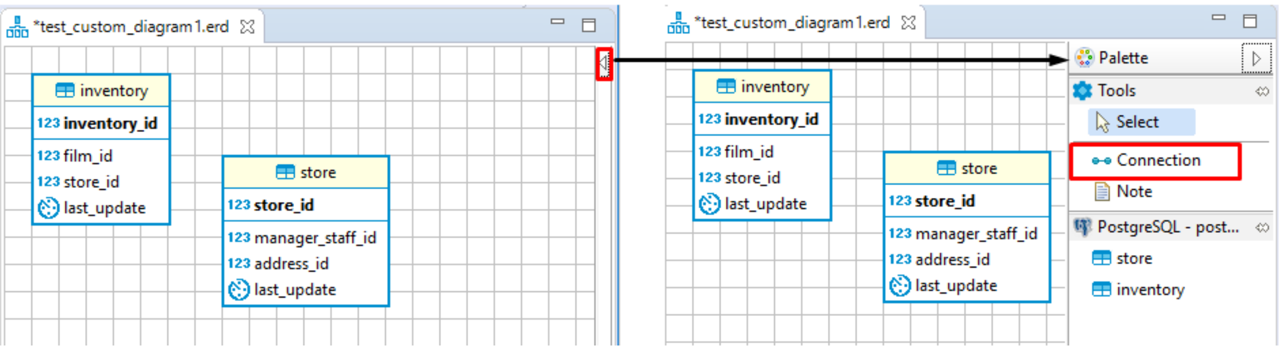
Structure Adjustment

NOTE: All changes to existing database schemas cannot be saved and are intended for exploration purposes only.
You can do the following structure changes in 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, and then to stop the connection line double-clicking the last table.

- (Available for Custom Diagrams only) - remove tables and connections: right-click the table or conection 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:

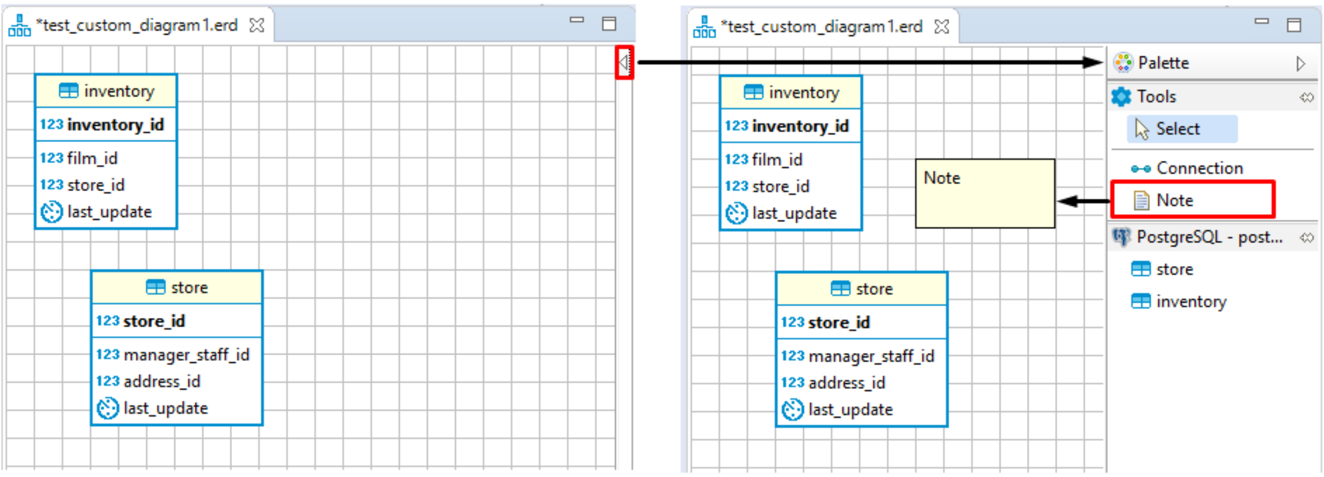
- Enable/disable the diagram grid: Click **Toggle Grid** ( ) in the toolbar.
- Modify attributes visibility: Right-click the diagram and, on the context menu, click **Show Attributes** and then one of the options:
  - **All** - all attributes
  - **Any keys** - primary and foreign keys
  - **Primary key** - only primary keys
  - **None** - no attributes
- Modify attributes presentation: Right-click the diagram and, on the context menu, click **View Styles** and then one of the options:
  - **Show Icons**
  - **Show Data Types**
  - **Show Nullability**
  - **Show Comments**
  - **Show Fully qualified names**
- Change the color of 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 front or send to 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 to the database schema by other users, you might need to refresh the diagram: click **Refresh Diagram** ( ) in the toolbar.

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 then click anywhere in the diagram tab. Now you can double-click **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.

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** ( ) in the toolbar.

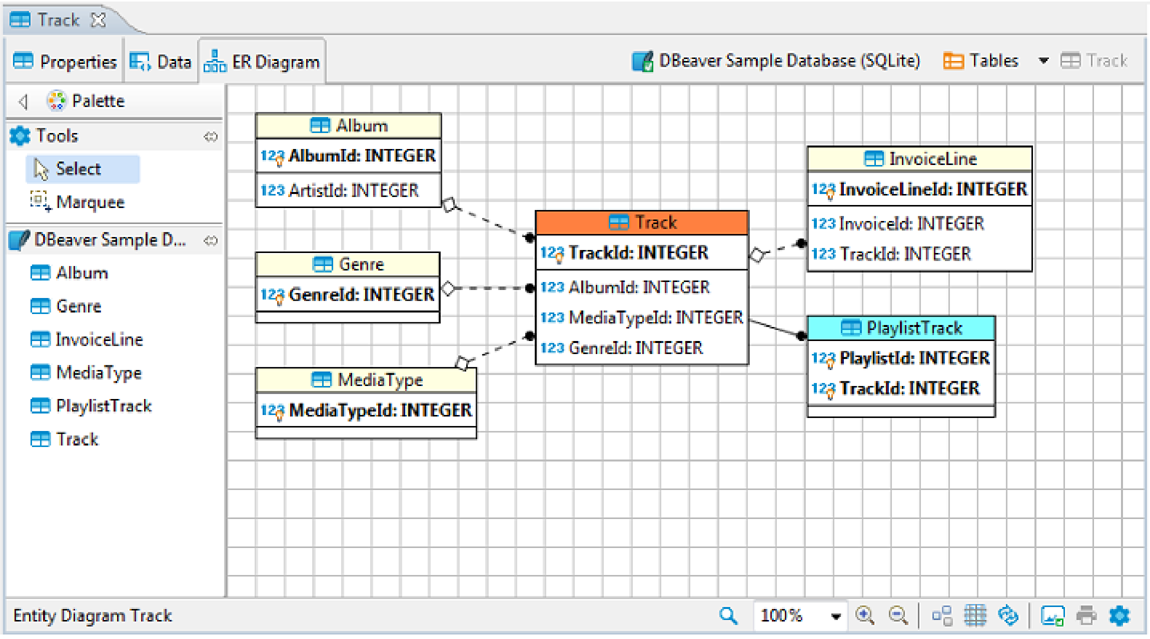
Settings

To modify the diagram settings, click **Configuration** ( ) in the toolbar.

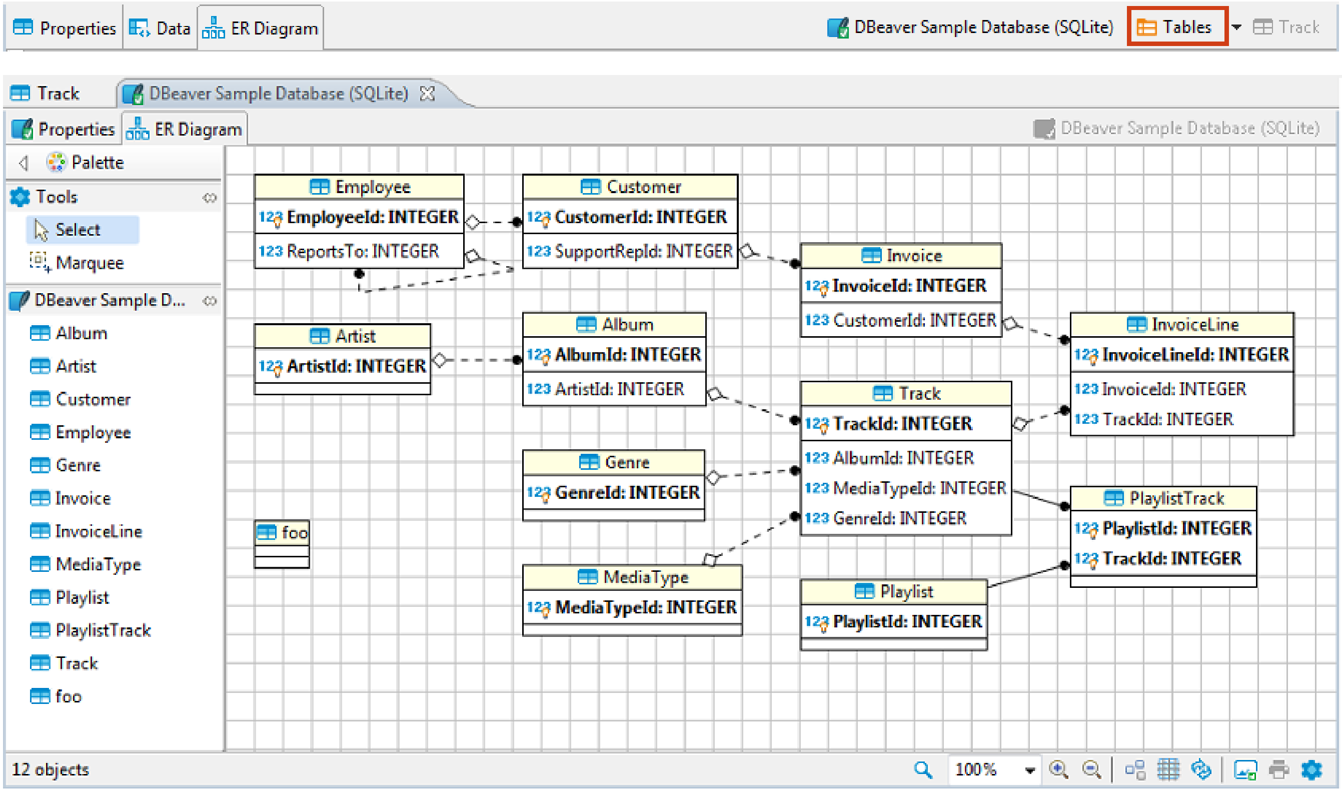
Database Structure Diagrams

You can view a database structure in the standard ERD (Entity Relation Diagram) form. ER diagrams are available for all tables and schemas (databases).

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:



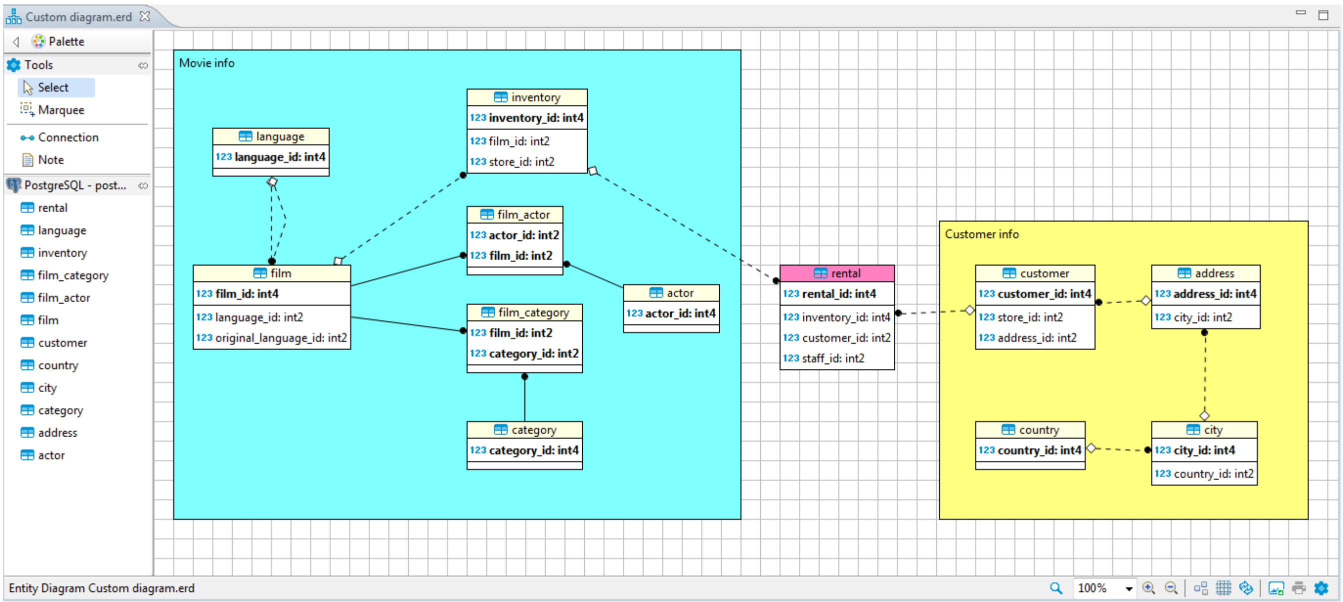
To view the ER diagram for a full database schema, double-click the schema name in the Database Navigator or the previous node in the path (usually - **Tables**):



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.

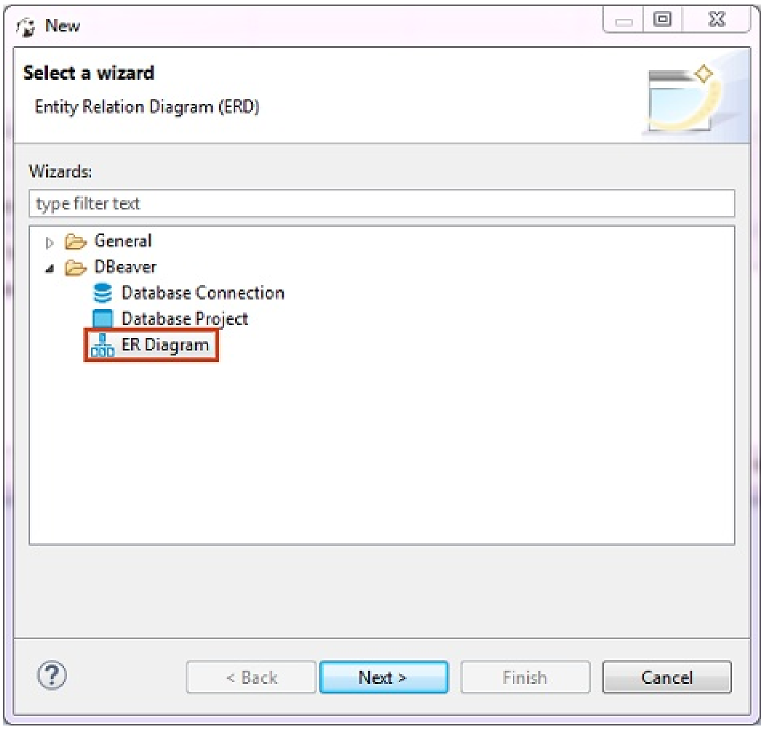
Custom Diagrams

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).

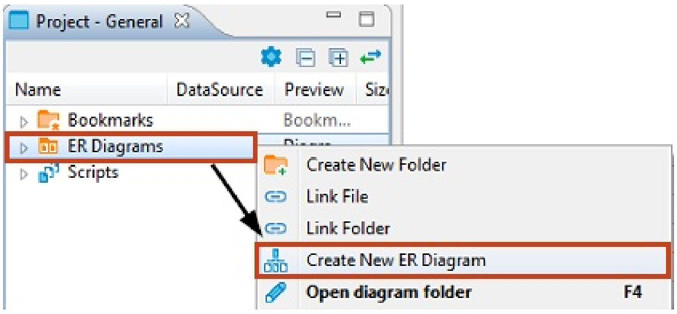


You can create a custom diagram in one of the ways:

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 (optionally) choose initial diagram contents (set of tables):

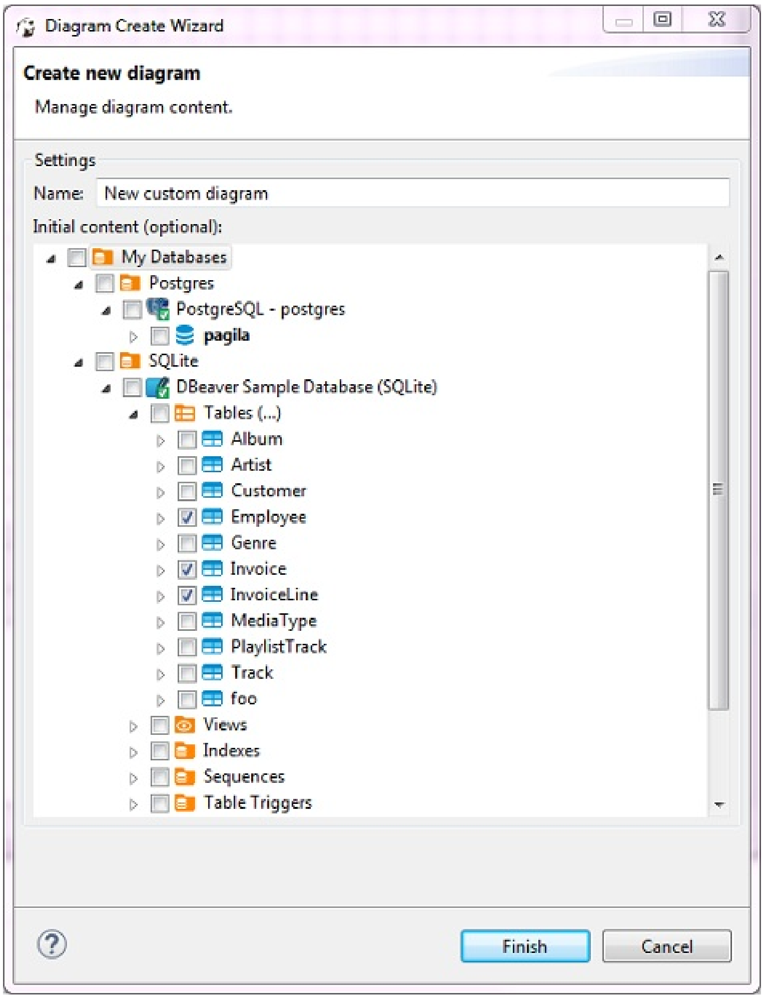


Diagram Create Wizard

Create new diagram

Manage diagram content.

Settings

Name: New custom diagram

Initial content (optional):

- □ My Databases
  - □ Postgres
    - □ PostgreSQL - postgres
      - □ **pagila**
  - □ SQLite
    - □ DBeaver Sample Database (SQLite)
      - □ Tables (...)
        - □ Album
        - □ Artist
        - □ Customer
        - ☑ Employee
        - □ Genre
        - ☑ Invoice
        - ☑ InvoiceLine
        - □ MediaType
        - □ PlaylistTrack
        - □ Track
        - □ foo
      - □ Views
      - □ Indexes
      - □ Sequences
      - □ Table Triggers

The new diagram appears in a separate editor. Now you can drag-n-drop any number of tables onto it. You can add tables from different connections and even different database type (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 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 back, and then set 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.

SQL Editor

You can create multiple SQL scripts for a single connection. Every script opens in its own SQL editor. To open an SQL editor for some connection:

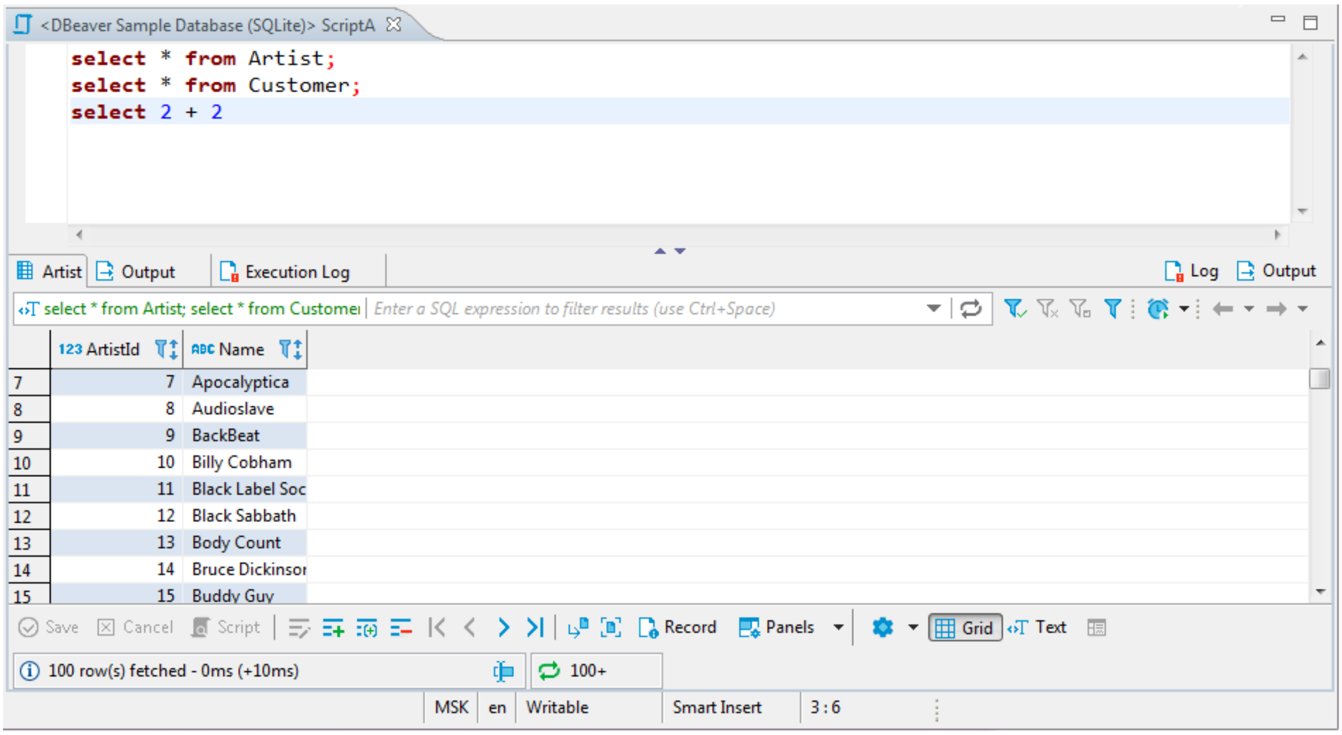
- Click this connection in the Database Navigator view and press `F3` or click **SQL Editor -> SQL Editor** on the main menu. Alternatively, you click **SQL Editor** on the context menu of this connection. DBeaver opens the Choose SQL script editor with saved SQL scripts linked to this connection. CLick the SQL script to open it in a separate tab.
- Click **Recent SQL Editor** on the context menu for this connection or on the main menu (**SQL Editor -> Recent SQL Editor**). This opens the latest used SQL editor. You can also open the most recent SQL editor using `Ctrl+Enter` shortcut in the Database Navigator view.
- If you need to create a new SQL script, on the main menu, click **SQL Editor -> New SQL Editor** or press `F3` and then click **New Script** in the Choose SQL Script window.

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.

NOTE: SQL Editor for a connection is different from SQL console for a table or view (right-click the table and click **Read data in SQL console**) in that, unlike the console, it can save scripts and changes to them.

You can see all your saved SQL scripts in the Project Explorer view in the **Scripts** folder.

The SQL editor includes the script panel at the top and results panel at the bottom:



<DBeaver Sample Database (SQLite)> ScriptA
Artist
Output
Execution Log
Log
Output
select * from Artist; select * from Custome
Enter a SQL expression to filter results (use Ctrl+Space)
Save
Cancel
Script
Record
Panels
Grid
Text
MSK
en
Writable
Smart Insert
3 : 6

```
select * from Artist;
select * from Customer;
select 2 + 2
```

	ArtistId	Name
7	7	Apocalyptica
8	8	Audioslave
9	9	BackBeat
10	10	Billy Cobham
11	11	Black Label Soc
12	12	Black Sabbath
13	13	Body Count
14	14	Bruce Dickinsor
15	15	Buddy Guy

100 row(s) fetched - 0ms (+10ms)

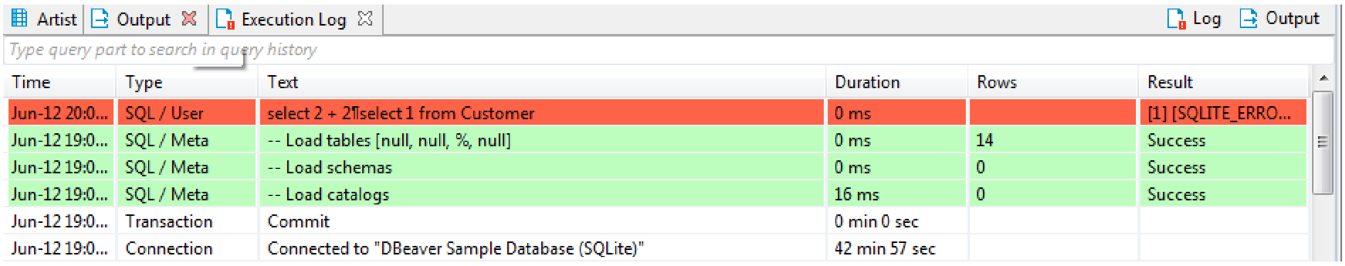
100+

You can open SQL editor preferences by pressing `Alt+Enter`.

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.

The results panel provides **Output** and **Log** views of results.
The execution **Log** tab contains all queries executed in the current SQL editor:



Time	Type	Text	Duration	Rows	Result
Jun-12 20:0...	SQL / User	select 2 + 2¶select 1 from Customer	0 ms		[1] [SQLITE_ERRO...
Jun-12 19:0...	SQL / Meta	-- Load tables [null, null, %, null]	0 ms	14	Success
Jun-12 19:0...	SQL / Meta	-- Load schemas	0 ms	0	Success
Jun-12 19:0...	SQL / Meta	-- Load catalogs	16 ms	0	Success
Jun-12 19:0...	Transaction	Commit	0 min 0 sec		
Jun-12 19:0...	Connection	Connected to "DBeaver Sample Database (SQLite)"	42 min 57 sec		

The **Output** tab contains all server-side database messages/warnings generated by a database when you execute queries. This feature is supported only by a few database engines (Oracle, SQL Server and some other ones).

Layout Adjustment

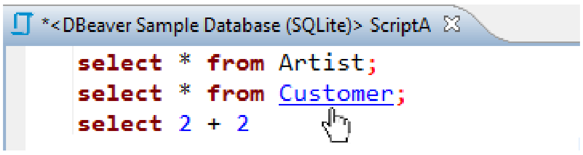
You can modify the layout of the SQL Editor by showing/hiding the results panel and changing the horizontal/vertical position of the panes.

- 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+6`, 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 `Alt+6` 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**.

Hyperlinks

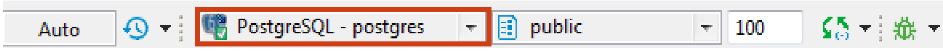
You can press and hold `Ctrl` and at the same time move the mouse over SQL text. If DBeaver recognizes some identifier as a table/view name, it presents it as a hyperlink. You can click the hyperlink to open this object`s editor:



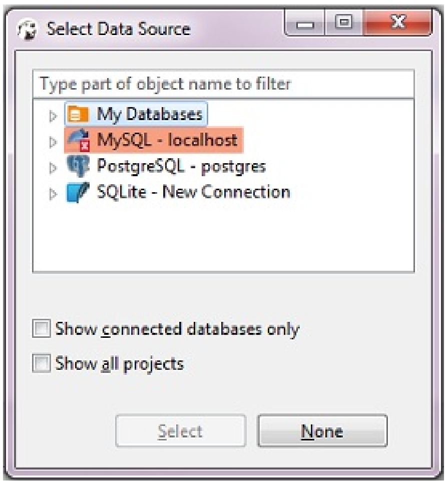
```
select * from Artist;
select * from Customer;
select 2 + 2
```

Active Database/Schema Selection

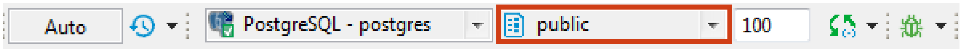
You can change the connection associated with the current SQL editor or change the active database/schema, at the same time retaining the SQL text.
To change the connection, press `Ctrl+9` or click the **Active datasource** box on DBeaver`s main toolbar:



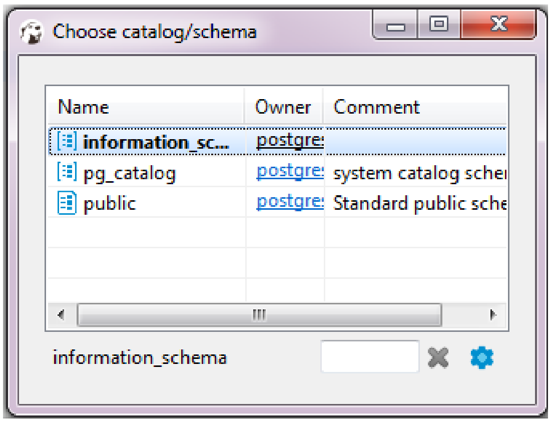
The Select Data Source dialog box opens. In the tree of connections, click the required connection and then click **Select**. To disassociate the SQL Editor with any connection, click **None**:



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:



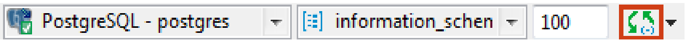
Name	Owner	Comment
information_sc...	postgre:	
pg_catalog	postgre:	system catalog schei
public	postgre:	Standard public sche

information_schema

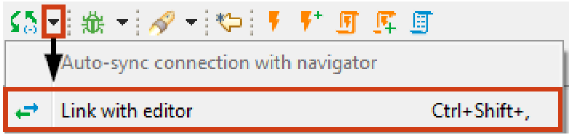
If there are many schemas and they do not fit in the dialog box use the search field to find the schema you need:

To configure the set of columns to be visible for each schema in the dialog box, click the **Configure columns** button ().

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 **Set connection from navigator** button in DBeaver`s main toolbar:



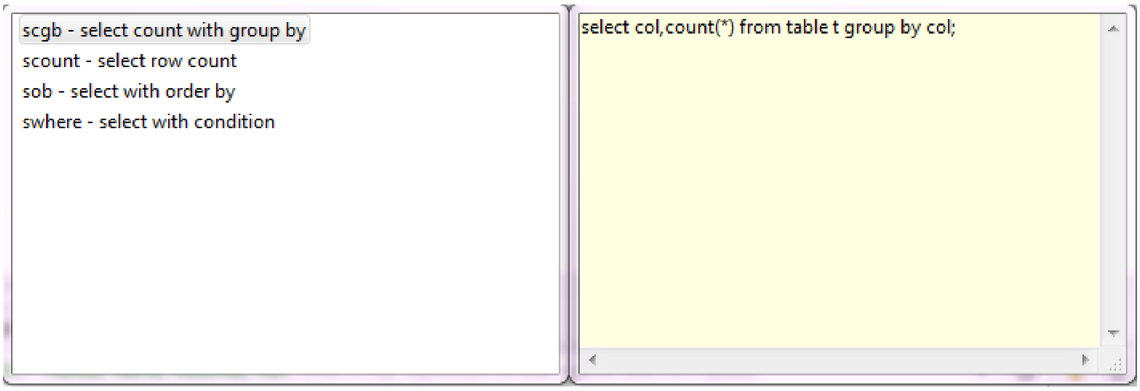
The reverse action is also possible: you can set the focus of the Database Navigator to the active connection of the SQL Editor - press `Ctrl+Shift+,` or click the arrow next to the **Set connection from navigator** button in DBeaver`s main toolbar and then click **Link with editor**:



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 appears with a list of available templates:



scgb - select count with group by
scount - select row count
sob - select with order by
swhere - select with condition

```
select col,count(*) from table t group by col;
```

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 (⚙) in the bottom toolbar, then click **Preferences -> SQL Editor -> Templates**. For more information about managing templates, please visit Eclipse Website.

Standard Eclipse templates:

	Variable	Description
	${cursor}	Specifies the cursor position when the template edit mode is left. This is useful when the cursor should jump to different place than to the end of the template upon leaving the template edit mode.
	${year}	Takes the current year value
	${date}	Takes the current date value
	${time}	Takes the current time value
	${dollar}	Takes the dollar sign $. Alternatively, two dollar signs can be used: $$.
	${user}	Takes the user name
	${word_selection}	Takes the content of the current text selection
	${line_selection}	Takes content of all currently selected lines

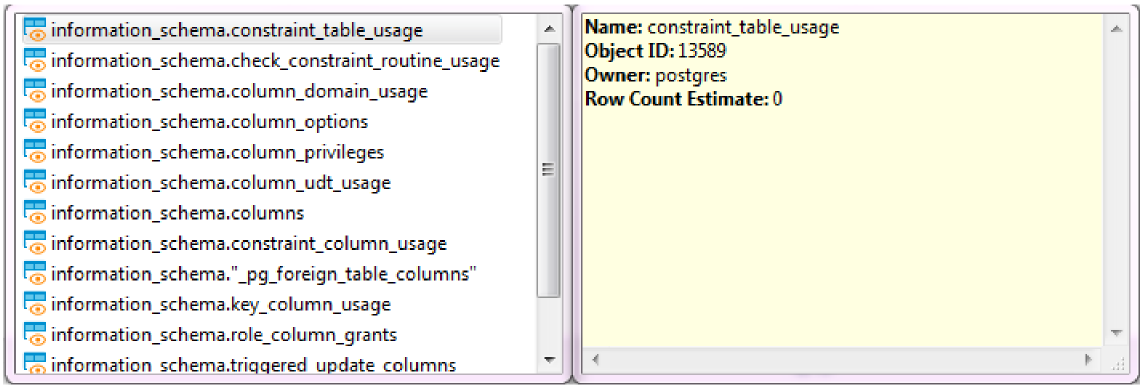
DBeaver-specific templates:

	Variable	Description
	${schema}	Takes the current schema name
	${catalog}	Takes the catalog name
	${table}	Takes the current table name (from the active catalog/schema)
	${column}	Takes the column name (from the current table)

SQL Assist and Auto-Complete

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 required place in the query and click **SQL Assist** on the context menu. DBeaver searches for potentially suitable objects in already loaded database metadata and in the database system tables.



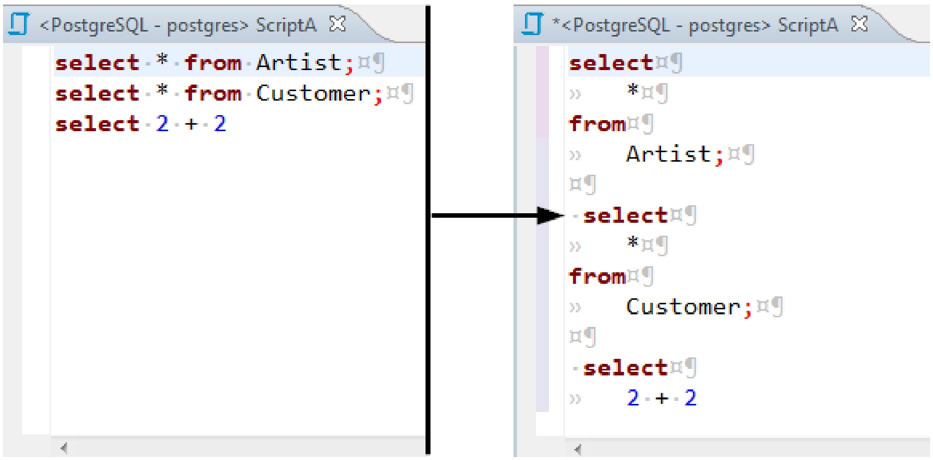
information_schema.constraint_table_usage
information_schema.check_constraint_routine_usage
information_schema.column_domain_usage
information_schema.column_options
information_schema.column_privileges
information_schema.column_udt_usage
information_schema.columns
information_schema.constraint_column_usage
information_schema."_pg_foreign_table_columns"
information_schema.key_column_usage
information_schema.role_column_grants
information_schema.triggered_update_columns

Name: constraint_table_usage
Object ID: 13589
Owner: postgres
Row Count Estimate: 0

When you start typing an SQL keyword in a statement, DBeaver offers auto-complete options as well.
Another auto-complete function is search for completion only within already entered identifiers - press `Ctrl+Shift+Space`.

SQL Formatting

To format SQL text, select it and press `Ctrl+Shift+F` or right-click the selected text and click **Format -> Format SQL** on the context menu.


<PostgreSQL - postgres> ScriptA
*<PostgreSQL - postgres> ScriptA

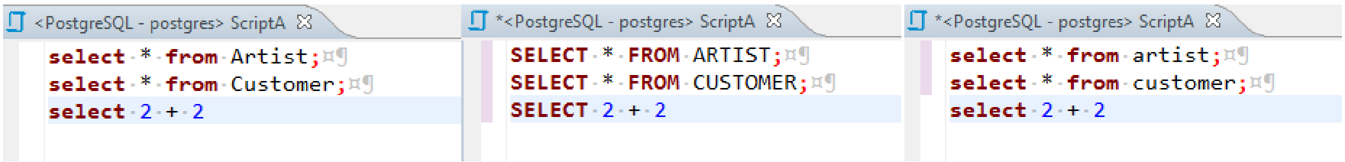
```
select * from Artist;
select * from Customer;
select 2 + 2
```

```
select
	*
from
	Artist;

select
	*
from
	Customer;

select
	2 + 2
```

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.

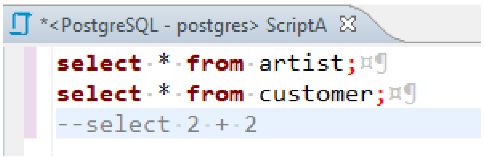

<PostgreSQL - postgres> ScriptA
*<PostgreSQL - postgres> ScriptA
*<PostgreSQL - postgres> ScriptA

```
select * from Artist;
select * from Customer;
select 2 + 2
```

```
SELECT * FROM ARTIST;
SELECT * FROM CUSTOMER;
SELECT 2 + 2
```

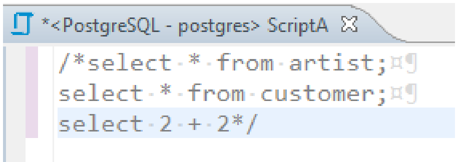
```
select * from artist;
select * from customer;
select 2 + 2
```

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.



```
select * from artist;
select * from customer;
--select 2 + 2
```

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, 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.

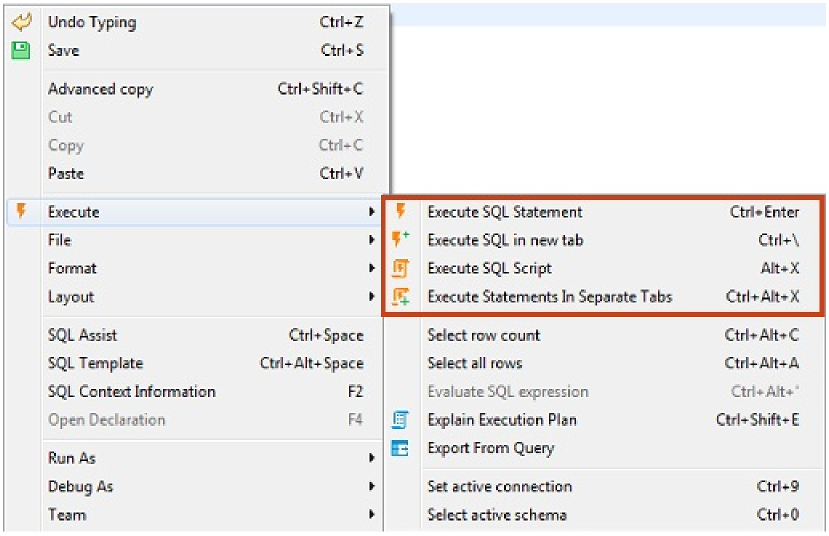


```
/*select * from artist;
select * from customer;
select 2 + 2*/
```

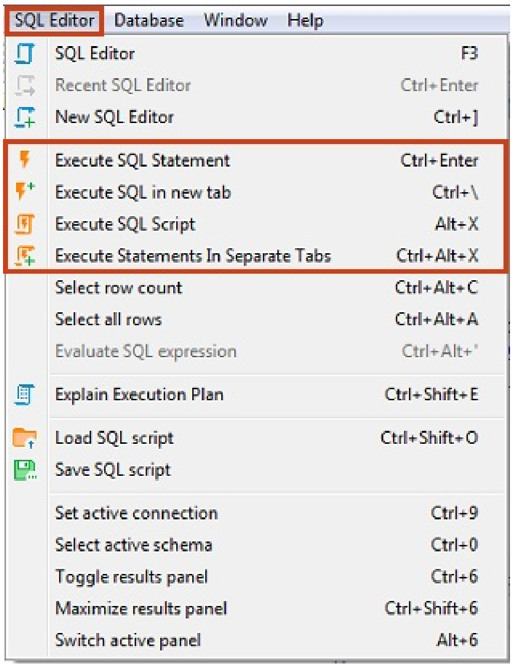
SQL Execution

You can execute one query, a highlighted portion of a script, or a whole script. You can execute them using:

- Shortcut key combinations (see details further in this article)
- Tools in the main toolbar:
- Context menu (right-click the query):


Undo Typing Ctrl+Z
Save Ctrl+S
Advanced copy Ctrl+Shift+C
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Execute
File
Format
Layout
SQL Assist Ctrl+Space
SQL Template Ctrl+Alt+Space
SQL Context Information F2
Open Declaration F4
Run As
Debug As
Team
Execute SQL Statement Ctrl+Enter
Execute SQL in new tab Ctrl+\
Execute SQL Script Alt+X
Execute Statements In Separate Tabs Ctrl+Alt+X
Select row count Ctrl+Alt+C
Select all rows Ctrl+Alt+A
Evaluate SQL expression Ctrl+Alt+'
Explain Execution Plan Ctrl+Shift+E
Export From Query
Set active connection Ctrl+9
Select active schema Ctrl+0

- DBeaver main menu:


SQL Editor
Database
Window
Help
SQL Editor F3
Recent SQL Editor Ctrl+Enter
New SQL Editor Ctrl+]
Execute SQL Statement Ctrl+Enter
Execute SQL in new tab Ctrl+\
Execute SQL Script Alt+X
Execute Statements In Separate Tabs Ctrl+Alt+X
Select row count Ctrl+Alt+C
Select all rows Ctrl+Alt+A
Evaluate SQL expression Ctrl+Alt+'
Explain Execution Plan Ctrl+Shift+E
Load SQL script Ctrl+Shift+O
Save SQL script
Set active connection Ctrl+9
Select active schema Ctrl+0
Toggle results panel Ctrl+6
Maximize results panel Ctrl+Shift+6
Switch active panel Alt+6

To execute a query under 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 cursor or selected text and fills the results pane with the query results.

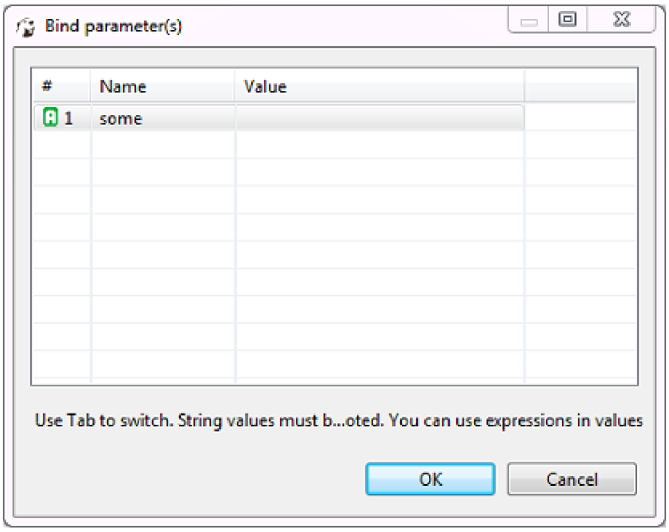
To execute a query under cursor in a separate tab, press `CTRL+\ `or right-click the query and click **Execute -> Execute SQL in new tab** on the context menu. The same can be done using the main toolbar or the main menu: **SQL Editor -> Execute SQL in new tab**. This executes the SQL query under cursor 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 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. The 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). NOTE: Be careful with this feature. If you execute a huge script with a large number of queries, it might cause unexpected problems.

Dynamic Parameter Bindings

You can use dynamic parameters in your SQL queries. The parameter format is :name. When you execute a query which contains dynamic parameters, DBeaver displays a dialog box in which you can fill the parameter values:

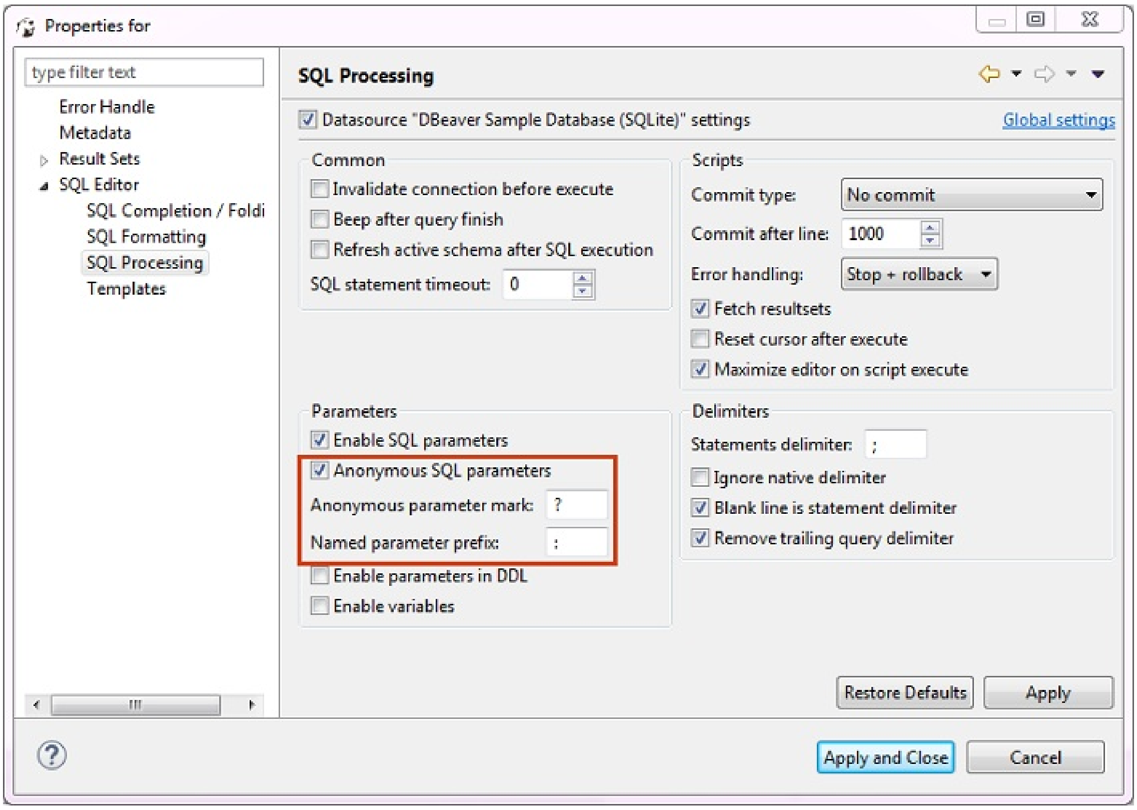


Bind parameter(s)

#	Name	Value
1	some	

Use Tab to switch. String values must b...oted. You can use expressions in values

You can also use anonymous parameters (?), but you will need to enable them in SQL editor preferences:



Properties for

SQL Processing

☑ Datasource "DBeaver Sample Database (SQLite)" settings

Common

☐ Invalidate connection before execute
☐ Beep after query finish
☐ Refresh active schema after SQL execution
SQL statement timeout: 0

Scripts

Commit type: No commit
Commit after line: 1000
Error handling: Stop + rollback
☑ Fetch resultsets
☐ Reset cursor after execute
☑ Maximize editor on script execute

Parameters

☑ Enable SQL parameters
☑ Anonymous SQL parameters
Anonymous parameter mark: ?
Named parameter prefix: :
☐ Enable parameters in DDL
☐ Enable variables

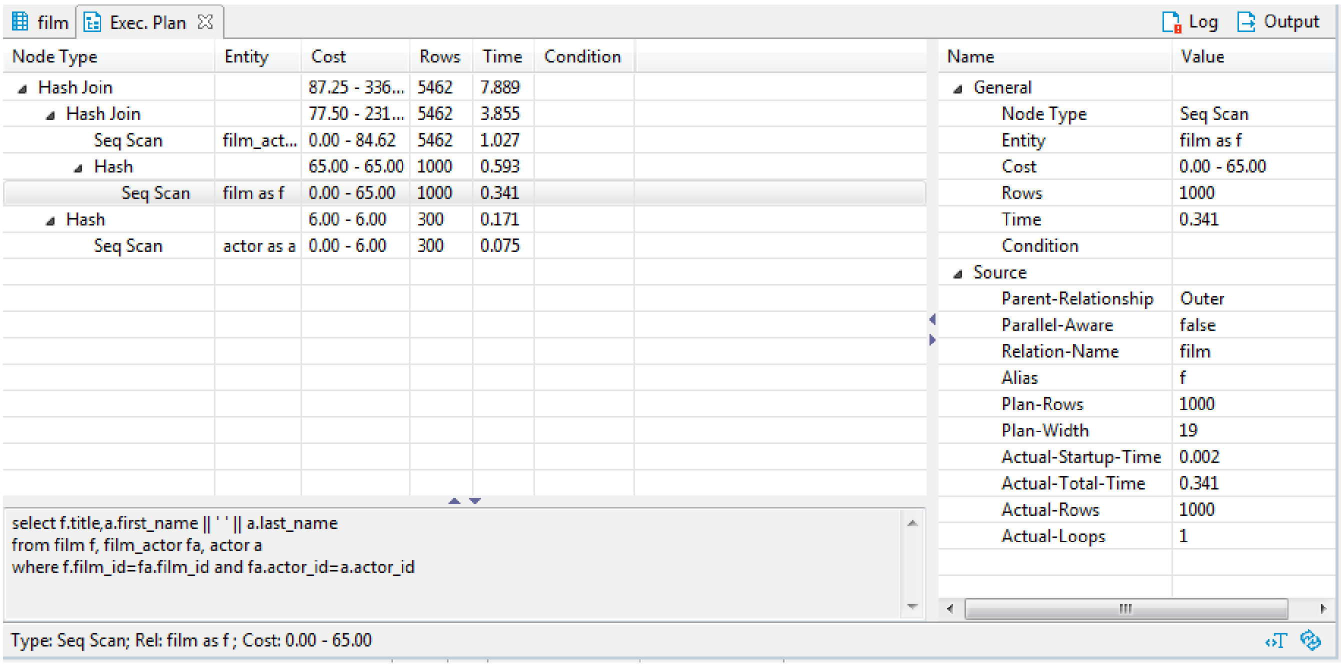
Delimiters

Statements delimiter: ;
☐ Ignore native delimiter
☑ Blank line is statement delimiter
☑ Remove trailing query delimiter

You can open SQL editor preferences by pressing `Alt+Enter`.

Execution Plan

If a database driver supports execution plan visualization, 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 main toolbar: 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:



Node Type	Entity	Cost	Rows	Time	Condition
Hash Join		87.25 - 336...	5462	7.889	
Hash Join		77.50 - 231...	5462	3.855	
Seq Scan	film_act...	0.00 - 84.62	5462	1.027	
Hash		65.00 - 65.00	1000	0.593	
Seq Scan	film as f	0.00 - 65.00	1000	0.341	
Hash		6.00 - 6.00	300	0.171	
Seq Scan	actor as a	0.00 - 6.00	300	0.075	

Name	Value
General	
Node Type	Seq Scan
Entity	film as f
Cost	0.00 - 65.00
Rows	1000
Time	0.341
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

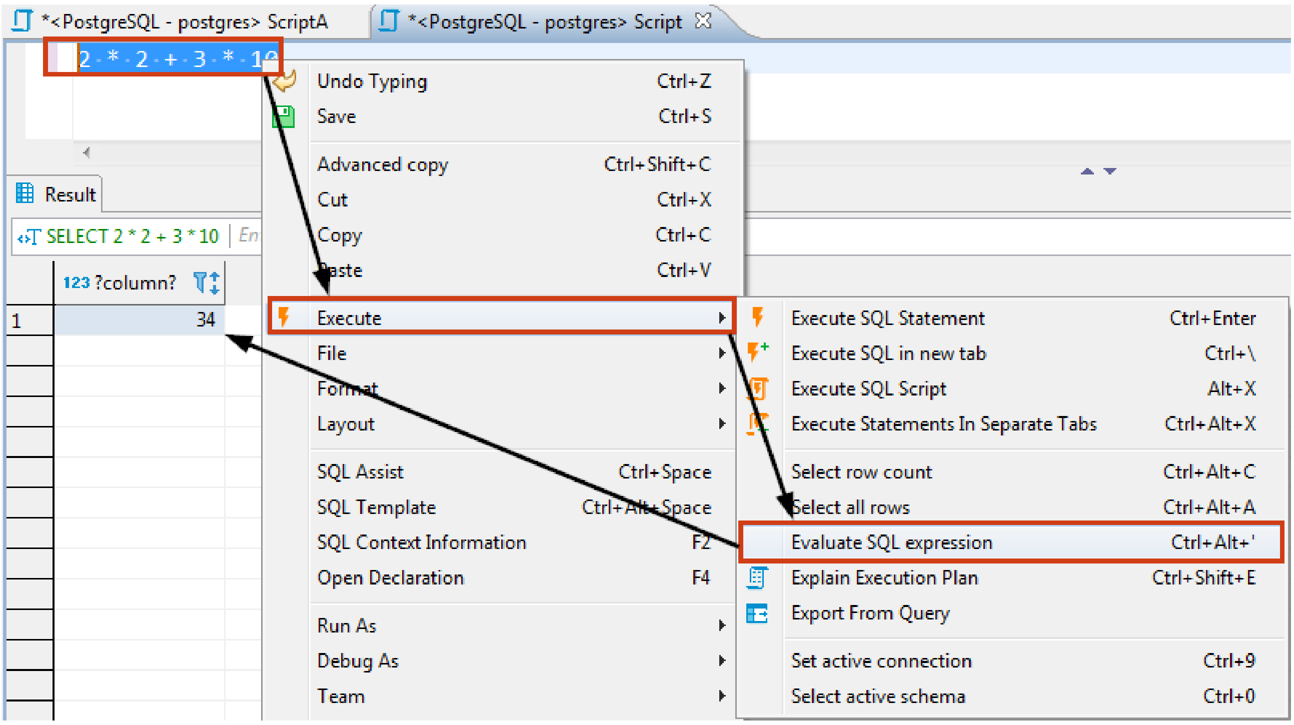
```
select f.title,a.first_name || ' ' || a.last_name
from film f, film_actor fa, actor a
where f.film_id=fa.film_id and fa.actor_id=a.actor_id
```

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 (‹›T).

SQL Expression Evaluation

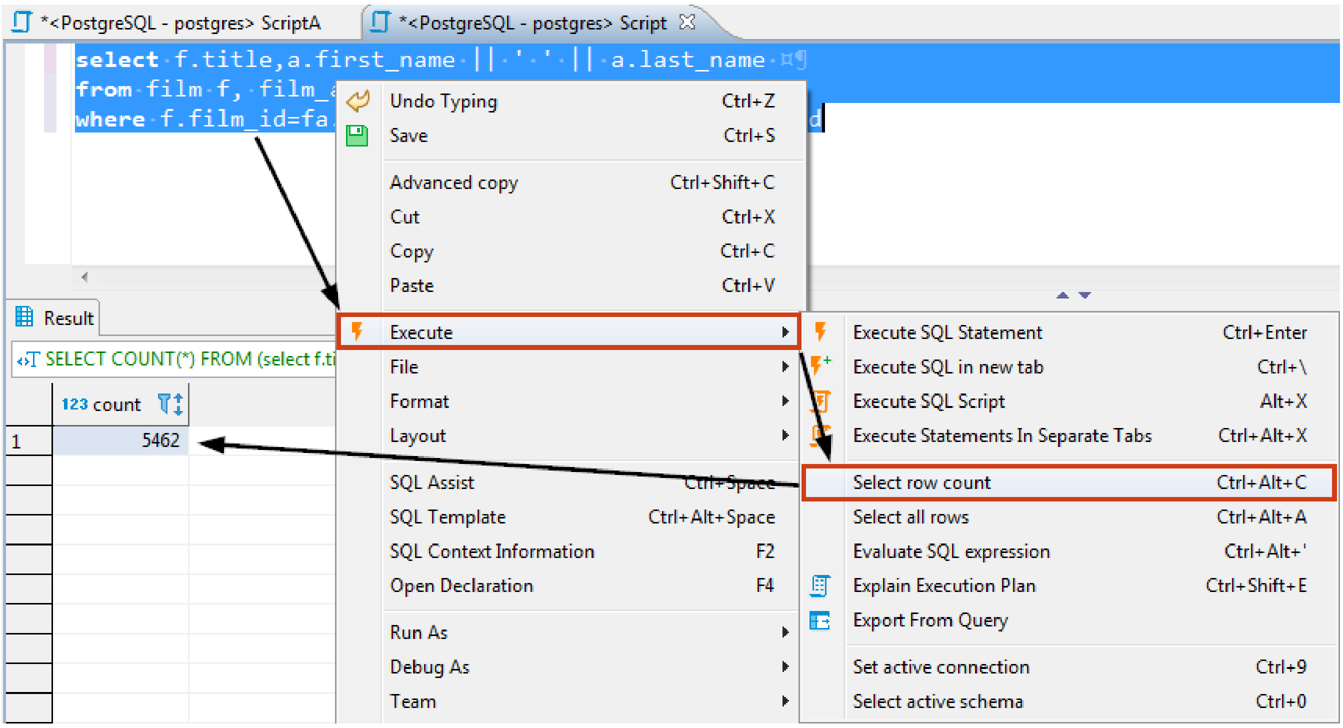
To evaluate an SQL expression, right-click the expression and click **Execute -> Evaluate SQL expression** on the context menu. This command basically performs a query of **SELECT [expression] FROM DUAL** type:



*<PostgreSQL - postgres> ScriptA
*<PostgreSQL - postgres> Script
2 * 2 + 3 * 10
Result
SELECT 2 * 2 + 3 * 10
123 ?column?
1
34
Undo Typing Ctrl+Z
Save Ctrl+S
Advanced copy Ctrl+Shift+C
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Execute
File
Format
Layout
SQL Assist Ctrl+Space
SQL Template Ctrl+Alt+Space
SQL Context Information F2
Open Declaration F4
Run As
Debug As
Team
Execute SQL Statement Ctrl+Enter
Execute SQL in new tab Ctrl+\
Execute SQL Script Alt+X
Execute Statements In Separate Tabs Ctrl+Alt+X
Select row count Ctrl+Alt+C
Select all rows Ctrl+Alt+A
Evaluate SQL expression Ctrl+Alt+'
Explain Execution Plan Ctrl+Shift+E
Export From Query
Set active connection Ctrl+9
Select active schema Ctrl+0

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:



```
select f.title,a.first_name || ' ' || a.last_name
from film f, film_a
where f.film_id=fa
```

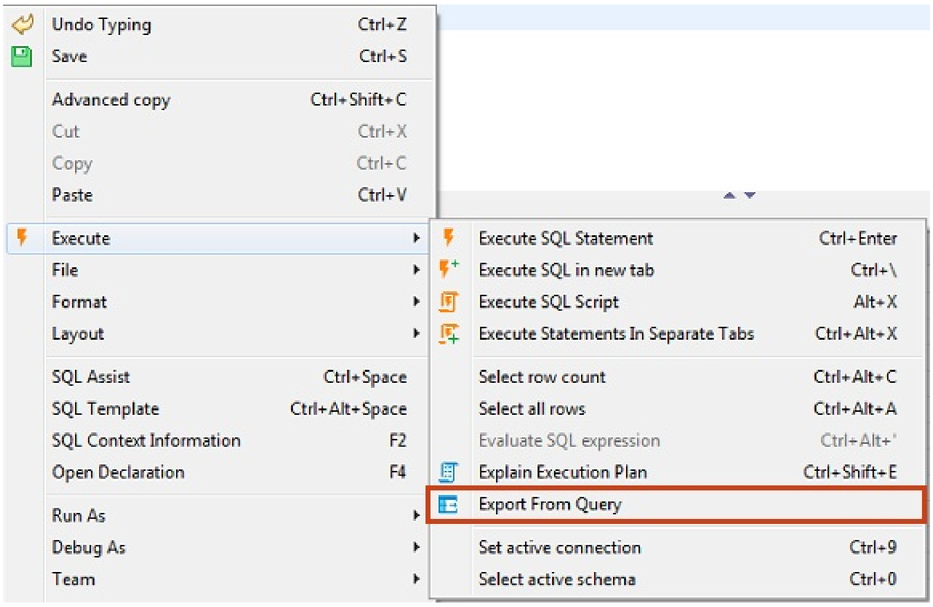
SELECT COUNT(*) FROM (select f.ti

count
5462

*<PostgreSQL - postgres> ScriptA
*<PostgreSQL - postgres> Script
Result
Undo Typing Ctrl+Z
Save Ctrl+S
Advanced copy Ctrl+Shift+C
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Execute
File
Format
Layout
SQL Assist Ctrl+Space
SQL Template Ctrl+Alt+Space
SQL Context Information F2
Open Declaration F4
Run As
Debug As
Team
Execute SQL Statement Ctrl+Enter
Execute SQL in new tab Ctrl+\
Execute SQL Script Alt+X
Execute Statements In Separate Tabs Ctrl+Alt+X
Select row count Ctrl+Alt+C
Select all rows Ctrl+Alt+A
Evaluate SQL expression Ctrl+Alt+'
Explain Execution Plan Ctrl+Shift+E
Export From Query
Set active connection Ctrl+9
Select active schema Ctrl+0

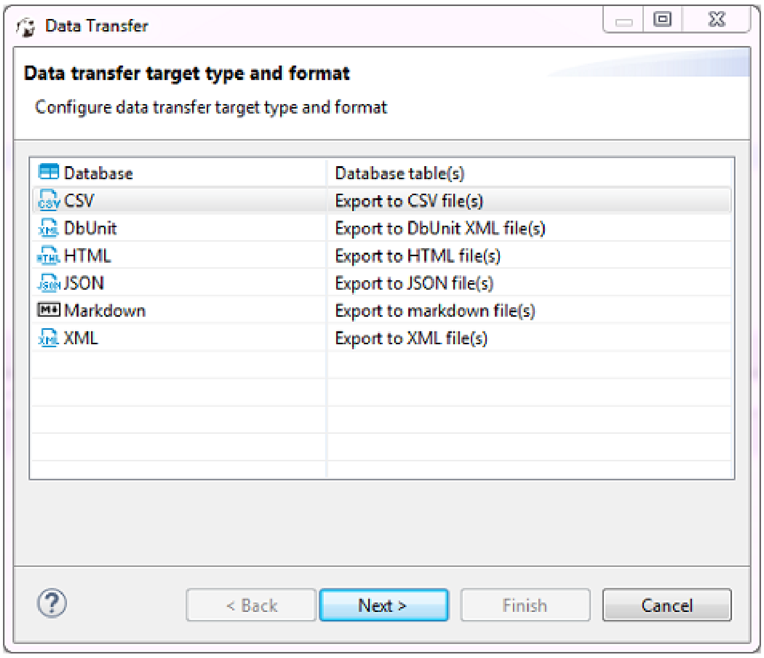
Query Export

It might be useful to export a query if you have a very 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:



Undo Typing Ctrl+Z
Save Ctrl+S
Advanced copy Ctrl+Shift+C
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Execute
File
Format
Layout
SQL Assist Ctrl+Space
SQL Template Ctrl+Alt+Space
SQL Context Information F2
Open Declaration F4
Run As
Debug As
Team
Execute SQL Statement Ctrl+Enter
Execute SQL in new tab Ctrl+\
Execute SQL Script Alt+X
Execute Statements In Separate Tabs Ctrl+Alt+X
Select row count Ctrl+Alt+C
Select all rows Ctrl+Alt+A
Evaluate SQL expression Ctrl+Alt+'
Explain Execution Plan Ctrl+Shift+E
Export From Query
Set active connection Ctrl+9
Select active schema Ctrl+0

The Data transfer wizard opens. Go through its steps to complete the export of the query.


Data Transfer

Data transfer target type and format

Configure data transfer target type and format

Database	Database table(s)
CSV	Export to CSV file(s)
DbUnit	Export to DbUnit XML file(s)
HTML	Export to HTML file(s)
JSON	Export to JSON file(s)
Markdown	Export to markdown file(s)
XML	Export to XML file(s)

Client-side commands

You can use special commands in SQL scripts.
These commands are executed on DBeaver side, not on server-side.

Name	Description
@set var = value	Sets default value for SQL parameter
@echo text	Prints string into server output viewer
@include file	Includes script file from file system

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`.

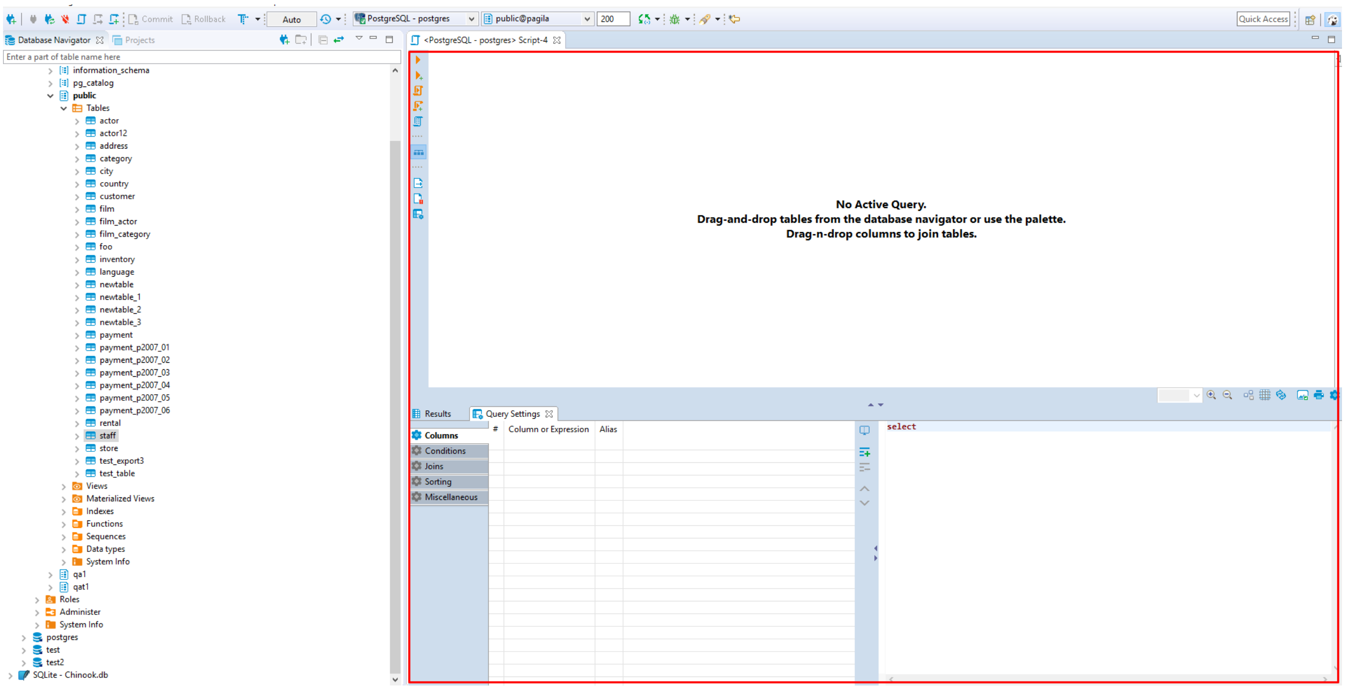
Visual Query Builder

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 is not much familiar with SQL scripting or if you don't want to insert script commands manually. The tool creates SQL scripts automatically based on visual schema you create.

*Note: Visual Query Builder presents only in DBeaver Enterprise Edition

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.

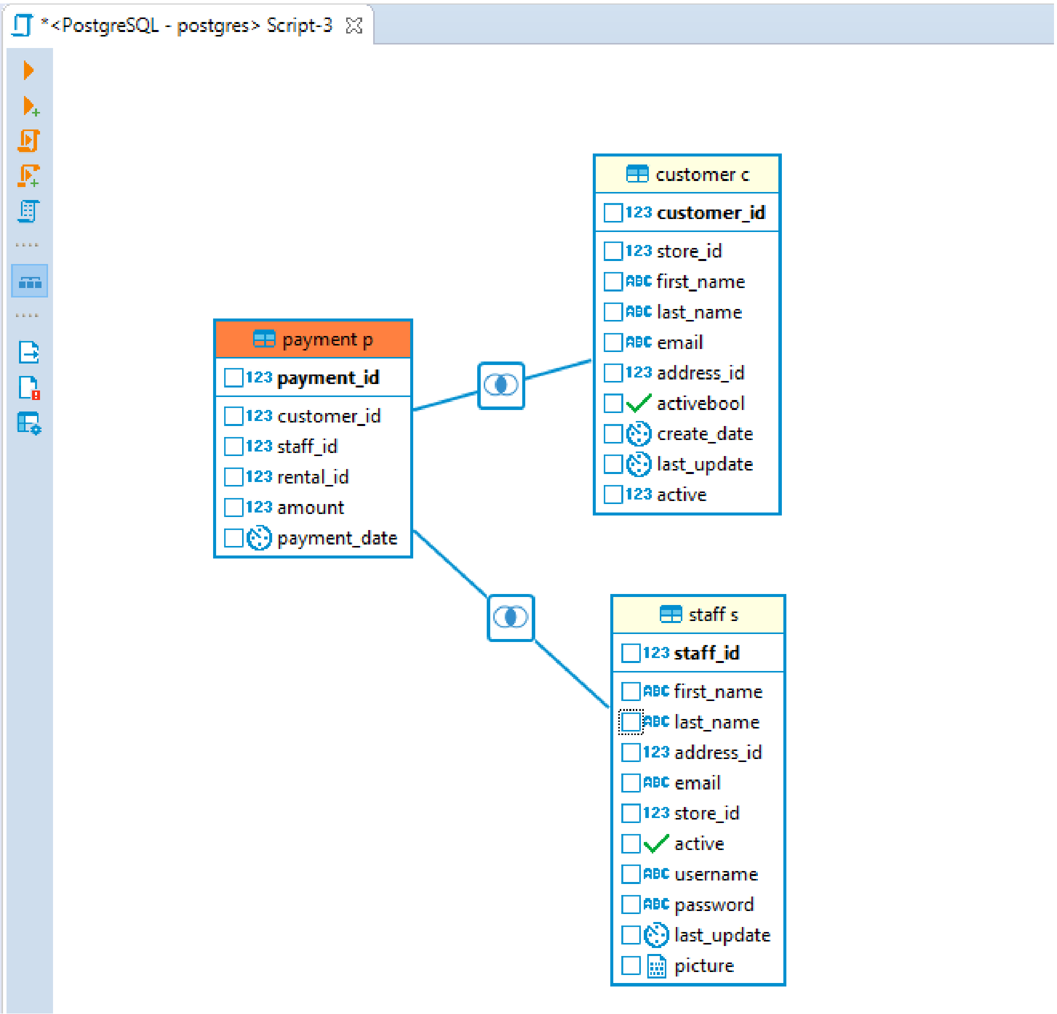


No Active Query.
Drag-and-drop tables from the database navigator or use the palette.
Drag-n-drop columns to join tables.

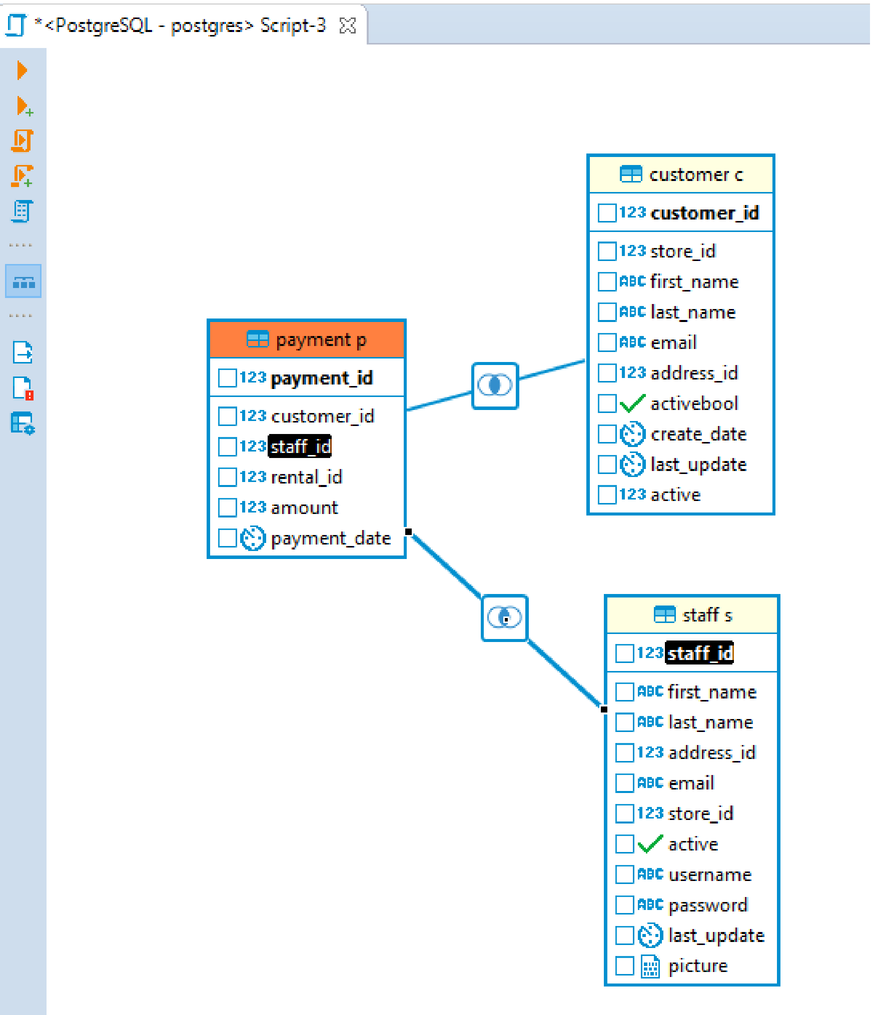
```
select
```

Creating Visual Query

Start creating a query from selecting 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 pressed 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 check-box 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.

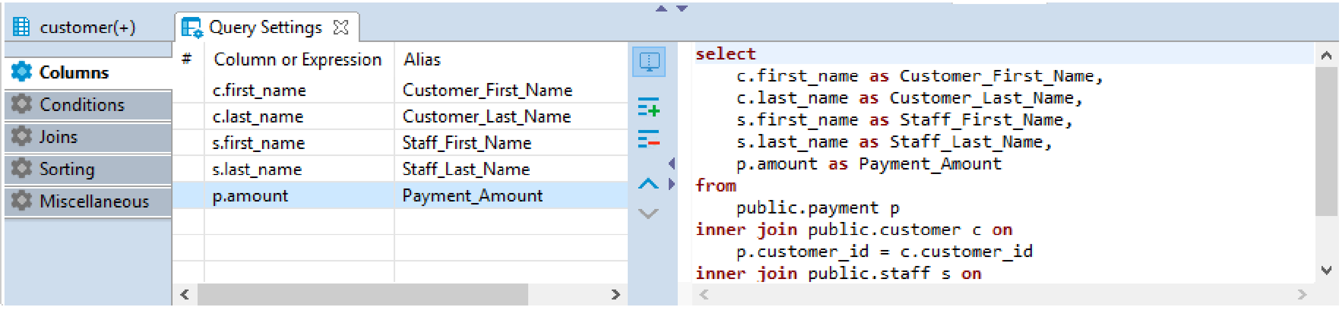
Adjusting Query Settings

Visual Query Builder also allows setting query conditions and adjust 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

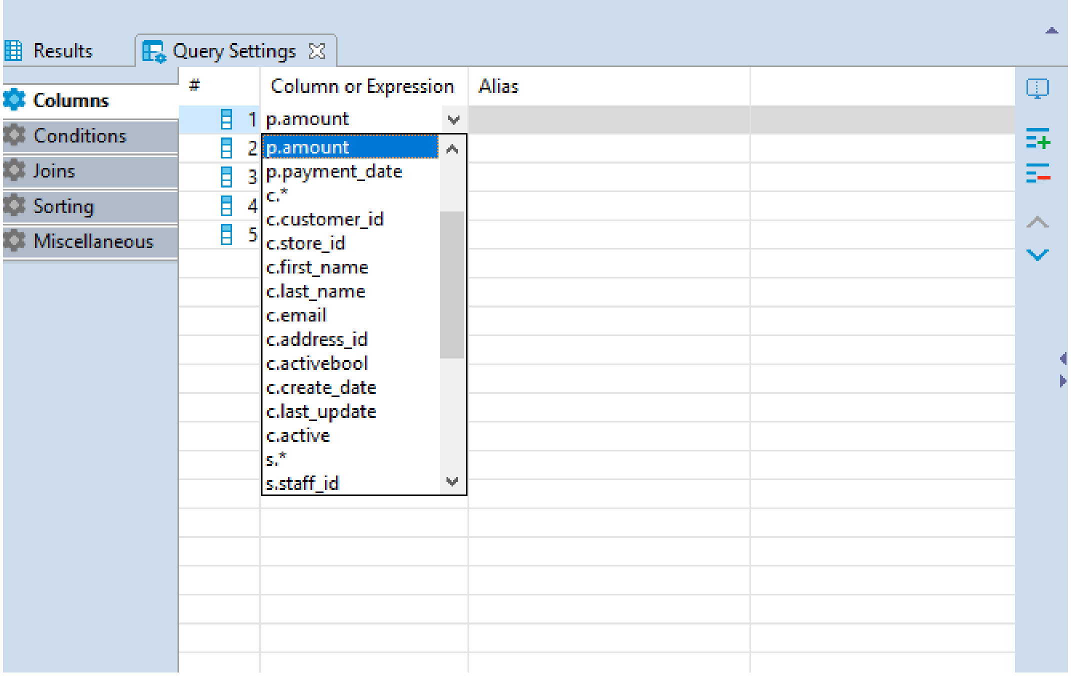


#	Column or Expression	Alias
	c.first_name	Customer_First_Name
	c.last_name	Customer_Last_Name
	s.first_name	Staff_First_Name
	s.last_name	Staff_Last_Name
	p.amount	Payment_Amount

```
select
    c.first_name as Customer_First_Name,
    c.last_name as Customer_Last_Name,
    s.first_name as Staff_First_Name,
    s.last_name as Staff_Last_Name,
    p.amount as Payment_Amount
from
    public.payment p
inner join public.customer c on
    p.customer_id = c.customer_id
inner join public.staff s on
```

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.

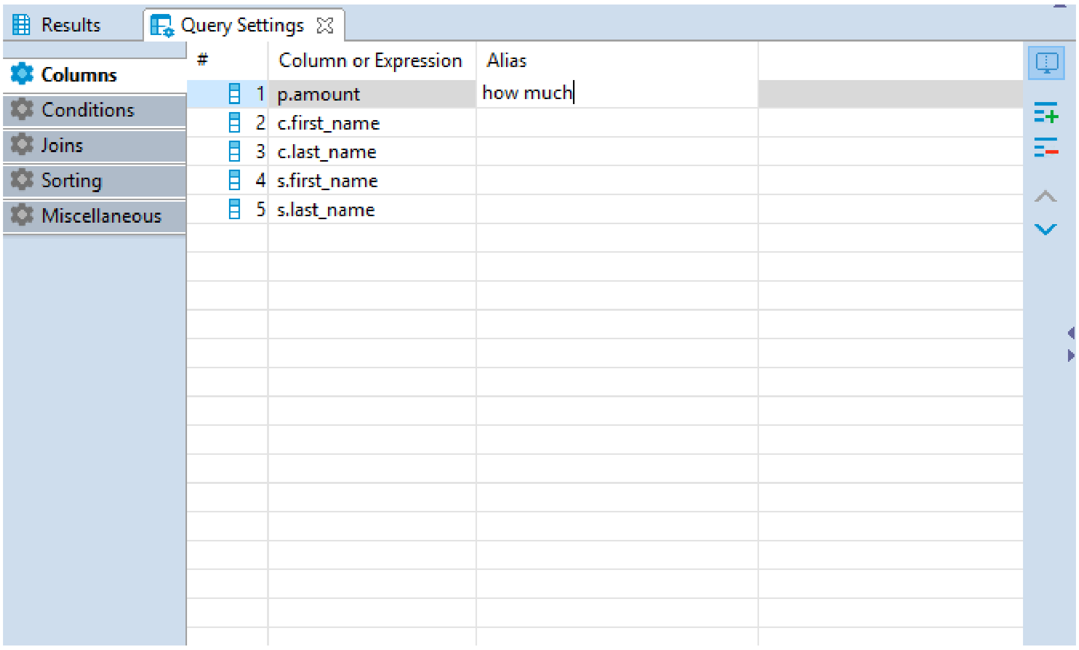


#	Column or Expression	Alias
1	p.amount	
2	p.amount	
3		
4		
5		

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 .

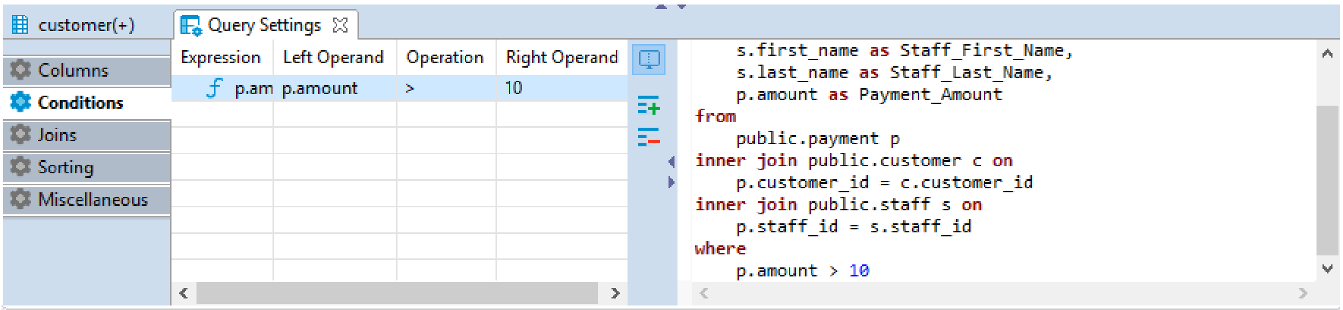
You can also define a user-friendly name of the column to be displayed in the result table. To set a user-friendly name click on a cell in **Alias** column and insert the name. The change will be immediately displayed in the SQL script area.



#	Column or Expression	Alias
1	p.amount	how much
2	c.first_name	
3	c.last_name	
4	s.first_name	
5	s.last_name	

Conditions

Conditions tab is used for managing query conditional expressions.

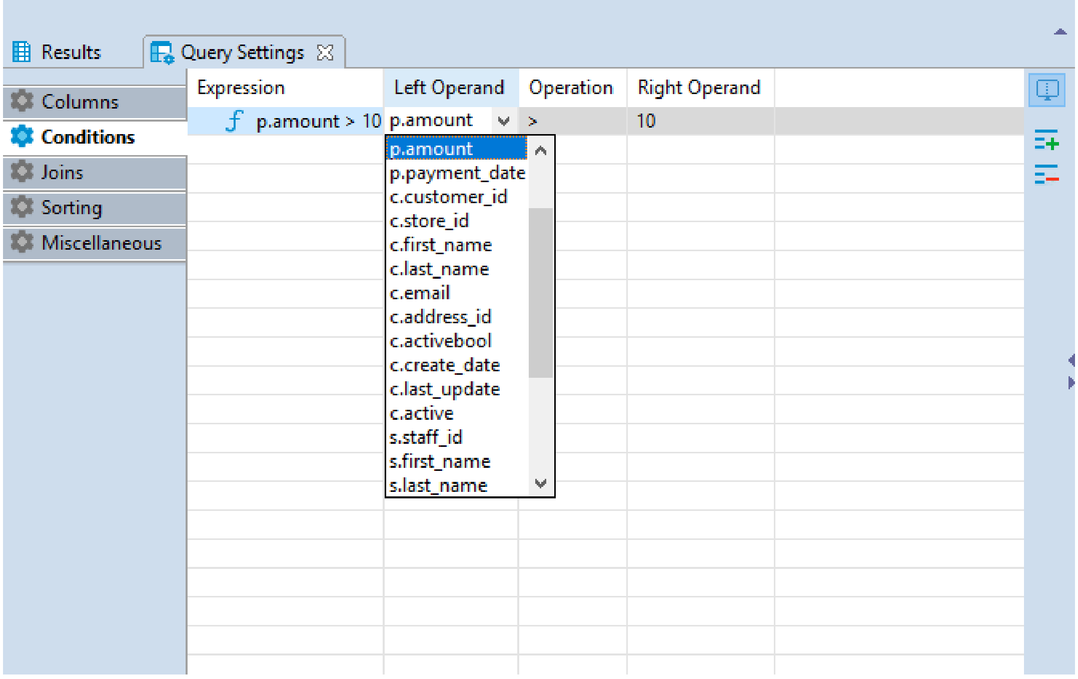


Expression	Left Operand	Operation	Right Operand
p.am	p.amount	>	10

```
        s.first_name as Staff_First_Name,
        s.last_name as Staff_Last_Name,
        p.amount as Payment_Amount
from
        public.payment p
inner join public.customer c on
        p.customer_id = c.customer_id
inner join public.staff s on
        p.staff_id = s.staff_id
where
        p.amount > 10
```

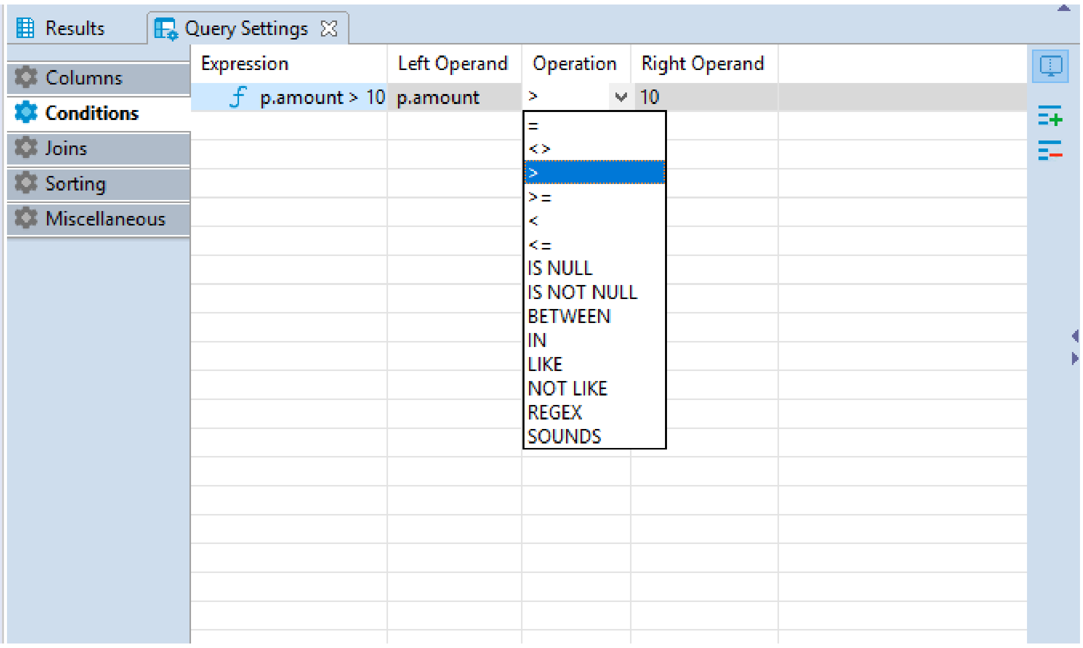
To add a new conditional expression use the **Add** button on the right - a new instance will be added and the default conditional expression WHERE will be added to the SQL script area automatically. This default conditional expression can be then adjusted to the one you need:

- **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.



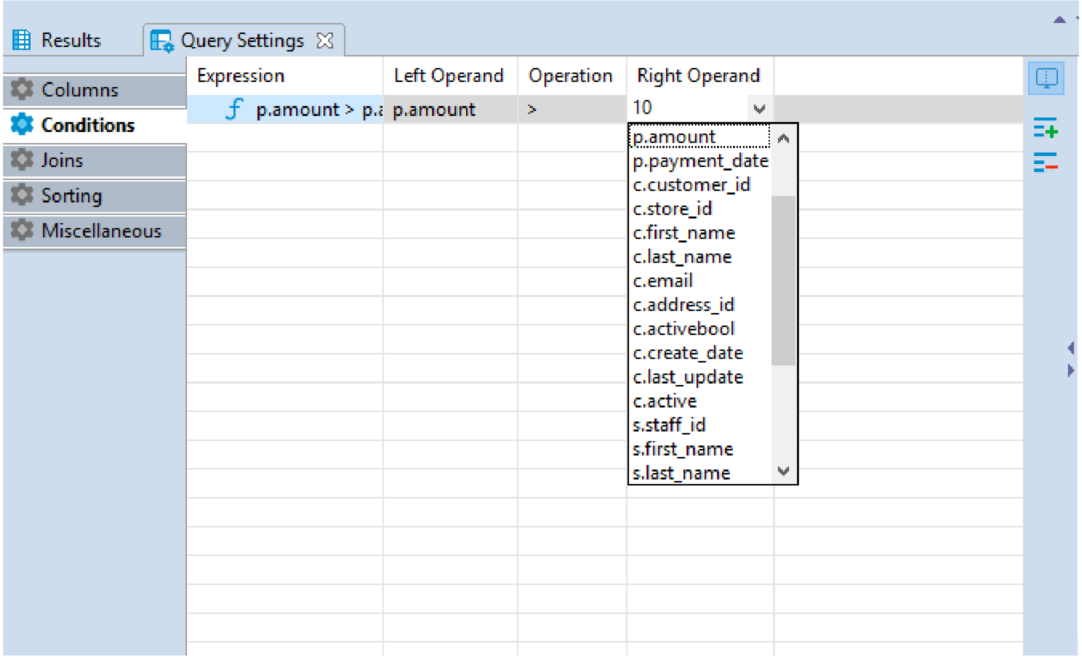
Expression	Left Operand	Operation	Right Operand
p.amount > 10	p.amount	>	10

- **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 appeared.



Expression	Left Operand	Operation	Right Operand
p.amount > 10	p.amount	>	10

- **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.



Expression	Left Operand	Operation	Right Operand
p.amount > p.a	p.amount	>	10

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**.

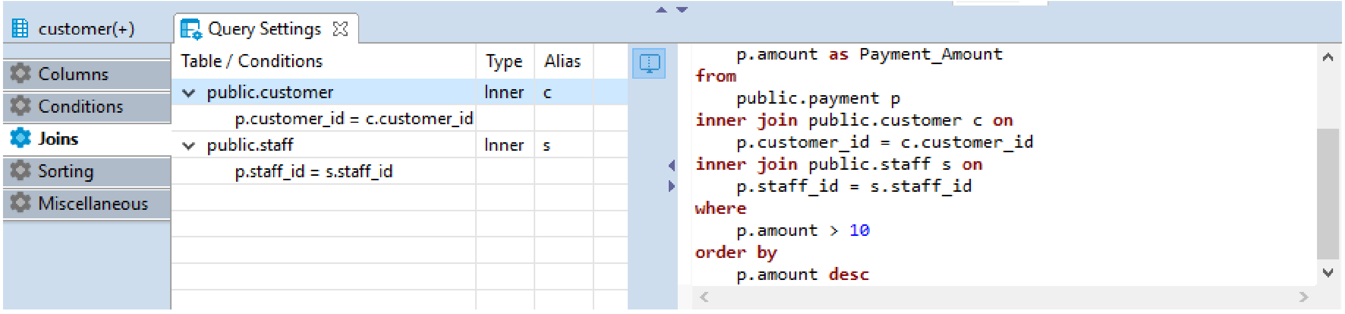


Table / Conditions	Type	Alias
public.customer	Inner	c
p.customer_id = c.customer_id		
public.staff	Inner	s
p.staff_id = s.staff_id		

```
     p.amount as Payment_Amount
from
     public.payment p
inner join public.customer c on
     p.customer_id = c.customer_id
inner join public.staff s on
     p.staff_id = s.staff_id
where
     p.amount > 10
order by
     p.amount desc
```

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.

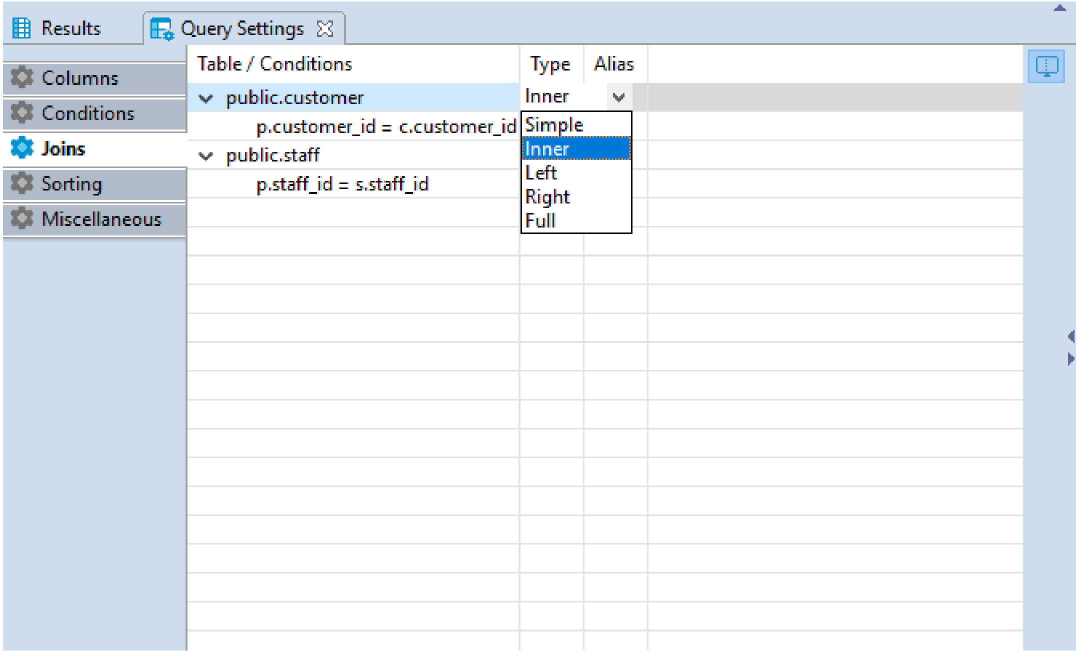


Table / Conditions	Type	Alias
public.customer	Inner	
p.customer_id = c.customer_id		
public.staff		
p.staff_id = s.staff_id		

- **Alias** - defines a user friendly name of the join. To define this setting click on the cell in **Alias** column and input the name.

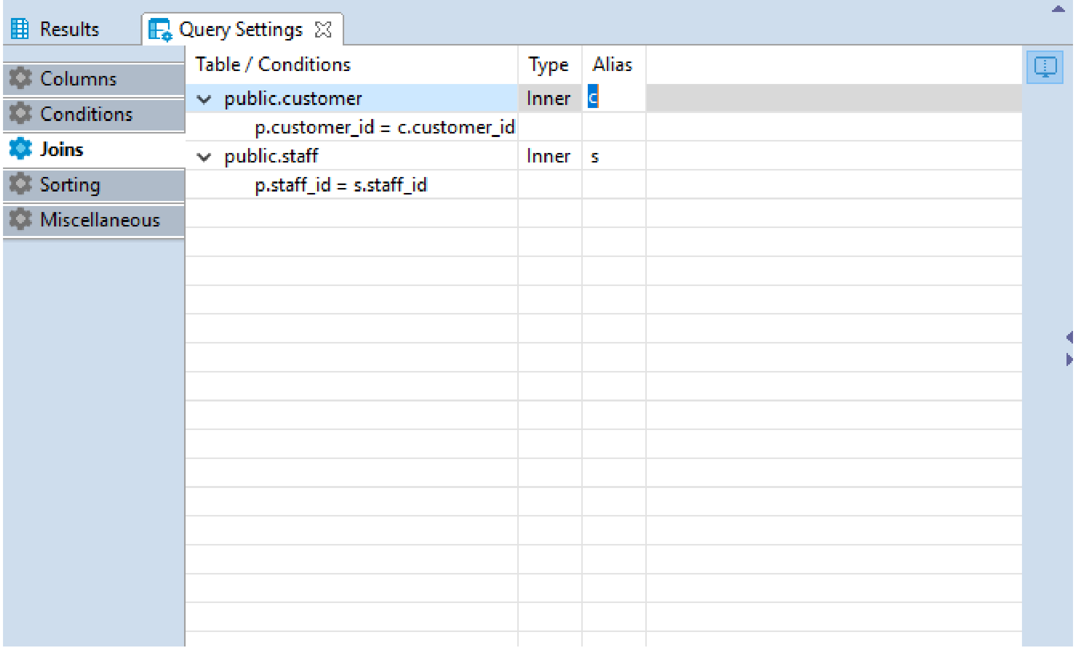
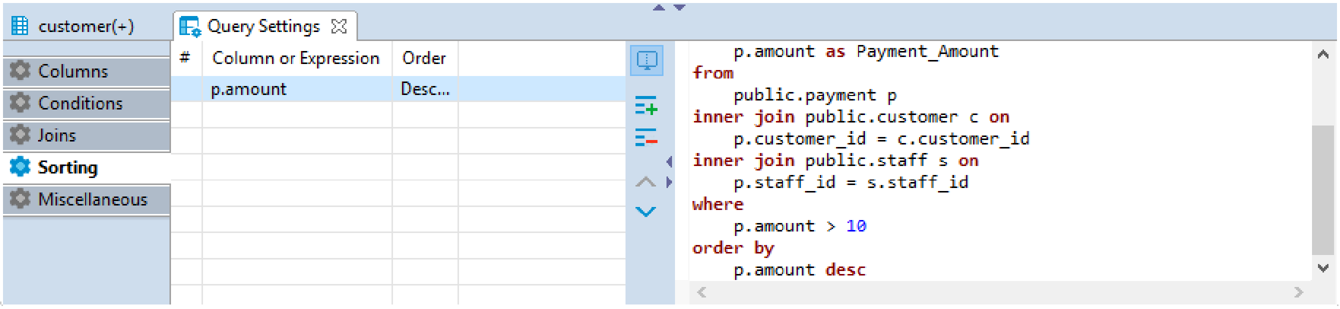


Table / Conditions	Type	Alias
public.customer	Inner	c
p.customer_id = c.customer_id		
public.staff	Inner	s
p.staff_id = s.staff_id		

Sorting

In **Sorting** tab you can set the order of rows in the result table.

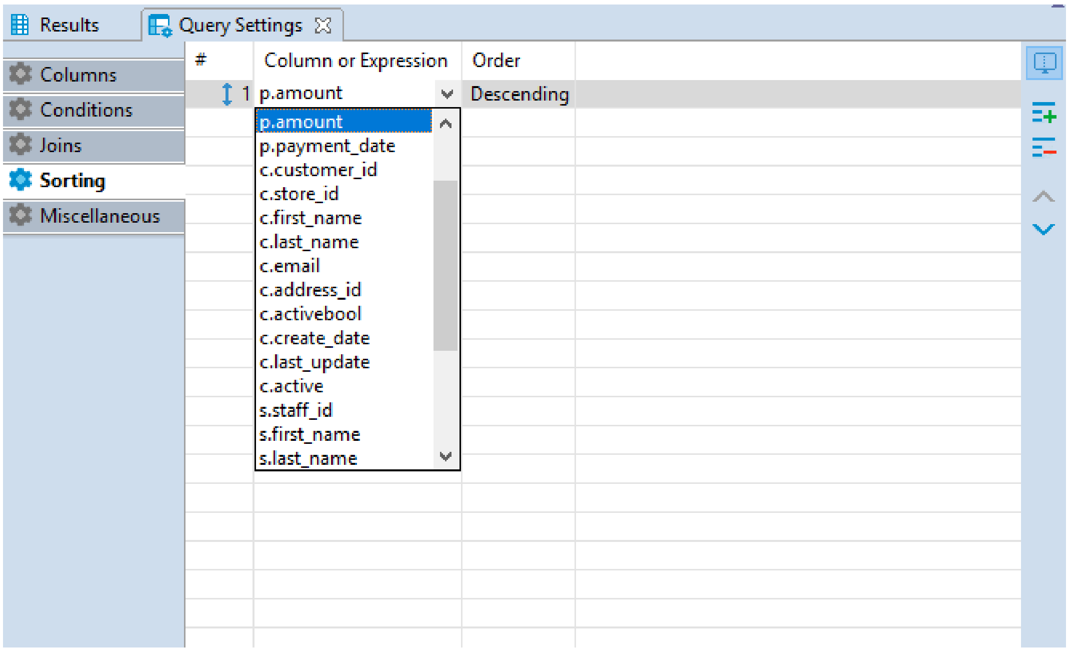


#	Column or Expression	Order
	p.amount	Desc...

```
    p.amount as Payment_Amount
from
    public.payment p
inner join public.customer c on
    p.customer_id = c.customer_id
inner join public.staff s on
    p.staff_id = s.staff_id
where
    p.amount > 10
order by
    p.amount desc
```

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 be then adjusted to the one you need:

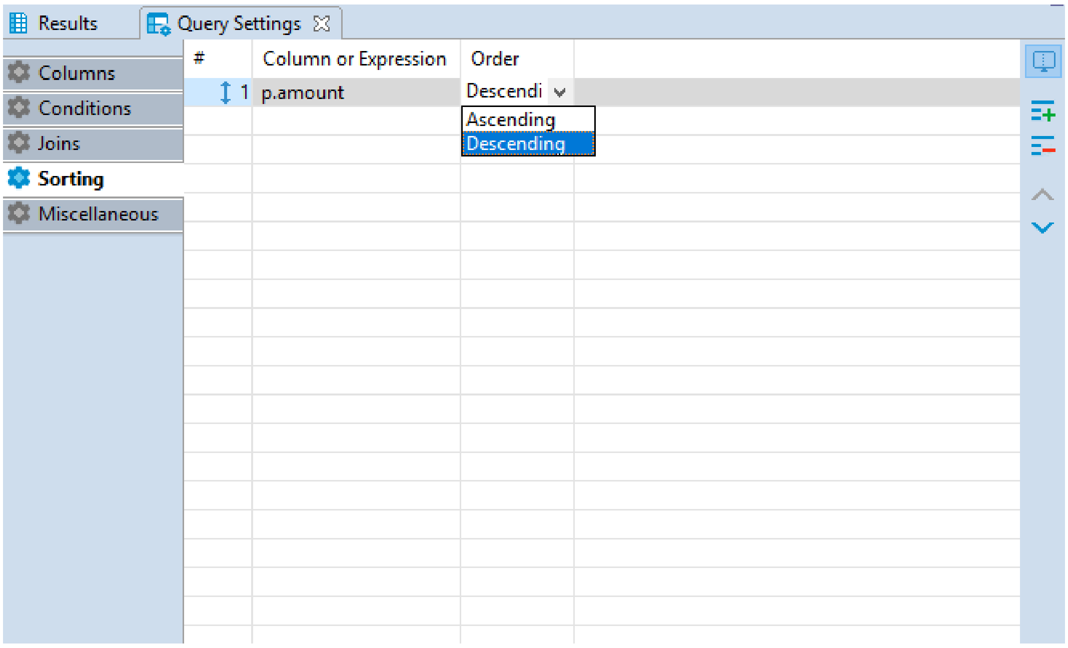
- Once a new condition is added, click the first cell in **Conditions or Expressions** column and a drop down list of all available columns will appear. Select the required column by clicking on its name.



#	Column or Expression	Order
1	p.amount	Descending

```
p.amount
p.payment_date
c.customer_id
c.store_id
c.first_name
c.last_name
c.email
c.address_id
c.activebool
c.create_date
c.last_update
c.active
s.staff_id
s.first_name
s.last_name
```

- In **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 order by command will be added to the script.

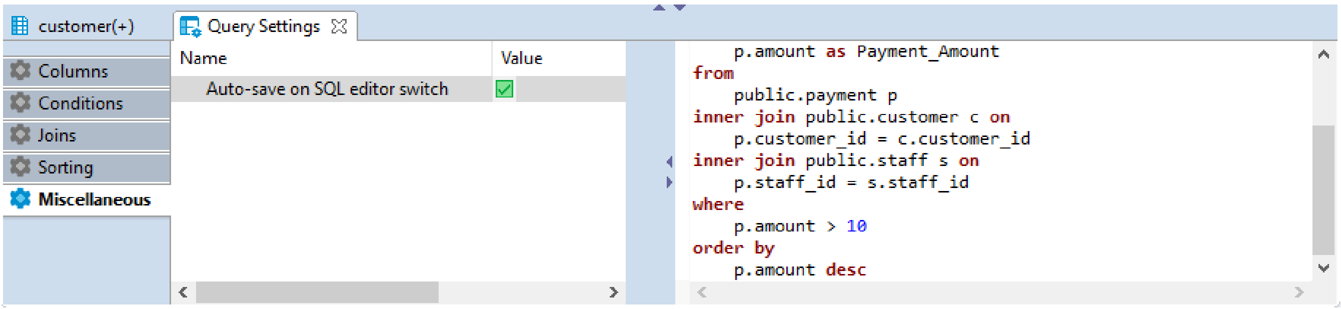


#	Column or Expression	Order
1	p.amount	Descendi

To remove a condition use the **Remove** button on the right.

Miscellaneous

In **Miscellaneous** tab it is possible to autosave on SQL-editor switch by selecting the **Autosave on SQL-editor switch** check-box.



Name	Value
Auto-save on SQL editor switch	☑

```
    p.amount as Payment_Amount
from
    public.payment p
inner join public.customer c on
    p.customer_id = c.customer_id
inner join public.staff s on
    p.staff_id = s.staff_id
where
    p.amount > 10
order by
    p.amount desc
```

Executing Visual Query

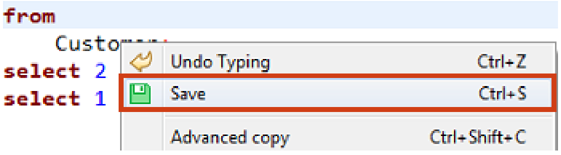
To execute a query, use **Execute SQL statement** button to get the results in the same tab or **Execute SQL statement in new tab** button to get the results in a new tab. Both buttons are located in the **Visual Query Builder** vertical toolbar.

Script Management

Saving Scripts

You can save scripts to a predefined space in the currently active project or somewhere in the file system.

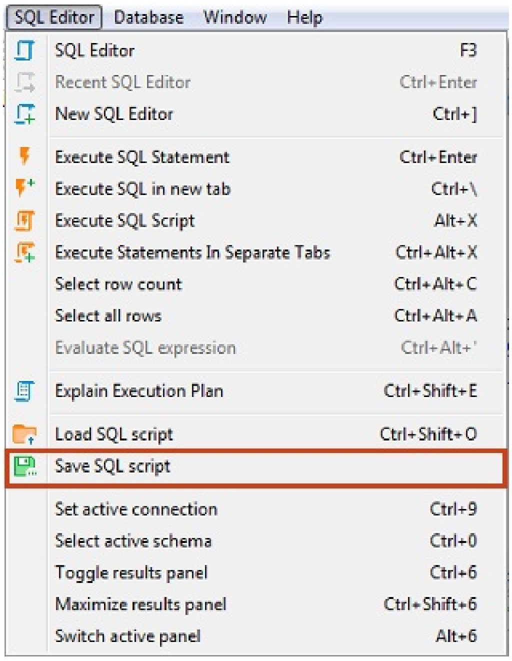
To save a script to the current project space, just press `Ctrl+S` or right-click the script and click **Save** on the context menu:



```
from
    Custo
select 2
select 1
```

You can find the script saved this way in the Project Explorer view in the **Scripts** folder.

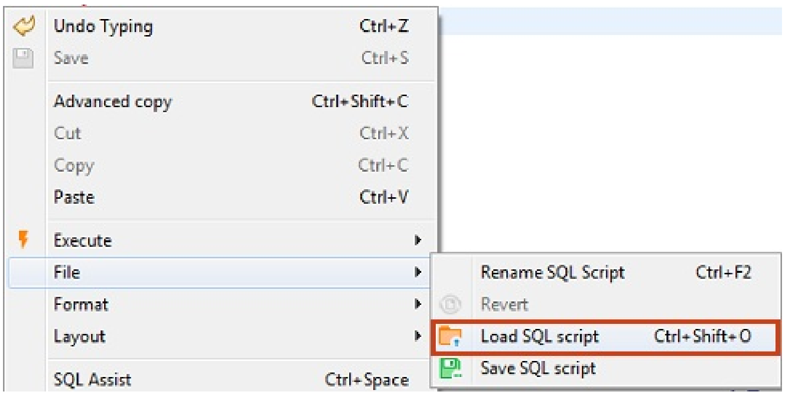
To save a script to the file system, right-click the script, click **File -> Save SQL script** on the context menu and then select the folder in the file system.
You can also, click **SQL Editor -> Save SQL script** on the main menu:



SQL Editor
Database
Window
Help
SQL Editor F3
Recent SQL Editor Ctrl+Enter
New SQL Editor Ctrl+]
Execute SQL Statement Ctrl+Enter
Execute SQL in new tab Ctrl+\
Execute SQL Script Alt+X
Execute Statements In Separate Tabs Ctrl+Alt+X
Select row count Ctrl+Alt+C
Select all rows Ctrl+Alt+A
Evaluate SQL expression Ctrl+Alt+'
Explain Execution Plan Ctrl+Shift+E
Load SQL script Ctrl+Shift+O
Save SQL script
Set active connection Ctrl+9
Select active schema Ctrl+0
Toggle results panel Ctrl+6
Maximize results panel Ctrl+Shift+6
Switch active panel Alt+6

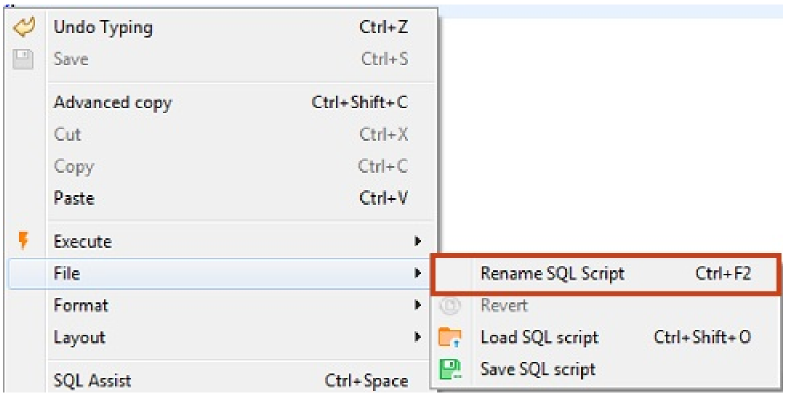
Loading Scripts

To load a script stored in the file system to the SQL Editor, press `CTRL+SHIFT+O`, or click **SQL Editor -> Load SQL script** on the main menu, or right-click the script panel and click **File -> Load SQL script** on the context menu:

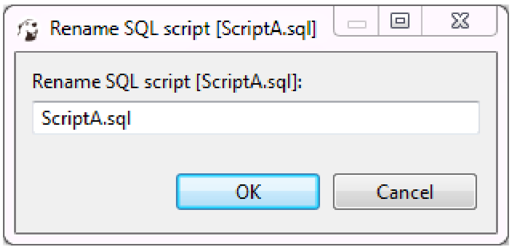


Renaming Scripts

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**:



Rename SQL script [ScriptA.sql]:

ScriptA.sql

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.

SQL Console

In some cases you might want to execute a query and do 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 doesn't has an associated .sql file. Its contents will be lost when you close it.

Client Side Commands

DBeaver supports the following commands:

	Command	Database	Description
@set var = value	All	Sets a script variable. You can use expressions as a value. Variables can be used as SQL queries input parameters.	
@echo message	All	Prints message to output log. You can use a macro in message (for example ${var}).	
@include fileName	All	- 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.	
source fileName	MySQL	The same as @include but in MySQL CLI syntax	
define var = value	Exasol	The same as @set but in Exasol EXAPlus syntax.	

PostgreSQL Debugger

Prerequisites for Debugging

In order to implement interactive debugging of PL/SQL procedures on a Postgres server you need to use plugin_debugger. Plugin_debugger is a typical interactive debugger delivered as an extension and requiring a shared library preload in Postgres to operate the shared_preload_libraries parameter in settings. The debugger is developed and maintained by the community. 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

As the first step, you need to install the **plugin_debugger** extension in your PostgreSQL database. It is free and open source. You can download the source code here. And you can find the installation README file here.

To install the debugger plugin:

1. Copy this directory to **contrib/** in your PostgreSQL source tree.
2. Run **make; make install**.
3. Edit your **postgresql.conf** file and modify the **shared_preload_libraries** config option to look like the following: **shared_preload_libraries = '$libdir/plugin_debugger'**.
4. Restart PostgreSQL for the new setting to take effect.
5. Run the following command in the database or databases that you wish to debug functions in:
   CREATE EXTENSION pldbgapi;
   NOTE: On server versions older than 9.1, instead of running the command, you need to run the **pldbgapi--1.0.sql** script directly using psql.

If you use binary builds from pgdg, you can use repository for your Linux distribution. For more information, please see [Official Postgres page] (https://www.postgresql.org/download/).

After installing the plugin on the database server, you need to install the debug plugin from either a marketplace or P2 repository, see this article for more information.

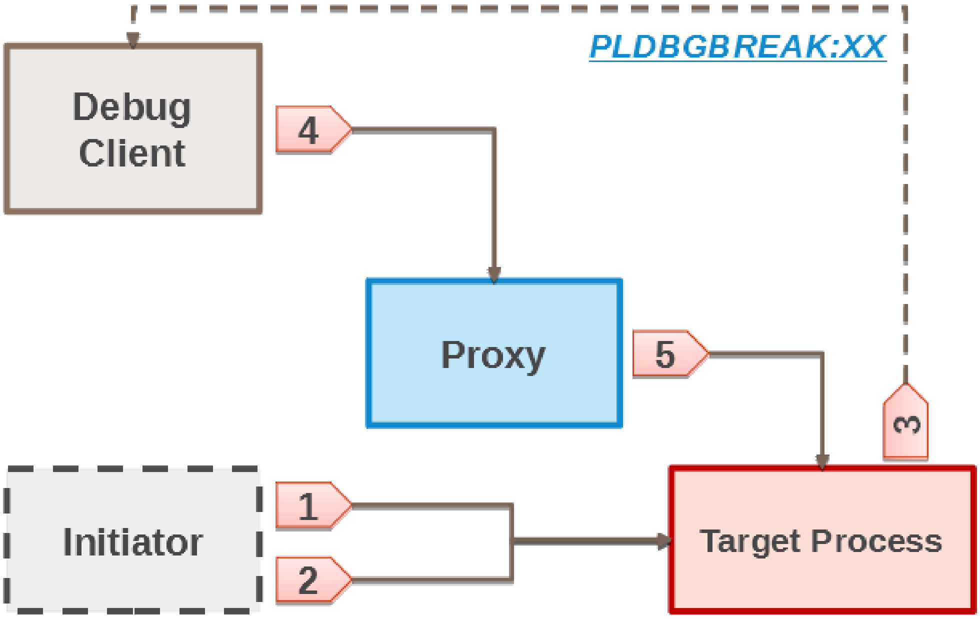
You can find a detailed DBeaverCE Debug plugin installation guide from Jkiss P2 repository here.

How to start debug with local breakpoint

For local breakpoints, stop will be done only for server process launched by the initiator. All other processes will run procedure/function as usual. When you create or run a previously defined debugging session in DBeaver with local breakpoint, the following steps take place:

1. DBeaver creates initiator session and local breakpoint.
2. DBeaver runs procedure\function (with parameters specified) selected for debugging in the initiator session.
3. The server process reaches breakpoint and displays NOTICE to initiator, in which the PORT is reported for PLDBGBREAK:XX debugging. Then the process opens the socket and becomes blocked, waiting for PROXY connection. If Debug Client cannot get PORT name from the process being debugged, an error message appears.
4. Debug Client creates a session, then tries to establish connection with the debugged process, using PORT from step 3 provided by the initiator. The connection is established via API call **attach_to_port**. If the connection cannot be established, the **Error rcv port number** message appears.
5. PROXY receives PORT from the client, then establishes connection to the process, and returns the session identifier to the client. After that, using the identificator, one can:
   - Send debugging commands to the process;
   - Receive responses from the process;

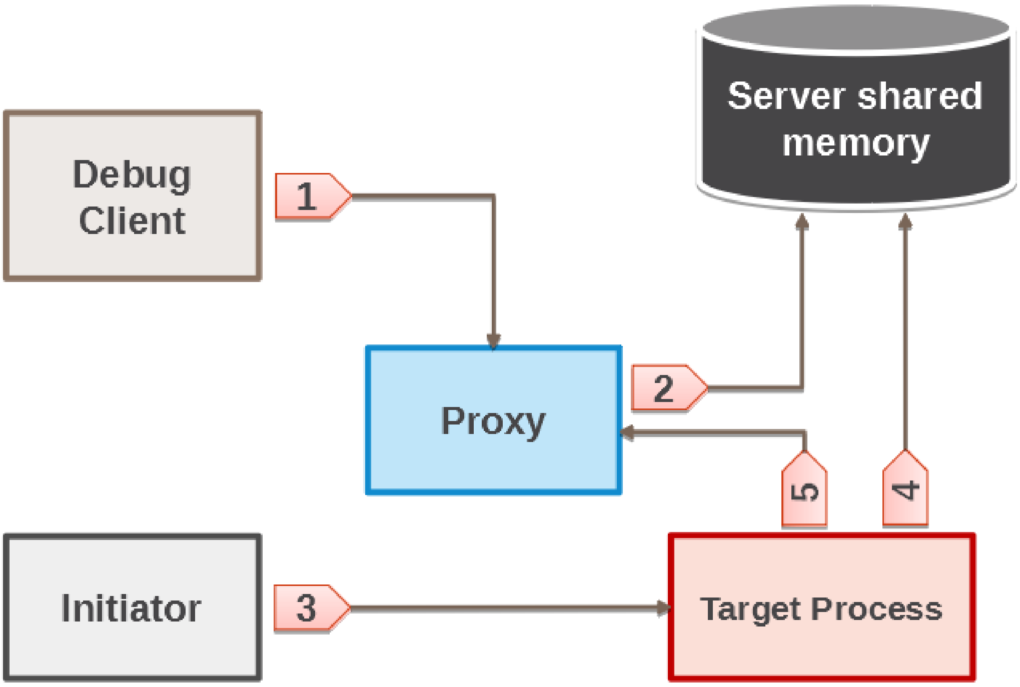
- Translate responses to the client.



How to start debug with global breakpoint

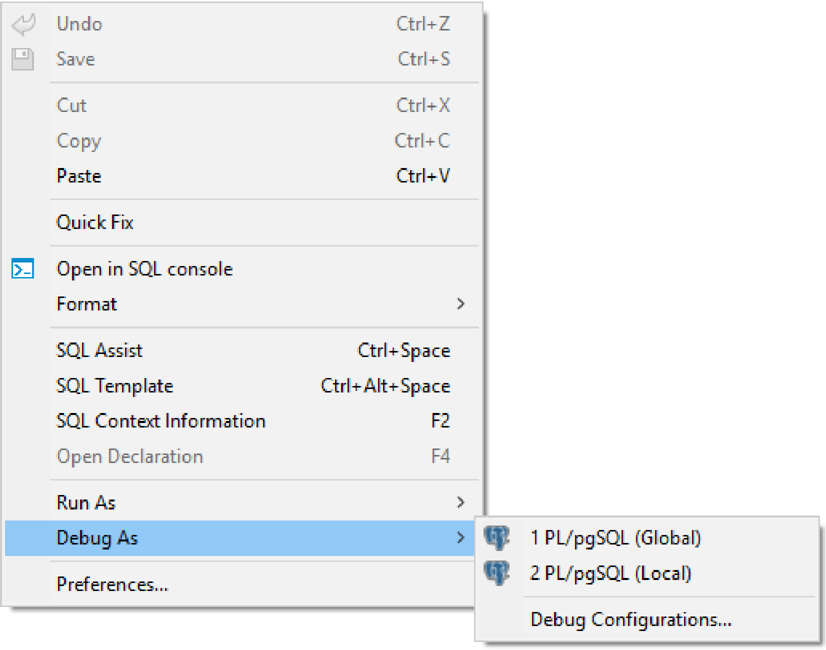
Whenever the breakpoint is global, stop will be done for any server process launched by anyone. No modifications or additional calls from the debugged process are required. When you create a session with a global breakpoint, the following steps take place:

1. The client creates a session and then establishes a connection with PROXY (receiving session). After that, the client creates a global breakpoint for the selected function, and then becomes blocked via the **pldbg_wait_for_target** API call waiting for the server process.
2. PROXY receives the global control point from the client and adds global control point to the shared memory, if the point of this type has not been added before.
3. The initiator creates a connection and runs the selected procedure.
4. On each executed line, the target process checks if the global breakpoint with corresponding conditions exists, and becomes blocked if the breakpoint exists and conditions are met.
5. The target process establishes a connection with PROXY and reports that the breakpoint is reached.
6. Afterwards, the process becomes blocked and waits for PROXY (client) commands. The process will be implementing commands unless the procedure finishes or an exception is thrown.



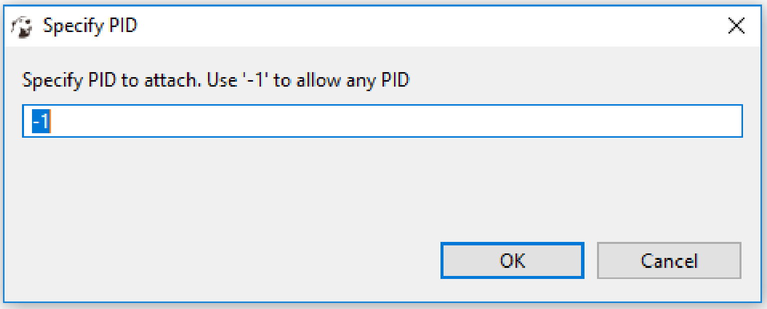
Running debugger in DBeaver interface

If you have successfully installed the debugger plugin, and the plugin is up and running, you can start Debug from procedure source page by right-clicking the procedure source text - a context menu appears:



You need to choose the type of your debug session: Local or Global.

For a **Global** session, you need to specify the target process PID filter (enter **-1** value for any process):

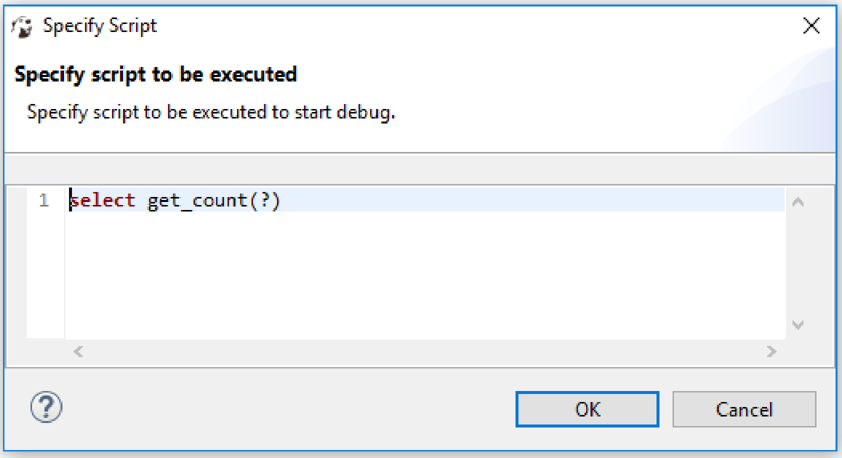

Specify PID

Specify PID to attach. Use '-1' to allow any PID

-1

After running a Global session, you need to wait for any (or specified PID) process to call your procedure.

For a **Local** session, you need to specify the executable SQL sentence for starting the target process:


Specify Script

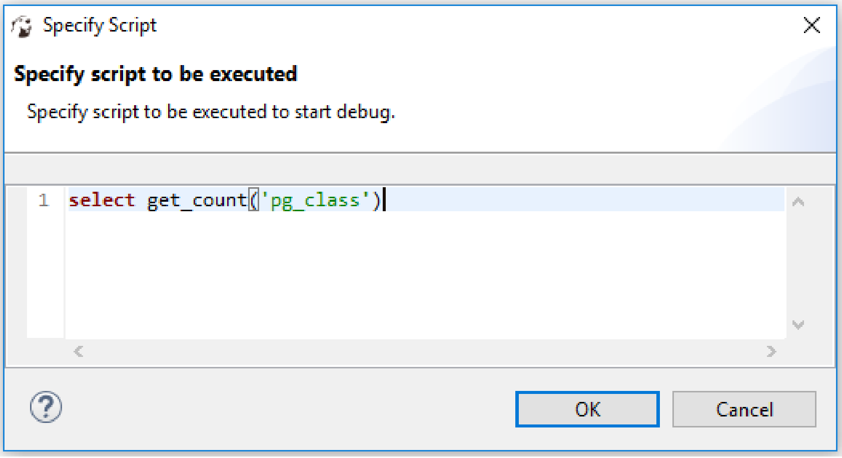
Specify script to be executed

Specify script to be executed to start debug.

```
select get_count(?)
```

In the script editor window, you can see the **select get_count(?)** text that is just a specification of how DBeaver must call your function(procedure) for debug.

DBeaver is not intended to know about the values of the variables in the procedure being debugged. You need to set the values for call arguments BEFORE you start debugging. Your settings might look as follows:


Specify Script

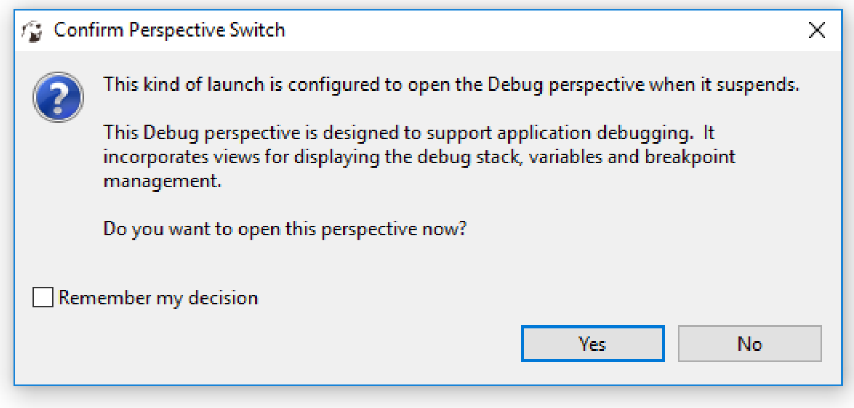
Specify script to be executed

Specify script to be executed to start debug.

```
select get_count('pg_class')
```

If you run debugging now, DBeaver creates a separate connection to the database server, tries to execute the given SQL sentence **select get_count('pg_class')** in this connection, and attaches the debug session to the running SQL context.

After successfully attaching to the target server process, the platform prompts you to switch to the debug perspective:



Confirm Perspective Switch

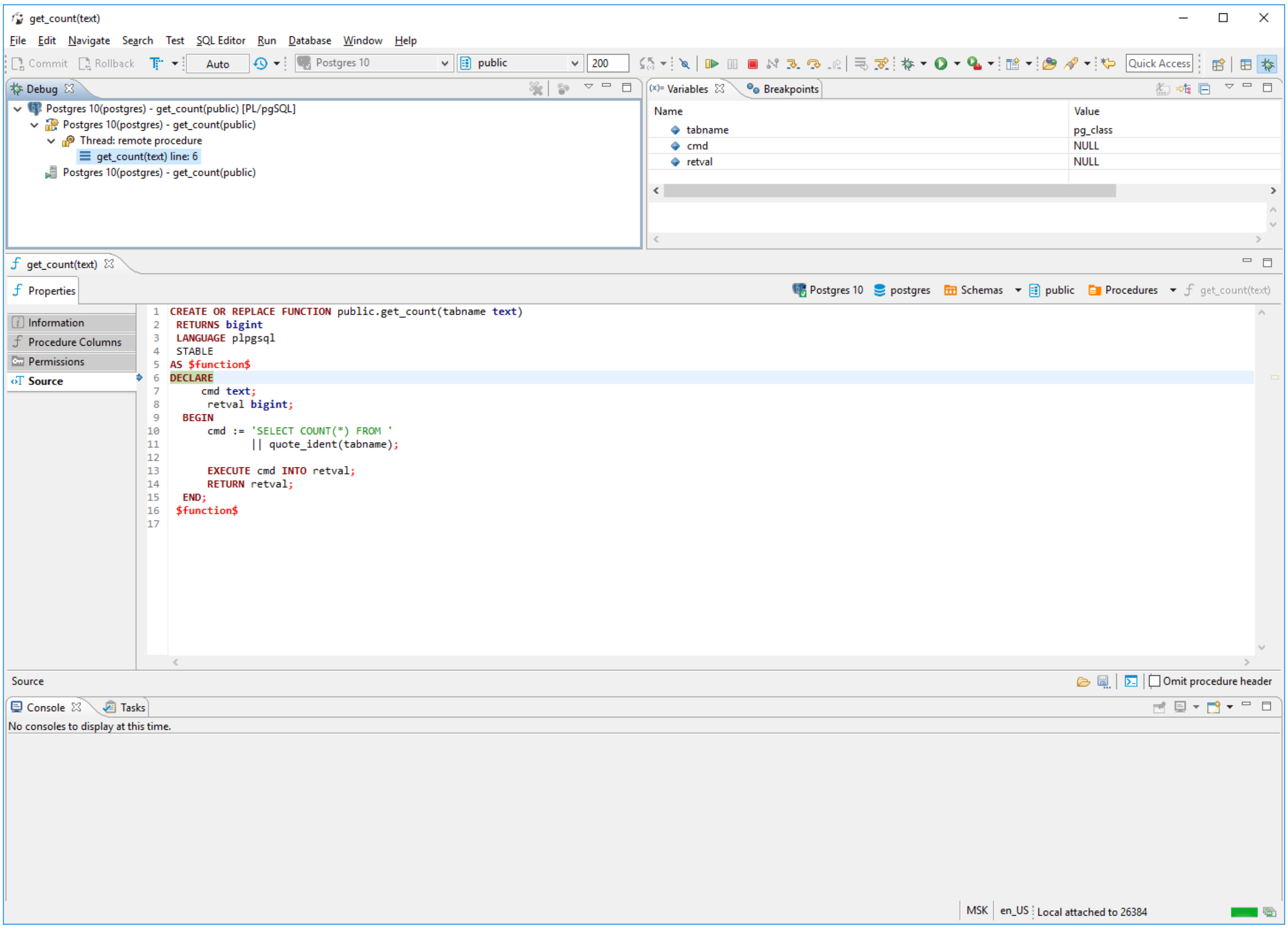
This kind of launch is configured to open the Debug perspective when it suspends.

This Debug perspective is designed to support application debugging. It incorporates views for displaying the debug stack, variables and breakpoint management.

Do you want to open this perspective now?

□ Remember my decision

After you click **OK**, the debug toolset (perspective) opens:



get_count(text)

- Postgres 10(postgres) - get_count(public) [PL/pgSQL]
  - Postgres 10(postgres) - get_count(public)
    - Thread: remote procedure
      - get_count(text) line: 6
  - Postgres 10(postgres) - get_count(public)

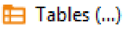
Name	Value
tabname	pg_class
cmd	NULL
retval	NULL

```
CREATE OR REPLACE FUNCTION public.get_count(tabname text)
 RETURNS bigint
 LANGUAGE plpgsql
 STABLE
AS $function$
DECLARE
     cmd text;
     retval bigint;
  BEGIN
     cmd := 'SELECT COUNT(*) FROM '
            || quote_ident(tabname);

     EXECUTE cmd INTO retval;
     RETURN retval;
  END;
 $function$
```

No consoles to display at this time.

Filter Database Objects

In Database Navigator and Database Object Editor you can filter database objects to include or exclude some of them from the view. You can filter 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 names of tables and views using the filter field above the tree of objects:

genre

To filter objects by the name of a table and view, type the name in the field. The tree dynamically updates to show tables / views with that name. To reset the filter, click the Clear icon ( ) on the right end of the field.

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 for folder or parent nodes of database objects - like 'Tables', 'Indexes', etc. Allows creating a complex filter with multiple filtering criteria, see Configure Filters.

A 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

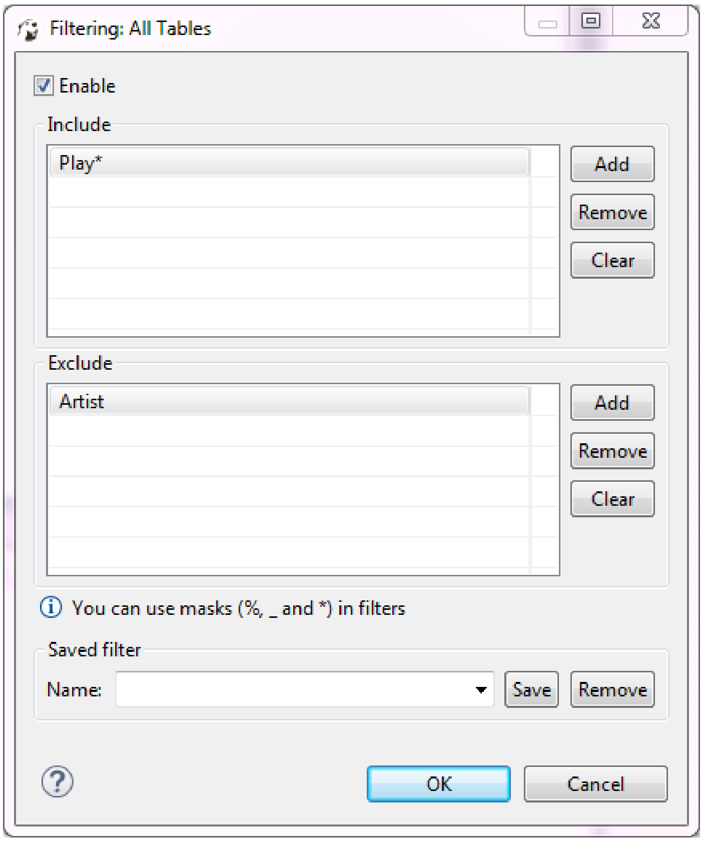
To reset such filters, right-click the parent (folder) node displaying the dots sign `(...)`, and then click **Filter -> Clear filter**.

Configure 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.



Filtering: All Tables

☑ Enable

Include

Play*

Exclude

Artist

ⓘ You can use masks (%, _ and *) in filters

Saved filter

Name:

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 filer 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: You can use masks with `%` and `*` to replace one or more symbols and `_` to replace one symbol in the search combination.

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 saves filters. To remove a filter, in the **Name** drop-down list, click the filter name and then click **Remove**.

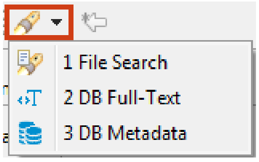
8. Click **OK** to apply the filtering criteria. Otherwise, click **Cancel**.

Search

DBeaver provides:

- File search (search among file contents)
- Database full-text search
- Database metadata search

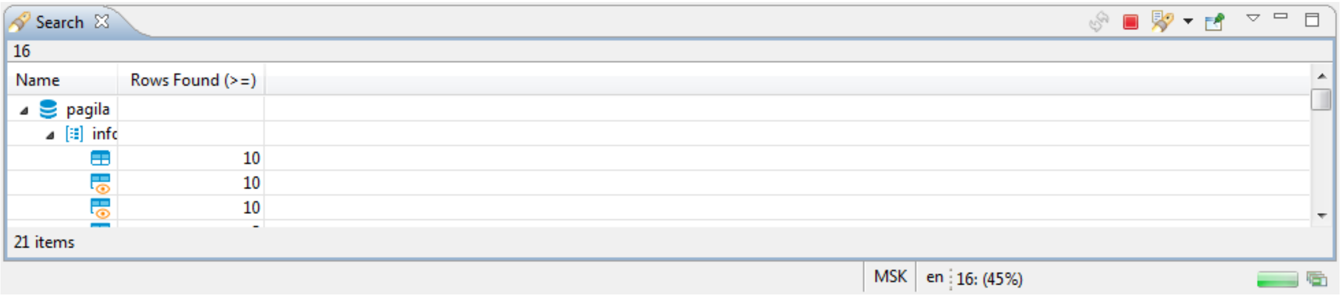
To use search, click the Search button in the main toolbar:



Please see dedicated articles for information about search of 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:



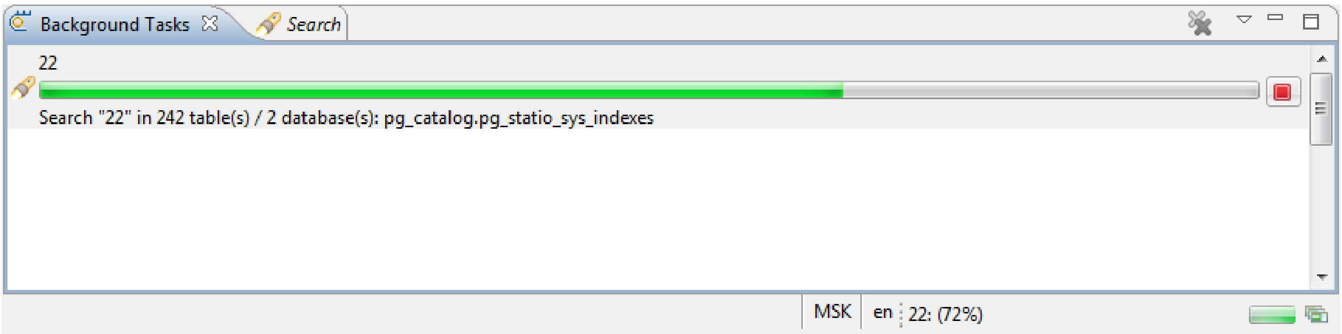
The view contains a toolbar that provides common tools for all types of search as well as specific tools for the File Search type. The following are common tools:

Button	Name	Description
↻	**Run the Current Search Again**	Repeats the search the results of which are displayed in the Search view
■/🟥	**Cancel Current Search**	- 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**	- Clicking the button itself opens the Previous Searches window. - Clicking the arrow next to the button opens a dropdown menu. See the *Search History* 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:

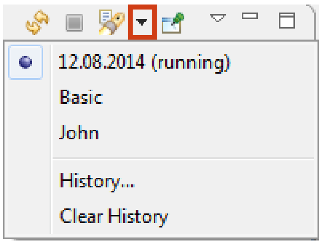

Background Tasks
Search
22
Search "22" in 242 table(s) / 2 database(s): pg_catalog.pg_statio_sys_indexes
MSK
en
22: (72%)

Search History

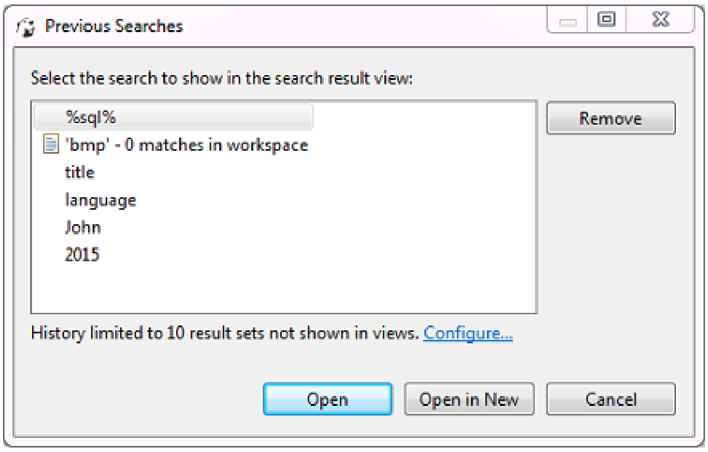
DBeaver stores the history of search queries made during the current session. You can reopen the Search view with results of a previous search query. You can also remove individual queries and clear the history. To manage the search history, use the **Show Previous Searches** button in the toolbar ( ).

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 **Open** to open it in the active Search view or **Open in New** to open it in a new view:


Previous Searches
Select the search to show in the search result view:
%sql%
'bmp' - 0 matches in workspace
title
language
John
2015
Remove
History limited to 10 result sets not shown in views. Configure...
Open
Open in New
Cancel

To remove one or more 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.

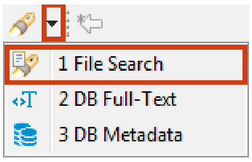
2. Click the query to remove or select several of them by clicking and simultaneously holding the `Ctrl` key.

3. Click **Remove**.

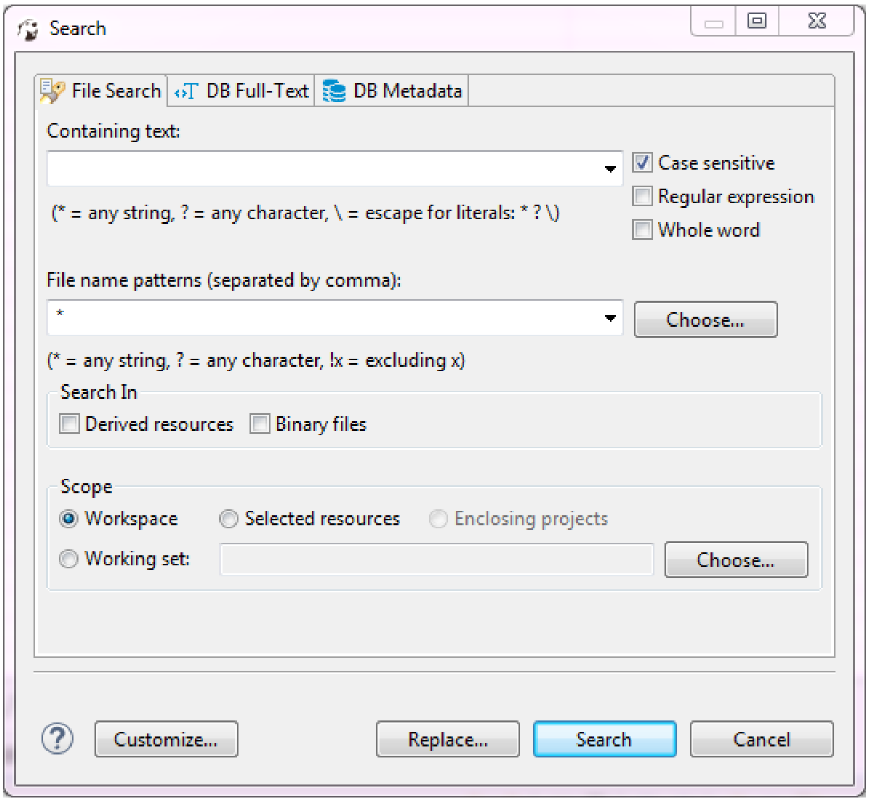
To clear the history by removing all previous queries, click the arrow next to the **Show Previous Searches** button in the toolbar and then click **Clear History** on the dropdown menu.

File Search

To search file contents for a string, click the Search button in 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:



Containing text:

(* = any string, ? = any character, \ = escape for literals: * ? \)

☑ Case sensitive
☐ Regular expression
☐ Whole word

File name patterns (separated by comma):

*

(* = any string, ? = any character, !x = excluding x)

Search In

☐ Derived resources ☐ Binary files

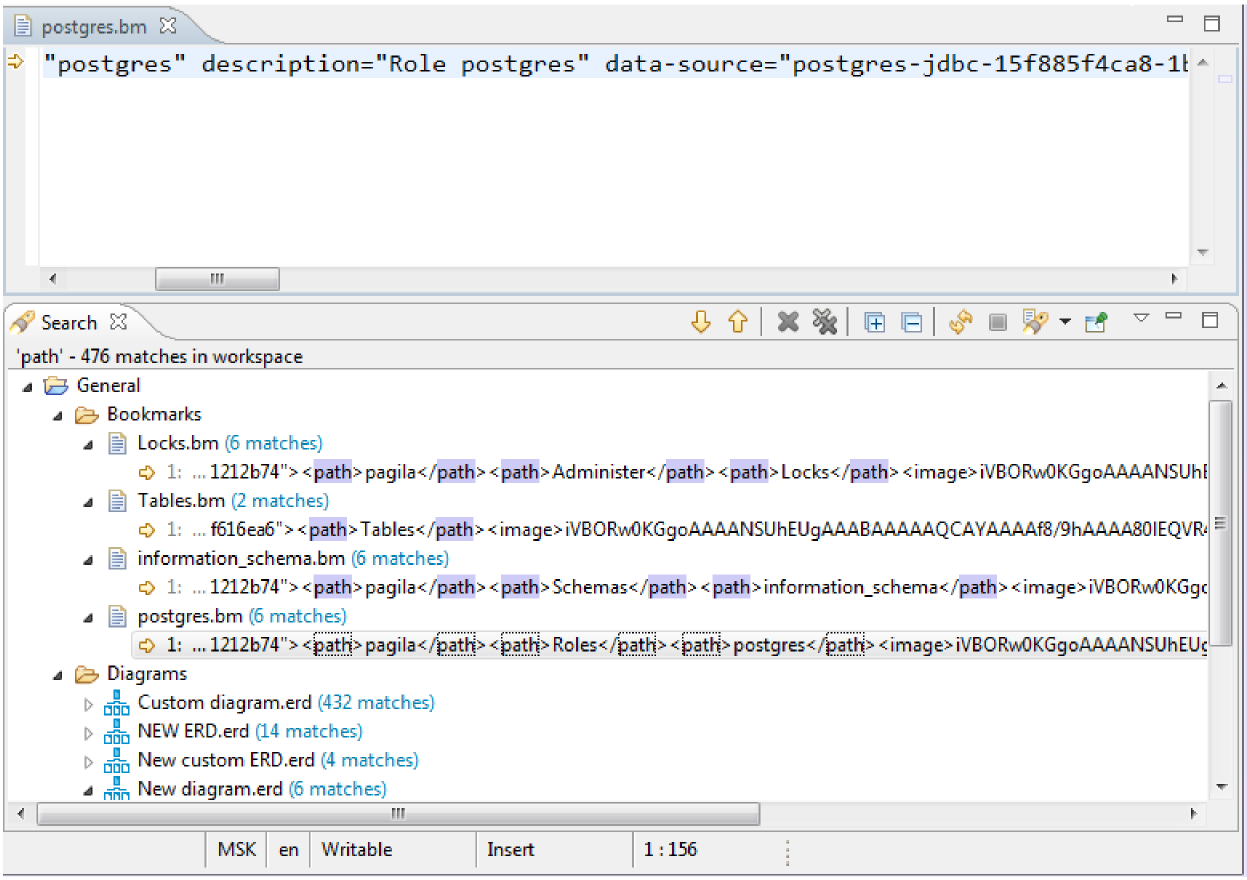
Scope

● Workspace ○ Selected resources ○ Enclosing projects

○ Working set:

You can apply 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:



postgres.bm

```
"postgres" description="Role postgres" data-source="postgres-jdbc-15f885f4ca8-1l
```

Search

'path' - 476 matches in workspace

- General
  - Bookmarks
    - Locks.bm (6 matches)
      - 1: ...1212b74"><path>pagila</path><path>Administer</path><path>Locks</path><image>iVBORw0KGgoAAAANSUhE
    - Tables.bm (2 matches)
      - 1: ... f616ea6"><path>Tables</path><image>iVBORw0KGgoAAAANSUhEUgAAABAAAAAQCAYAAAAf8/9hAAAA80IEQVR
    - information_schema.bm (6 matches)
      - 1: ...1212b74"><path>pagila</path><path>Schemas</path><path>information_schema</path><image>iVBORw0KGgo
    - postgres.bm (6 matches)
      - 1: ...1212b74"><path>pagila</path><path>Roles</path><path>postgres</path><image>iVBORw0KGgoAAAANSUhEUg
  - Diagrams
    - Custom diagram.erd (432 matches)
    - NEW ERD.erd (14 matches)
    - New custom ERD.erd (4 matches)
    - New diagram.erd (6 matches)

MSK | en | Writable | Insert | 1 : 156

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

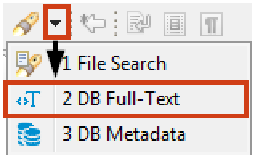
The view also provides a view menu (click the **View Menu** button (▽) in the upper-right corner of the view) that contains the following items:

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

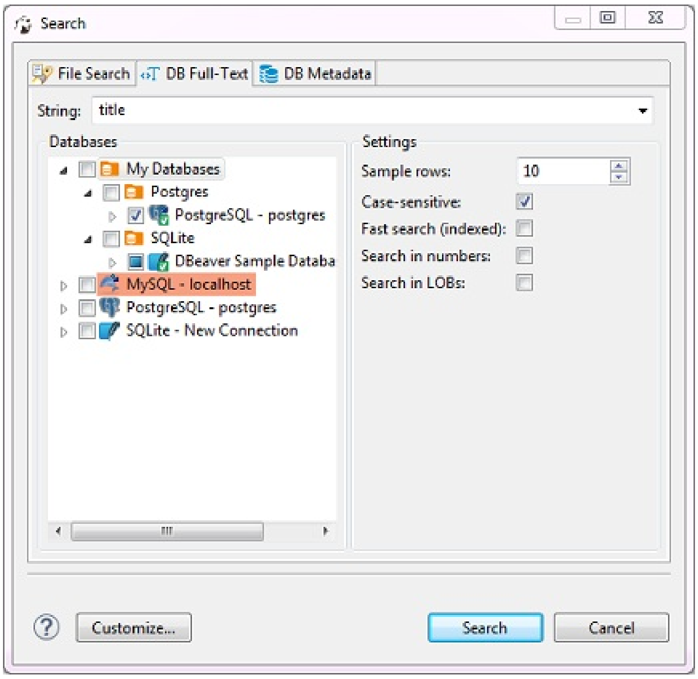
Double-clicking a results row opens it in a separate viewer.

DB Full-Text Search

To do full text search in the database contents, 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:



String: title

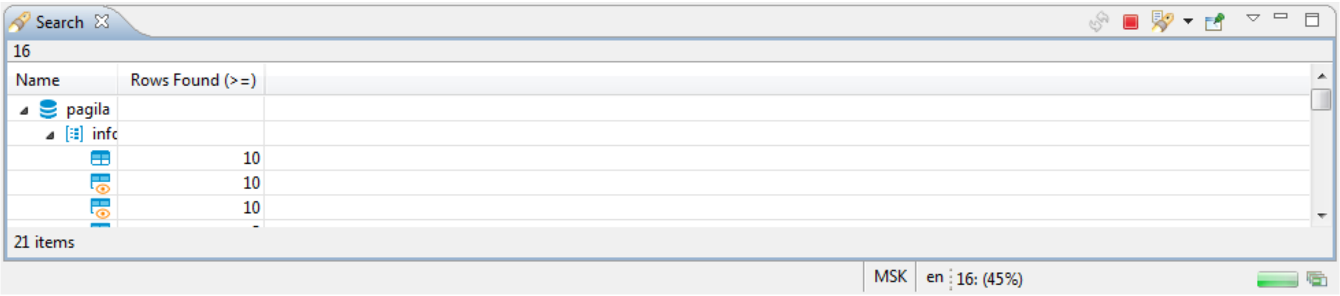
Databases:
- □ My Databases
  - □ Postgres
    - ☑ PostgreSQL - postgres
  - □ SQLite
    - ■ DBeaver Sample Databa
- □ MySQL - localhost
- □ PostgreSQL - postgres
- □ SQLite - New Connection

Settings:
- Sample rows: 10
- Case-sensitive: ☑
- Fast search (indexed): □
- Search in numbers: □
- Search in LOBs: □

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 case-sensitive search, fast search and search in numbers and LOBs.

After you click **Search**, the results open in a Search view:



16

Name	Rows Found (>=)
pagila	
infc	
	10
	10
	10

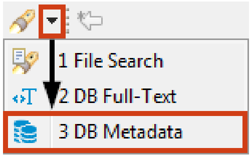
21 items

MSK en 16: (45%)

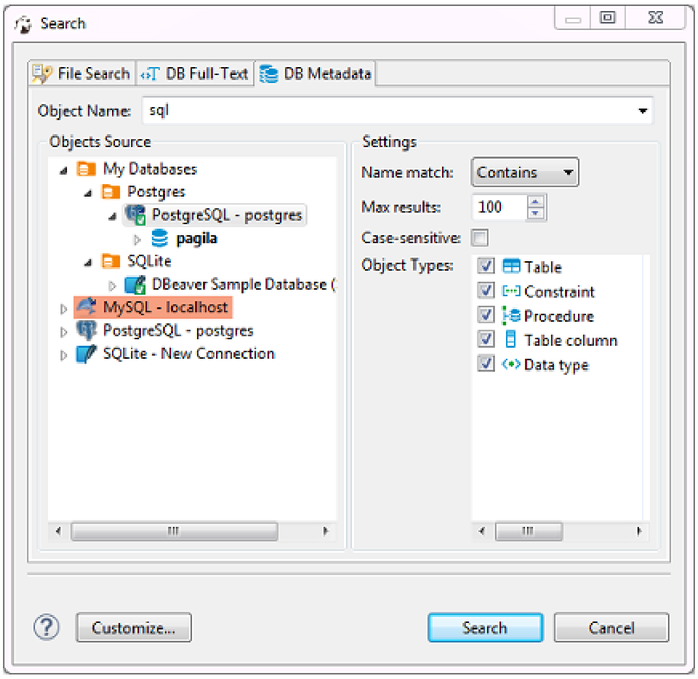
Double-clicking a row in the Search view opens the respective object in a dedicated Database Object editor.

DB 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:



Object Name: sql

Name match: Contains

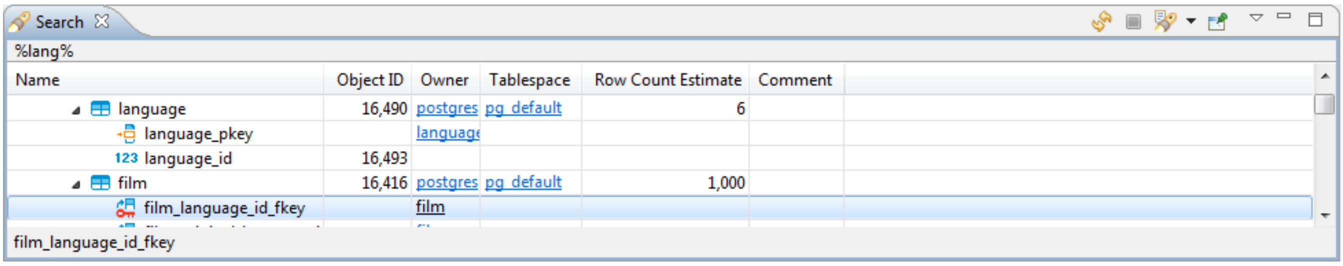
Max results: 100

Case-sensitive: ☐

Object Types: ☑ Table ☑ Constraint ☑ Procedure ☑ Table column ☑ Data type

Now you need to choose the database connection against which to run the search – expand the tree in the **Objects Source** field to the database connections level and click the required database connection.
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 if the metadata should start with or contain or be 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 open in a Search view:



%lang%

Name	Object ID	Owner	Tablespace	Row Count Estimate	Comment
language	16,490	postgres	pg_default	6	
language_pkey		language			
language_id	16,493				
film	16,416	postgres	pg_default	1,000	
film_language_id_fkey		film			

film_language_id_fkey

Double-clicking a row in the Search view opens the respective object in a dedicated Database Object editor.

Dashboards, DB monitoring

Dashboards tool allows DBAs and programmers to quickly identify performance, disk space issues, 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 data bases as PostgreSQL, MySQL, Oracle and Exasol. Custom dashboards are also supported. To learn more about custom dashboards, see Managing Dashboards section below.

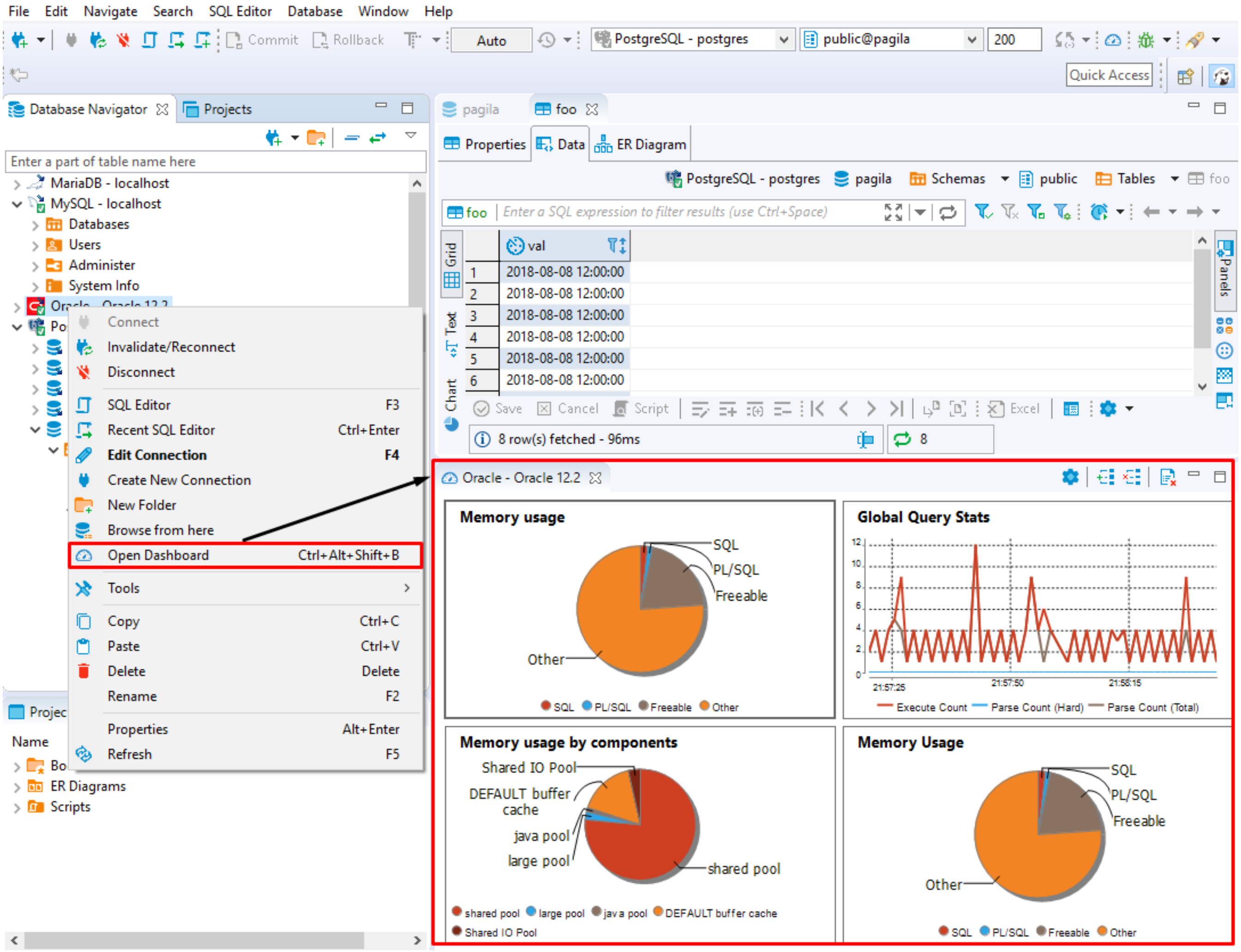
Managing Dashboards Panel

Dashboards panel is a collection of real-time dashboards, that is dashboards that are continuously updated. Dashboards displayed on the dashboards panel are actually a combination of continiously run SQL SELECT queries and charts continiously built on the data fetched.

Opening Dashboard Panel

To open dashboards panel **Open Dashboard** button in 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.

You can also right-click a connection name in the **Database Navigator** editor and select **Open Dashboard** menu option or use keyboard shortcut Ctrl+Alt+Shift+B and the dashboards panel will be opened.



	val
1	2018-08-08 12:00:00
2	2018-08-08 12:00:00
3	2018-08-08 12:00:00
4	2018-08-08 12:00:00
5	2018-08-08 12:00:00
6	2018-08-08 12:00:00

8 row(s) fetched - 96ms

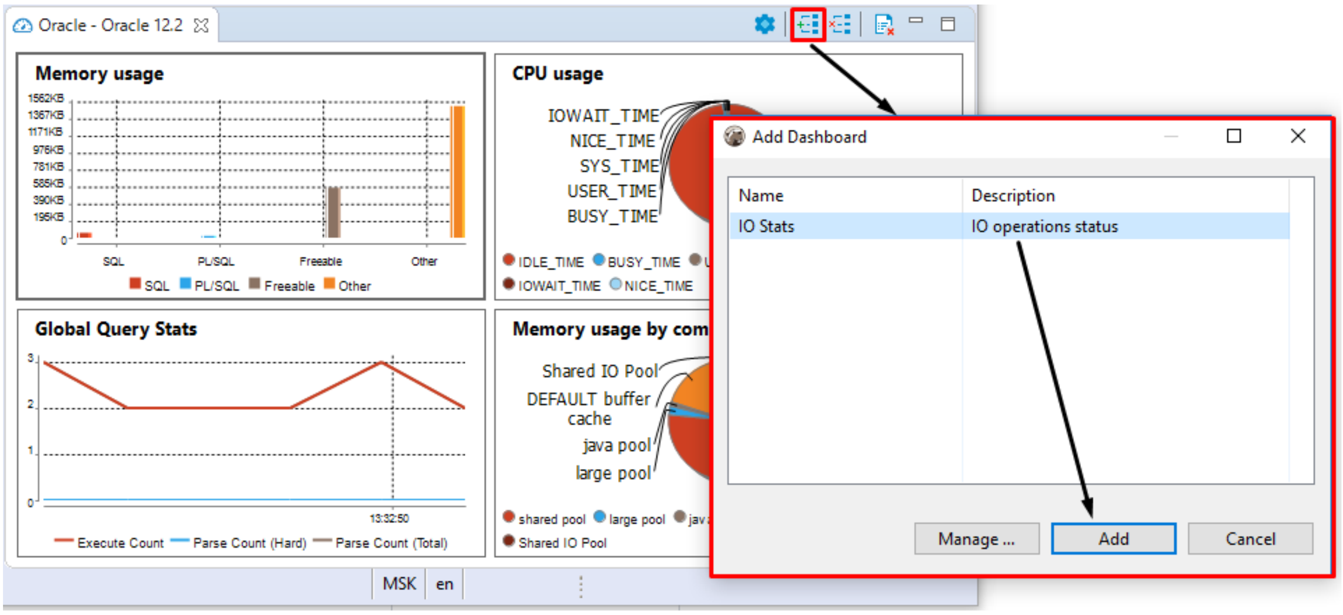
The following controls are available in the dashboards panel toolbar:

Icon	Name	Description
⚙	**Settings**	Allows managing dashboards' configuration.
⊞	**Add dashboard**	Allows to add dashboards to the dashboard panel.
⊟	**Remove dashboard**	Allows to remove dashboards from the dashboard panel.

Icon	Name	Description
	Reset dashboards	Allows to restart dashboard calculation.

Adding Dashboards

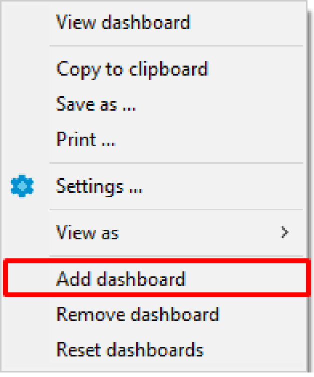
To add a dashboard to the dashboards panel, press **Add dashboard** button in the dashboards panel's toolbar, choose one of the dashboards from the list of available dashboards and press **Add** button.



Name	Description
IO Stats	IO operations status

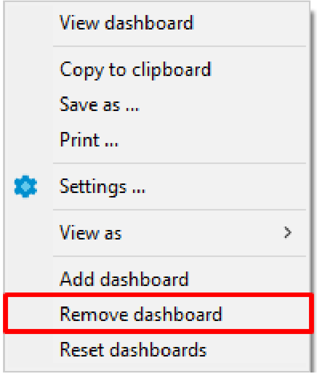
Note: Different databases have different sets of predefined dashboards. DBeaver is delivered with sets of predefined dashboards for such databases as Postgress SQL, MySQL, Oracle, and Exasol. It is also possible to create new custom dashboards, for more details see Managing Dashboards.

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.

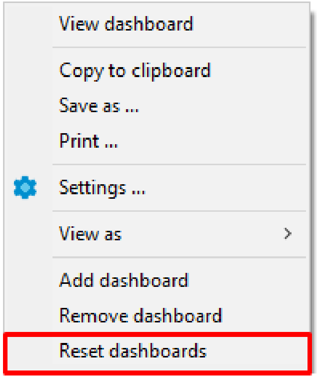


Resetting Dashboards

If you want to restart dashboard's calculation you can reset it.

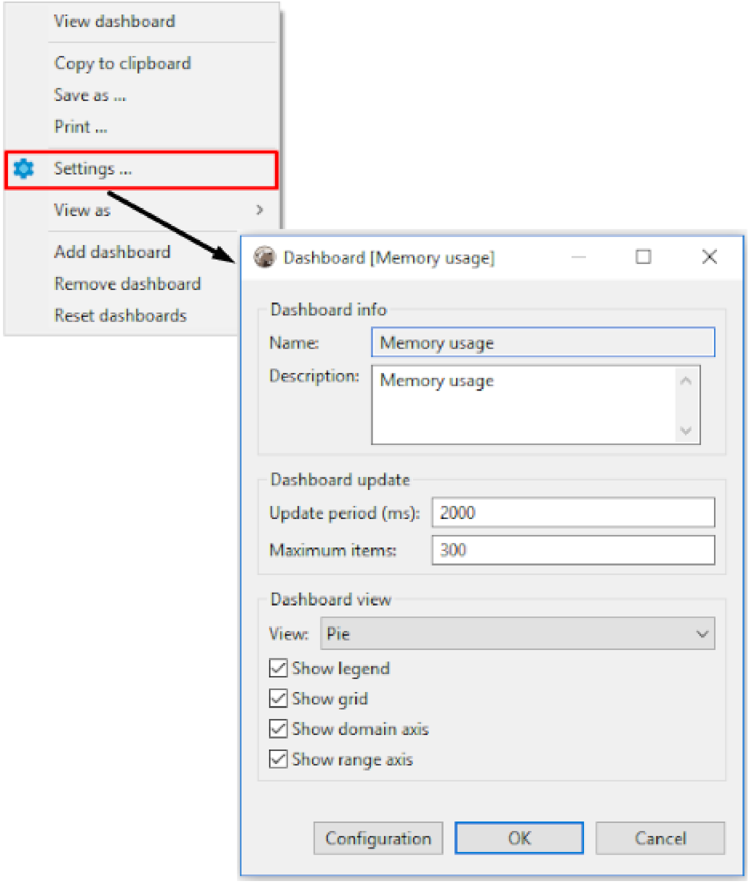
You can reset all the dashboards displayed in the dashboards panel by a single click on **Reset dashboards** button in the dashboard panel's toolbar.

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 in 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.



Dashboard info

Name: Memory usage

Description: Memory usage

Dashboard update

Update period (ms): 2000

Maximum items: 300

Dashboard view

View: Pie

☑ Show legend

☑ Show grid

☑ Show domain axis

☑ Show range axis

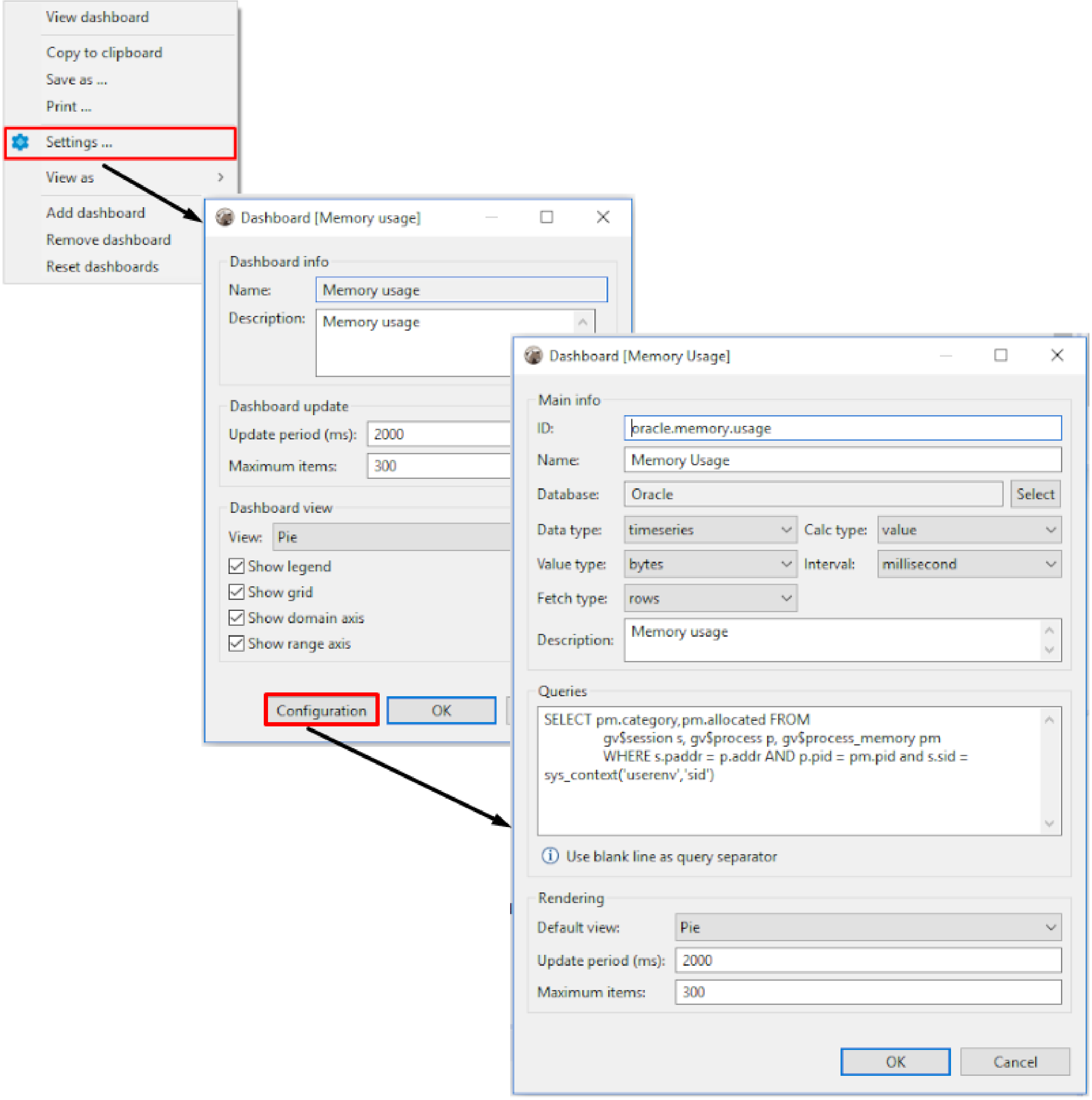
The following dashboard representation parameters can be adjusted:

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.

Parameter	Description
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.



Dashboard info

Name: Memory usage

Description: Memory usage

Dashboard update

Update period (ms): 2000

Maximum items: 300

Dashboard view

View: Pie

☑ Show legend

☑ Show grid

☑ Show domain axis

☑ Show range axis

Main info

ID: oracle.memory.usage

Name: Memory Usage

Database: Oracle

Data type: timeseries

Calc type: value

Value type: bytes

Interval: millisecond

Fetch type: rows

Description: Memory usage

Queries

```
SELECT pm.category,pm.allocated FROM
        gv$session s, gv$process p, gv$process_memory pm
        WHERE s.paddr = p.addr AND p.pid = pm.pid and s.sid =
sys_context('userenv','sid')
```

ⓘ Use blank line as query separator

Rendering

Default view: Pie

Update period (ms): 2000

Maximum items: 300

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 dasboard's name.
Database	Defines the database driver. To learn moe about database drivers, see Database Drivers.
Data type	Defines the data type. The following options are availabe: timerseries (the default option) and statistics. Select timeseries type if you want to track the actual value returned by the server. Select 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're 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 convinient to be tracked in KBytes.
Interval	Defines time interval to be shown on the domain axis. The following time intervals are available: millicecond(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 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 as templates to create new dashboards with any query and other settings. To learn about creating new dashboards, see Managing Dashboards section.

Setting Connection Prefereces

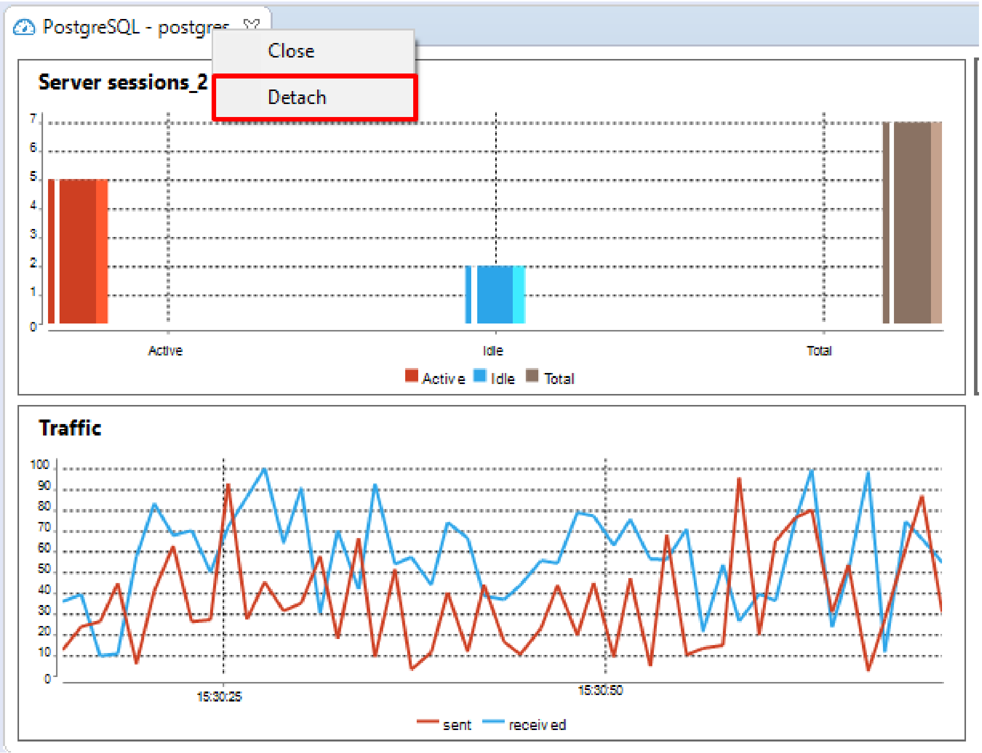
By default, if there is no active connection to the database and you open its dashboards panel, all the dashboards on the panel will be empty.

You can force database connection on the dashboard panel's activation by pressing **Settings** button ⚙ on the dashboards panel's toolbar and then selecting the **Connect on activation** check-box.

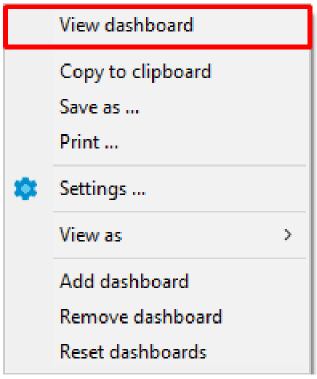
Detaching Dashboards

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.

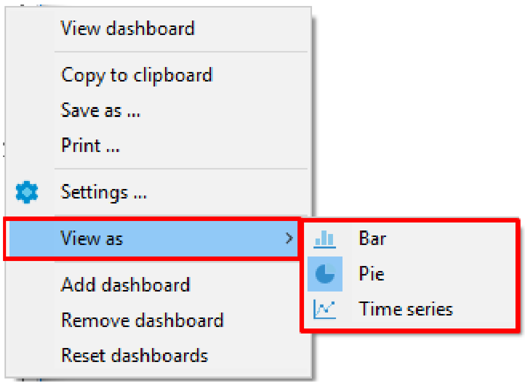


To detach a single dashboard make a double left click over it. You can also right click the dashboard and then, select the **View Dashboard** menu option, the dashboard will be detached from the panel and you will be able to move it to any place of your screen.



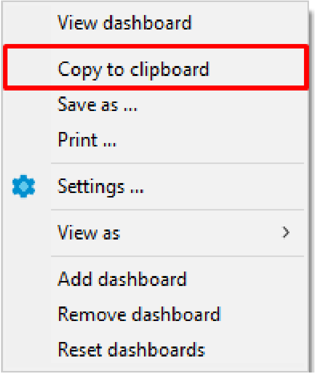
Changing Dashboard View

You can change the representation of a dashboard and view it as a Pie, Bar or Time series. To change the view of a dashboard, right click on it and select **View as** menu option.



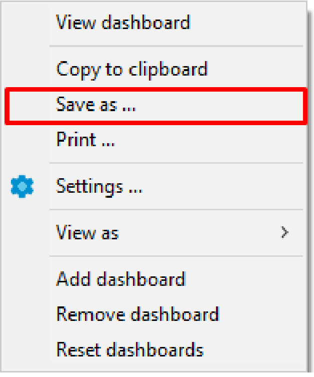
Copying Dashboards to Clipboard

To copy a dashboard into the clipboard, right click on the dashboard and use **Copy to Clipboard** menu option, the screenshot of the dashboard will be placed to the clipboard.



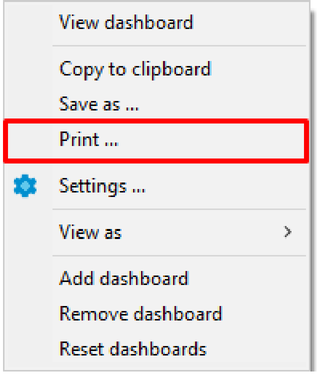
Saving Dashboards

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

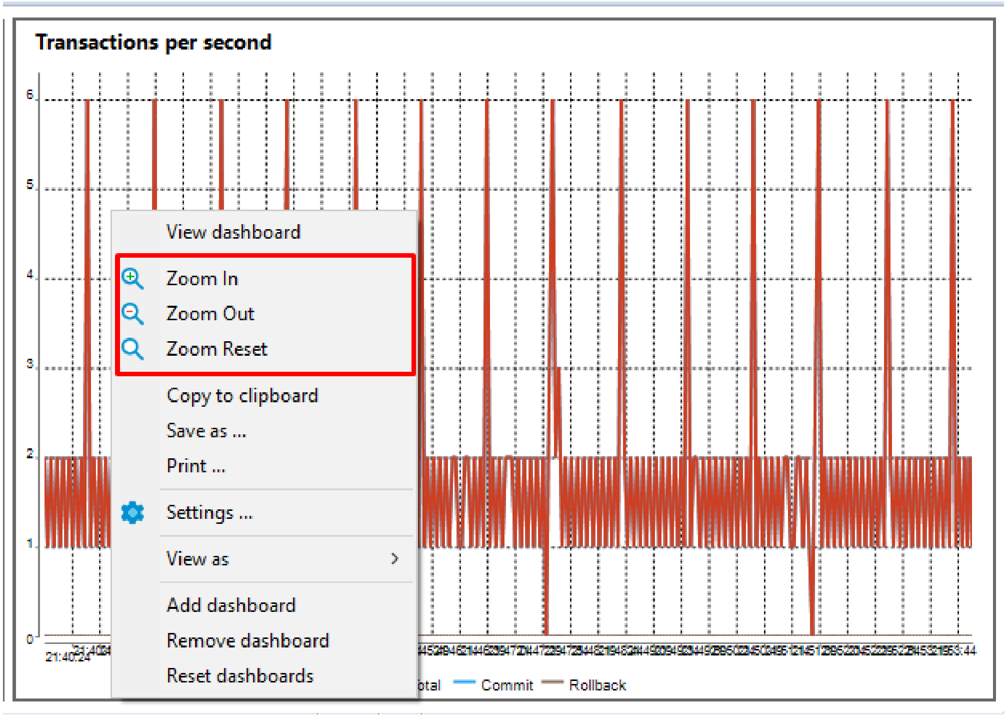
If you want to print out a screenshot of a dashboard, right-click the dashboard to be printed and select the **Print…** option.



Zooming

For Time series and Bar dashboard representations the following zooming options are available in the dashboard's context menu:

- Zoom In
- Zoom Out
- Zoom Reset



Managing Dashboards

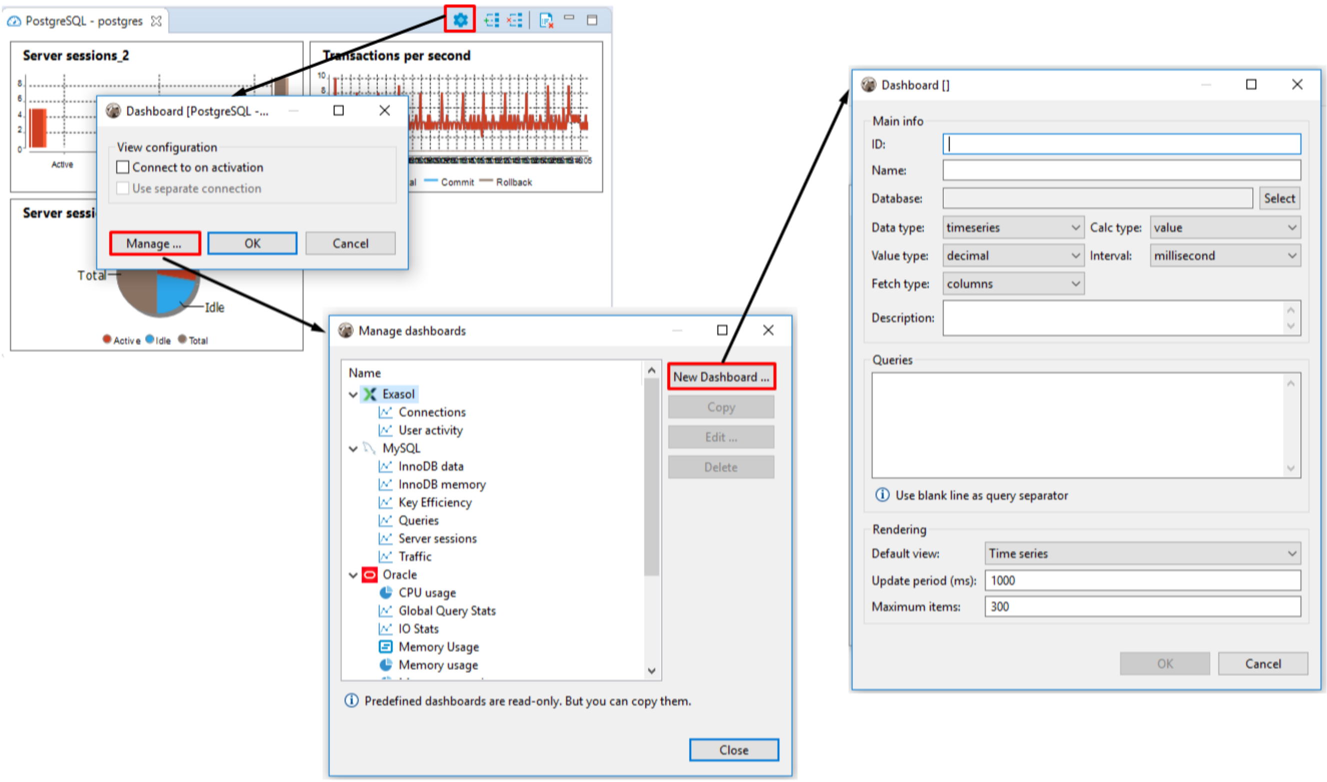
You can extend the list of predefined default dashboards by creating your own custom dashboards. This section describes dashboards' list management.

Creating Dashboards

You can create a new custom dashboard either from scratch or from any of already existing dashboards.

To create a dashboard from scratch:

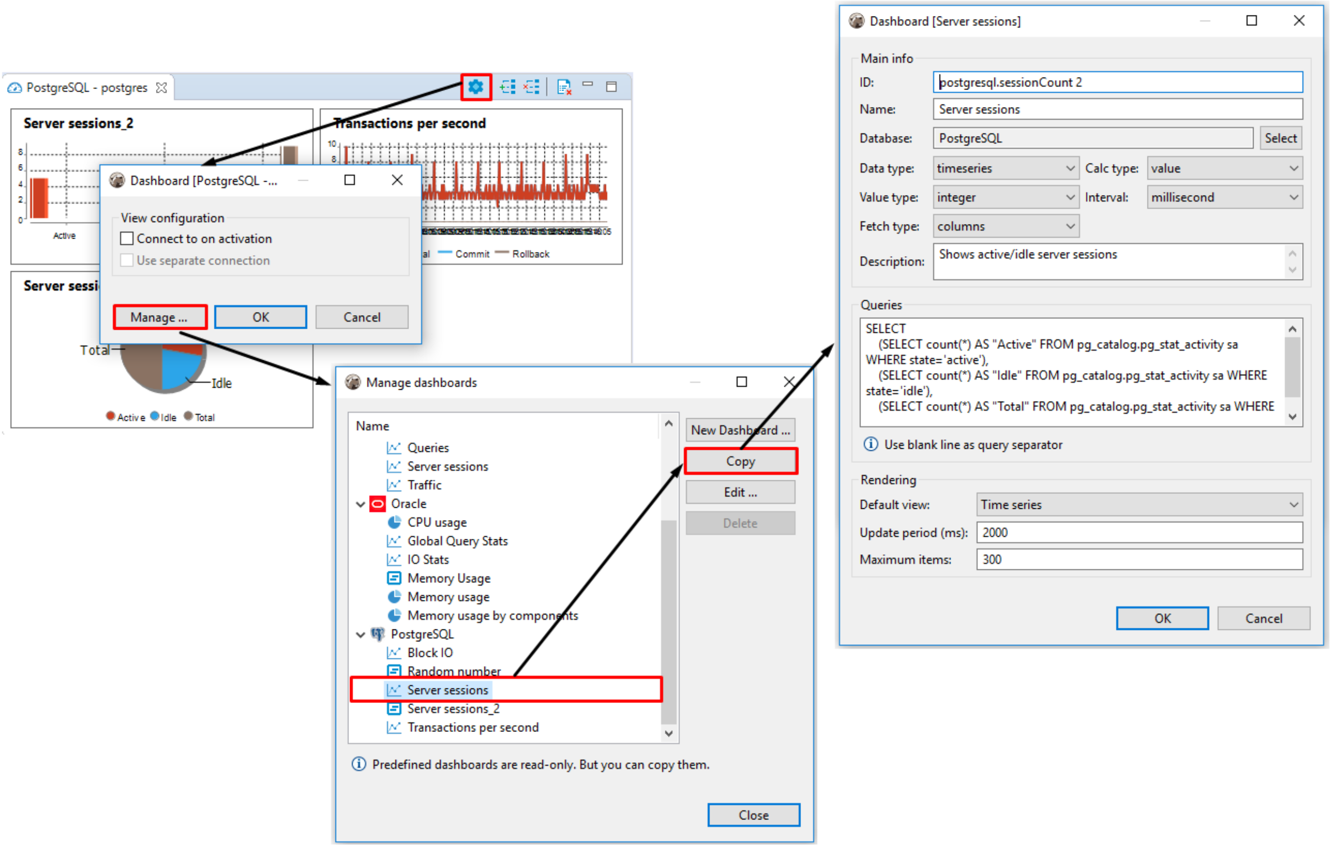
1. Press **Settings** button in the dashboards panel toolbar.
2. In the opened dialog box click **Manage...** button.
3. In the **Manage dashboards** window click **New dashboard...** button.
4. Set up all configurational parameters as required and press **OK**. To learn more about dashboard's configuration parameters, see Adjusting Dashboard Configuration.



Data type: timeseries
Calc type: value
Value type: decimal
Interval: millisecond
Fetch type: columns
Default view: Time series
Update period (ms): 1000
Maximum items: 300

To create a dashboard from template:

1. Press **Settings** button in the dashboards panel toolbar.
2. In the opened dialog box click **Manage...** button.
3. In the **Manage dashboards** window select any of the existing dashboards from the list and click **Copy**.
4. Adjust all configurational parameters as required and press **OK**. To learn more about dashboard's configuration parameters, see Adjusting Dashboard Configuration.



ID: postgresql.sessionCount 2
Name: Server sessions
Database: PostgreSQL
Data type: timeseries
Calc type: value
Value type: integer
Interval: millisecond
Fetch type: columns
Description: Shows active/idle server sessions

```
SELECT
   (SELECT count(*) AS "Active" FROM pg_catalog.pg_stat_activity sa
WHERE state='active'),
   (SELECT count(*) AS "Idle" FROM pg_catalog.pg_stat_activity sa WHERE
state='idle'),
   (SELECT count(*) AS "Total" FROM pg_catalog.pg_stat_activity sa WHERE
```

Default view: Time series
Update period (ms): 2000
Maximum items: 300

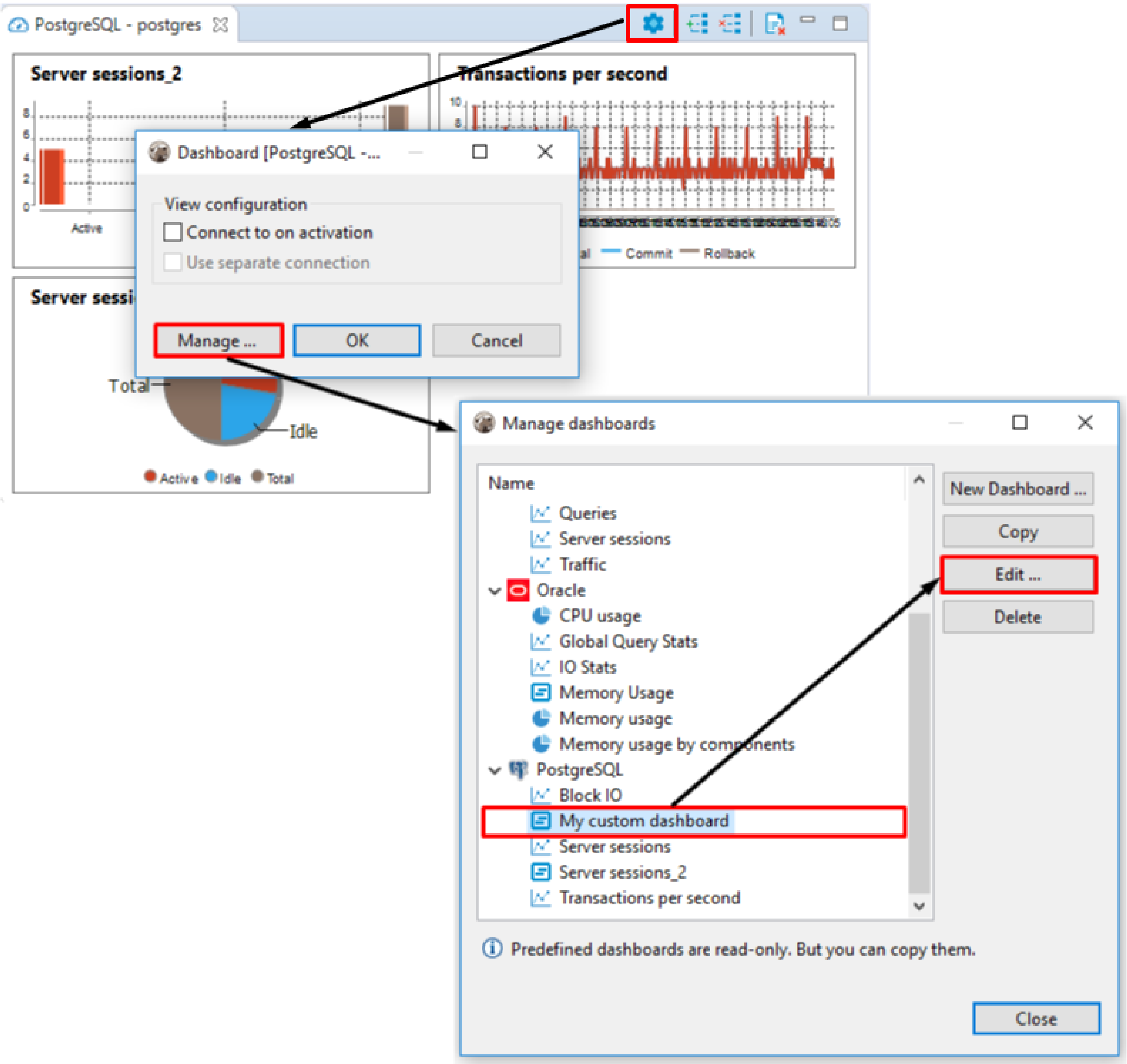
Editing Dashboards

If you need to change dashboard's name , ID or any other configurational 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 **Settings** button ⚙ in the dashboards panel toolbar.
2. In the opened dialog box click **Manage...** button.
3. In the **Manage dashboards** window select any of the custom dashboards from the list and click **Edit...**.
4. Adjust all configurational parameters as required and press **OK**. To learn more about dashboard's configuration parameters, see Adjusting Dashboard Configuration.



ⓘ Predefined dashboards are read-only. But you can copy them.

Deleting Dashboards

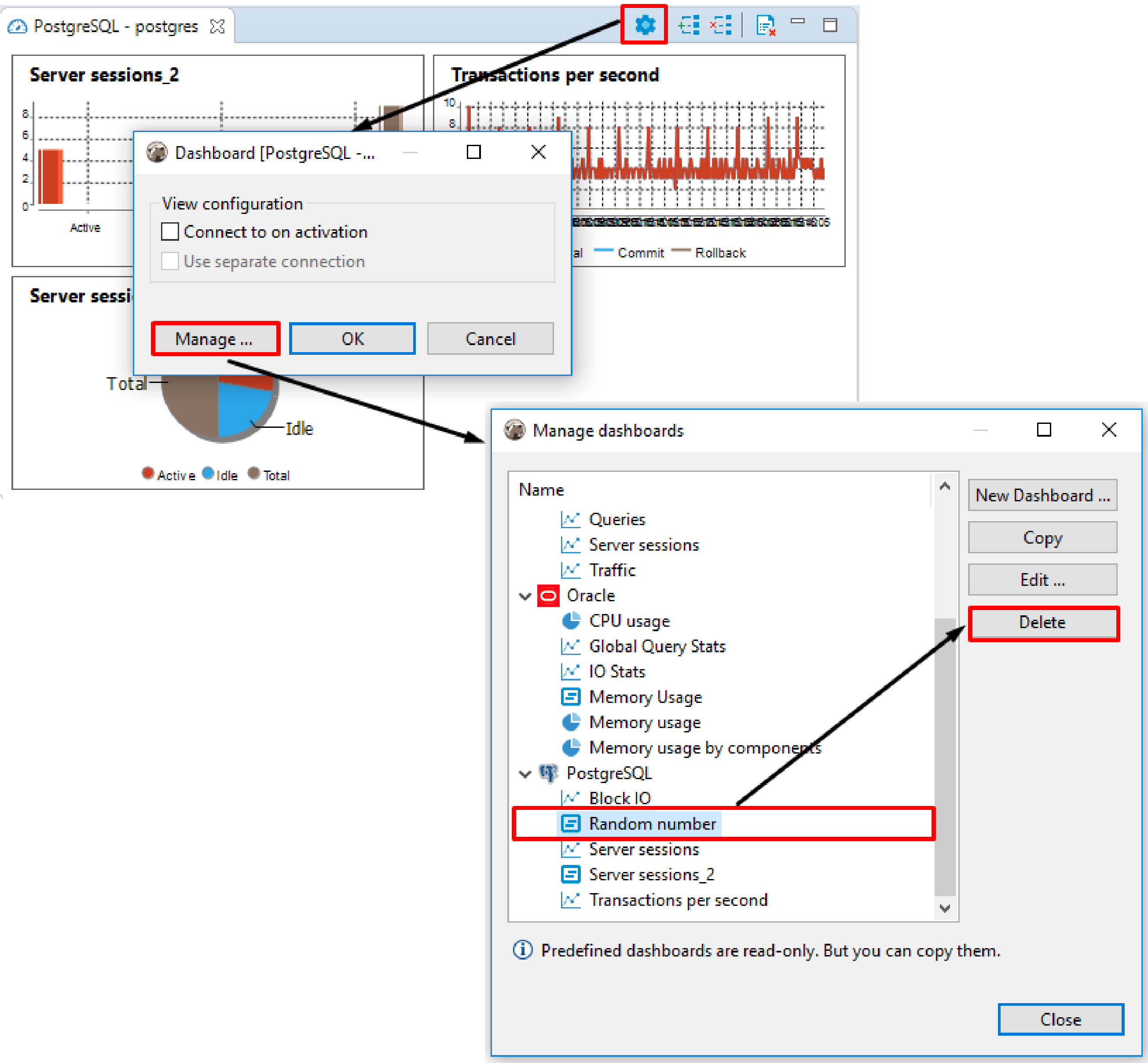
Note: Predefined dashboards cannot be deleted, but any of the custom dashboards can be deleted.

If you want to delete a dashboard, follow the steps described below.

To delete a dashboard:

1. Press **Settings** button ⚙ in 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**.



Dashboard [PostgreSQL -...

View configuration

□ Connect to on activation

□ Use separate connection

Manage dashboards

Name
- Queries
- Server sessions
- Traffic
- Oracle
  - CPU usage
  - Global Query Stats
  - IO Stats
  - Memory Usage
  - Memory usage
  - Memory usage by components
- PostgreSQL
  - Block IO
  - Random number
  - Server sessions
  - Server sessions_2
  - Transactions per second

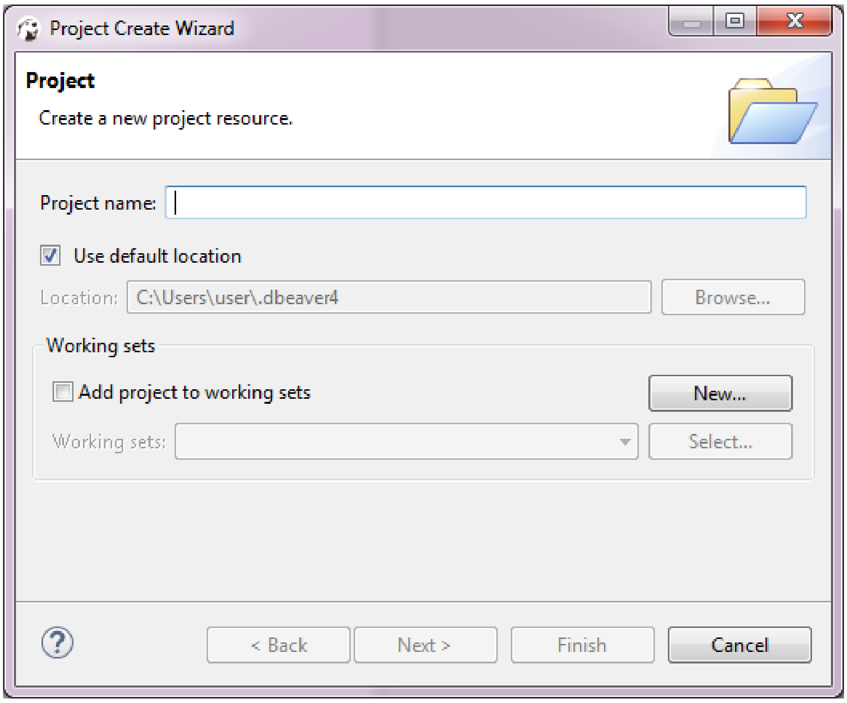
ⓘ Predefined dashboards are read-only. But you can copy them.

Projects

The Projects view allows creating 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.



Project Create Wizard

Project

Create a new project resource.

Project name:

☑ Use default location

Location: C:\Users\user\.dbeaver4

Working sets

□ Add project to working sets

Working sets:

1. In the Project screen, in the **Project name** field, specify the name of the project.

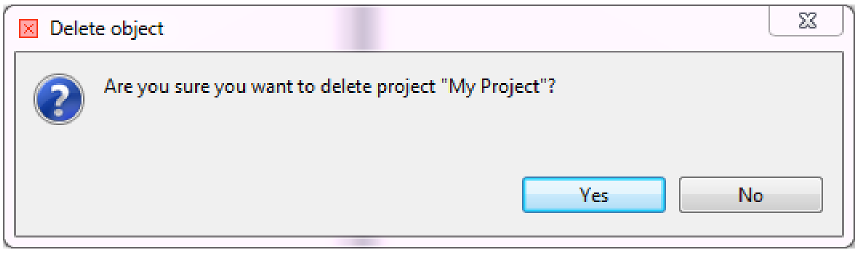
2. To keep the default location to store the project, leave the **Use default location** checkbox selected. If you want to change the location, clear the checkbox and enter the name of the new directory into the **Location** field or click **Browse** and select the directory in the folder tree.

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 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**.



Delete object

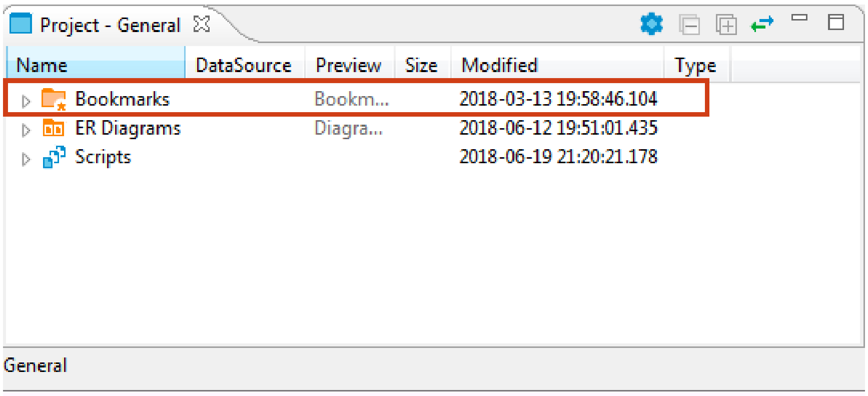
Are you sure you want to delete project "My Project"?

2. **Delete project** dialog box is to confirm the deletion of the project`s contents: these are the data stored in the file system, database connections are not affected. 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.

Bookmarks

Bookmarks are quick access links to objects of a database. They appear in the project tree inside the Projects or Project Explorer views.



Project - General

Name	DataSource	Preview	Size	Modified	Type
Bookmarks		Bookm...		2018-03-13 19:58:46.104	
ER Diagrams		Diagra...		2018-06-12 19:51:01.435	
Scripts				2018-06-19 21:20:21.178	

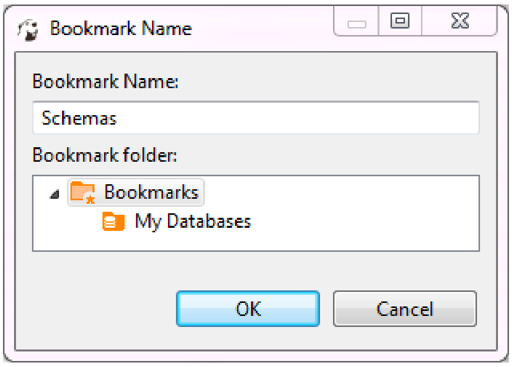
General

To create a bookmark:

1. In the Database Navigator or under **Connections** node of the Projects view, click the database object of interest to set focus on it.

2. Press `CTRL+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**:



Bookmark Name

Bookmark Name: Schemas

Bookmark folder:
- Bookmarks
  - My Databases

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.

Shortcuts

Brief list of the most important DBeaver shortcuts.
Of course you can redefine any (or almost any) of these shortcuts, here is the list of the default values.
Most of the following commands are accessible from DBeaver main menu, context menu or editor toolbar (or from all of them).
Use context menu wherever it is possible - it usually shows all actions accessible at this moment.

SQL Editor

Shortcut	Action
`CTRL+Enter`	Execute current query (*)
`CTRL+\`	Execute current query in a new tab
`ALT+X`	Execute current script (**)
`CTRL+ALT+'`	Execute selected SQL expression and print results
`CTRL+SHIFT+E`	Explain current query execution plan
`CTRL+ALT+SHIFT+X`	Execute queries of current script simultaneously, showing results in separate tabs
`CTRL+9`	Switch active connection (for SQL script)
`CTRL+Space` `Option+Space`	SQL completion proposals popup
`CTRL+ALT+Space`	SQL templates proposals popup
`CTRL+SHIFT+F`	Format current script (**) using current formatter
`CTRL+/` `CTRL+SHIFT+/`	Toggle single/multi line comment
`ALT+Up` `ALT+Down`	Jump to previous/next query
`CTRL+6` `CTRL+SHIFT+6` `ALT+6`	Toggle editor/results panels (maximize/minimize/switch)
`CTRL+SHIFT+X` `CTRL+SHIFT+Y`	Convert selected text into upper/lower case

Data viewer

Shortcut	Action
`TAB`	Switch to record/grid mode
`CTRL+~`	Switch presentation (grid, plain text, json ,etc)
`CTRL+1`	Foreign keys navigation menu
`ALT+Space`	Navigate to the link in active cell
`ALT+Left`	Navigate backward in history
`ALT+Right`	Navigate forward in history
`CTRL+2`	Toggle sorting by current column
`F11`	Current column filters menu
`CTRL+F11`	Current column filter dictionary panel
`F7` `CTRL+7`	Toggle right panels on/off
`F5`	Refresh results (re-run query)

Data editor

Shortcut	Action

Shortcut	Action
`Enter`	Activate inline editor
`SHIFT+Enter`	Open value editor dialog or separate value editor (for LOB values)
`Delete` `ALT+Delete`	Delete row
`ALT+Insert`	Add new row
`CTRL+ALT+Insert`	Copy current row
`Escape`	Cancel changes in current cell/row

Database Navigator

Shortcut	Action
`F2`	Rename current element (if supported)
`F4`	Open editor of selected element(s)
`F5`	Refresh selected element(s)
`Delete`	Delete selected element(s) (if supported)
`CTRL+ALT+SHIFT+D`	Add bookmark on selected element
`Alt+Enter`	Show properties of selected element
`F3` `CTRL+[`	Open SQL editor for current connection (***). Shows script selector popup.
`CTRL+F3` `CTRL+]`	Open new SQL editor for current connection (***). Always creates new script.
`CTRL+Enter`	Open recent SQL editor for current connection (***). Opens last modified script or creates a new script.

Other

Shortcut	Action
`ALT+~`	Shows database tools context menu
`CTRL+0`	Switch active schema/catalog (available if SQL/database editor is open)
`CTRL+SHIFT+C`	Advanced copy. Works in different contexts and performs "smart copy" operation (usually with parameters).
`CTRL+SHIFT+V`	Advanced paste. Same as "smart copy" but for "paste".

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 current connection is the same as in this editor. If active window is database navigator then active connection is "owner" connection of currently selected element. In other cases there is no current connection and DBeaver will ask you to choose connection explicitly.

Database Connections

To be able to manage your database in DBeaver, you need to create a connection to this database – see Creating Connections. A connection includes a driver and a number of configuration parameters including the location of the database and credentials to access it. You need to create a separate connection to every database you want to manage. Every database type requires its own set of connection parameters.

Connections reside in the Database Navigator and in the Projects views. In these views, you can:

- Edit connections, see Editing Connections
- Rename and delete connections - via corresponding context menu items, see Database Navigator
- Connect to and disconnect from databases using connections, see Connect to Database and Disconnect from Database.

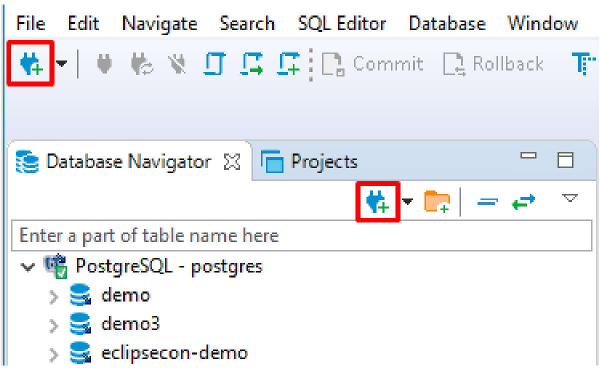
Database connections might have the following states:

- not connected
- has network settings specified (such as SSH tunnel, etc.)
- connected
- connection error

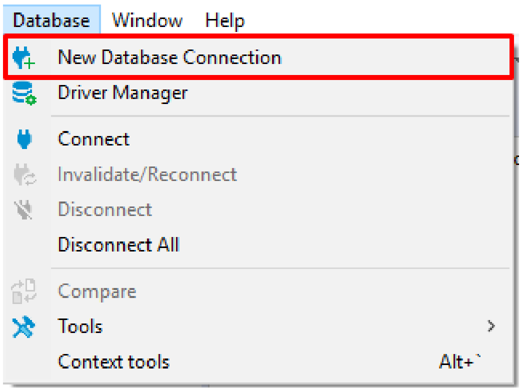
Create Connection

DBeaver provides a wizard that guides you through the steps to create a connection. If you run DBeaver for the first time (standalone version), the new connection wizard appears automatically. In other cases, to create a connection, do one of the following:

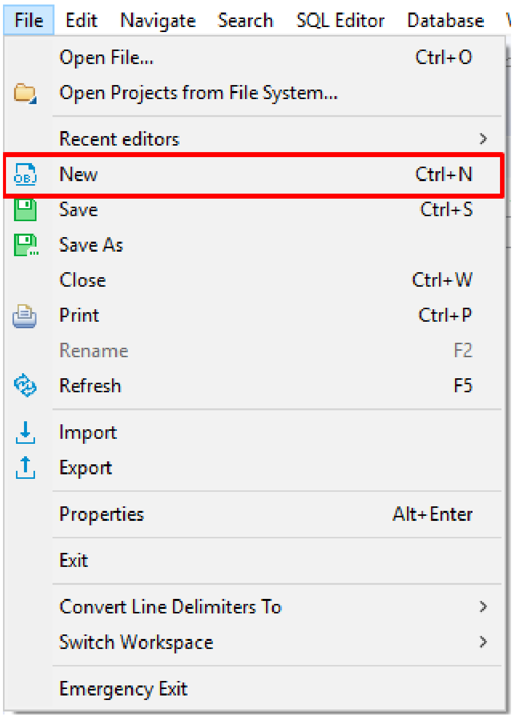
- Click the **New Connection Wizard** button in the application toolbar or in the Database Navigator view toolbar:



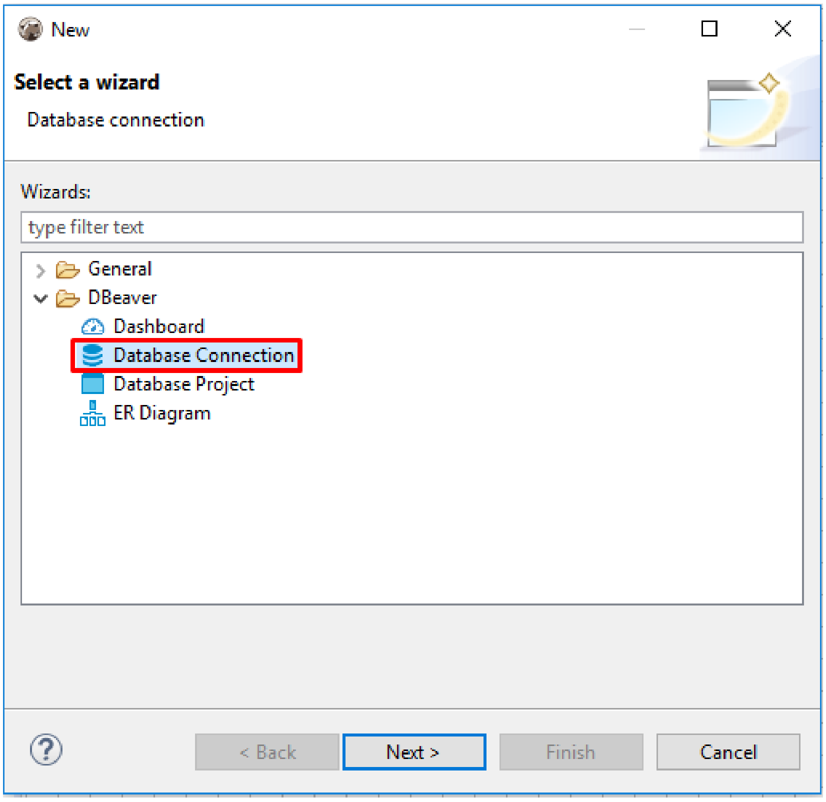
- Click **Database -> New Connection** in the menu bar:



- Press `Ctrl+N` or click **File -> New** in the menu bar:

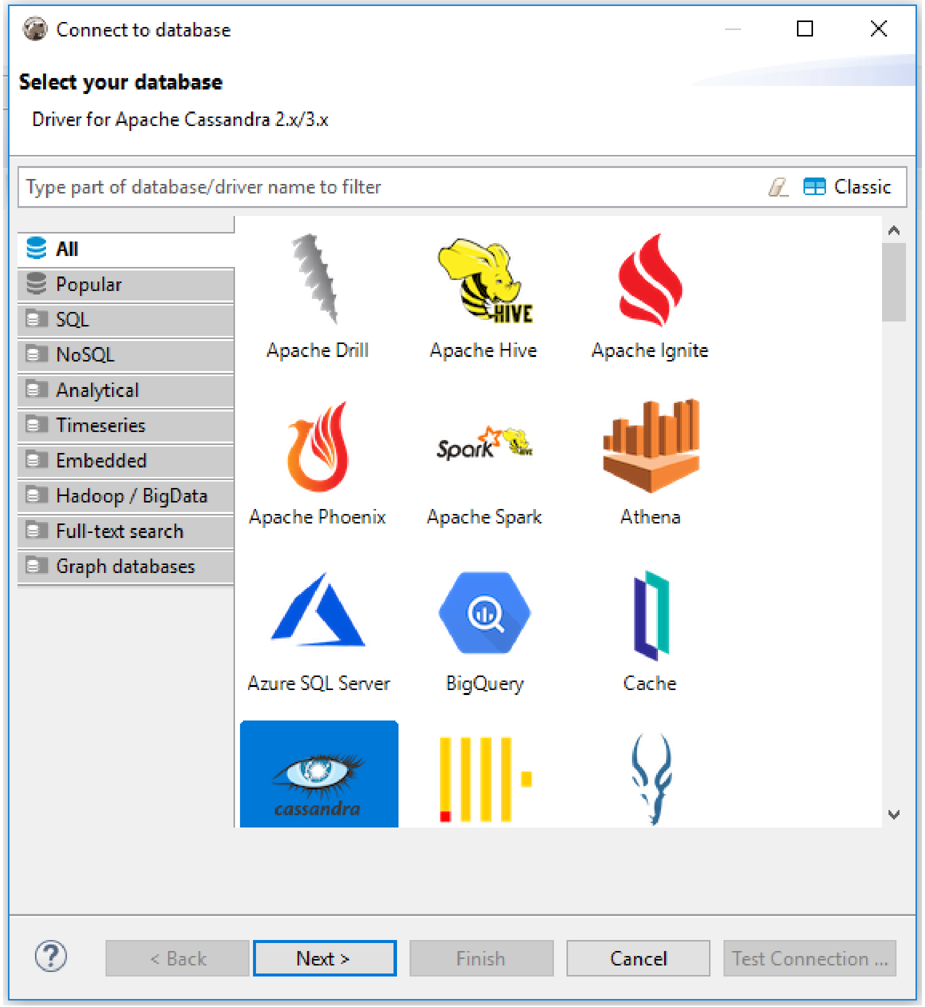


  Then, in the wizard, click **Database connection** and then click **Next**:



Then, in the **Create new connection** wizard:

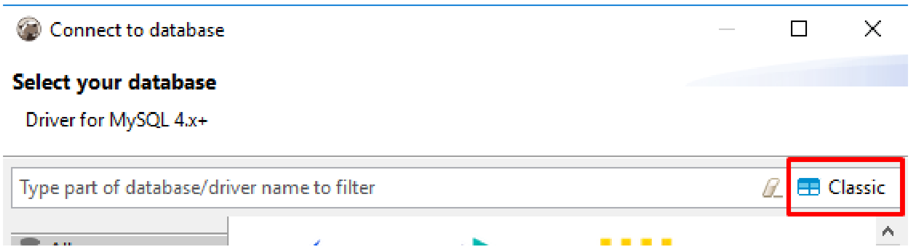
1. Choose a driver for the new connection: click the name of the suitable database type in the gallery. Then click **Next**.



To quickly find the needed driver, you can type a hint in the text field above the list of drivers.
If you cannot find a driver for your database then probably there is no suitable driver and you need to create one. Please see Database Drivers article.

NOTE: The list of database drivers diaplays the number of exising connections next to each driver. No number is displayed if there are no connections.

If you prefer the classic list view of the available drivers, use the **Classic** button.



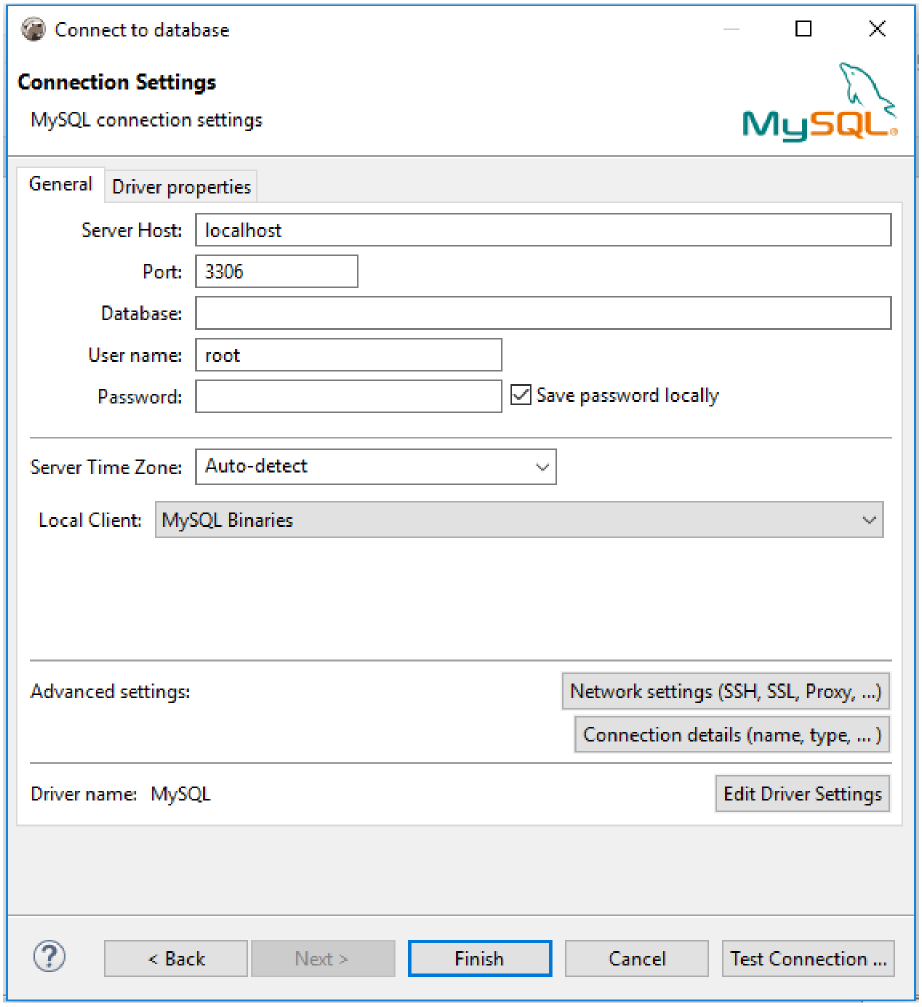
Connect to database

Select your database

Driver for MySQL 4.x+

Type part of database/driver name to filter

2. In the Connection Settings screen, on the General tab, set all primary connection settings:



Connect to database

Connection Settings

MySQL connection settings

General | Driver properties

Server Host: localhost

Port: 3306

Database:

User name: root

Password:

☑ Save password locally

Server Time Zone: Auto-detect

Local Client: MySQL Binaries

Advanced settings:

Driver name: MySQL

MySQL
Network settings (SSH, SSL, Proxy, ...)
Connection details (name, type, ...)
Edit Driver Settings
< Back
Next >
Finish
Cancel
Test Connection ...

For most drivers required settings include:

- Host
- Port
- Database name
- User name and password

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.

3. If necessary, specify advanced settings, see **Advanced Settings** section below, and click **Next**.

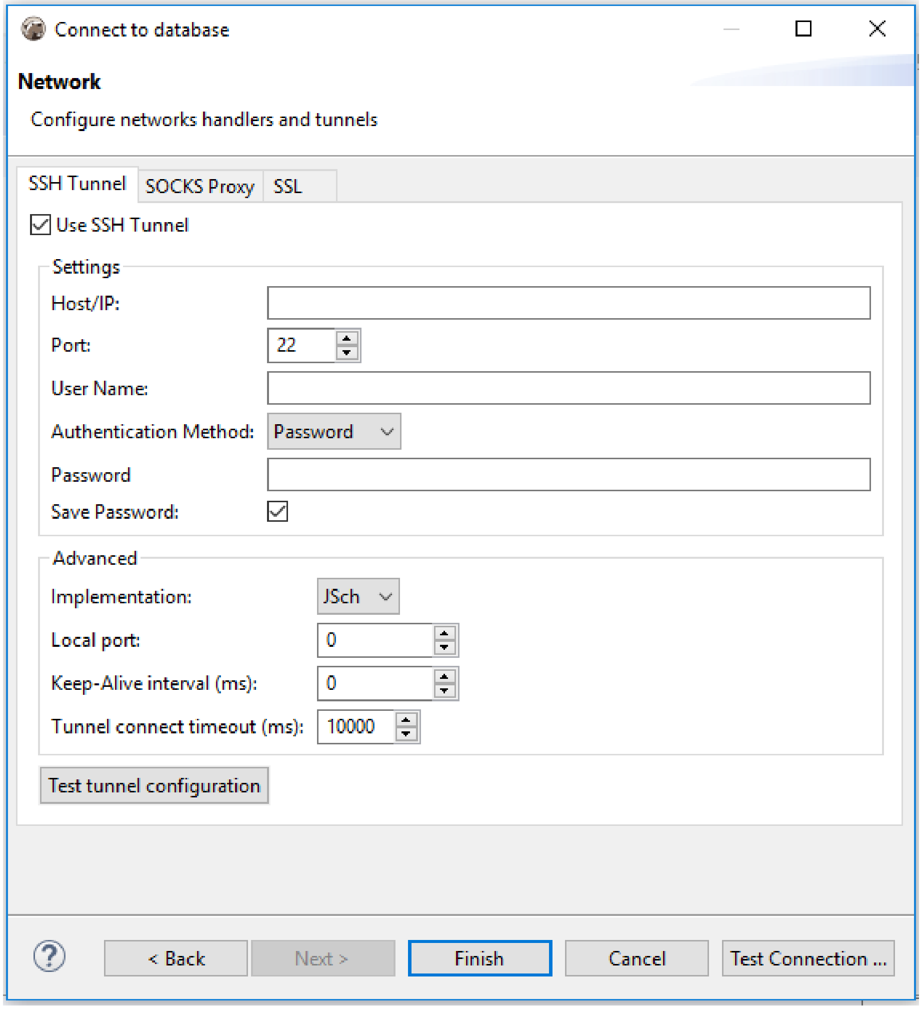
4. To test if the connection works, click **Test Connection**.

5. Click **Finish**. The connection appears in the tree of connections in the Database Navigator and DBeaver actually connects to the database.

Advanced Settings

Network Settings (SSH, SOCKS, SSL)

If your database cannot be accessed directly, you can use SSH tunnel:



Connect to database

Network

Configure networks handlers and tunnels

☑ Use SSH Tunnel

Settings

Host/IP:	
Port:	22
User Name:	
Authentication Method:	Password
Password	
Save Password:	☑

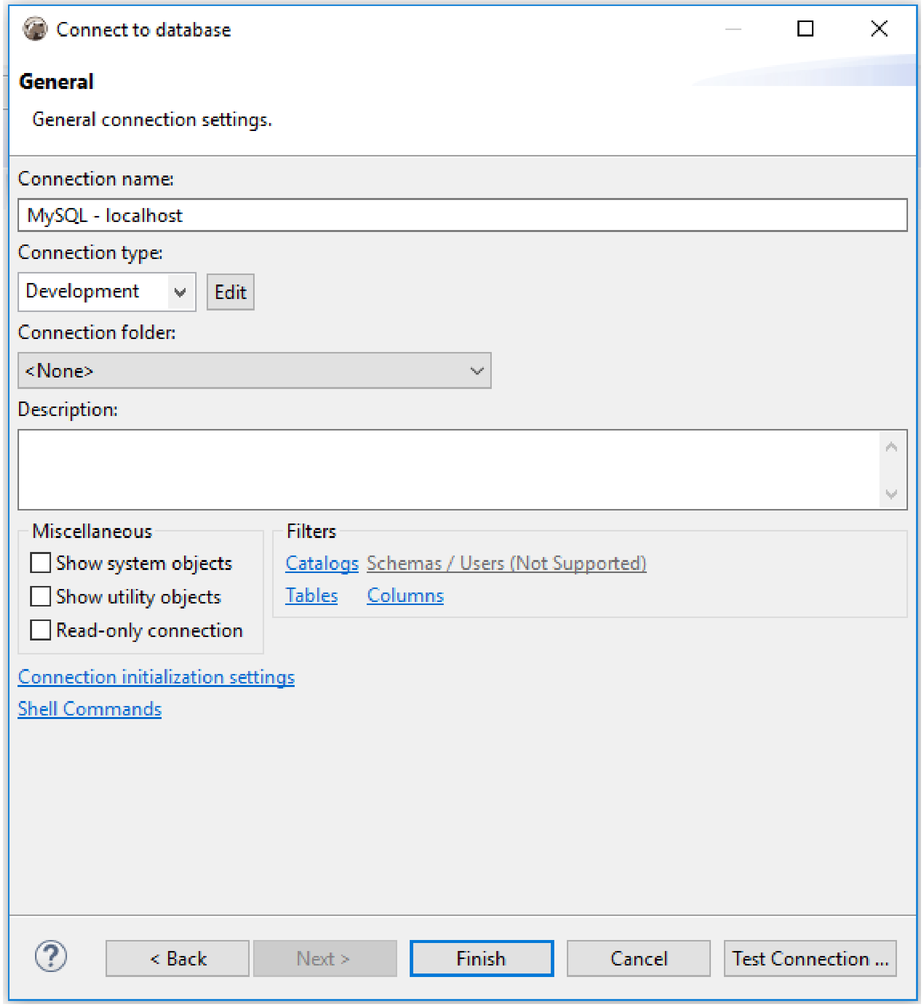
Advanced

Implementation:	JSch
Local port:	0
Keep-Alive interval (ms):	0
Tunnel connect timeout (ms):	10000

If a connection has network settings specified, such a connection appears in the application with a special 'arrow' icon such as this:

Connection Details (name, type, etc.)

You can also set the connection name, type and initial settings (such as bootstrap queries, transaction state, global filters, etc.).



Connect to database

General

General connection settings.

Connection name: MySQL - localhost

Connection type: Development

Connection folder: <None>

Description:

Miscellaneous

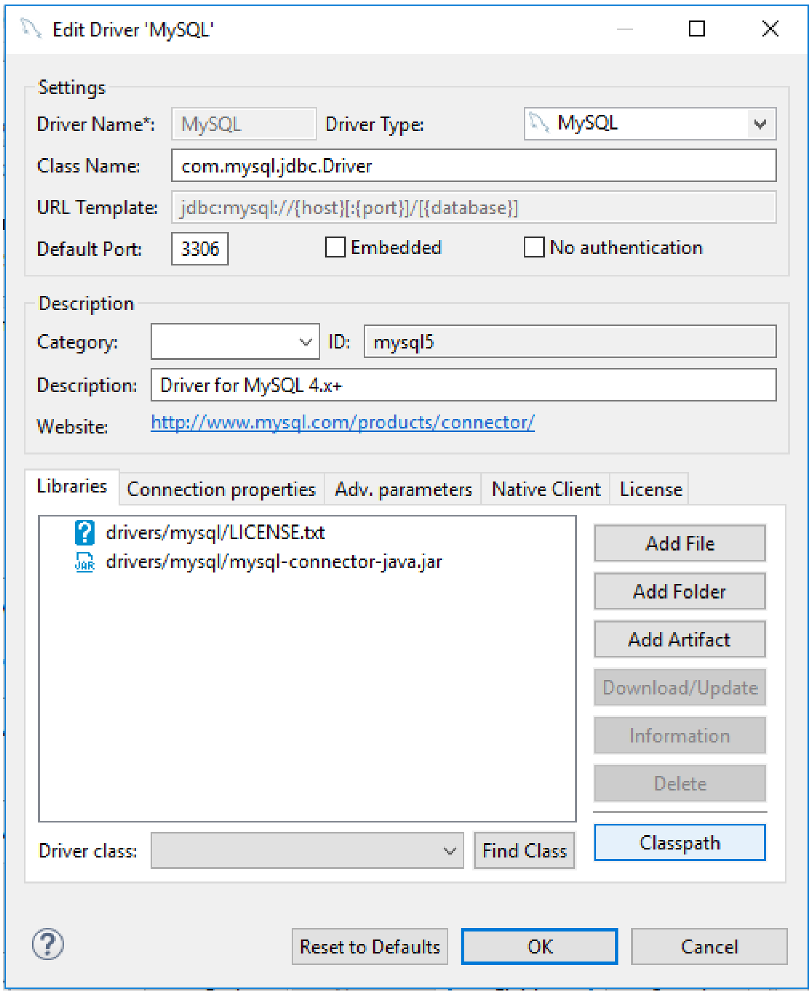
□ Show system objects

□ Show utility objects

□ Read-only connection

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.



Edit Driver 'MySQL'

Settings

Driver Name*: MySQL

Driver Type: MySQL

Class Name: com.mysql.jdbc.Driver

URL Template: jdbc:mysql://{host}[:{port}]/[{database}]

Default Port: 3306

□ Embedded □ No authentication

Description

Category:

ID: mysql5

Description: Driver for MySQL 4.x+

Website: http://www.mysql.com/products/connector/

drivers/mysql/LICENSE.txt

drivers/mysql/mysql-connector-java.jar

Driver class:

Variables in parameters

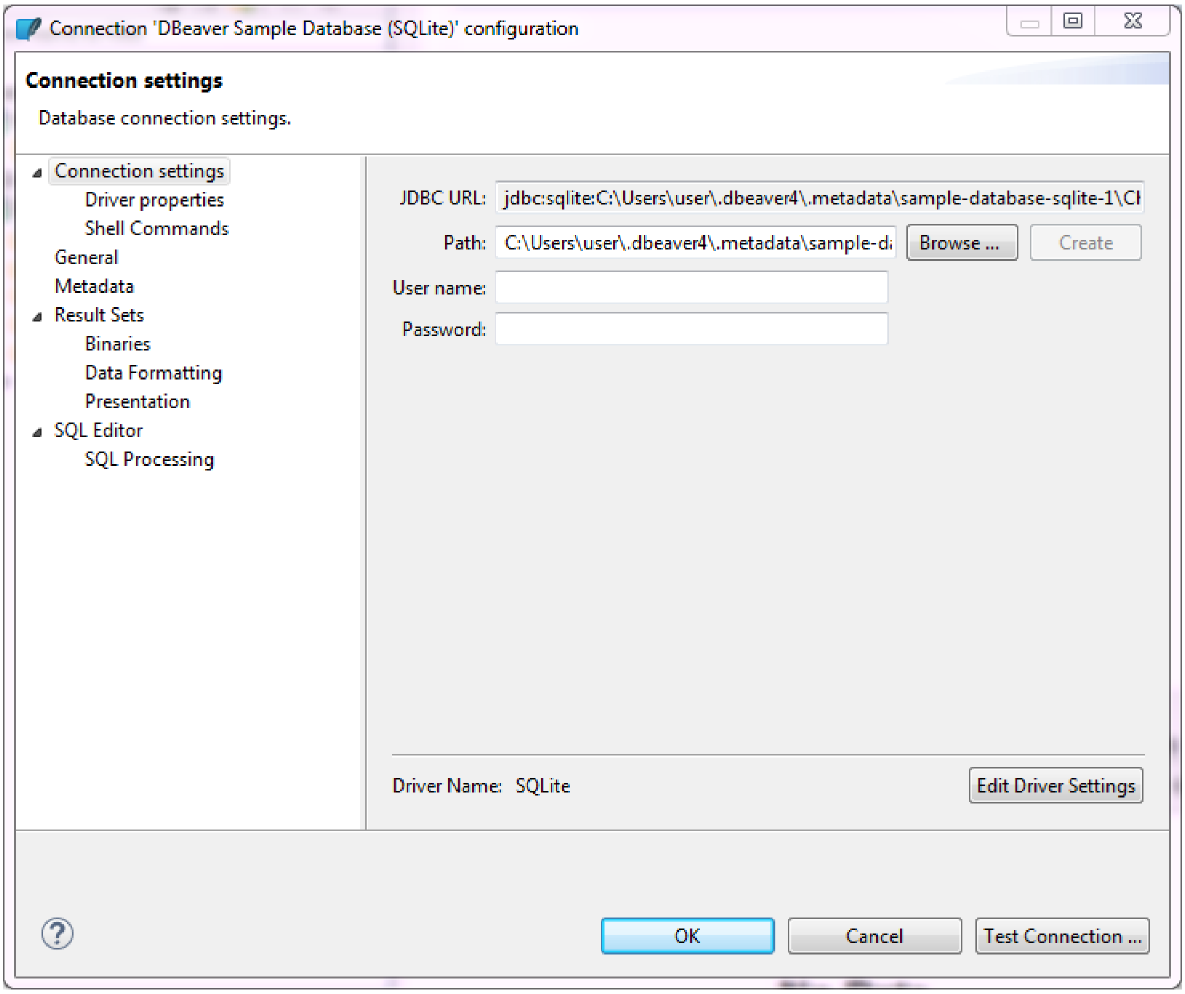
You can use variables in all connection parameters and in the driver properties. Variables are system environment variables or one of the following list:

Name	Value
${host}	Host name
${port}	Port number
${database}	Database name
${server}	Server name
${url}	Connection URL
${user}	User name
${password}	User password

Note: option `Use environment variables in connection parameters` must be turned on (see preferences).

Edit Connection

To edit 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 opens:



Connection 'DBeaver Sample Database (SQLite)' configuration

Connection settings

Database connection settings.

JDBC URL: jdbc:sqlite:C:\Users\user\.dbeaver4\.metadata\sample-database-sqlite-1\Cł

Path: C:\Users\user\.dbeaver4\.metadata\sample-d

User name:

Password:

Driver Name: SQLite

The navigation pane on the left displays 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

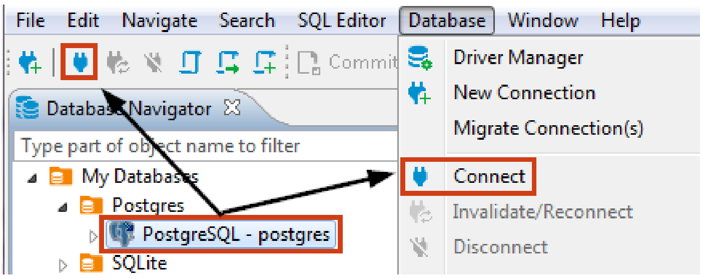
- In development

JDBC Time Zones

Connect to Database

To be able to work with the content and structure of a database, you need to connect to it. When you create a new connection to a database, DBeaver automatically connects to the new database, see Create Connection.

To connect to a database using an existing connection, in the Database Navigator or Projects view, click the connection and then click the Connect button in the toolbar or click **Database -> Connect** on the main menu:



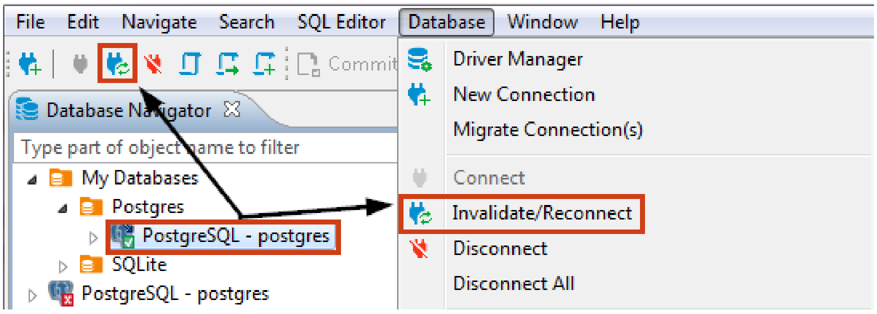
You can also right-click the connection and click **Connect** on the context menu.

If a database connection exists but DBeaver is not connected to the database, the connection appears with its original icon (for example, for PostgreSQL database). When DBeaver connects to the database, the icon changes to signal the connected status: .

If DBeaver cannot connect to a database, the connection appears with an error sign: . If you attempt to connect to such a database, DBeaver displays an error message describing the cause for the error.

Invalidate/Reconnect to Database

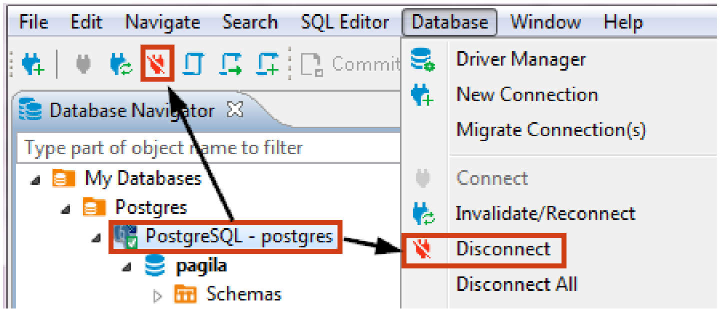
You might need to invalidate a database connection and then to reconnect to it again in such cases as connection to the server being lost, etc. To invalidate a database connection and then reconnect to the database, in the Database Navigator or Projects view, click the database connection and then click 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.

Disconnect from Database

You might need to disconnect from a database to free up resources or close transactions. To disconnect from a database, in the Database Navigator or Projects view, click the connection 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.

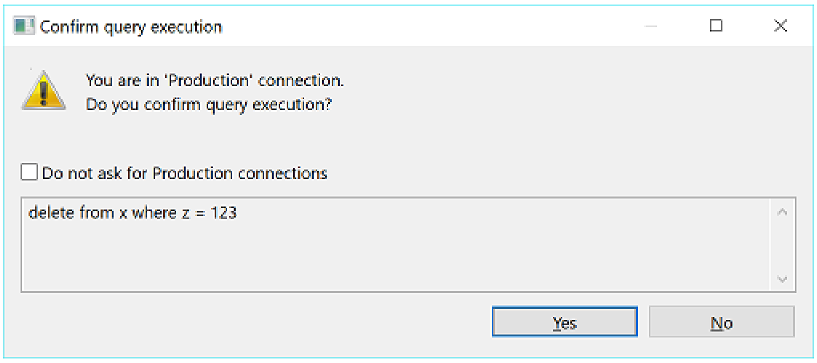
Connection Types

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 require 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.):



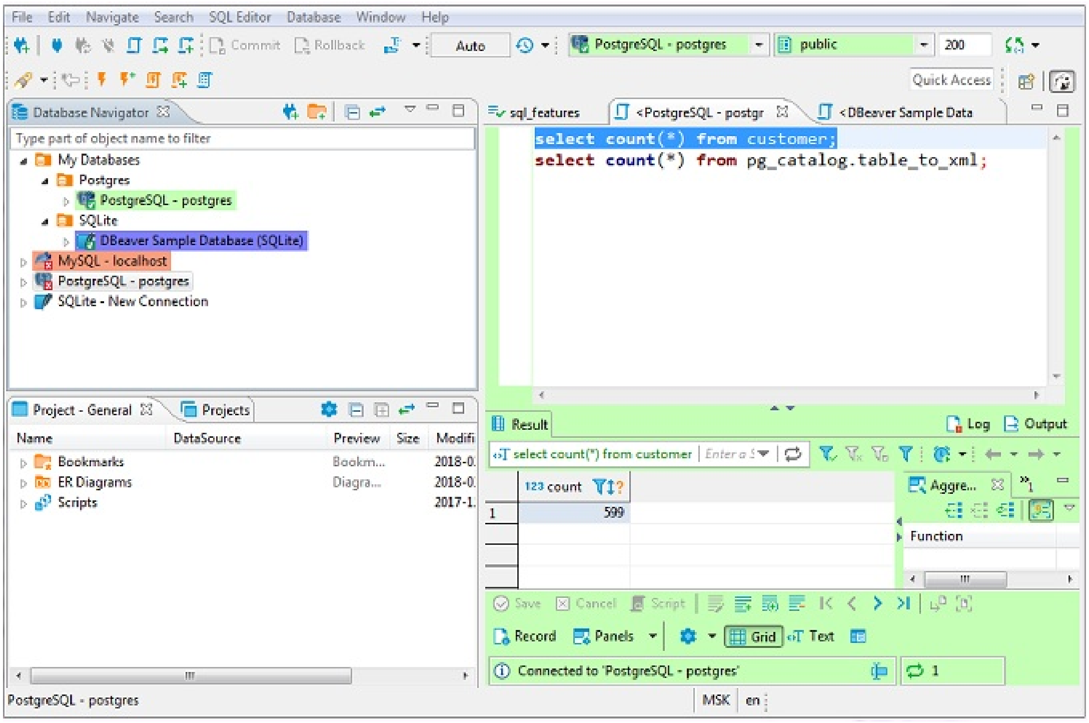
Confirm query execution

You are in 'Production' connection.
Do you confirm query execution?

□ Do not ask for Production connections

```
delete from x where z = 123
```

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. Colored are database connections that use a certain connection type in the Database Navigator and Projects views as well as editors related to these connections:

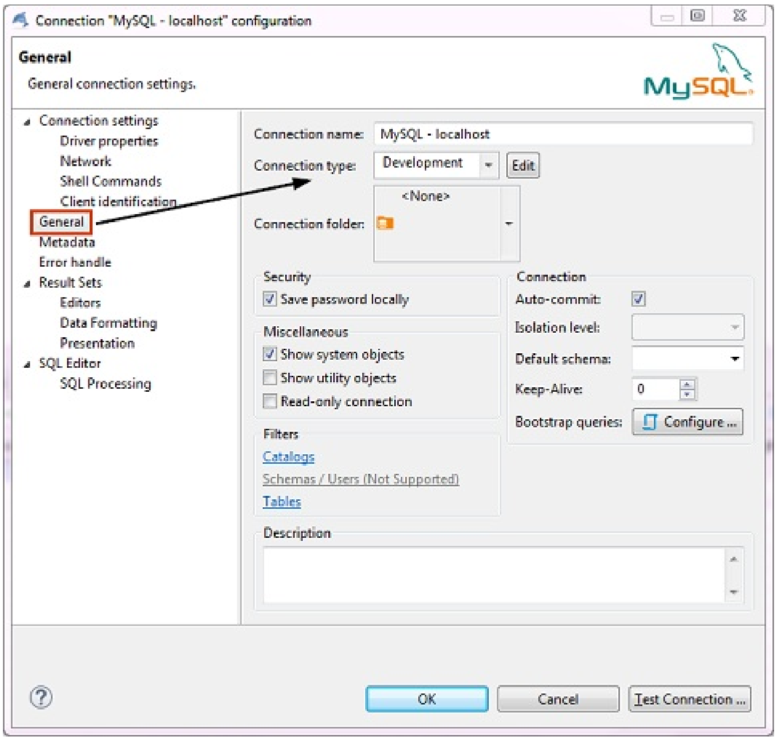


```
select count(*) from customer;
select count(*) from pg_catalog.table_to_xml;
```

count
599

Connected to 'PostgreSQL - postgres'

To manage connection types for a database connection, in the Database Navigator or Projects view, click the connection to set focus to 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:



Connection name: MySQL - localhost

Connection type: Development Edit

Connection folder: <None>

☑ Save password locally

☑ Show system objects

□ Show utility objects

□ Read-only connection

Auto-commit: ☑

Isolation level:

Default schema:

Keep-Alive: 0

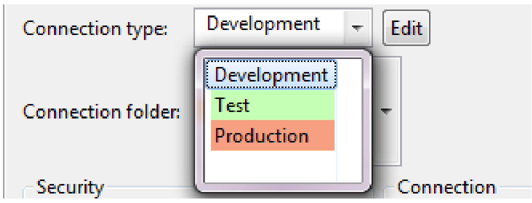
There are three default connection types – **Development**, **Test**, and **Production**. You can change the connection type for your database connection as well as you can create a new connection type, edit or delete an existing one.

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:

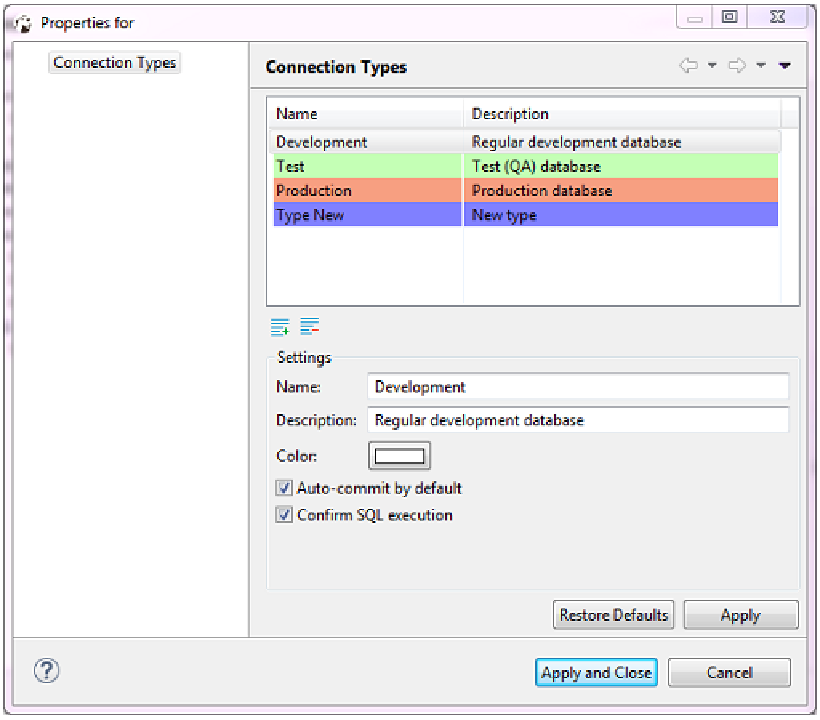


2. To test the connection, click **Test Connection**. To confirm the change, click **OK**.

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:



Name	Description
Development	Regular development database
Test	Test (QA) database
Production	Production database
Type New	New type

Settings

Name: Development

Description: Regular development database

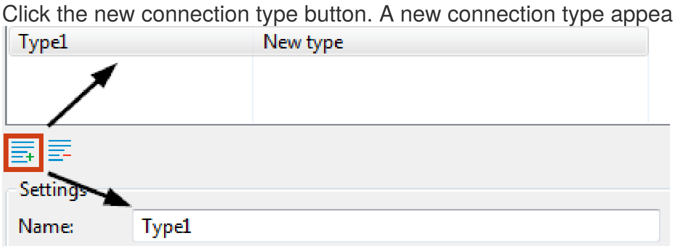
Color:

☑ Auto-commit by default

☑ Confirm SQL execution

The window displays existing connection types and their settings.

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



Type1	New type

Settings

Name: Type1

3. Now you can specify the settings for the new connection type:
   - Enter the connection type`s name into the **Name** field.
   - Enter a description into the **Description** field, if needed.
   - Click the **Color** box and select the color for the new connection type.
   - To set DBeaver to automatically commit changes to the database when connections use this connection type, select the **Auto-commit by default** checkbox. Otherwise, leave it empty.
   - To set DBeaver to ask for your confirmation at each execution of SQL statement of ‘transaction’ type, select the **Confirm SQL execution** checkbox. Otherwise, leave it empty.
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. To discard all changes and return to the previous state, click **Restore Defaults**.

Edit Connection Type

To edit 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.
2. Specify the settings for the new connection type the same way as when you create a connection type, see 'Create Connection Types' section above.
3. 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 focus to it and then click the delete button under the list:

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.

Auto and Manual Commit Modes

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: Auto.

Clicking the statistics field opens the Transaction Log.

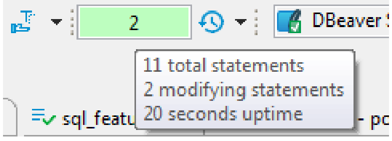
Manual Commit Mode

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** (Commit) and **Rollback** (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: 2.

If you hover you mouse over the field, you can see statistics of your SQL statements:



11 total statements
2 modifying statements
20 seconds uptime

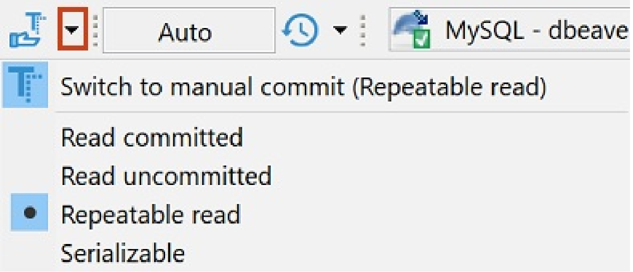
To commit statements to the database, click the **Commit** button in the toolbar. To discard them, click **Rollback**.

If no modifying statements have been made, the statistics field shows **None**: None

Clicking the statistics field opens the Transaction Log.

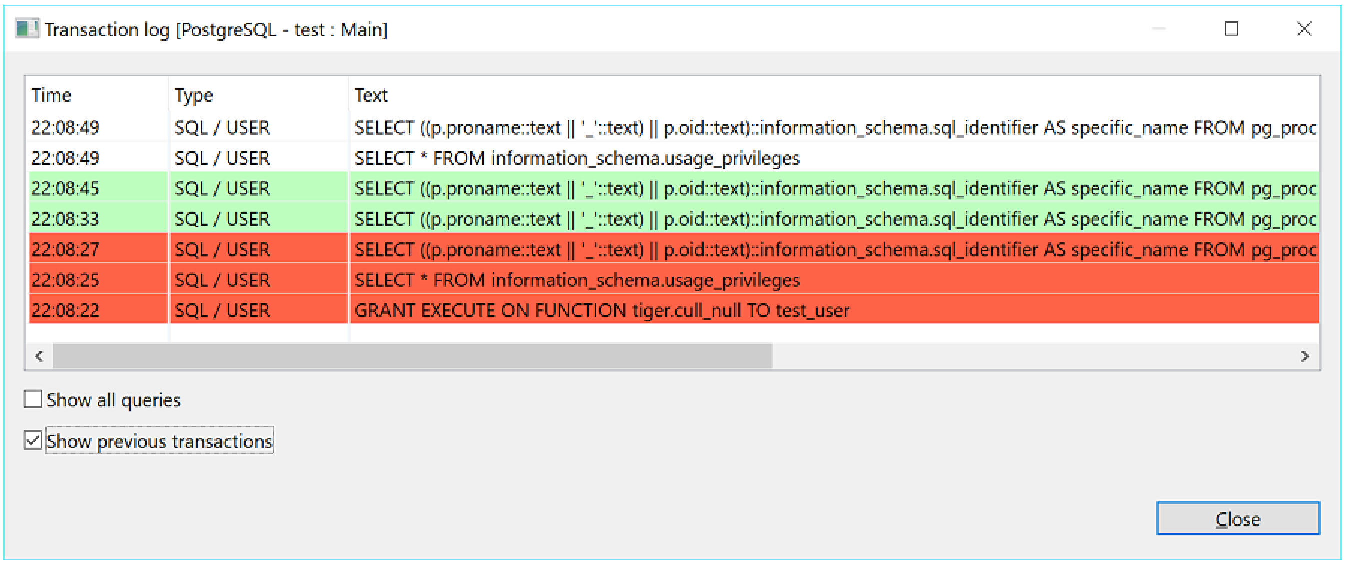
Transaction Isolation Level

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:



Transaction Log

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.



Transaction log [PostgreSQL - test : Main]

Time	Type	Text
22:08:49	SQL / USER	SELECT ((p.proname::text || '_'::text) || p.oid::text)::information_schema.sql_identifier AS specific_name FROM pg_proc
22:08:49	SQL / USER	SELECT * FROM information_schema.usage_privileges
22:08:45	SQL / USER	SELECT ((p.proname::text || '_'::text) || p.oid::text)::information_schema.sql_identifier AS specific_name FROM pg_proc
22:08:33	SQL / USER	SELECT ((p.proname::text || '_'::text) || p.oid::text)::information_schema.sql_identifier AS specific_name FROM pg_proc
22:08:27	SQL / USER	SELECT ((p.proname::text || '_'::text) || p.oid::text)::information_schema.sql_identifier AS specific_name FROM pg_proc
22:08:25	SQL / USER	SELECT * FROM information_schema.usage_privileges
22:08:22	SQL / USER	GRANT EXECUTE ON FUNCTION tiger.cull_null TO test_user

☐ Show all queries

☑ Show previous transactions

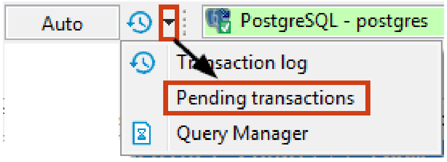
The Transaction Log window shows transactions that are:

- In progress or pending - shown without any special color
- Successfully committed – in green: 🟩
- Rolled back – in orange or red: 🟥

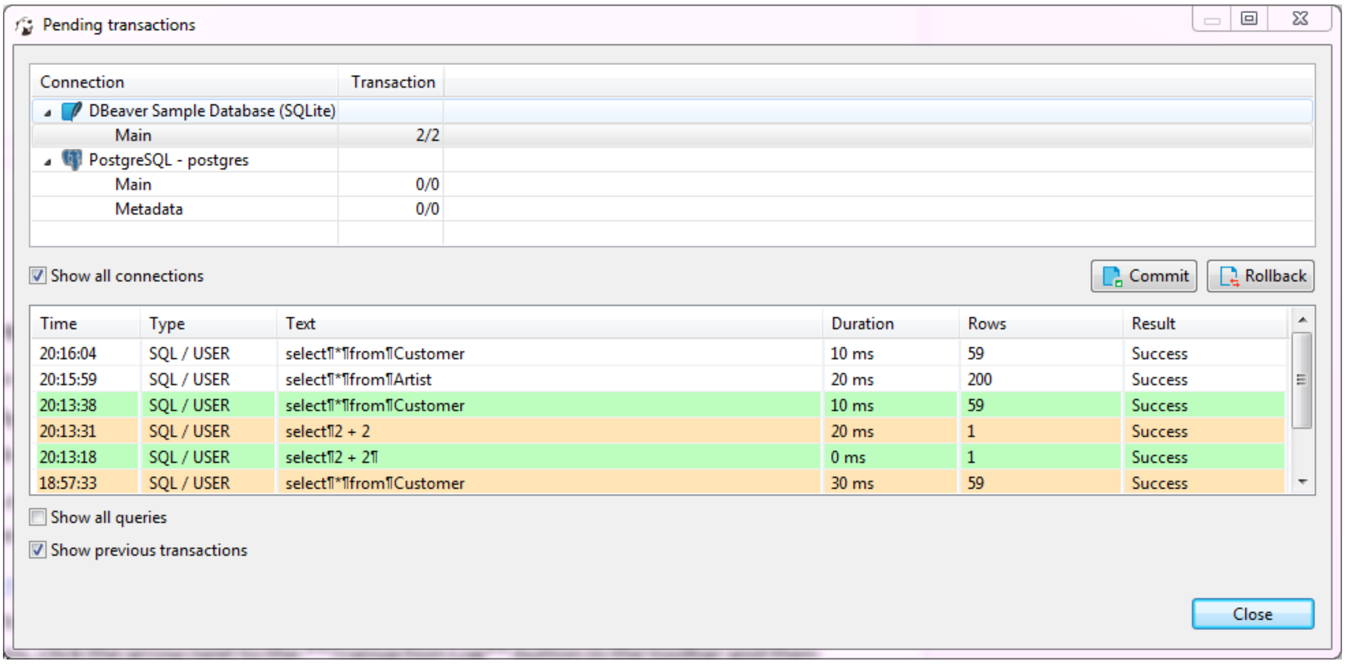
To see all previous transactions during the current session, select the **Show previous transactions** checkbox. To see all queries including non-transactional ones, select the **Show all queries** checkbox.

Pending transactions

It might be useful to check pending transactions because they might lock your database. To see 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:



Connection	Transaction
DBeaver Sample Database (SQLite)	
Main	2/2
PostgreSQL - postgres	
Main	0/0
Metadata	0/0

☑ Show all connections

Time	Type	Text	Duration	Rows	Result
20:16:04	SQL / USER	select¶*¶from¶Customer	10 ms	59	Success
20:15:59	SQL / USER	select¶*¶from¶Artist	20 ms	200	Success
20:13:38	SQL / USER	select¶*¶from¶Customer	10 ms	59	Success
20:13:31	SQL / USER	select¶2 + 2	20 ms	1	Success
20:13:18	SQL / USER	select¶2 + 2¶	0 ms	1	Success
18:57:33	SQL / USER	select¶*¶from¶Customer	30 ms	59	Success

☐ Show all queries

☑ Show previous transactions

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 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, orange (or red) ones are rolled back, rows without a special color are non-transactional or pending transactions.

Database drivers

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

DBeaver has a lot of pre-configured driver 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 JDBC driver of your database. The rest is easy.

Obtaining JDBC driver

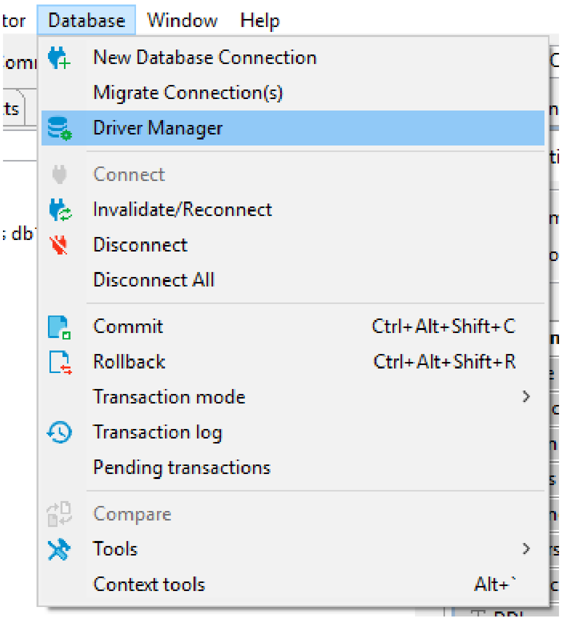
JDBC driver is a program (in Java) which can connect and operate with some local or remote database server. It usually provides all needed functionality to cover 100% of database functionality. Usually, JDBC driver are provided by database vendors to let customers ability to work with their databases.

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

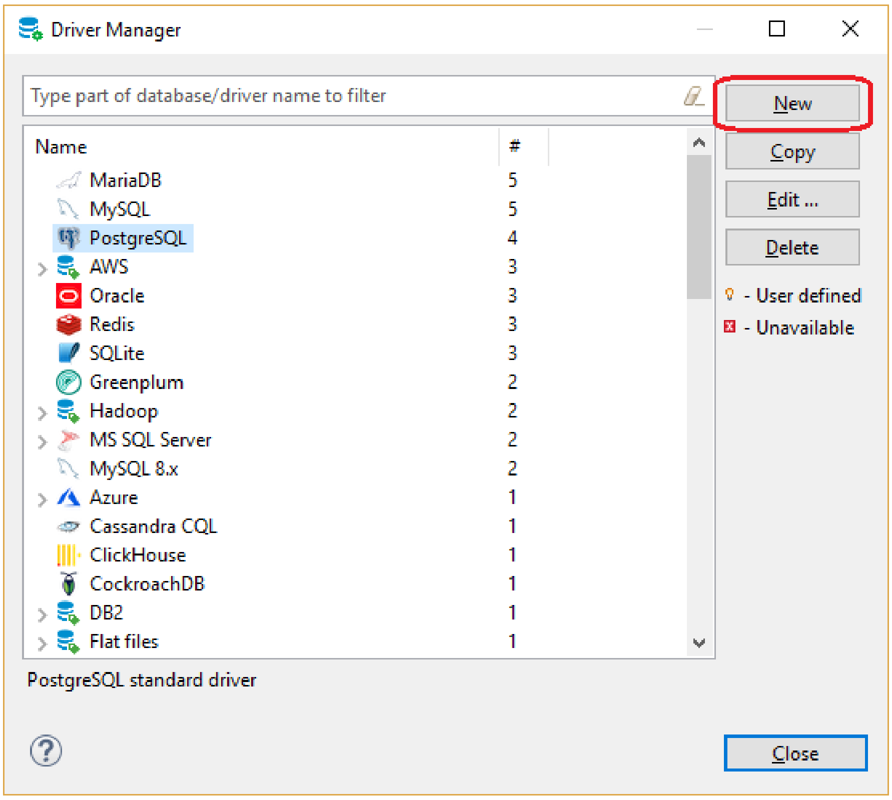
Adding driver configuration in DBeaver

Open driver manager dialog

You can open driver manager from main menu:



or from Database Navigator drop-down menu.



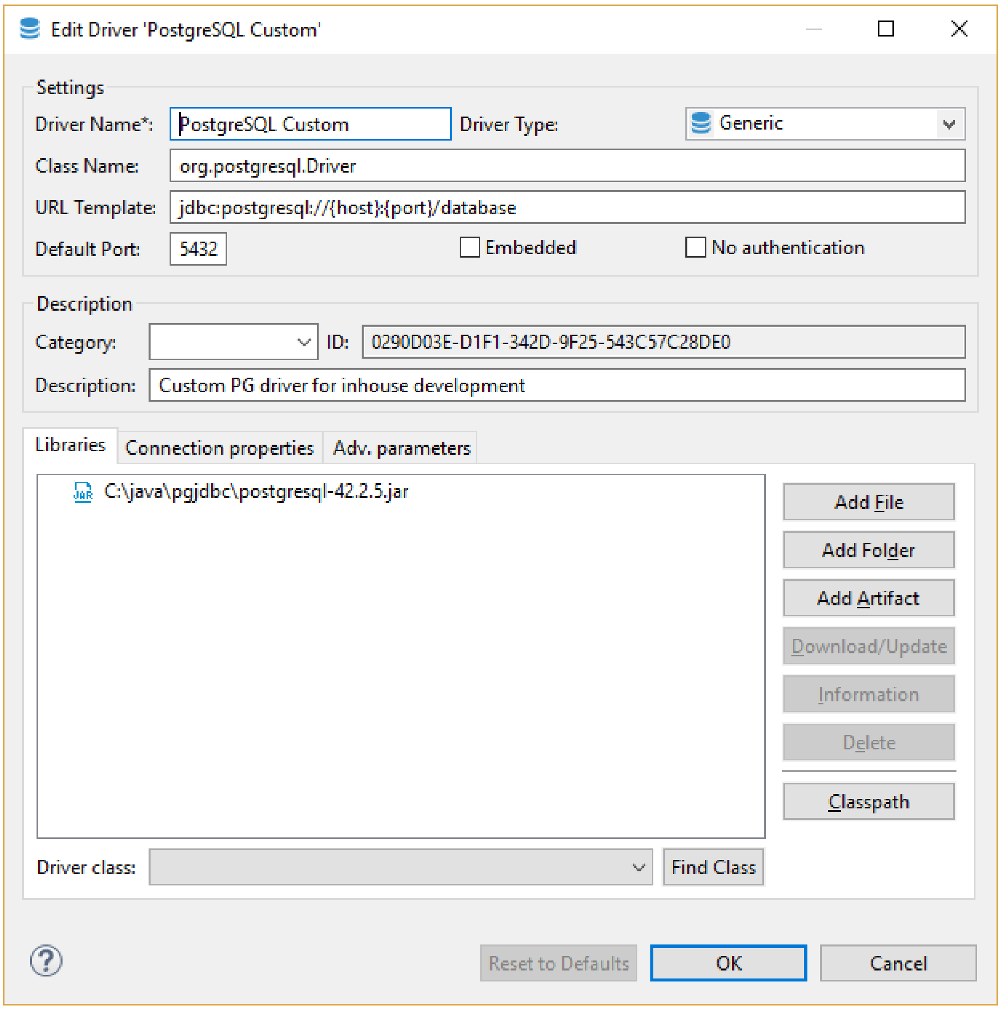
Name	#
MariaDB	5
MySQL	5
PostgreSQL	4
AWS	3
Oracle	3
Redis	3
SQLite	3
Greenplum	2
Hadoop	2
MS SQL Server	2
MySQL 8.x	2
Azure	1
Cassandra CQL	1
ClickHouse	1
CockroachDB	1
DB2	1
Flat files	1

PostgreSQL standard driver

Add 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



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

Category:

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

Description: Custom PG driver for inhouse development

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

Driver class:

Parameter	Description

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 Generic driver (JDBC provider)
Class Name	JDBC driver class name. You can get it from the documentation or find it in jar files (see "Find Class" 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, this will greatly simplify connections creation. See "URL Templates" for the detailed description
Default Port	Default database port. You can get it from documentation or leave it empty
Embedded	Enable it for server-less databases. This flag affects a few config options related to network/connections management
No Authentication	This means that driver doesn't 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 on 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 driver. In most cases you need only jar files.
Click "Add File" to add single jar file, "Add Folder" to add folder with Java classes/resources and "Add Artifact" to add Maven artifact (see below).

After you add jar files you will be able to find all JDBC driver classes which present 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 then you need to refer to the driver's documentation.

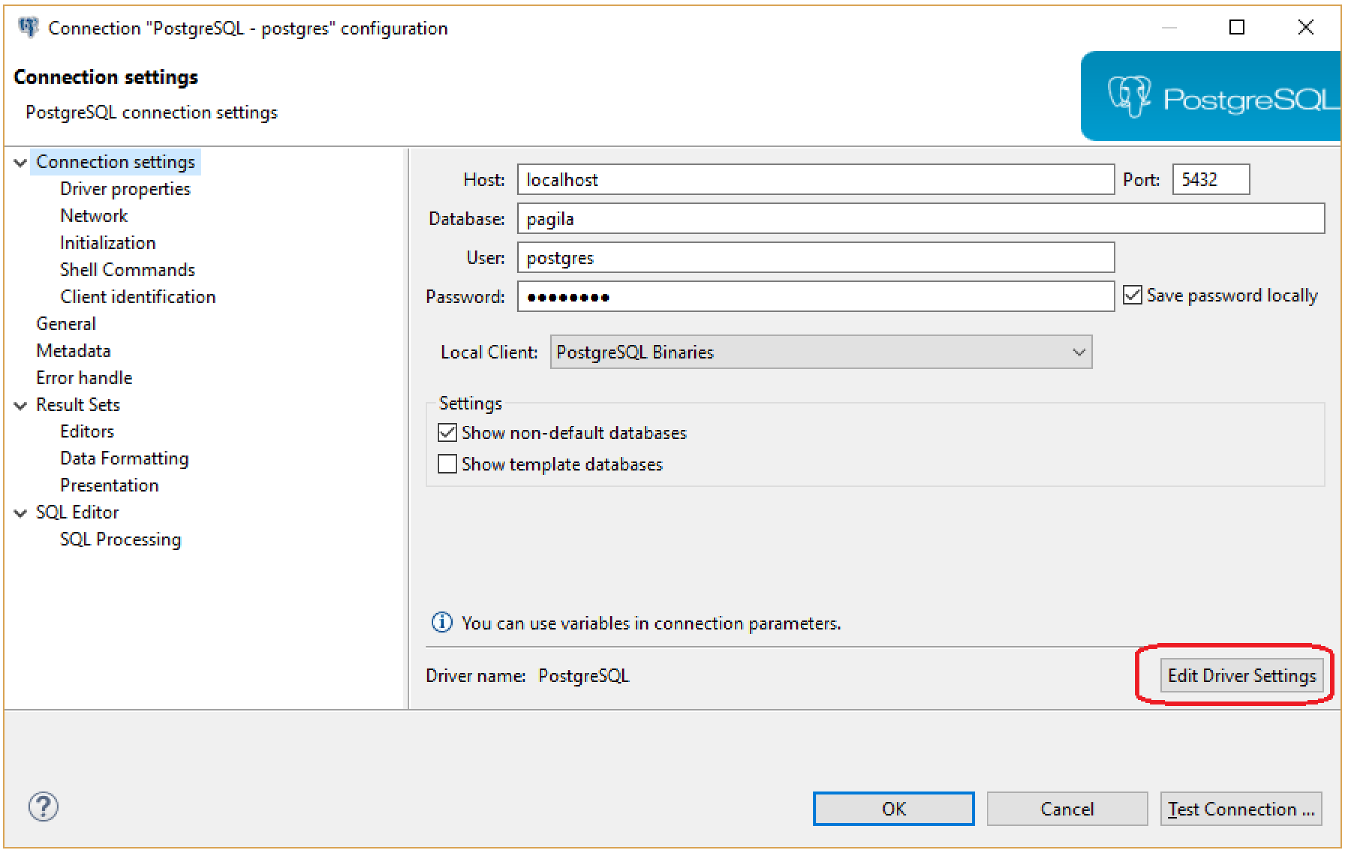
Maven artifacts

DBeaver can download driver jars directly from Maven repository (it is a global public repository of Java libraries, usually 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 driver version in runtime without any driver properties reconfiguration.

Saving driver, adding connection

After you finished configuring your driver just press 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:



Connection "PostgreSQL - postgres" configuration

Connection settings

PostgreSQL connection settings

- Connection settings
  - Driver properties
  - Network
  - Initialization
  - Shell Commands
  - Client identification
- General
- Metadata
- Error handle
- Result Sets
  - Editors
  - Data Formatting
  - Presentation
- SQL Editor
  - SQL Processing

Host: localhost Port: 5432

Database: pagila

User: postgres

Password: •••••••• ☑ Save password locally

Local Client: PostgreSQL Binaries

Settings

☑ Show non-default databases

□ Show template databases

You can use variables in connection parameters.

Driver name: PostgreSQL

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 long and unobvious string. DBeaver can construct this URL from connection parameters (like host, port, etc).

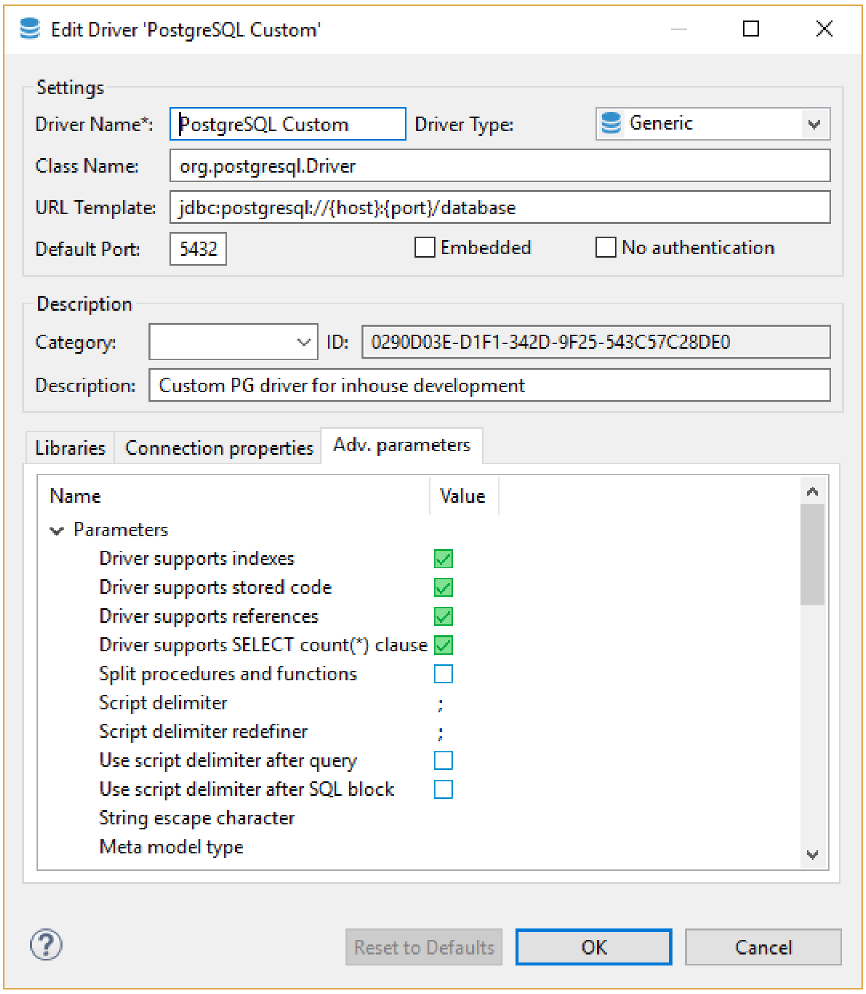
For example above the URL template is: jdbc:postgresql://{host}:{port}/{database}
Host, port and database are parameters which you will need to enter on the connection configuration page.

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 driver you don't need to change any advanced properties. But in some cases you can use this as a driver tuning, e.g. for better performance or for structure fixing.



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		

Name	Value
Parameters	
Driver supports indexes	☑
Driver supports stored code	☑
Driver supports references	☑
Driver supports SELECT count(*) clause	☑
Split procedures and functions	□
Script delimiter	;
Script delimiter redefiner	;
Use script delimiter after query	□
Use script delimiter after SQL block	□
String escape character	
Meta model type	

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
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 single schema	Hide schema if there is only one schema on server
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

Parameter	Description
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) driverss

DDL (DDL generation options)

Parameter	Description
Drop column short syntax	Use 'ALTER TABLE DROP column-name' instead of standard syntax
Use legacy SQL dialect for DDL	Use legacy SQL dialect for DDL

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

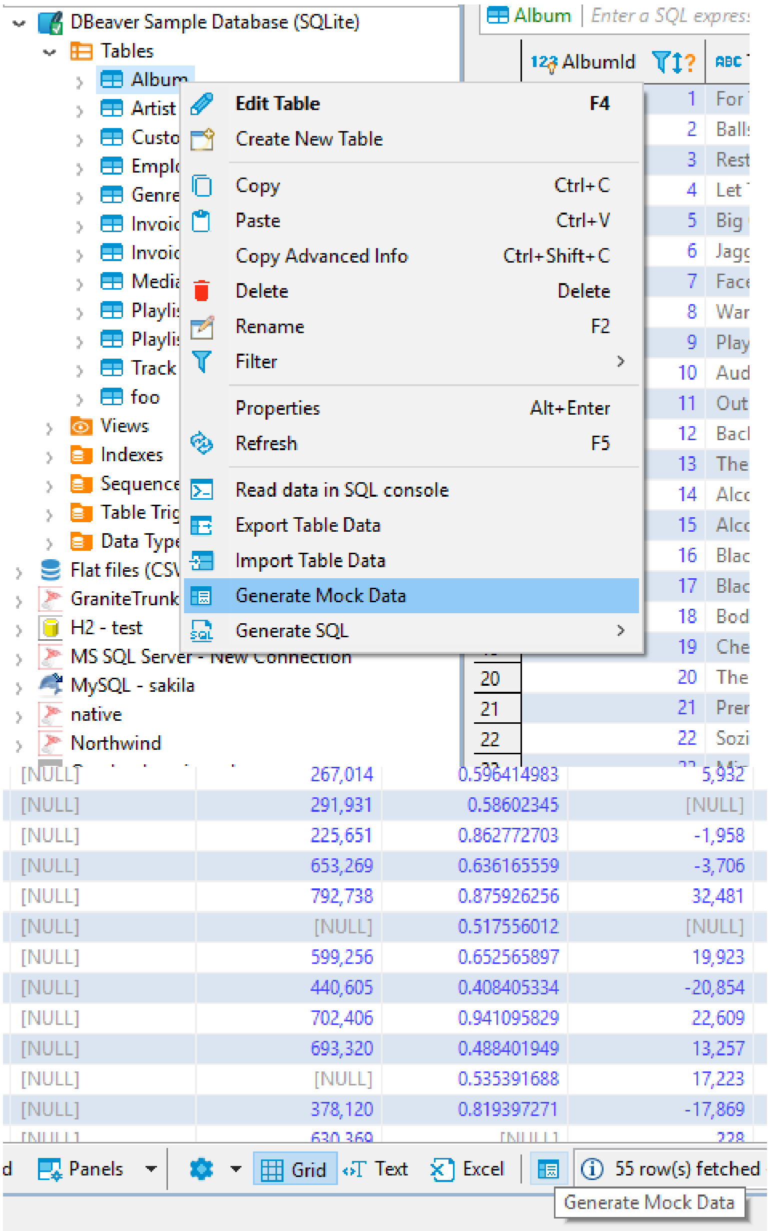
If you have configured some driver, it works good and you think that it makes sense to have this driver configuration in standard DBeaver, please send your configuration to us. Just create a feature request issue on GitHub and copy/paste driver description to the ticket (in any suitable form).

Thank you :)

Mock Data generation

Note: since version 6.2 Mock data generators were moved to EE version, see https://github.com/dbeaver/dbeaver/issues/6592

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.

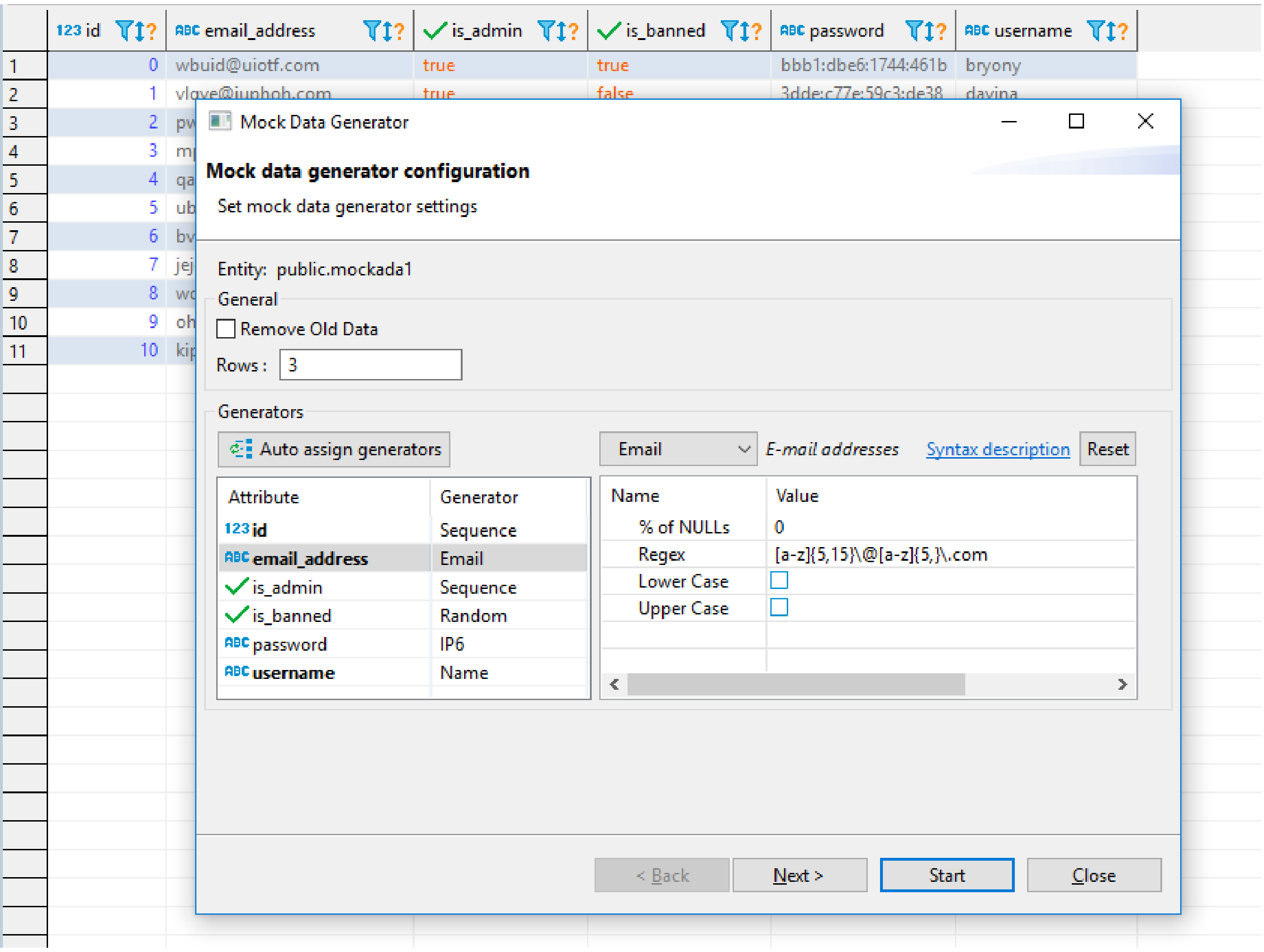


[NULL]	267,014	0.596414983	5,932
[NULL]	291,931	0.58602345	[NULL]
[NULL]	225,651	0.862772703	-1,958
[NULL]	653,269	0.636165559	-3,706
[NULL]	792,738	0.875926256	32,481
[NULL]	[NULL]	0.517556012	[NULL]
[NULL]	599,256	0.652565897	19,923
[NULL]	440,605	0.408405334	-20,854
[NULL]	702,406	0.941095829	22,609
[NULL]	693,320	0.488401949	13,257
[NULL]	[NULL]	0.535391688	17,223
[NULL]	378,120	0.819397271	-17,869

> *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.

Th following are features of 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



	id	email_address	is_admin	is_banned	password	username
1	0	wbuid@uiotf.com	true	true	bbb1:dbe6:1744:461b	bryony
2	1	vlqve@iunhoh.com	true	false	3dde:c77e:59c3:de38	davina
3	2	pw				
4	3	mj				
5	4	qa				
6	5	ub				
7	6	bv				
8	7	jej				
9	8	wc				
10	9	oh				
11	10	kip				

Mock Data Generator

Mock data generator configuration

Set mock data generator settings

Entity: public.mockada1

General

□ Remove Old Data

Rows : 3

Generators

Auto assign generators

Email · *E-mail addresses* · Syntax description · Reset

Attribute	Generator
id	Sequence
email_address	Email
is_admin	Sequence
is_banned	Random
password	IP6
username	Name

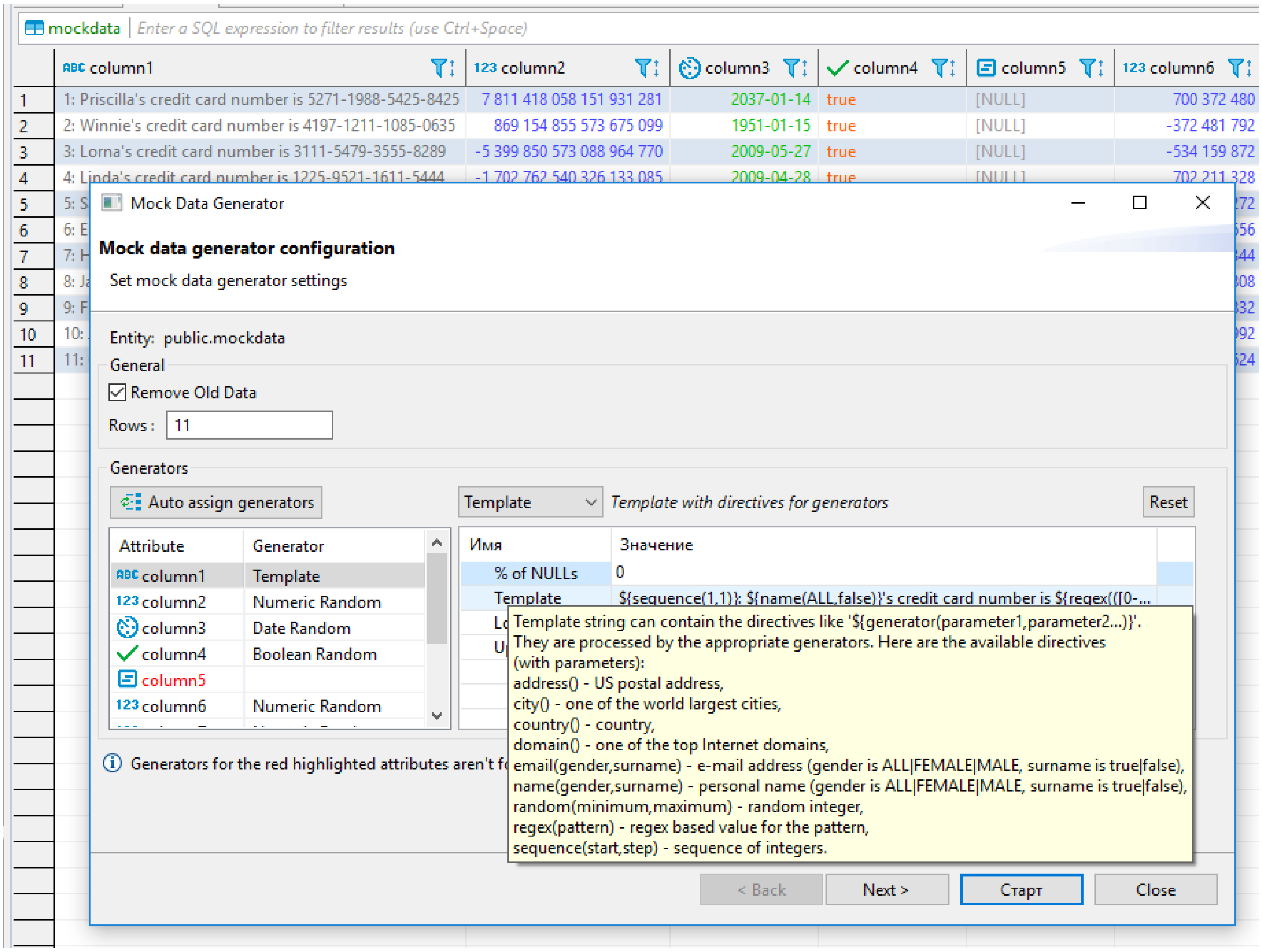
Name	Value
% of NULLs	0
Regex	[a-z]{5,15}\@[a-z]{5,}\.com
Lower Case	□
Upper Case	□

< Back · Next > · Start · Close

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 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



mockdata | *Enter a SQL expression to filter results (use Ctrl+Space)*

	column1	column2	column3	column4	column5	column6
1	1: Priscilla's credit card number is 5271-1988-5425-8425	7 811 418 058 151 931 281	2037-01-14	true	[NULL]	700 372 480
2	2: Winnie's credit card number is 4197-1211-1085-0635	869 154 855 573 675 099	1951-01-15	true	[NULL]	-372 481 792
3	3: Lorna's credit card number is 3111-5479-3555-8289	-5 399 850 573 088 964 770	2009-05-27	true	[NULL]	-534 159 872
4	4: Linda's credit card number is 1225-9521-1611-5444	-1 702 762 540 326 133 085	2009-04-28	true	[NULL]	702 211 328
5	5: S					72
6	6: E					56
7	7: H					44
8	8: Ja					08
9	9: F					32
10	10:					92
11	11:					24

Mock Data Generator

Mock data generator configuration

Set mock data generator settings

Entity: public.mockdata

General

☑ Remove Old Data

Rows : 11

Generators

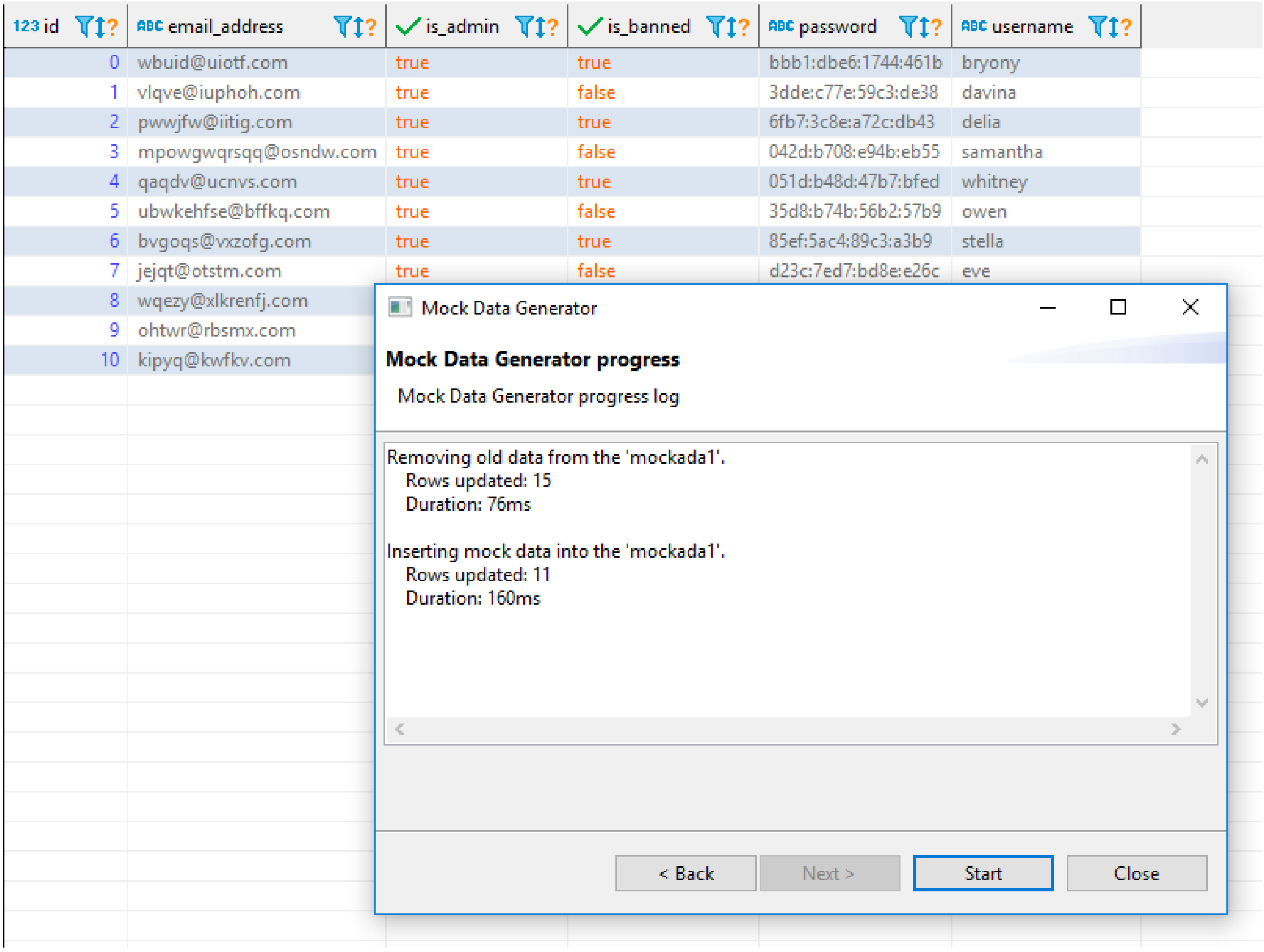
Template | *Template with directives for generators* | Reset

Attribute	Generator
column1	Template
column2	Numeric Random
column3	Date Random
column4	Boolean Random
column5	
column6	Numeric Random

Имя	Значение
% of NULLs	0
Template	${sequence(1,1)}: ${name(ALL,false)}'s credit card number is ${regex(([0-...

Template string can contain the directives like '${generator(parameter1,parameter2...)}'.
They are processed by the appropriate generators. Here are the available directives
(with parameters):
address() - US postal address,
city() - one of the world 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.

ⓘ Generators for the red highlighted attributes aren't f



id	email_address	is_admin	is_banned	password	username
0	wbuid@uiotf.com	true	true	bbb1:dbe6:1744:461b	bryony
1	vlqve@iuphoh.com	true	false	3dde:c77e:59c3:de38	davina
2	pwwjfw@iitig.com	true	true	6fb7:3c8e:a72c: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	wqezy@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
```

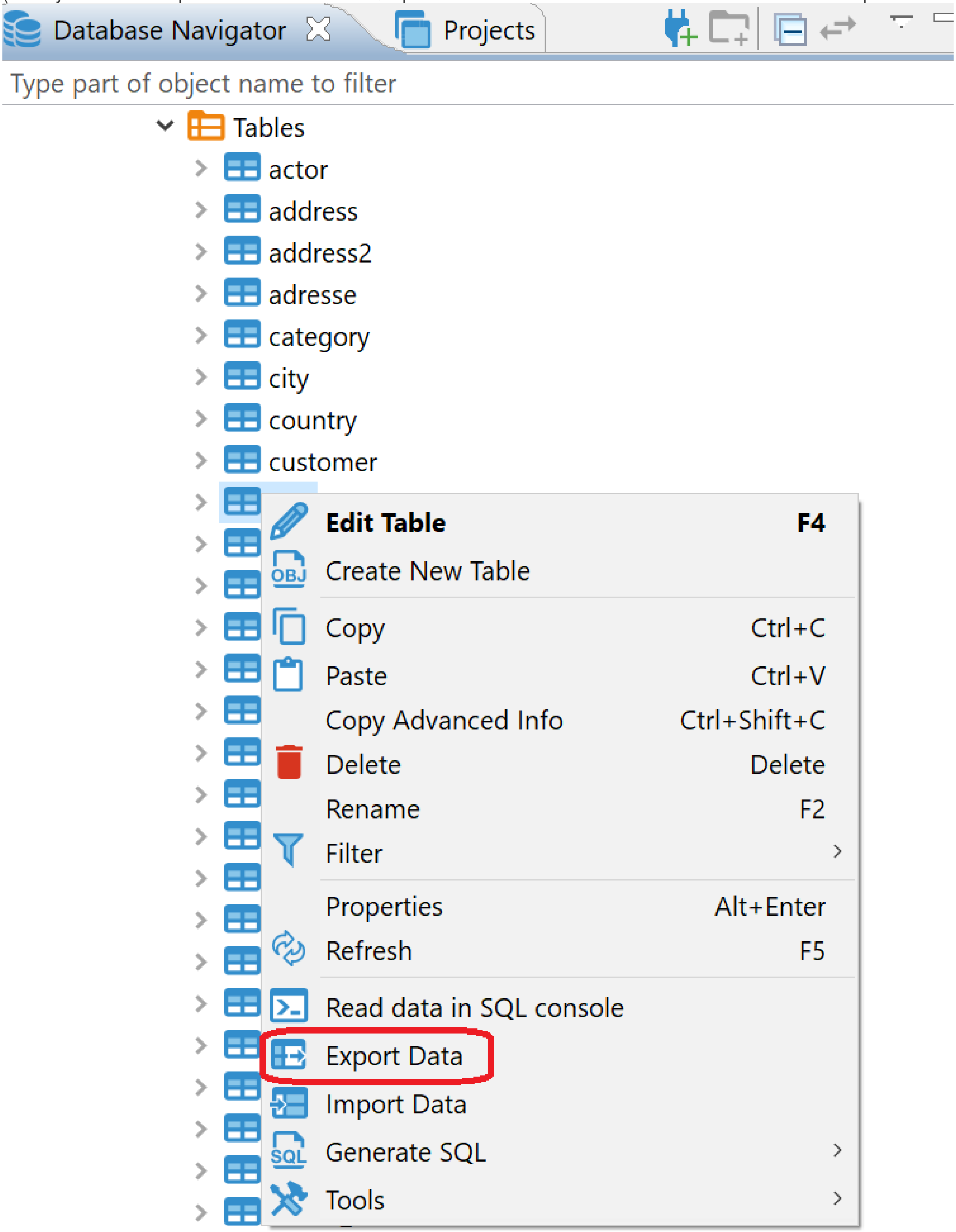
[*] These features are available in the DBeaver Enterprise Edition only.

Data export/import

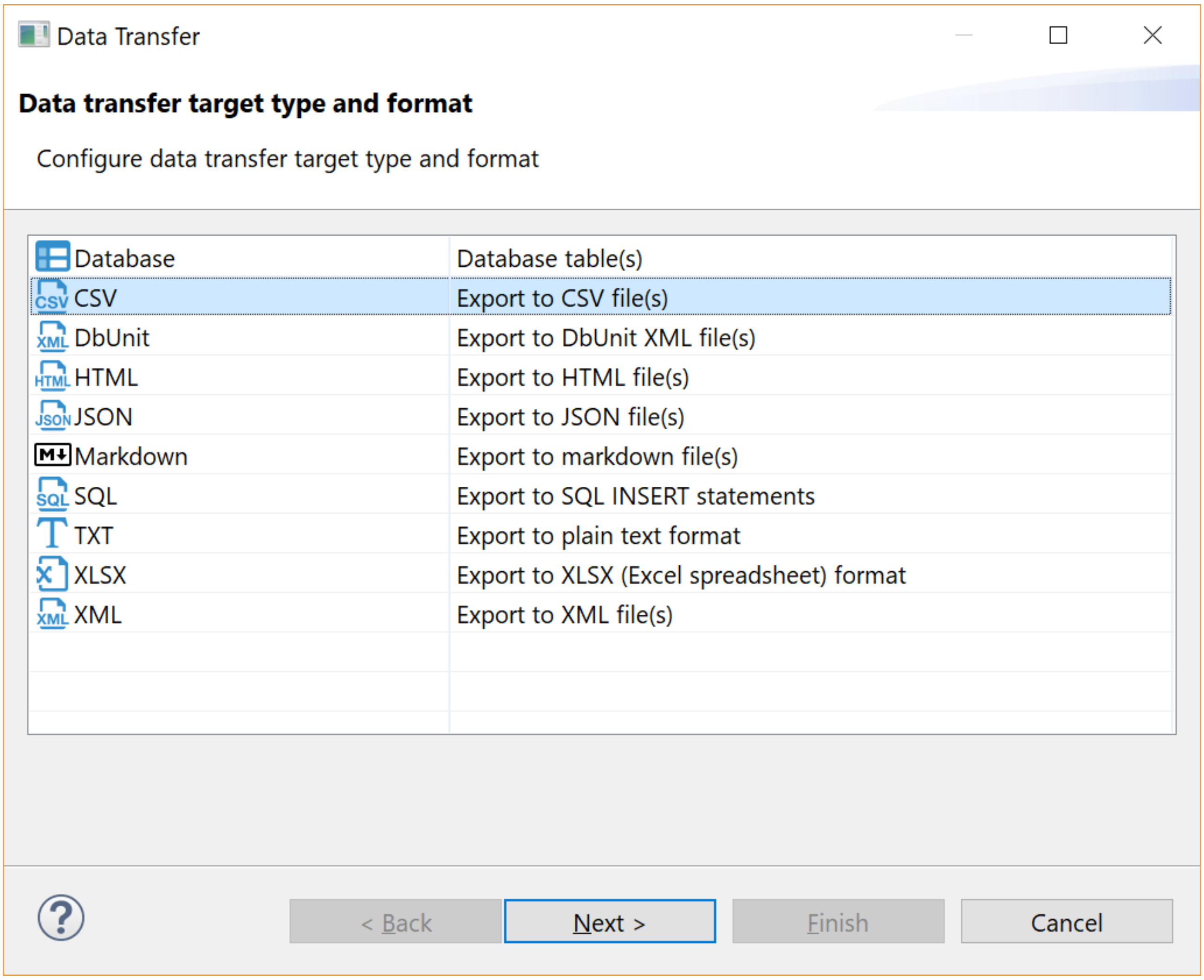
You can perform data export/import or migration for database table(s). We'll describe most typical use cases.

Exporting table data to CSV format

1. Select a table(s) you want to export. In the context menu choose "Export Data".
   (Note: you also can export data from custom SQL queries results. For that in results context menu choose "Export results").



2. Choose export format. DBeaver support many different output formats including CSV, HTML, XLSX, etc:

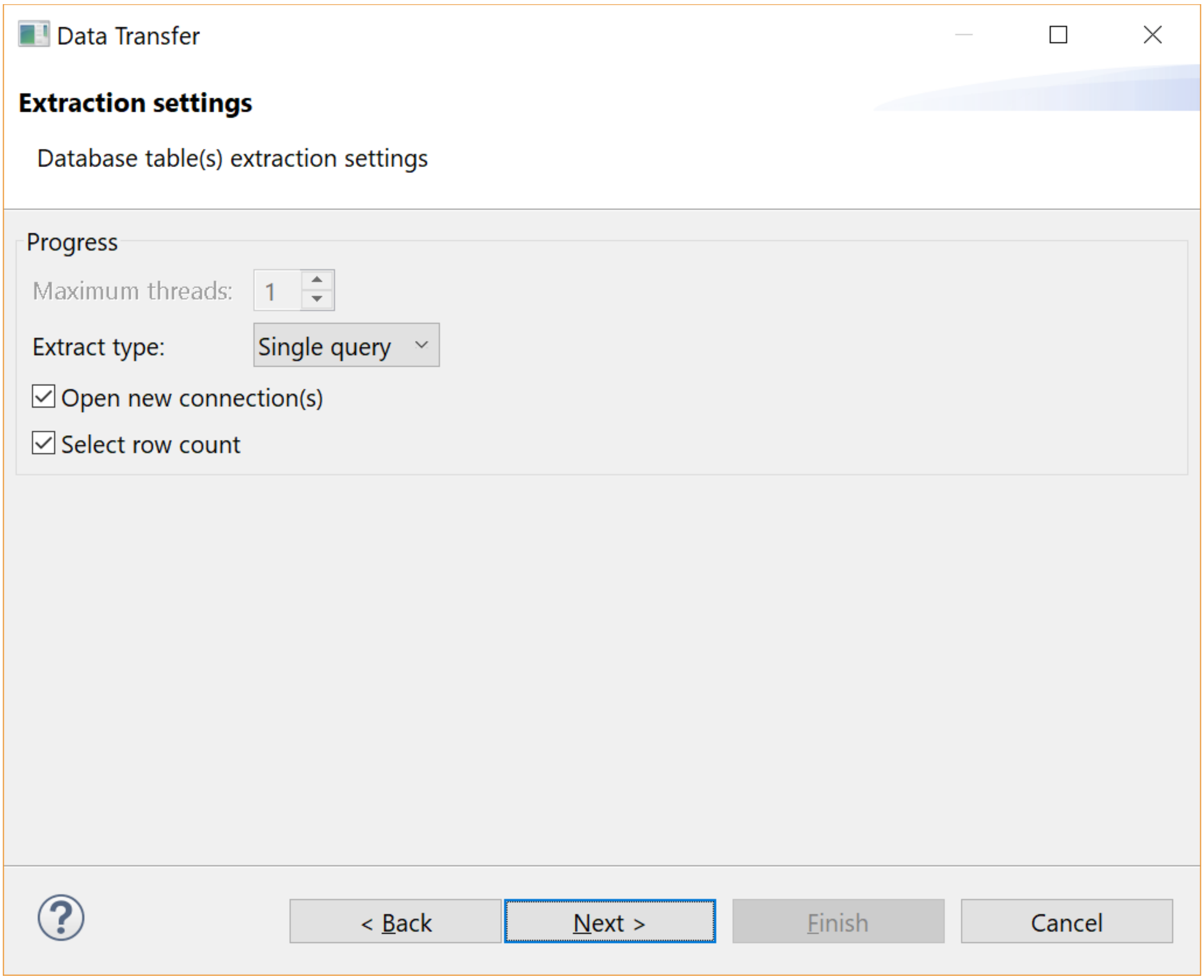


Data transfer target type and format

Configure data transfer target type and format

Database	Database table(s)
CSV	Export to CSV file(s)
DbUnit	Export to DbUnit XML file(s)
HTML	Export to HTML file(s)
JSON	Export to JSON file(s)
Markdown	Export to markdown file(s)
SQL	Export to SQL INSERT statements
TXT	Export to plain text format
XLSX	Export to XLSX (Excel spreadsheet) format
XML	Export to XML file(s)

3. Set data extraction options (how data will be read from tables). This may affect extraction performance:



Extraction settings

Database table(s) extraction settings

Progress

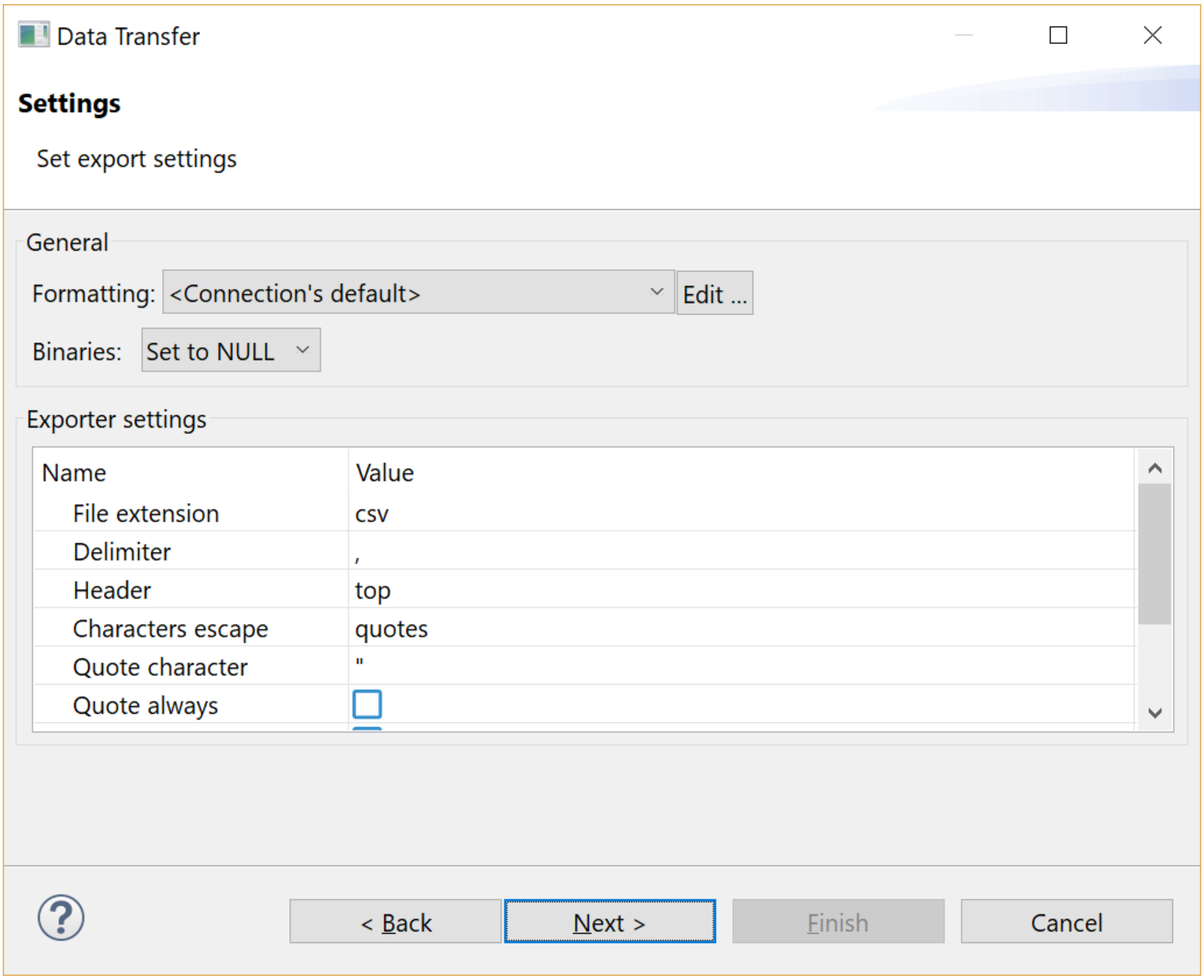
Maximum threads: 1

Extract type: Single query

☑ Open new connection(s)

☑ Select row count

4. Set export format option. They are specific to the data format you chose on step 2:



Settings

Set export settings

General

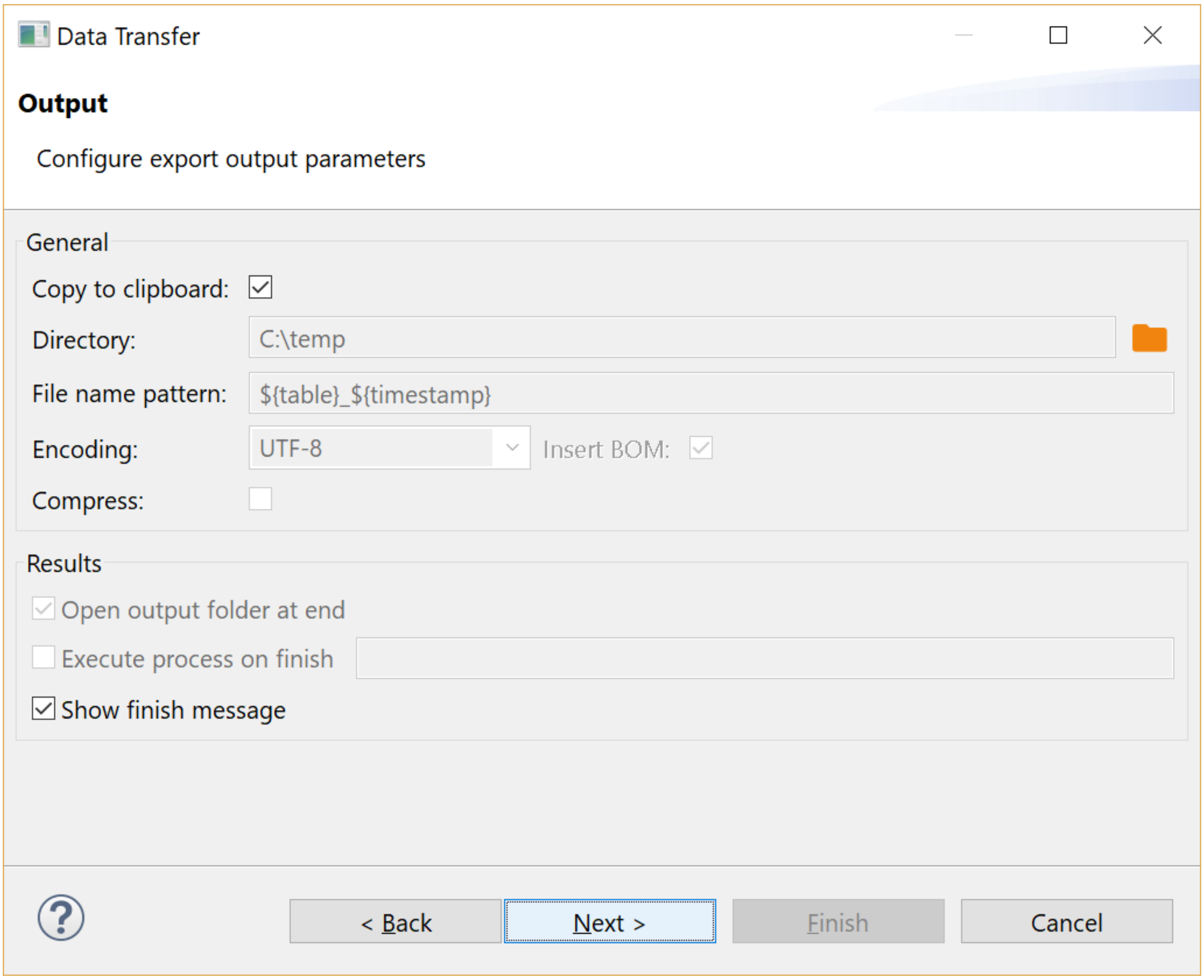
Formatting: <Connection's default> Edit ...

Binaries: Set to NULL

Exporter settings

Name	Value
File extension	csv
Delimiter	,
Header	top
Characters escape	quotes
Quote character	"
Quote always	□

5. Set options for output files or clipboard:



Data Transfer

Output

Configure export output parameters

General

Copy to clipboard: ☑

Directory: C:\temp

File name pattern: ${table}_${timestamp}

Encoding: UTF-8 Insert BOM: ☑

Compress: □

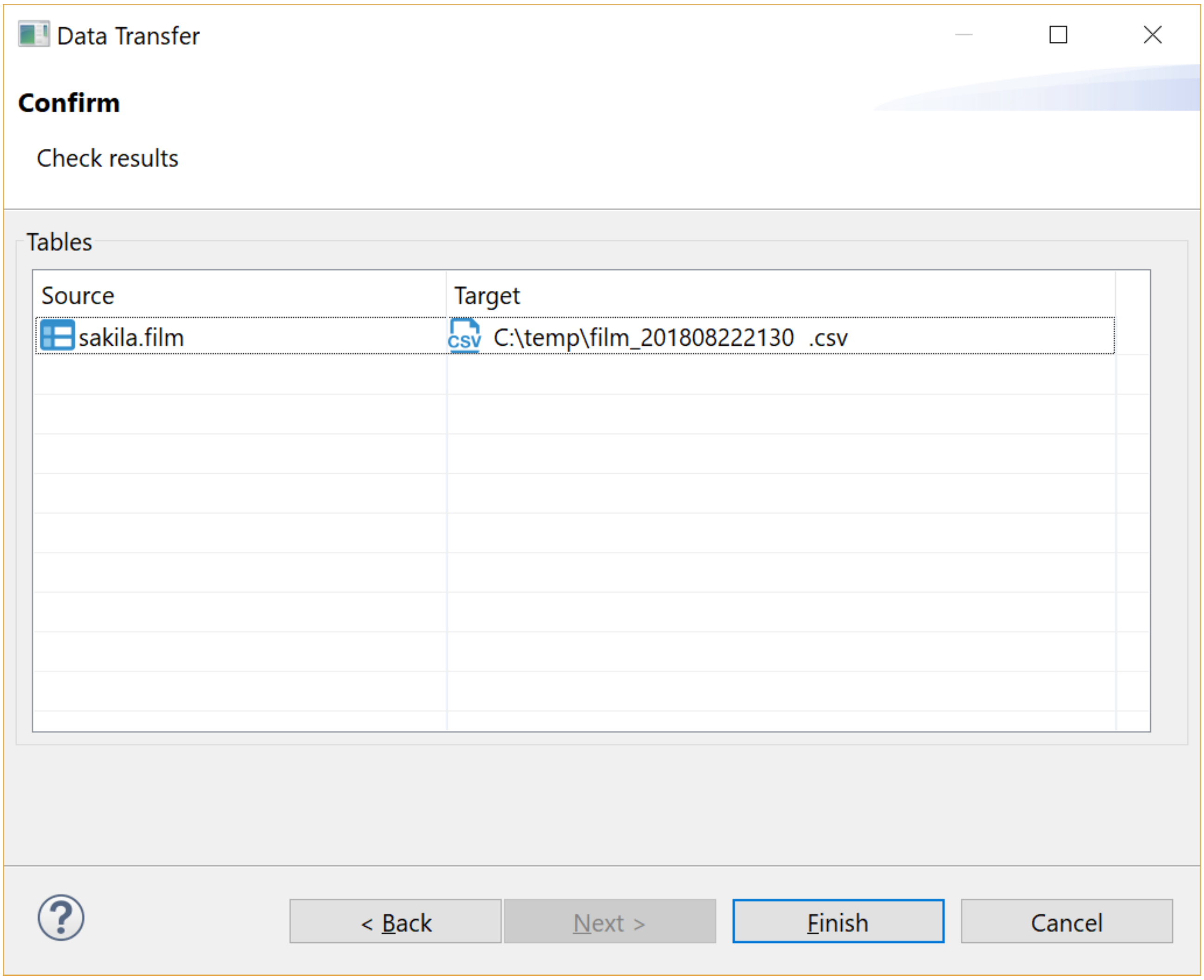
Results

☑ Open output folder at end

□ Execute process on finish

☑ Show finish message

6. Review what and to what format you will export:



Data Transfer

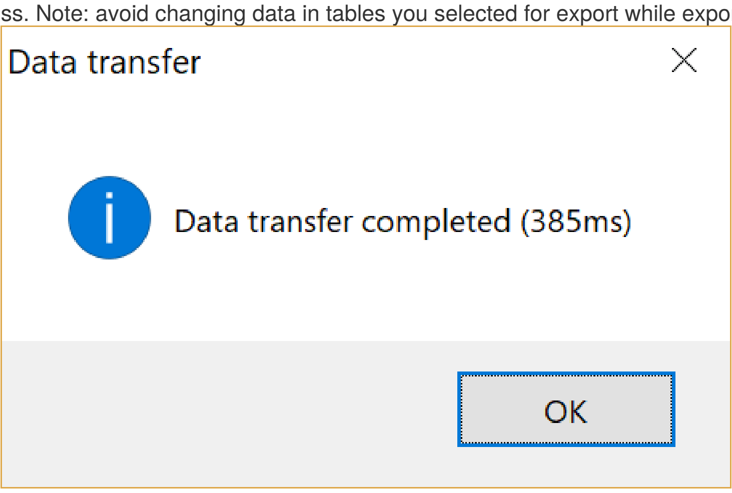
Confirm

Check results

Tables

Source	Target
sakila.film	C:\temp\film_201808222130 .csv

7. Press finish. See extraction progress. Actual data extraction will be performed in background, you can keep working with your database during export process. Note: avoid changing data in tables you selected for export while export is running. In the end you will see status message:



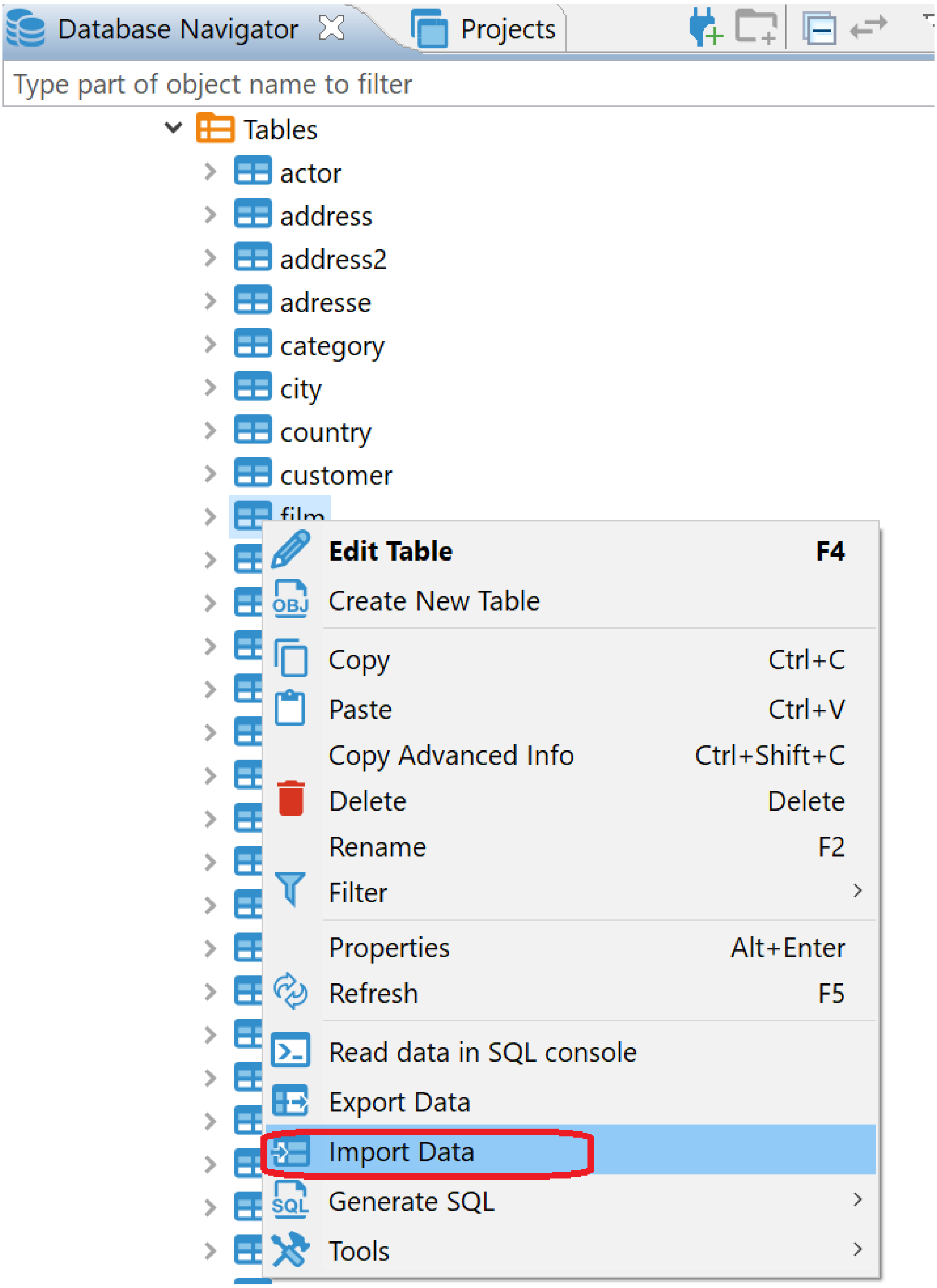
Data transfer

Data transfer completed (385ms)

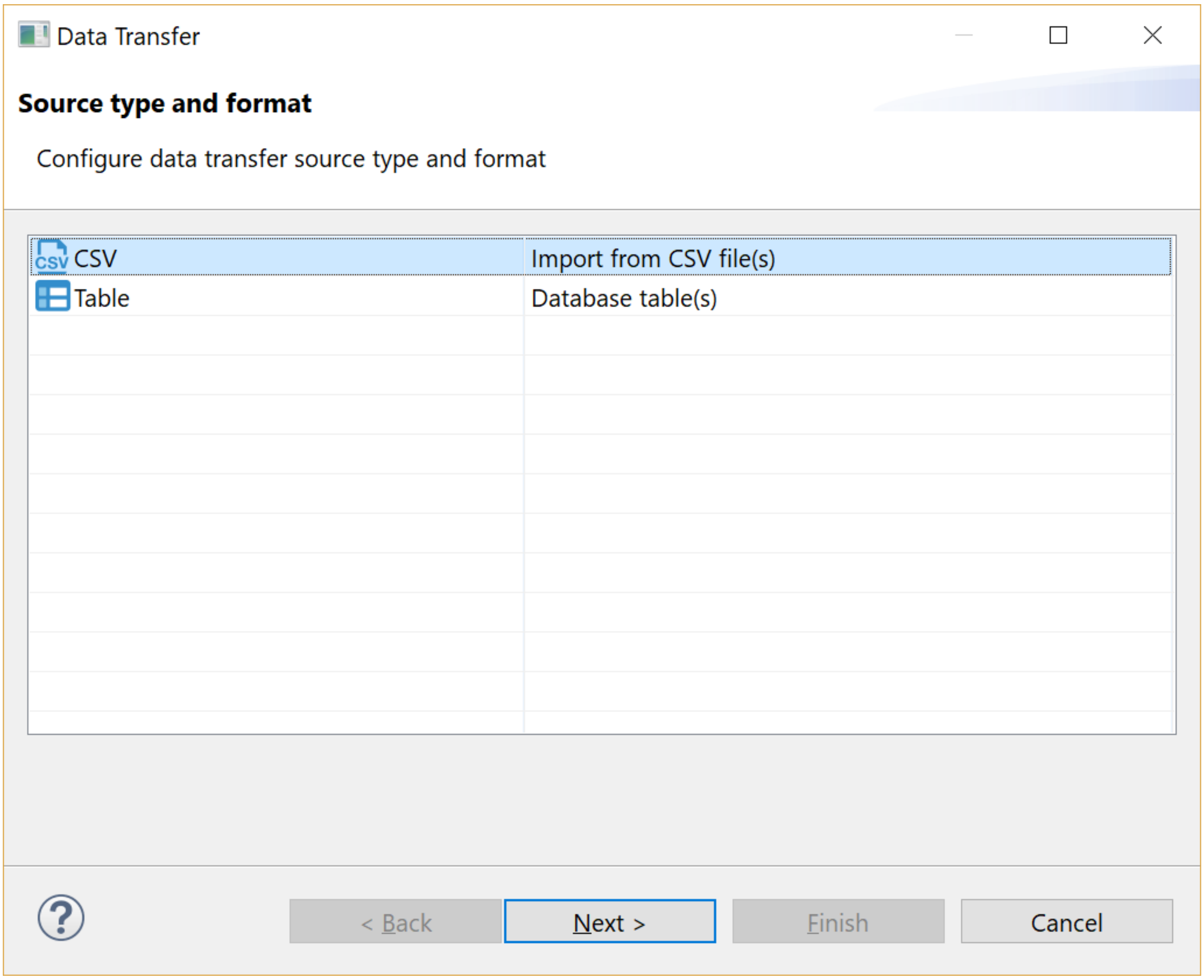
Importing data from CSV format

You can import data from CSV file(s) directly into your database table(s).

1. Select a table(s) to which you want to import data. In the context menu choose "Import Data":



2. Choose import format (CSV):

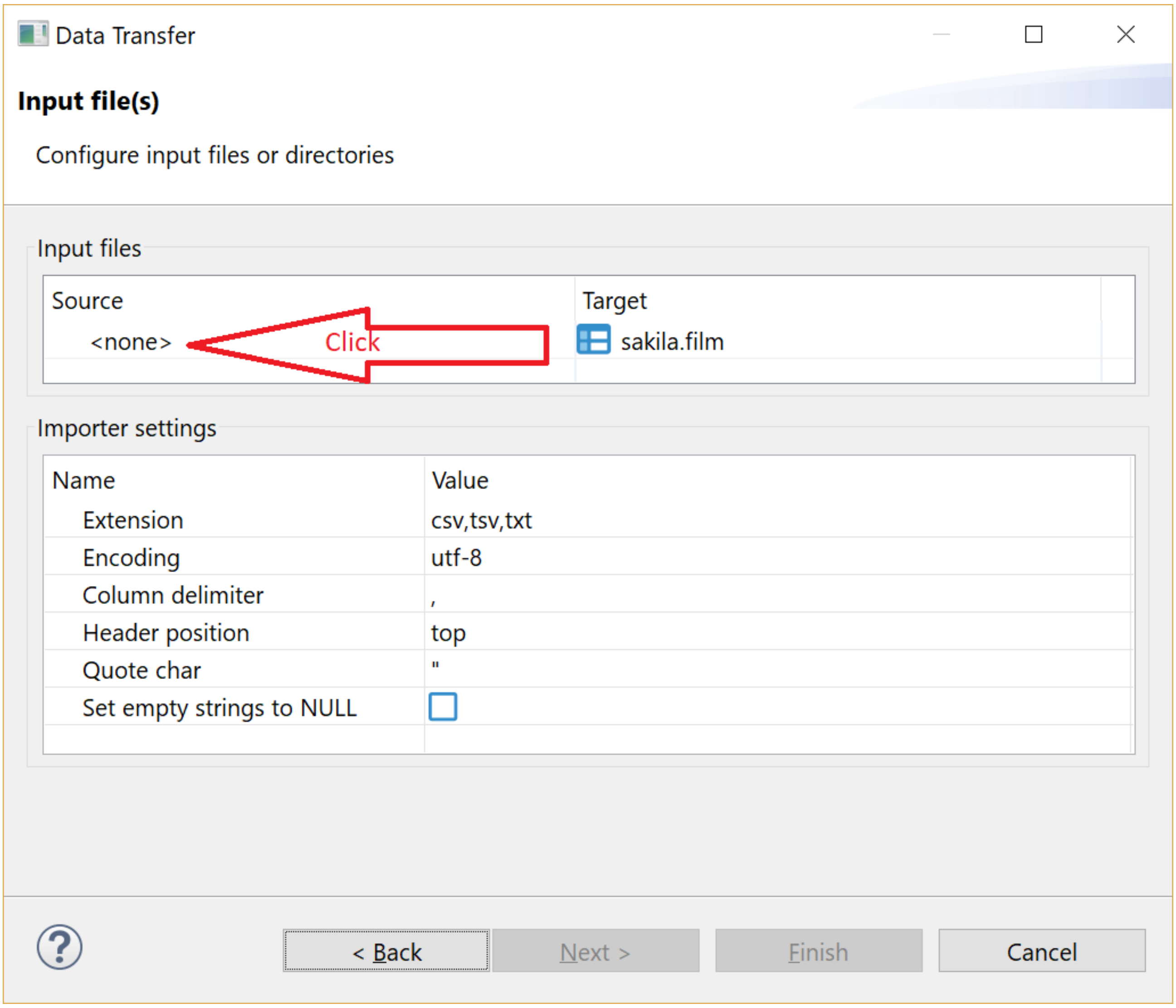

Data Transfer

Source type and format

Configure data transfer source type and format

CSV	Import from CSV file(s)
Table	Database table(s)

3. Select input CSV file for each table you want to import:



Input file(s)

Configure input files or directories

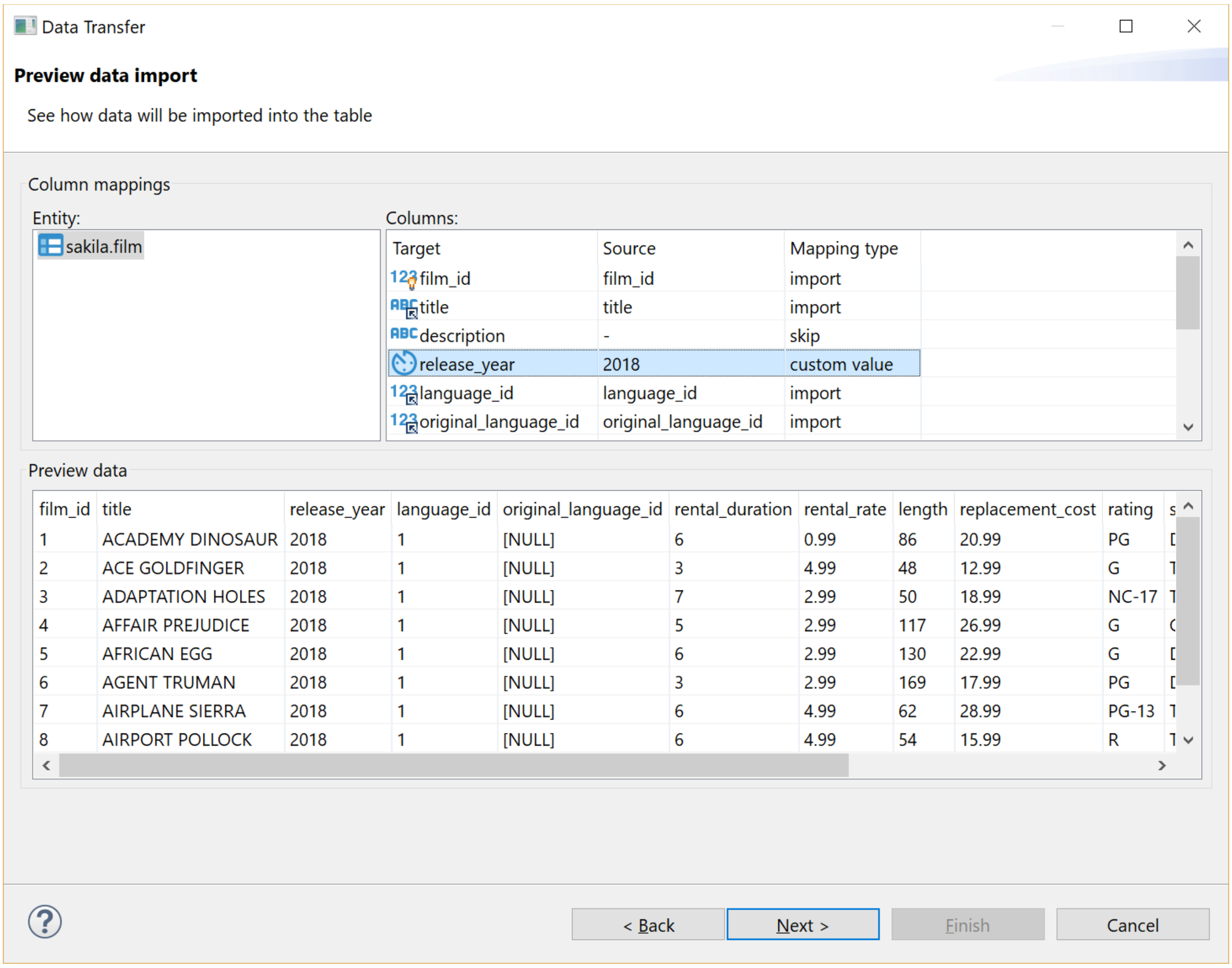
Input files

Source	Target
<none>	sakila.film

Importer settings

Name	Value
Extension	csv,tsv,txt
Encoding	utf-8
Column delimiter	,
Header position	top
Quote char	"
Set empty strings to NULL	□

4. Set CSV-to-table mappings. You need to set some column in CSV file for each database table column. You can skip some column at all (in target table column value will be set to NULL). You can set some constant value for table column if there is no source column for it in CSV.



Preview data import

See how data will be imported into the table

Column mappings

Entity:

sakila.film

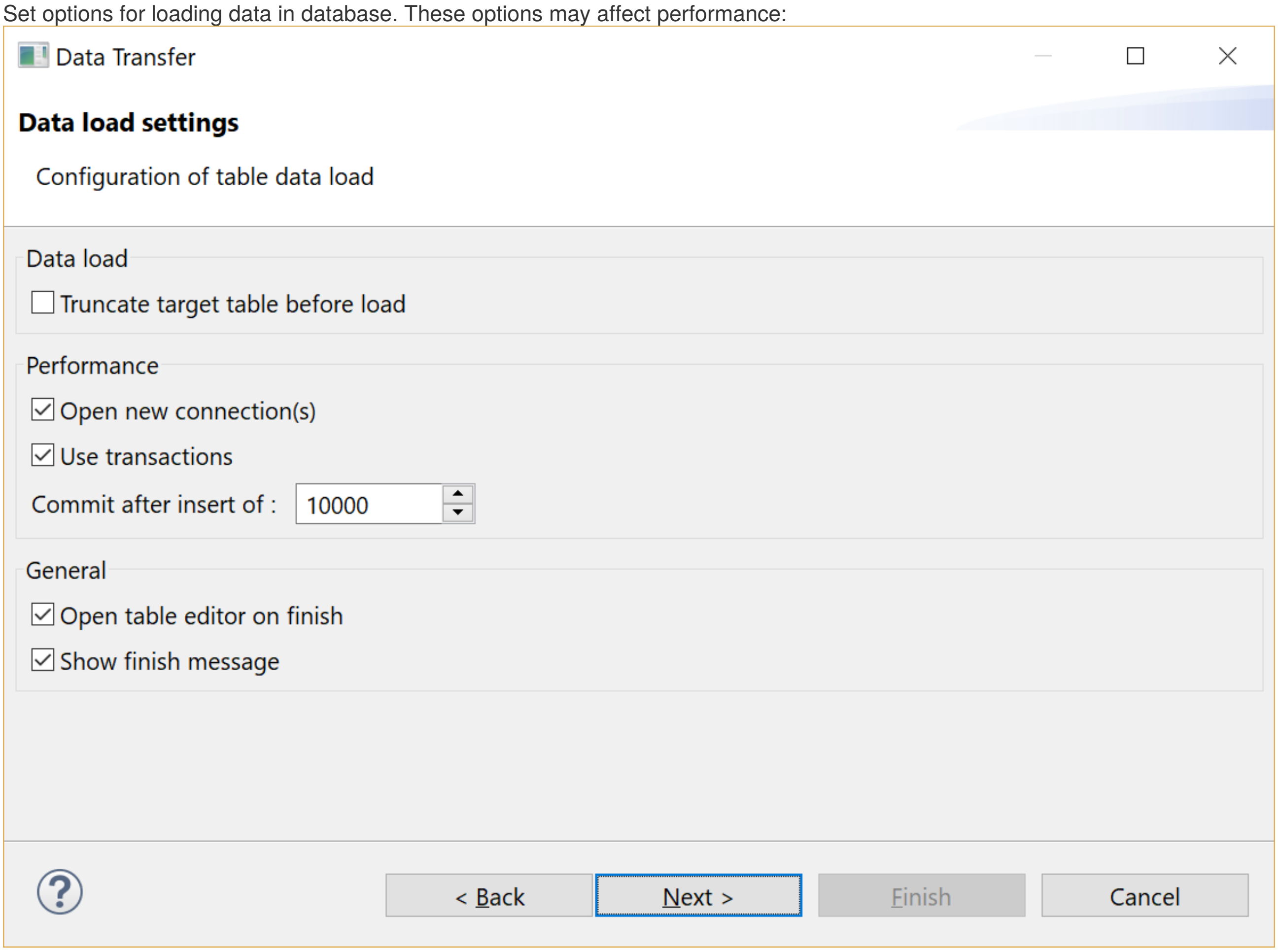
Columns:

Target	Source	Mapping type
film_id	film_id	import
title	title	import
description	-	skip
release_year	2018	custom value
language_id	language_id	import
original_language_id	original_language_id	import

Preview data

film_id	title	release_year	language_id	original_language_id	rental_duration	rental_rate	length	replacement_cost	rating	s
1	ACADEMY DINOSAUR	2018	1	[NULL]	6	0.99	86	20.99	PG	D
2	ACE GOLDFINGER	2018	1	[NULL]	3	4.99	48	12.99	G	T
3	ADAPTATION HOLES	2018	1	[NULL]	7	2.99	50	18.99	NC-17	T
4	AFFAIR PREJUDICE	2018	1	[NULL]	5	2.99	117	26.99	G	C
5	AFRICAN EGG	2018	1	[NULL]	6	2.99	130	22.99	G	D
6	AGENT TRUMAN	2018	1	[NULL]	3	2.99	169	17.99	PG	D
7	AIRPLANE SIERRA	2018	1	[NULL]	6	4.99	62	28.99	PG-13	T
8	AIRPORT POLLOCK	2018	1	[NULL]	6	4.99	54	15.99	R	T

5. Set options for loading data in database. These options may affect performance:



Data load settings

Configuration of table data load

Data load

□ Truncate target table before load

Performance

☑ Open new connection(s)

☑ Use transactions

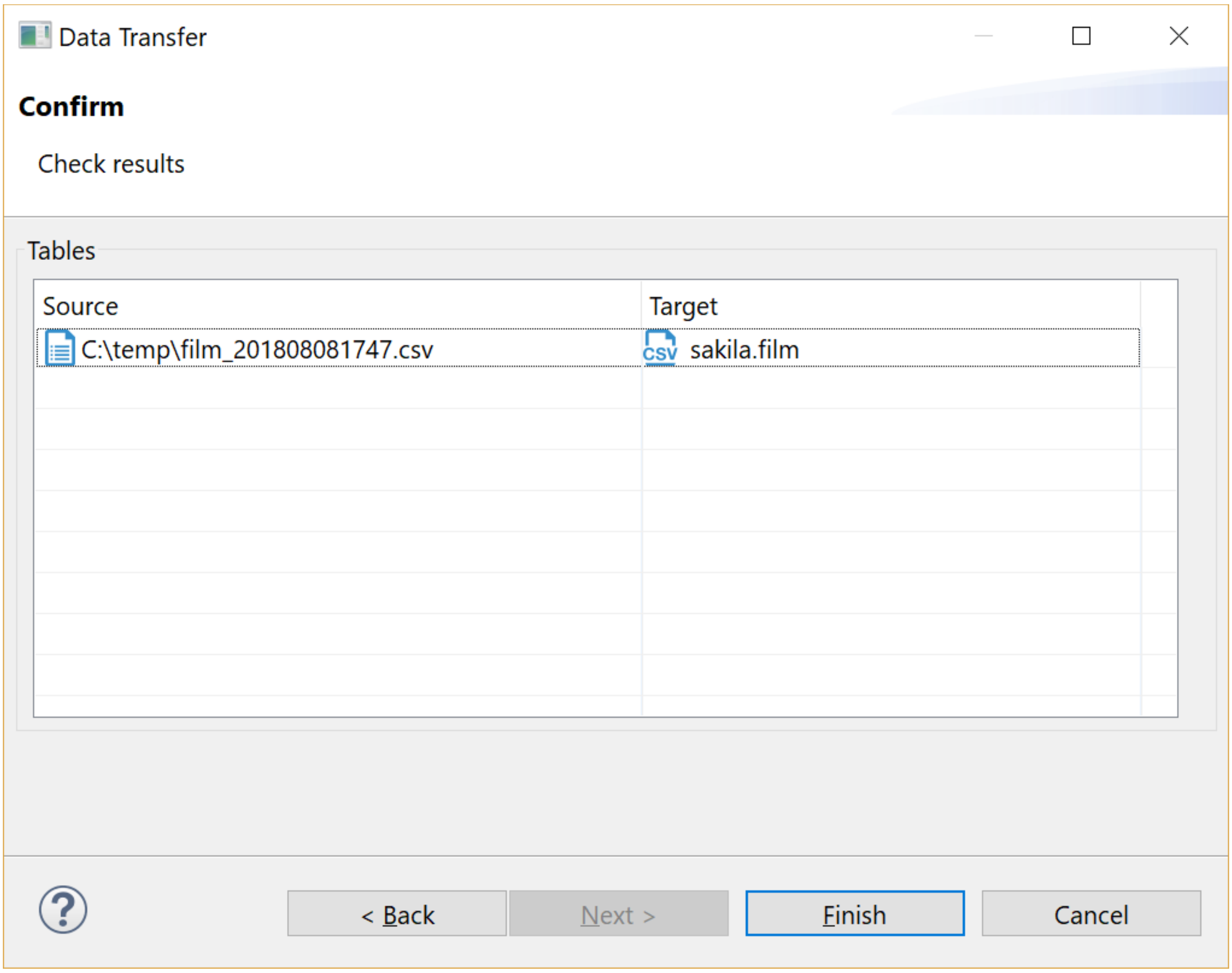
Commit after insert of : 10000

General

☑ Open table editor on finish

☑ Show finish message

6. Review what file(s) and to what table(s) you will import:

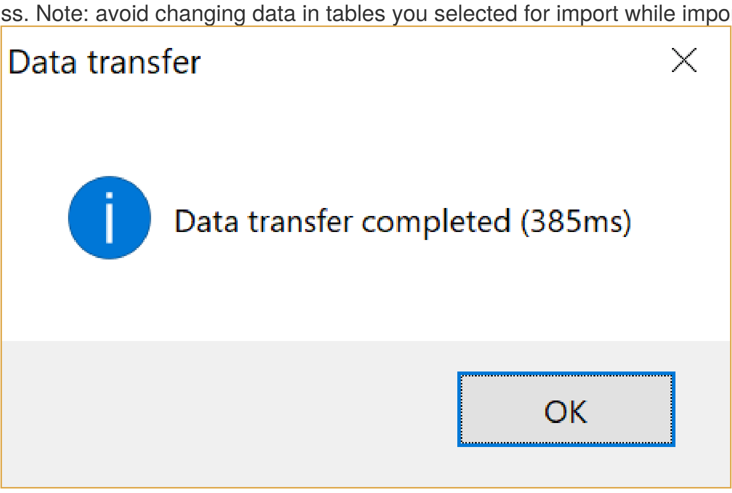

Confirm

Check results

Tables

Source	Target
C:\temp\film_201808081747.csv	sakila.film

7. Press finish. See extraction progress. Actual data loading will be performed in background, you can keep working with your database during export process. Note: avoid changing data in tables you selected for import while import is running. In the end you will see status message:


Data transfer completed (385ms)

Migrating table(s) data to another database table(s)

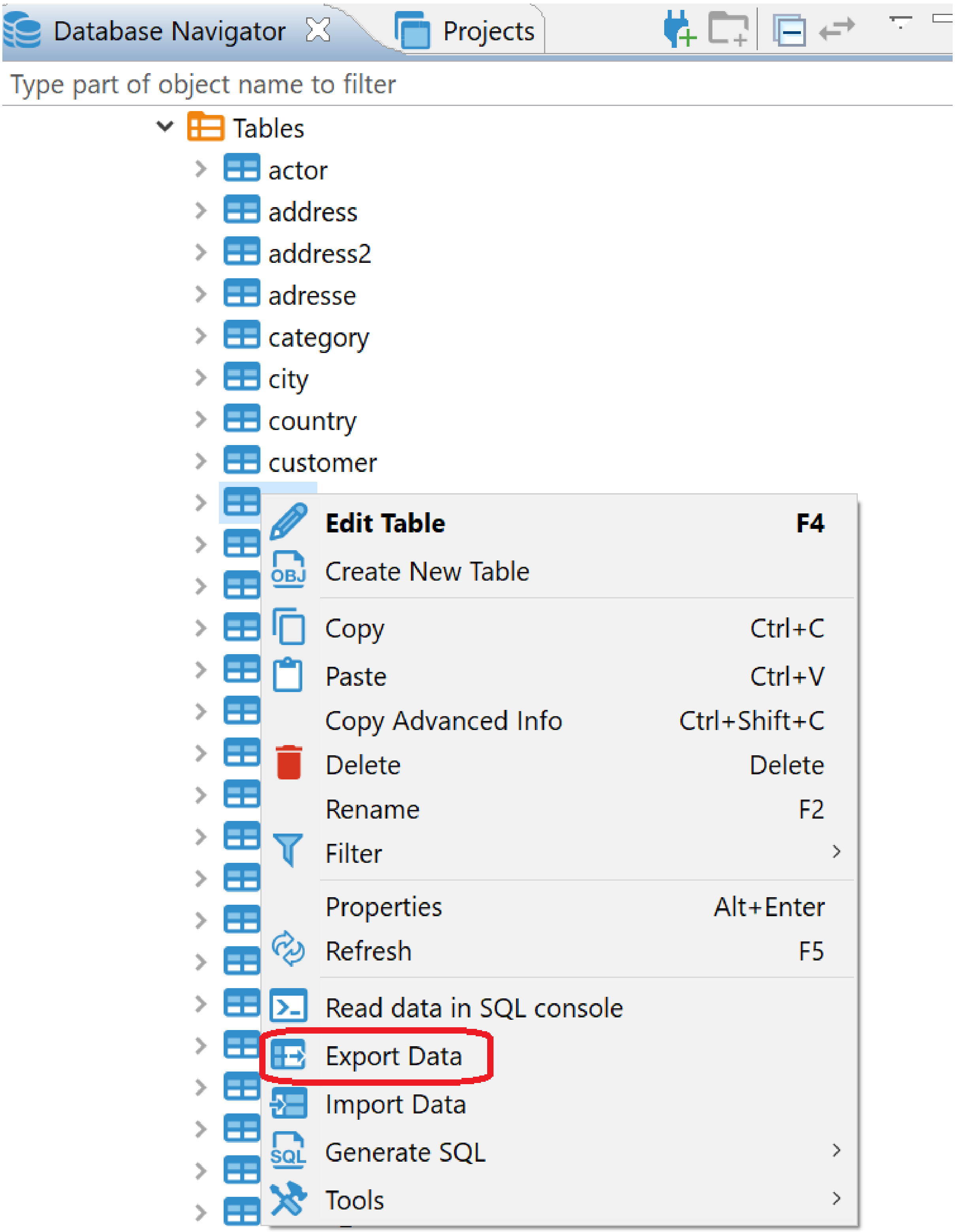
Data migration

DBeaver supports data migration from tables of one database to tables of another one.

To perform data transfer, please, follow the steps below.

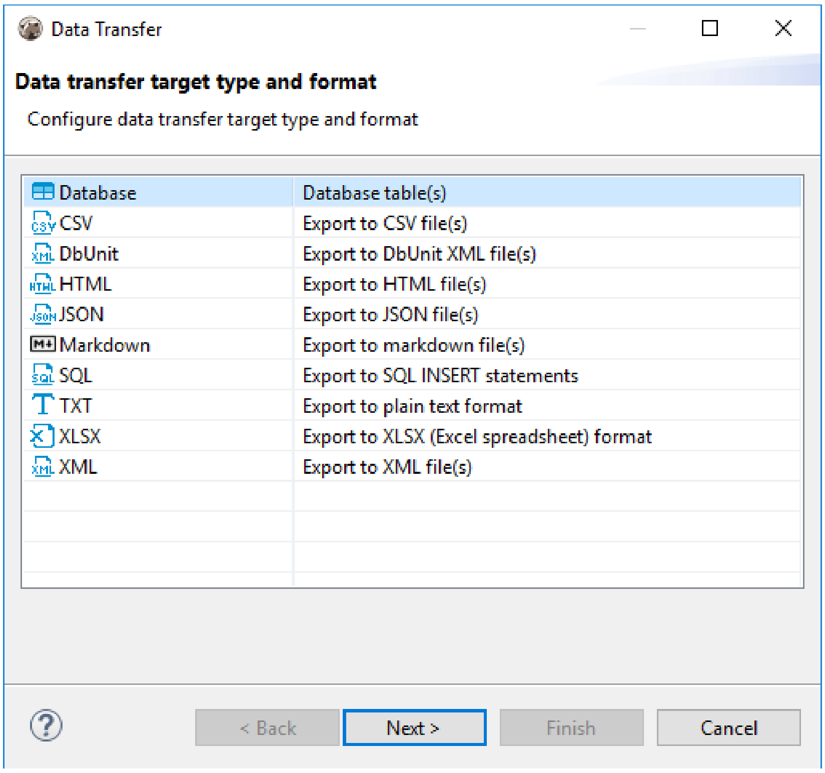
Step 1: Define the data source

In the **Database Navigator** select one or more tables you want to export. In the context menu choose "Export Data". (Note: you also can export data from the custom SQL queries results. For that in the results context menu choose "Export results").



Step 2: Define data transfer target type

In the opened dialog box choose **Database** type as the data transfer target and press **Next**.



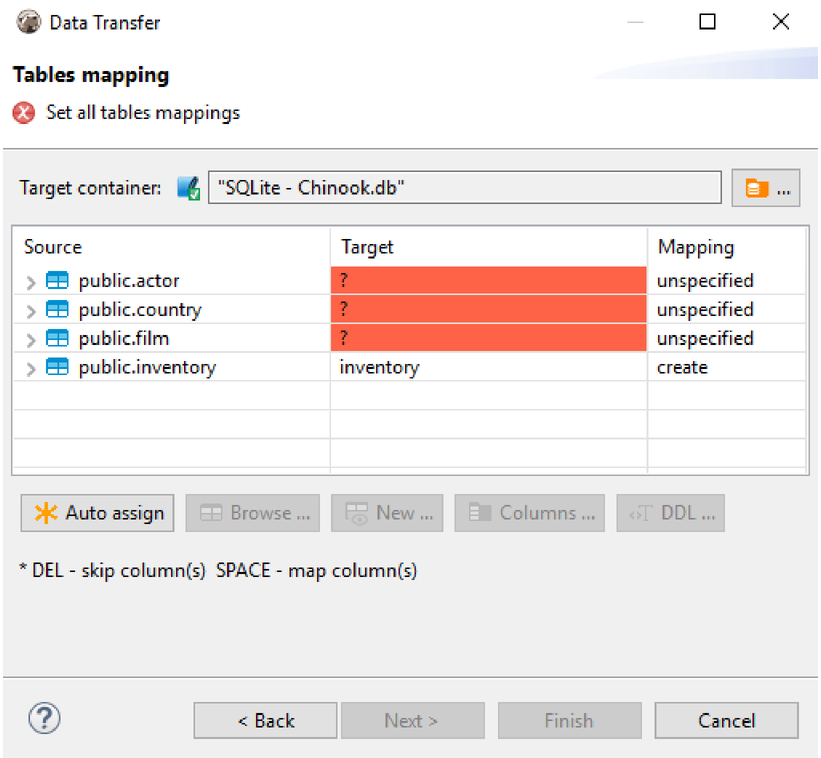
Data transfer target type and format

Configure data transfer target type and format

Database	Database table(s)
CSV	Export to CSV file(s)
DbUnit	Export to DbUnit XML file(s)
HTML	Export to HTML file(s)
JSON	Export to JSON file(s)
Markdown	Export to markdown file(s)
SQL	Export to SQL INSERT statements
TXT	Export to plain text format
XLSX	Export to XLSX (Excel spreadsheet) format
XML	Export to XML file(s)

Step 3: Define data mapping

For proper table mapping the following options are available:



Tables mapping

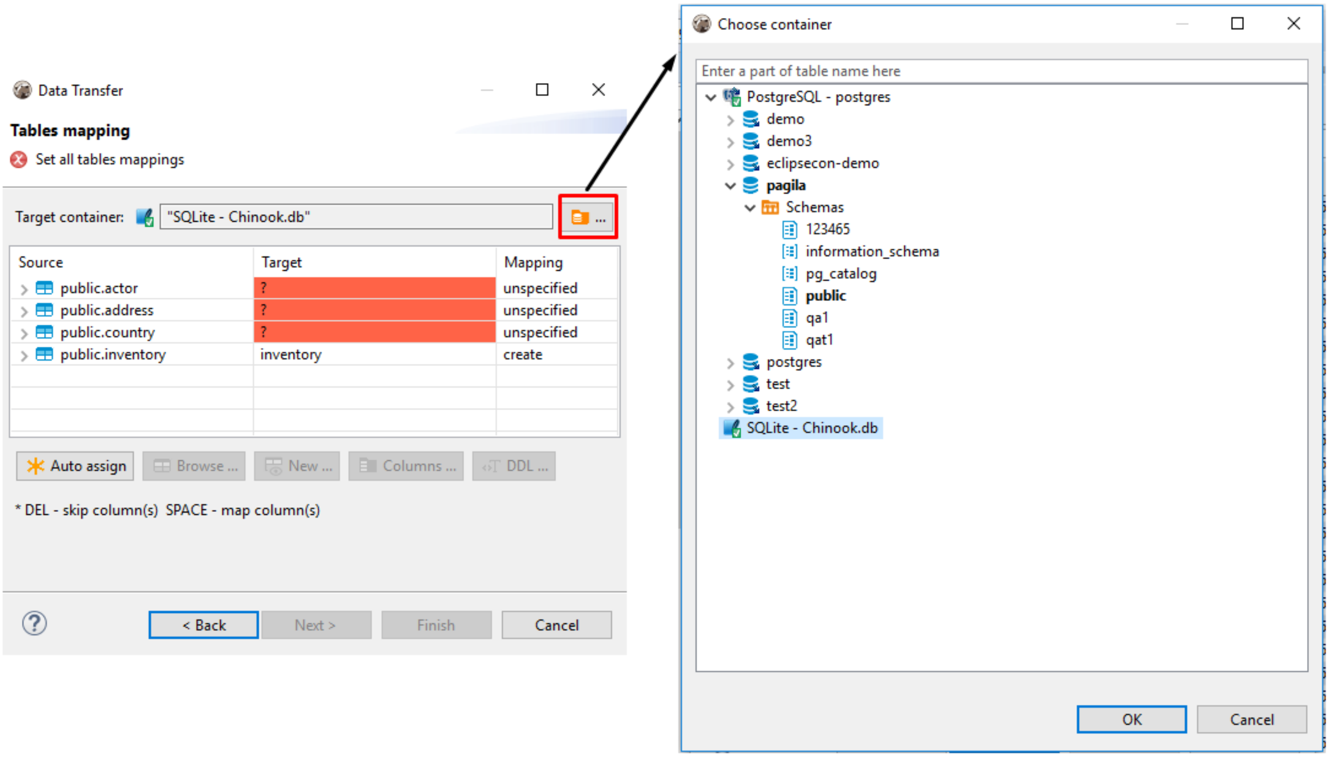
Set all tables mappings

Target container: "SQLite - Chinook.db"

Source	Target	Mapping
public.actor	?	unspecified
public.country	?	unspecified
public.film	?	unspecified
public.inventory	inventory	create

* DEL - skip column(s) SPACE - map column(s)

- **Target container** - defines a database or a scheme where the data will be transferred to. Press ... button and choose the container.



Tables mapping

Set all tables mappings

Target container: "SQLite - Chinook.db"

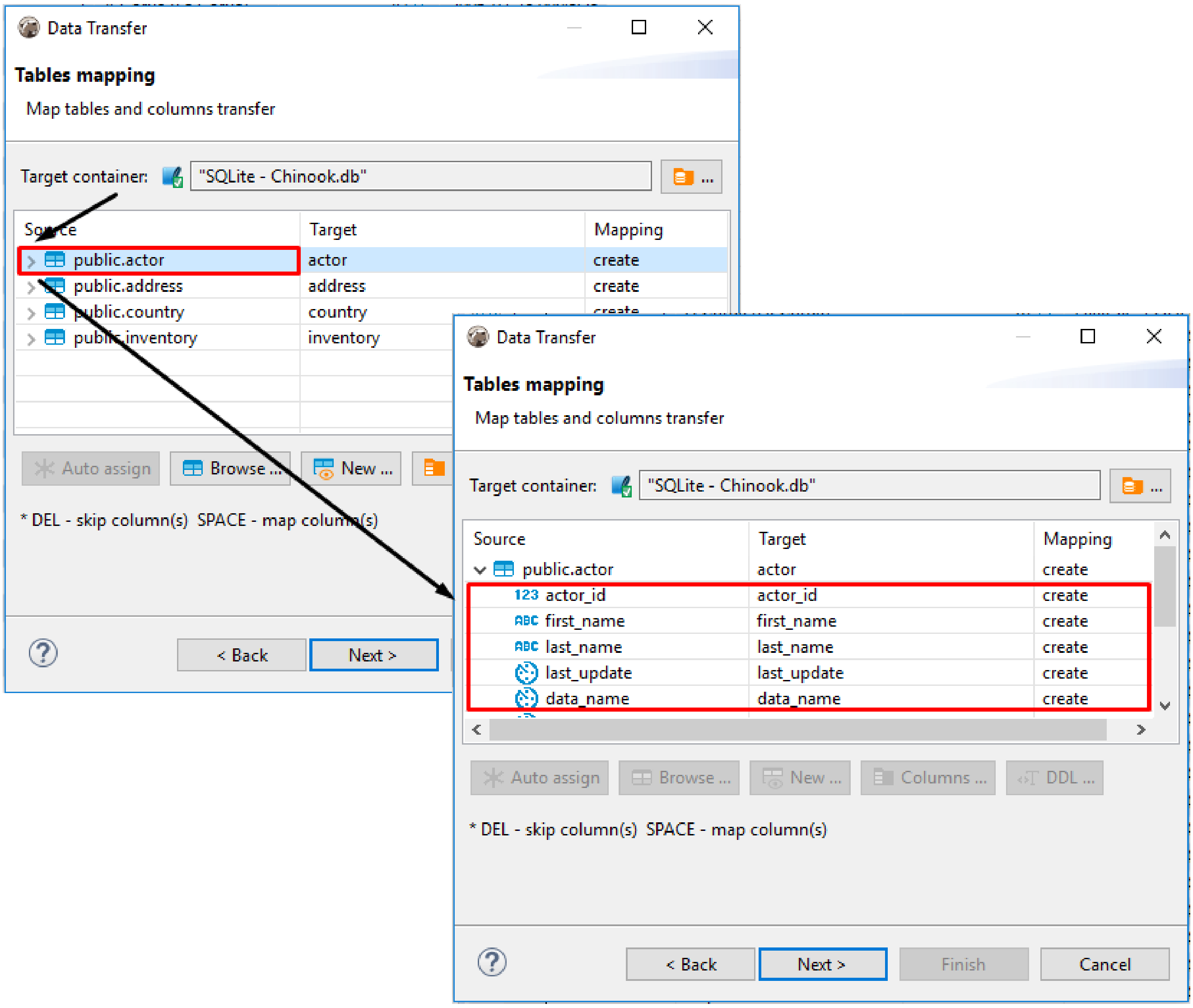
Source	Target	Mapping
public.actor	?	unspecified
public.address	?	unspecified
public.country	?	unspecified
public.inventory	inventory	create

* DEL - skip column(s) SPACE - map column(s)

Enter a part of table name here

- PostgreSQL - postgres
  - demo
  - demo3
  - eclipsecon-demo
  - pagila
    - Schemas
      - 123465
      - information_schema
      - pg_catalog
      - public
      - qa1
      - qat1
  - postgres
  - test
  - test2
- SQLite - Chinook.db

- **Source** - contains names of all the tables selected at step 1. You can also see the list of columns existing in the source table by pressing >



Data Transfer

Tables mapping

Map tables and columns transfer

Target container: "SQLite - Chinook.db"

Source	Target	Mapping
public.actor	actor	create
public.address	address	create
public.country	country	create
public.inventory	inventory	

* DEL - skip column(s) SPACE - map column(s)

Data Transfer

Tables mapping

Map tables and columns transfer

Target container: "SQLite - Chinook.db"

Source	Target	Mapping
public.actor	actor	create
actor_id	actor_id	create
first_name	first_name	create
last_name	last_name	create
last_update	last_update	create
data_name	data_name	create

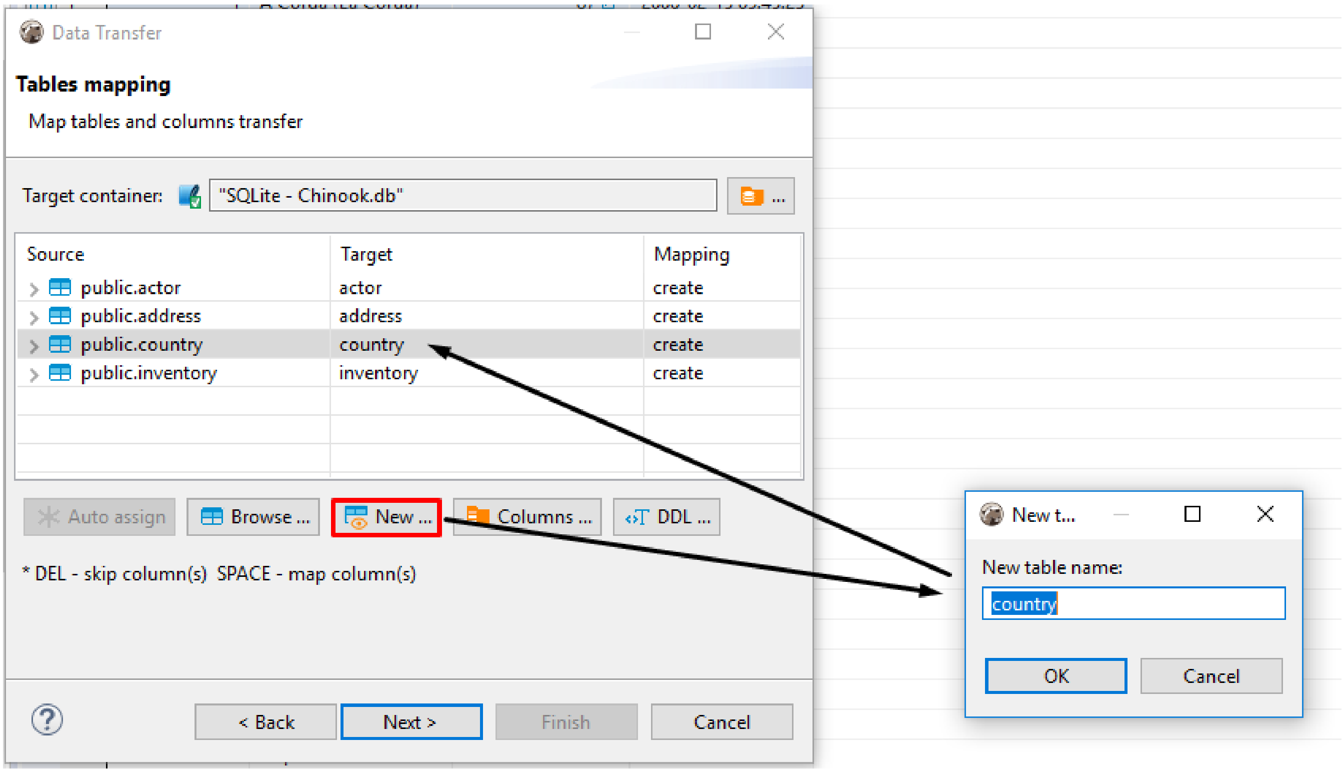
* DEL - skip column(s) SPACE - map column(s)

- **Target** - contains names of the tables where the data will be transferred to.
- **Mapping** - contains the list of actions to be applied to the source data on data transfer. The following options are available:

  Create- the source data will be populated into a newly created table or column of the target container.
  Skip - the source data will not be transferred to the target container.
  Existing - the source data will be transferred to the table that already exists in the target container.
  Unassigned - this value is set by default when there is no target defined.

If cells are marked as ? it means that in the target table there are no source tables with matching names, otherwise the names will be filled in automatically.

You can define a target table by clicking on a cell in the **Target** column and entering its name, or press the **New** button New ... and enter a new name in the opened dialog box.



Data Transfer

Tables mapping

Map tables and columns transfer

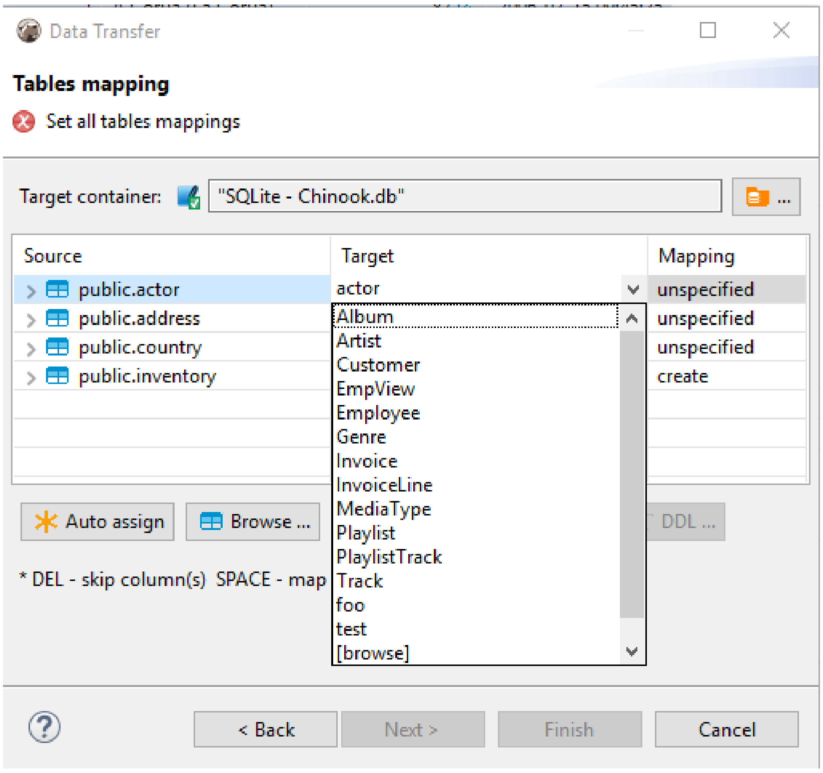
Target container: "SQLite - Chinook.db"

Source	Target	Mapping
public.actor	actor	create
public.address	address	create
public.country	country	create
public.inventory	inventory	create

* DEL - skip column(s) SPACE - map column(s)

New table name: country

You can also choose a name for a target table from the drop-down list.



Data Transfer

Tables mapping

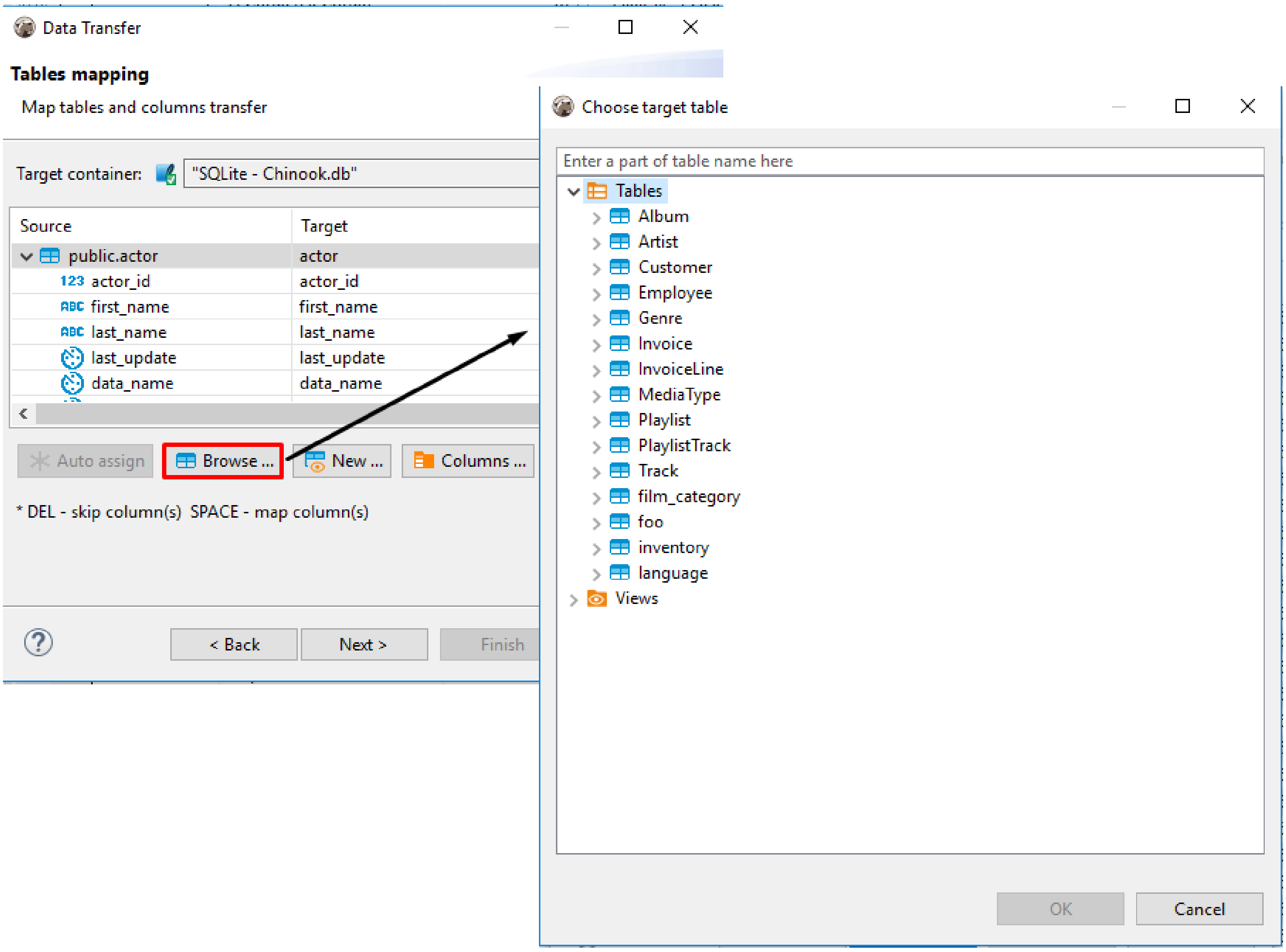
Set all tables mappings

Target container: "SQLite - Chinook.db"

Source	Target	Mapping
public.actor	actor	unspecified
public.address		unspecified
public.country		unspecified
public.inventory		create

* DEL - skip column(s) SPACE - map

Or select from the list of tables already existing in the target container by pressing the **Browse** button Browse ...



Data Transfer

Tables mapping

Map tables and columns transfer

Target container: "SQLite - Chinook.db"

Source	Target
public.actor	actor
actor_id	actor_id
first_name	first_name
last_name	last_name
last_update	last_update
data_name	data_name

* DEL - skip column(s) SPACE - map column(s)

Choose target table

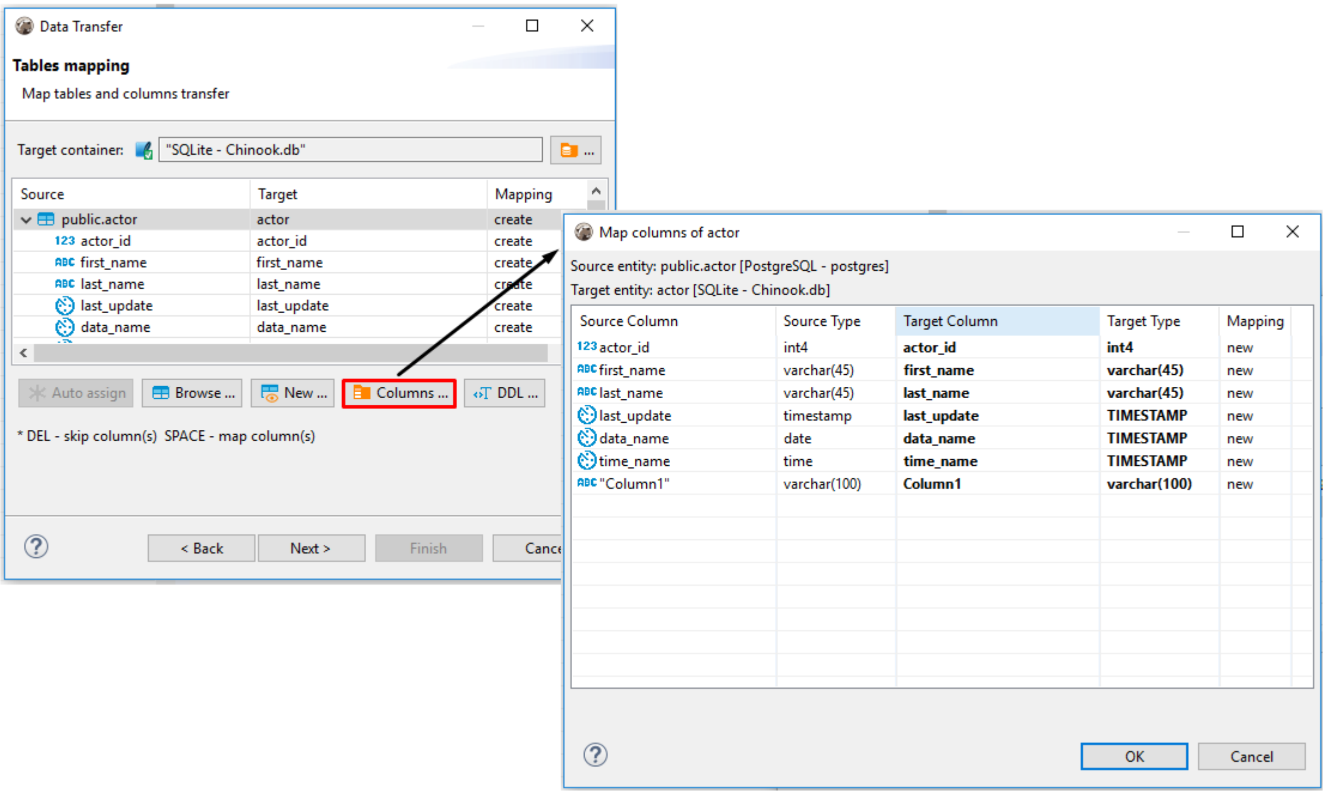
Enter a part of table name here

- Tables
  - Album
  - Artist
  - Customer
  - Employee
  - Genre
  - Invoice
  - InvoiceLine
  - MediaType
  - Playlist
  - PlaylistTrack
  - Track
  - film_category
  - foo
  - inventory
  - language
- Views

To define mapping setting for a column in a target table, release the list of source table columns by pressing > next to the table's name in the **Source** column, then click the name of the target column and enter a new one or select one from the dropdown list. To collaps the list, press ⌄

If you want tables of the target container to be named like those of source, press the **Auto assign** button ✱ Auto assign and the **Target** column will be automatically populated.

You can also define the names of target columns, as well as their data types, by clicking a row with a table name and pressing the **Columns** button Columns ...



Tables mapping

Map tables and columns transfer

Target container: "SQLite - Chinook.db"

Source	Target	Mapping
public.actor	actor	create
actor_id	actor_id	create
first_name	first_name	create
last_name	last_name	create
last_update	last_update	create
data_name	data_name	create

* DEL - skip column(s) SPACE - map column(s)

Source entity: public.actor [PostgreSQL - postgres]

Target entity: actor [SQLite - Chinook.db]

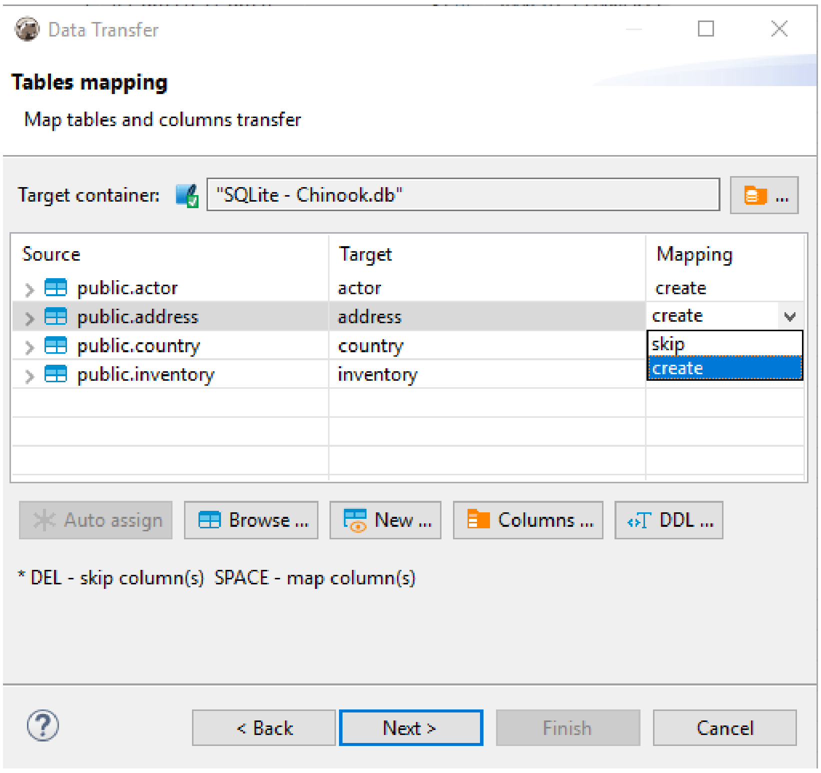
Source Column	Source Type	Target Column	Target Type	Mapping
actor_id	int4	actor_id	int4	new
first_name	varchar(45)	first_name	varchar(45)	new
last_name	varchar(45)	last_name	varchar(45)	new
last_update	timestamp	last_update	TIMESTAMP	new
data_name	date	data_name	TIMESTAMP	new
time_name	time	time_name	TIMESTAMP	new
"Column1"	varchar(100)	Column1	varchar(100)	new

The following elements are available here:

- **Source column** -this column contains names of columns existing in the selected source table;
- **Source type** - this column contains the list of data types assigned to the columns in the selected source table;
- **Target column** - this column contains names of target table columns where the data from source column will be transferred to. To change the name, click the cell and enter a new name.
- **Target type** - this column contains the list of data types that will be assigned to the columns in the target table.

  IMPORTANT: Sometimes data types that are supported on the source database are not supported on the target or vice versa. To set a data type for a column in a target table, click the cell in the **Target Type** column and select one from the dropdown list of data types supported on the target.
- **Mapping** - this column contains the list of actions to be applied to the data on data transfer.

To change mapping type click a cell in **Mapping** column of **Table mapping dialog box** and select the required mapping type.



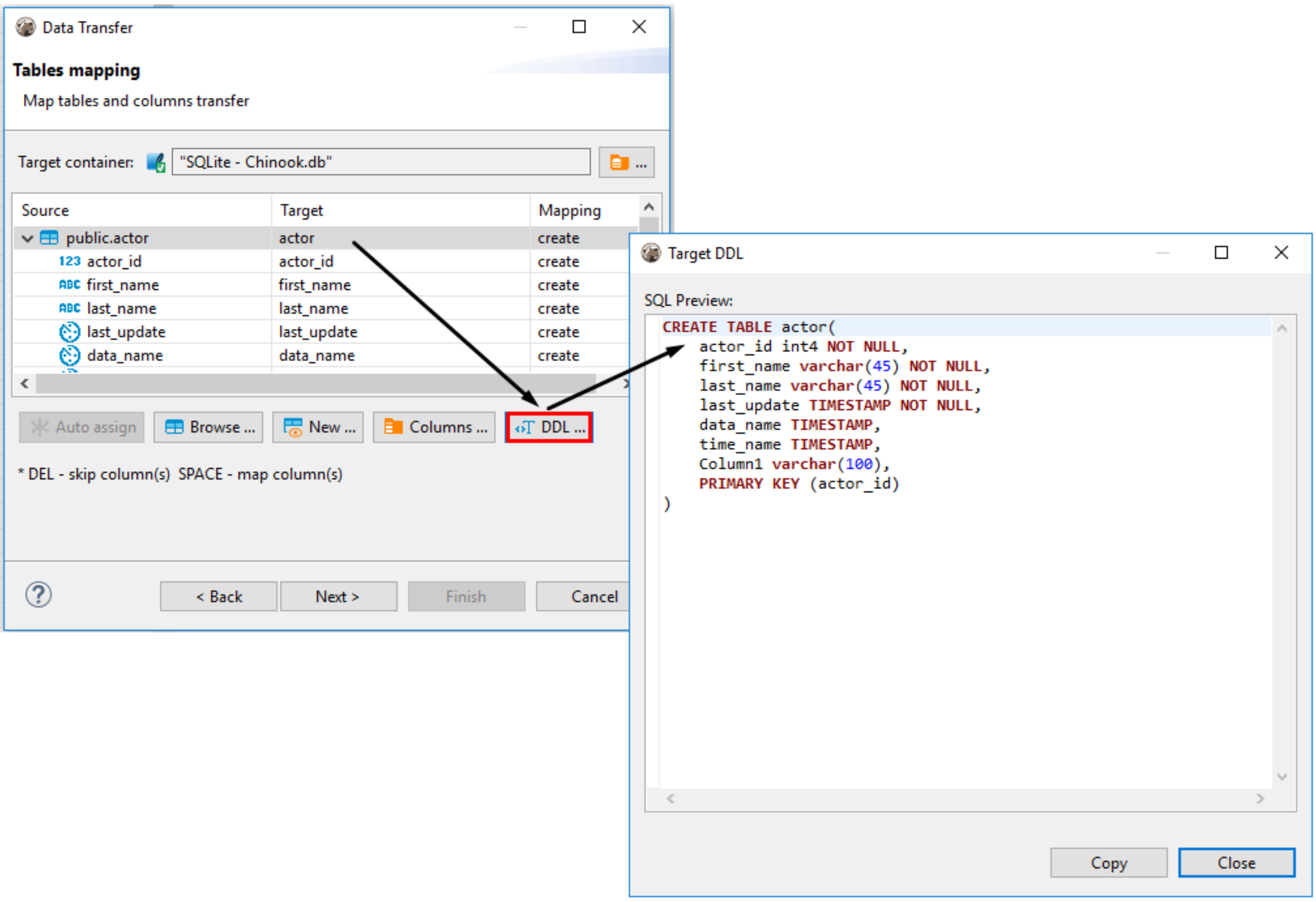
Target container: "SQLite - Chinook.db"

Source	Target	Mapping
public.actor	actor	create
public.address	address	create
public.country	country	
public.inventory	inventory	

skip
create

* DEL - skip column(s) SPACE - map column(s)

You can also view the SQL script that will be run on data transfer by pressing the DDL button DDL ... .



Target container: "SQLite - Chinook.db"

Source	Target	Mapping
public.actor	actor	create
actor_id	actor_id	create
first_name	first_name	create
last_name	last_name	create
last_update	last_update	create
data_name	data_name	create

* DEL - skip column(s) SPACE - map column(s)

```
CREATE TABLE actor(
    actor_id int4 NOT NULL,
    first_name varchar(45) NOT NULL,
    last_name varchar(45) NOT NULL,
    last_update TIMESTAMP NOT NULL,
    data_name TIMESTAMP,
    time_name TIMESTAMP,
    Column1 varchar(100),
    PRIMARY KEY (actor_id)
)
```

The following keyboard shortcuts for easy navigation within the mapping table area of **Table mapping** screen are supported:

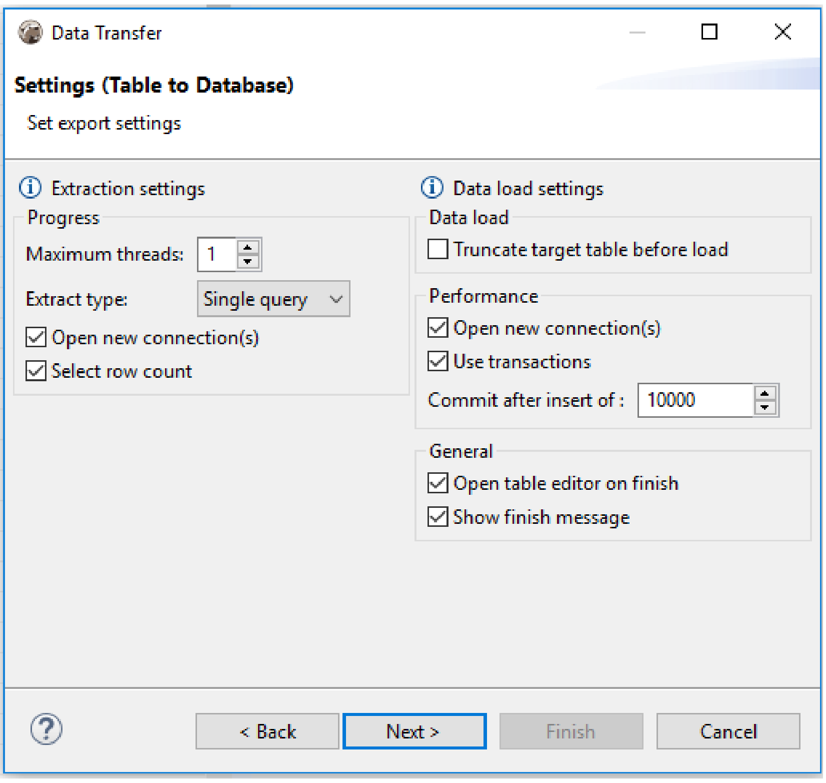
Shortcut	Action
Up	Moves one row up.
Down	Moves one row down.
Right	Releases the list of source table columns.

Shortcut	Action
Left	Swaps the list of source table columns.
Space	Auto-assigns the target.
Del	Sets mapping type to skip.

Configure data mapping and press **Next**.

Step 4: Define export settings

Data export settings are grouped into **Extraction settings** and **Data load settings**.



Extraction Settings

Extraction settings define how the data will be pulled from the source. The following options are available:

Option	Description
Maximum threads	Defines a number of threads to be used for data transfer.
Extract type	Select **Single query** option if your data load is not too big. Select **By segments** option if you need to migrate a solid amount of data. When this options is selected you can set the **Segment size** value, that is to define a number of rows to be transferred in each segment.
Open new connections	If selected, a new connection will be opened and data transfer will not interfere with other calls to the database whose data is being transferred.
Select row count	If selected, a progress bar displaying data migration process will be shown.

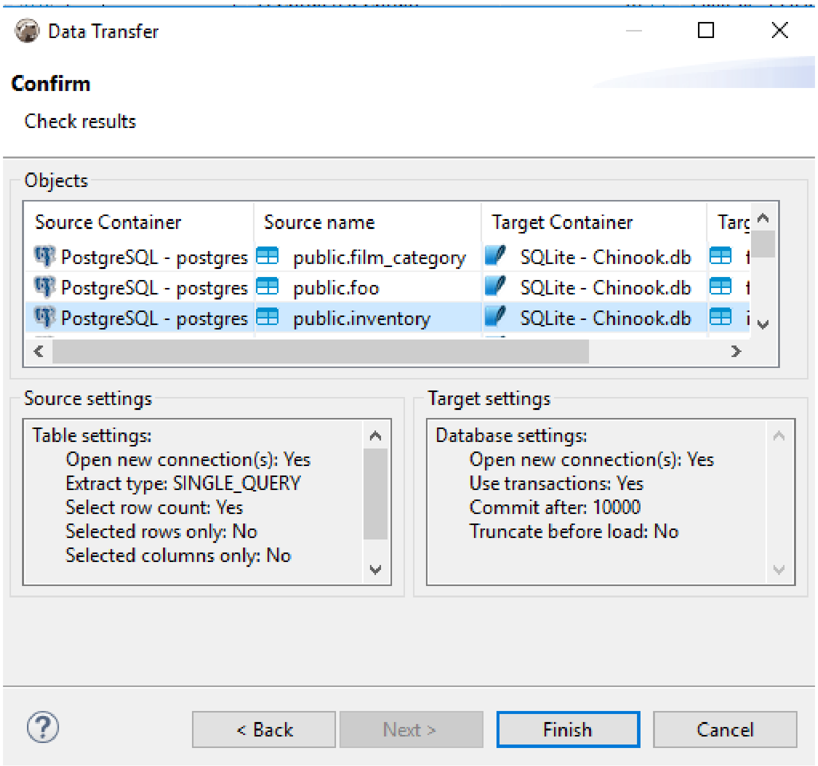
Data load settings

Data load settings define how the extracted data will be pushed to the target. The following options are available.

Option	Description
Truncate data load table before load	Select this check-box only if you want all the data be cleared from the target table. Be very careful with this option!

Option	Description
Open new connections	Use this option to speed up data transfer. If selected, a new connection will be opened and data transfer will not interfere with other calls to the database where data is being transferred to.
Use transactions	This option allows to speed up data transfer and to define the number of rows for each transaction by setting **Commit after insert of** parameter.
Open table editor on finish	If selected, the table editor to be opened when data tanfer is over.
Show finish message	If selected, a notification message will be shown when transfer is over.

Step 5: Confirm



Source Container	Source name	Target Container	Targ
PostgreSQL - postgres	public.film_category	SQLite - Chinook.db	f
PostgreSQL - postgres	public.foo	SQLite - Chinook.db	f
PostgreSQL - postgres	public.inventory	SQLite - Chinook.db	i

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

```
Database settings:
    Open new connection(s): Yes
    Use transactions: Yes
    Commit after: 10000
    Truncate before load: No
```

Check out data transfer settings and press **Finish**.

DBeaver extensions - Office, Debugger, SVG, SSH

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: installation won't work if DBeaver is installed in 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 Administrator/root.

Extension installation in EE

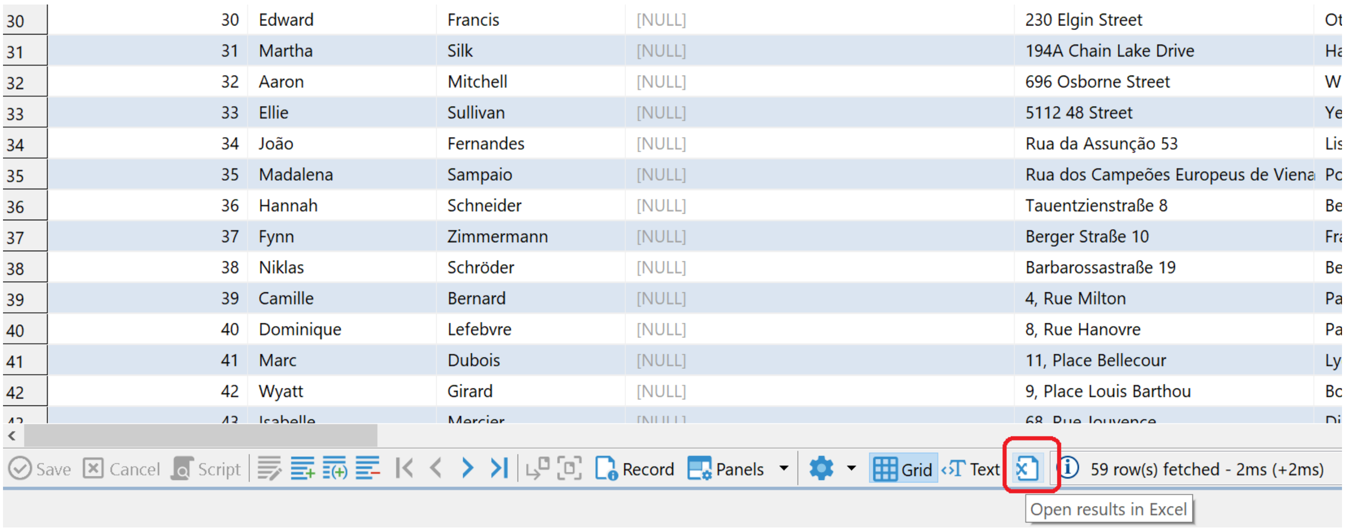
You can do this from online Eclipse Marketplace (open Marketplace UI from main menu). OR you can install it manually the same way as in CE version (see above)

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 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	Ot
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	Pc
36	36	Hannah	Schneider	[NULL]	Tauentzienstraße 8	Be
37	37	Fynn	Zimmermann	[NULL]	Berger Straße 10	Fra
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	Mercier	[NULL]	68, Rue Jouvence	Di

59 row(s) fetched - 2ms (+2ms)

Debugger support

This extension supports SQL debugger in PostgreSQL.
It is included in 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 EE version.

Marketplace URL: https://marketplace.eclipse.org/content/dbeaver-svg-support
P2 repository URL: https://dbeaver.io/update/svg/latest/

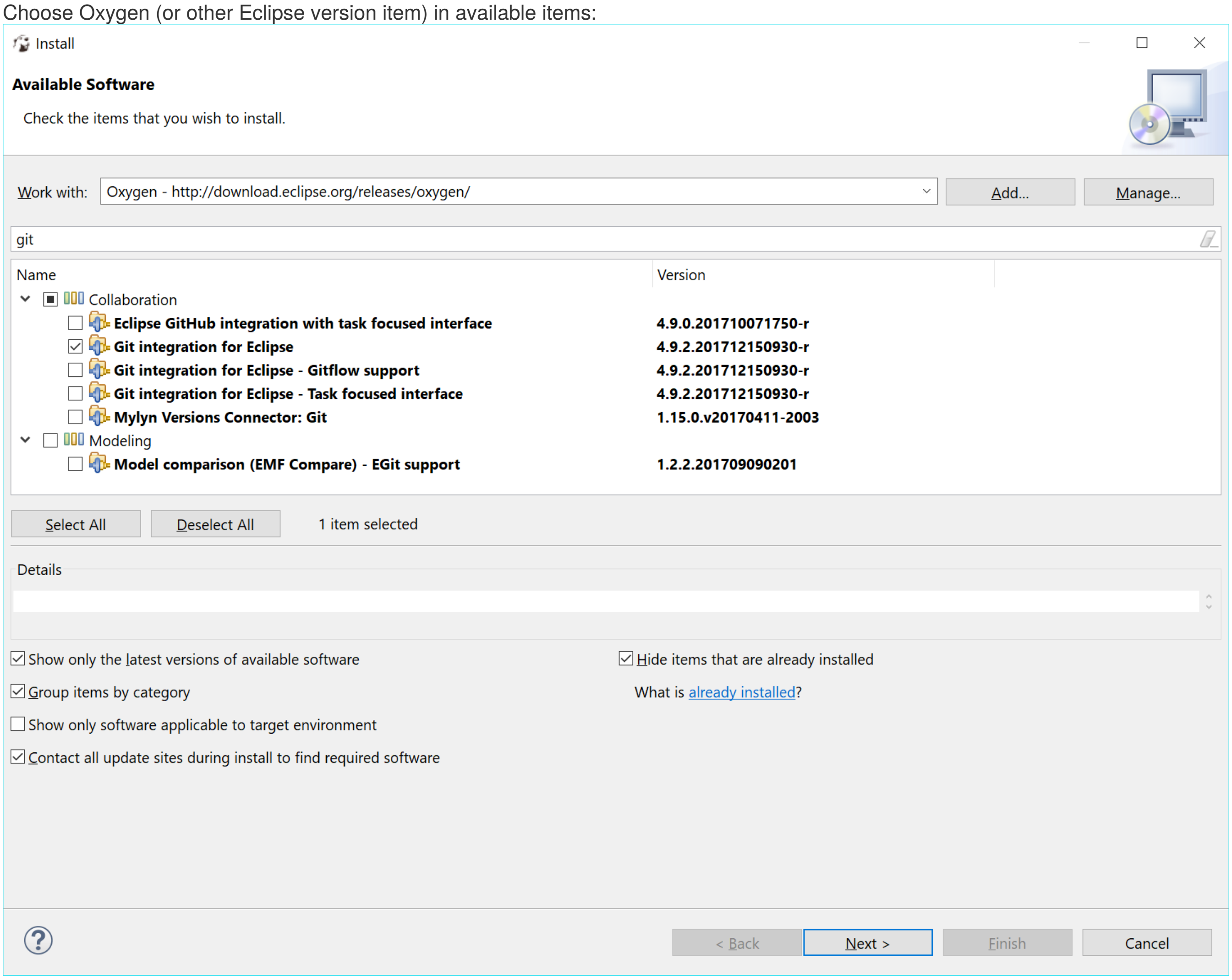
Advanced SSH tunnel libraries

Needed to support more key formats (e.g. ed25519).

P2 repository URL: https://dbeaver.io/update/sshj/latest/

Git integration

Choose Oxygen (or other Eclipse version item) in available items:



Install

Available Software

Check the items that you wish to install.

Work with: Oxygen - http://download.eclipse.org/releases/oxygen/

git

Name	Version
■ Collaboration	
□ Eclipse GitHub integration with task focused interface	4.9.0.201710071750-r
☑ Git integration for Eclipse	4.9.2.201712150930-r
□ Git integration for Eclipse - Gitflow support	4.9.2.201712150930-r
□ Git integration for Eclipse - Task focused interface	4.9.2.201712150930-r
□ Mylyn Versions Connector: Git	1.15.0.v20170411-2003
□ Modeling	
□ Model comparison (EMF Compare) - EGit support	1.2.2.201709090201

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

Installing extensions - Themes, version control, etc

You can install a lot of optional extensions (plugins) in DBeaver. Most of extensions can be found on Eclipse Marketplace website.

Popular extensions for DBeaver

- Darkest Dark theme - the best Dark theme for DBeaver
- Eclipse Color Theme - use it if you don't like Darkest Dark theme by some reason
- Git support - Git version control integration
- Subversion support - Subversion integration
- Embedded Shell - Allows to run shell commands directly from DBeaver
- Editor vertical indents - Adds vertical indents to all text editors

DBeaver-specific extensions

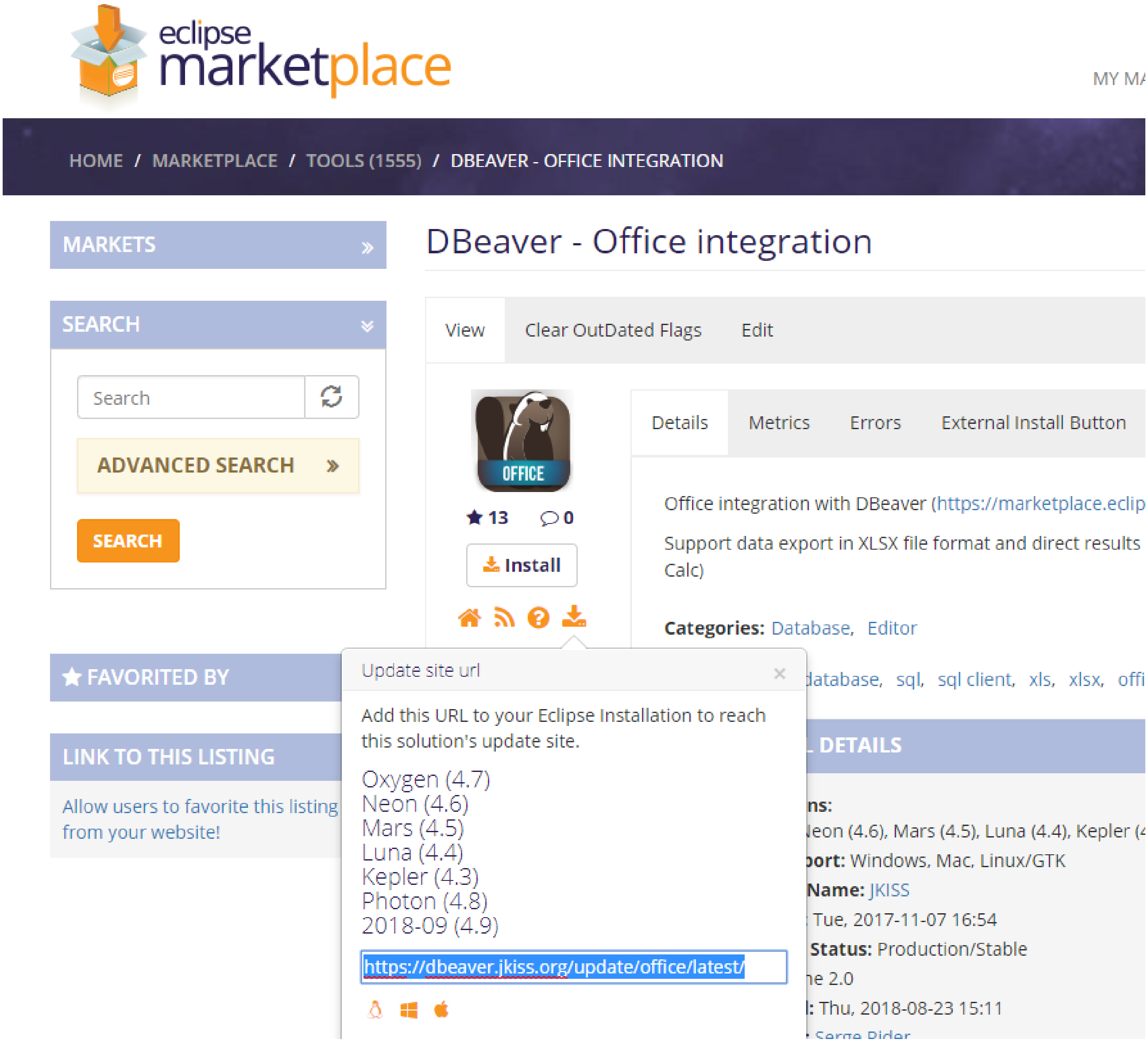
- Office formats support (XLSX)
- Vector graphics support (SVG)
- SSHJ and advanced cryptography
- SQL debugger

Install Process

In DBeaver EE you can use drag-n-drop from Marketplace web site (see button `Install`) into DBeaver main window. This will launch Marketplace installation wizard automatically. In DBeaver Community or other DBeaver-based products which do not include marketplace clients you can use following instructions:

Extension installation in CE version:

1. Copy URL of extension update site:



DBeaver - Office integration

Allow users to favorite this listing from your website!

Office integration with DBeaver (https://marketplace.eclip

Support data export in XLSX file format and direct results Calc)

Categories: Database, Editor

database, sql, sql client, xls, xlsx, offi

Update site url

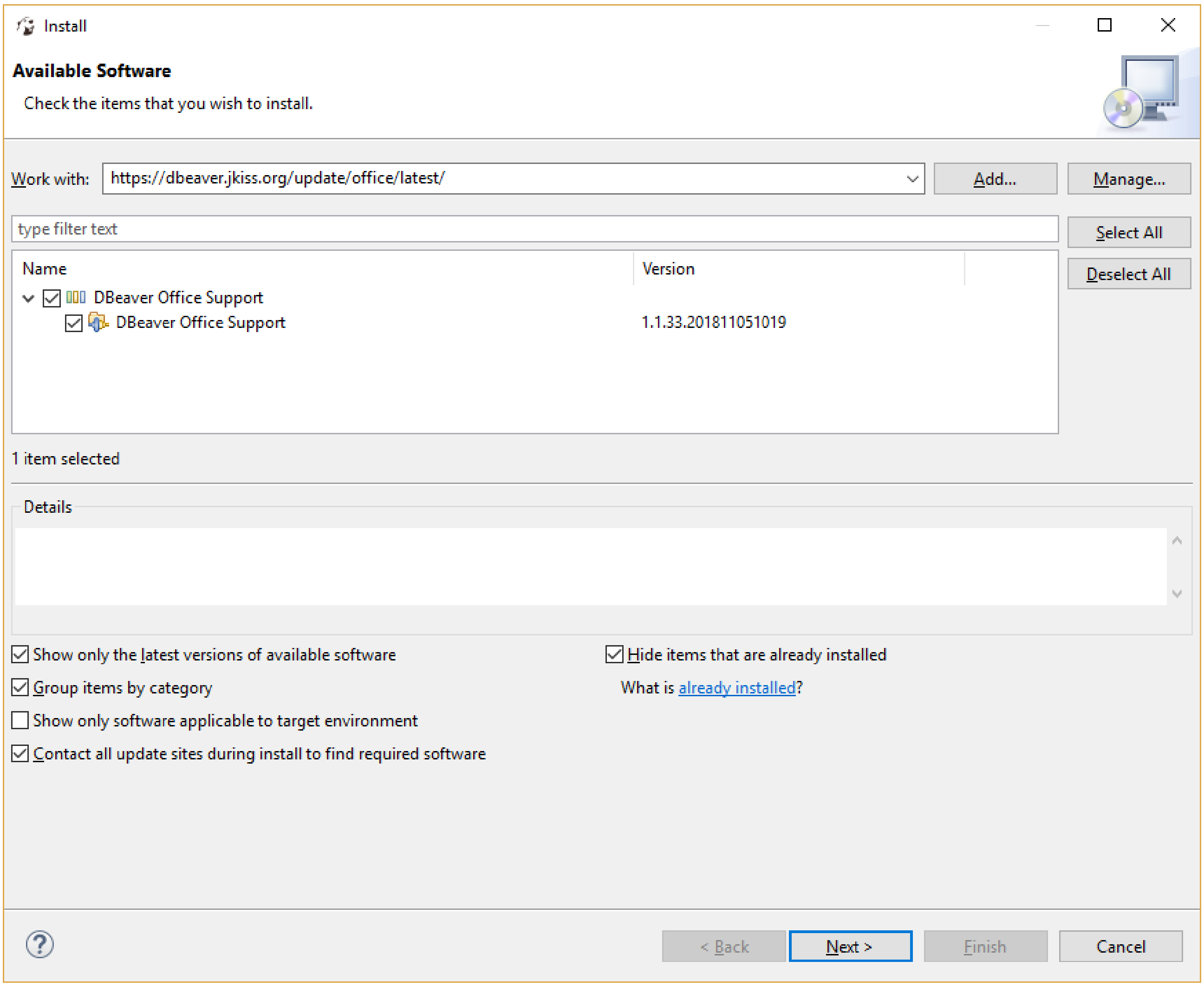
Add this URL to your Eclipse Installation to reach this solution's update site.

Oxygen (4.7)
Neon (4.6)
Mars (4.5)
Luna (4.4)
Kepler (4.3)
Photon (4.8)
2018-09 (4.9)

https://dbeaver.jkiss.org/update/office/latest/

ns: Neon (4.6), Mars (4.5), Luna (4.4), Kepler (4
port: Windows, Mac, Linux/GTK
Name: JKISS
: Tue, 2017-11-07 16:54
Status: Production/Stable
ne 2.0
: Thu, 2018-08-23 15:11
: Serge Rider

2. In 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)


Install

Available Software

Check the items that you wish to install.

Work with: https://dbeaver.jkiss.org/update/office/latest/

Name	Version
☑ DBeaver Office Support	
☑ DBeaver Office Support	1.1.33.201811051019

1 item selected

Details

☑ Show only the latest versions of available software

☑ Group items by category

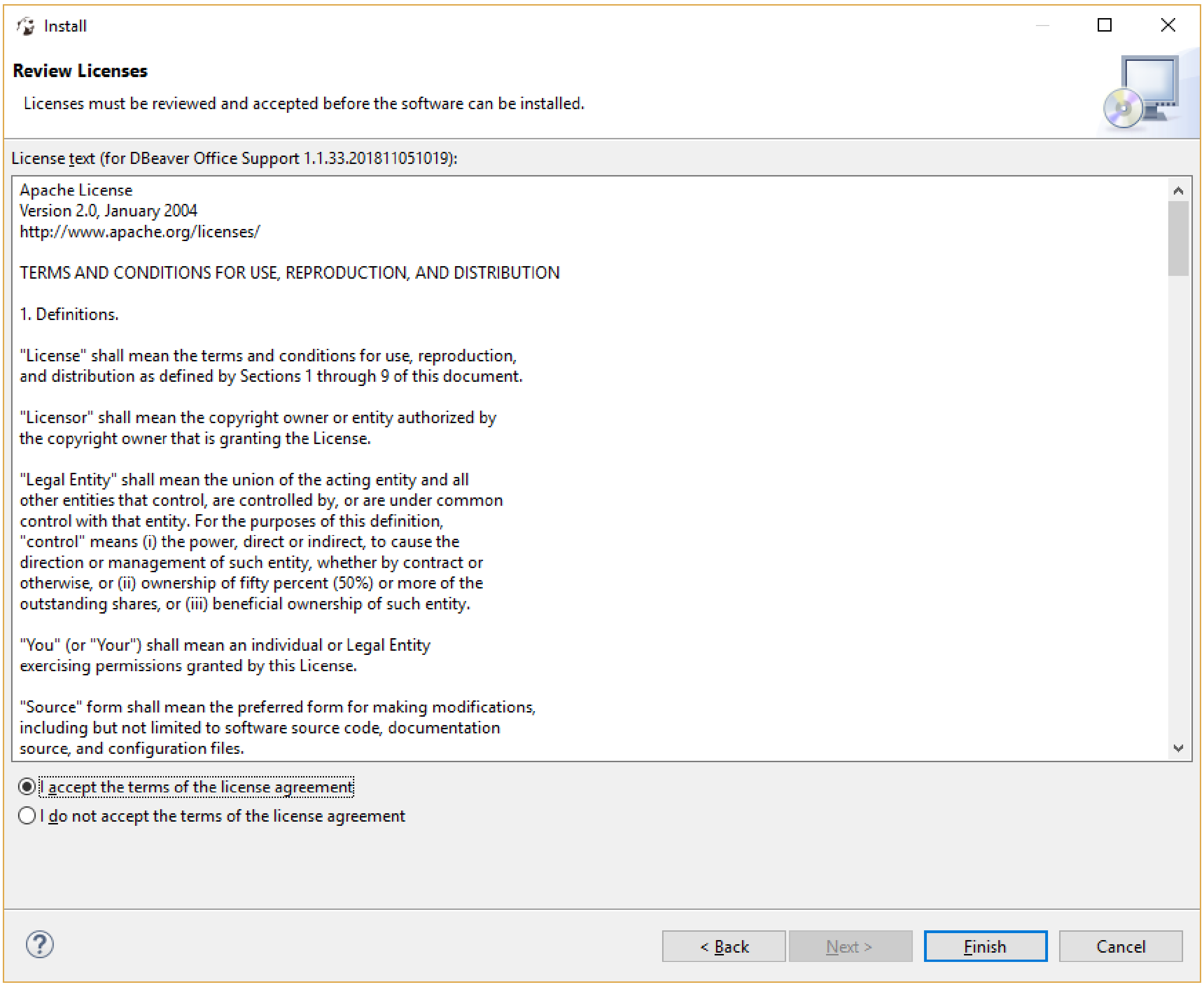
□ Show only software applicable to target environment

☑ Contact all update sites during install to find required software

☑ Hide items that are already installed

What is already installed?

5. Click Next. You may need to accept extension license before installation



Install

Review Licenses

Licenses must be reviewed and accepted before the software can be installed.

License text (for DBeaver Office Support 1.1.33.201811051019):

Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

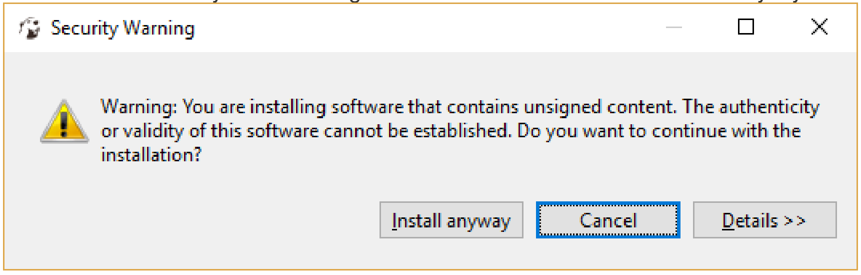
"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

● I accept the terms of the license agreement

○ I do not accept the terms of the license agreement

6. Some extensions may contain unsigned bundles. Install such extensions only if you really trust author.



Security Warning

Warning: You are installing software that contains unsigned content. The authenticity or validity of this software cannot be established. Do you want to continue with the installation?

7. Click Next->Finish. Installation will take some time. Restart DBeaver.

Apache Hive/Spark/Impala

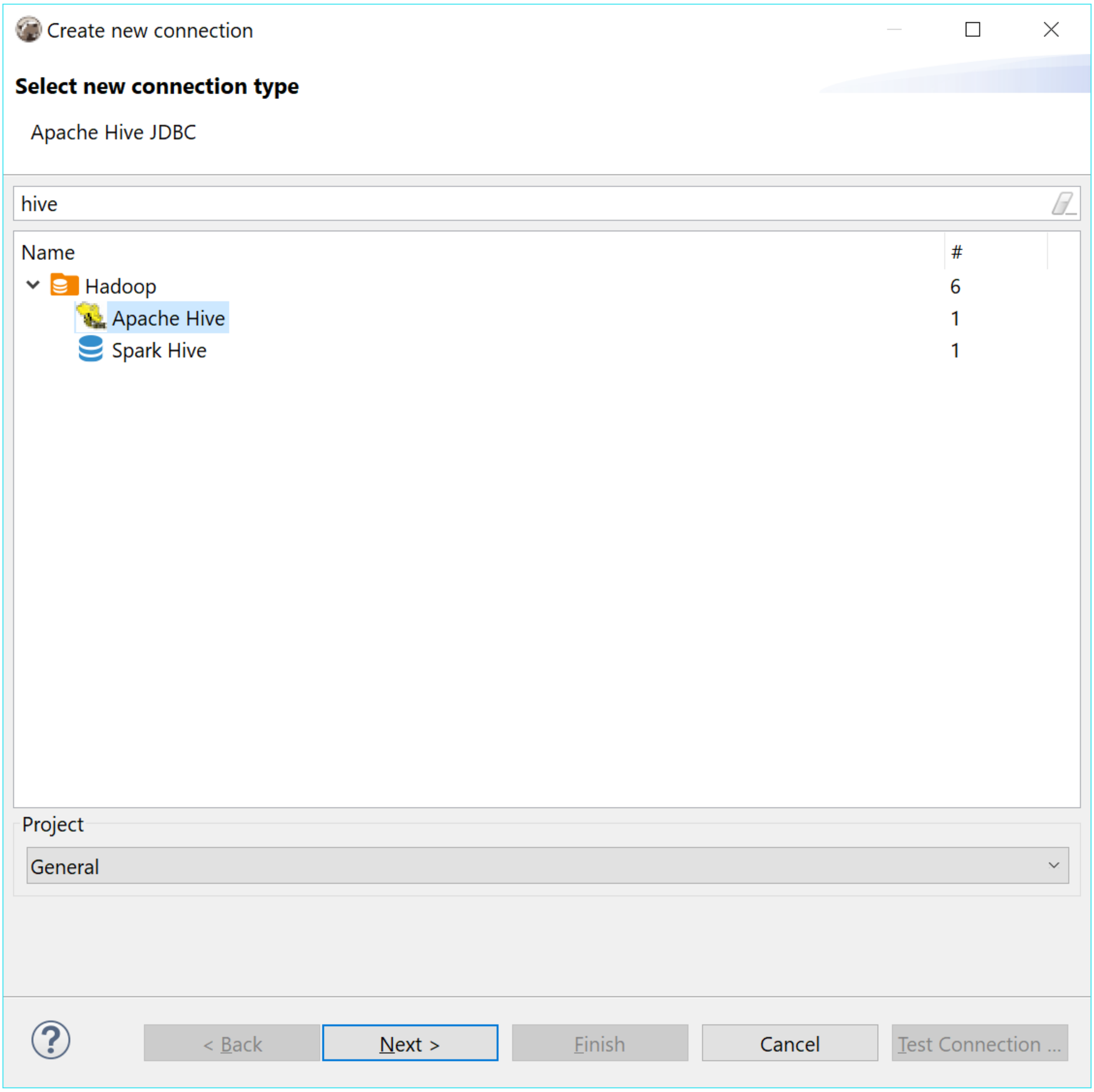
Apache Hive

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 is multiple implementation of storage systems which utilize Hive on server-side - including Apache Spark, Impala, etc. Most of them support standard Hive JDBC driver which is used in DBeaver to communicate with the server.

DBeaver uses so-called Hive JDBC Uber Jar driver (https://github.com/timveil/hive-jdbc-uber-jar) which includes all necessary dependencies. You don't need to download anything - DBeaver will download everything automatically (if you have internet access).

Connection setup



Create new connection

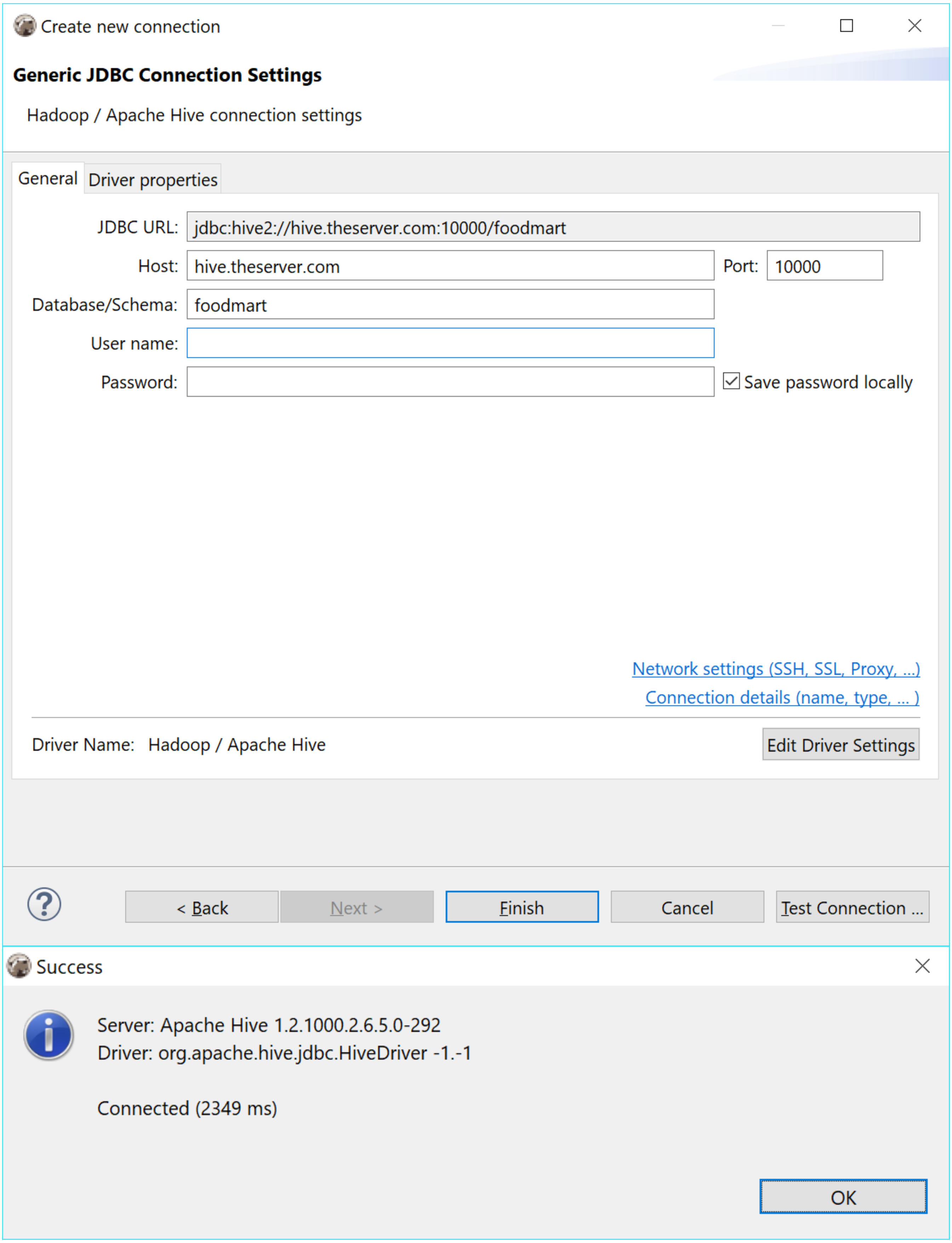
Select new connection type

Apache Hive JDBC

hive

Name	#
Hadoop	6
Apache Hive	1
Spark Hive	1

Project: General



Generic JDBC Connection Settings

Hadoop / Apache Hive connection settings

JDBC URL: jdbc:hive2://hive.theserver.com:10000/foodmart

Host: hive.theserver.com Port: 10000

Database/Schema: foodmart

User name:

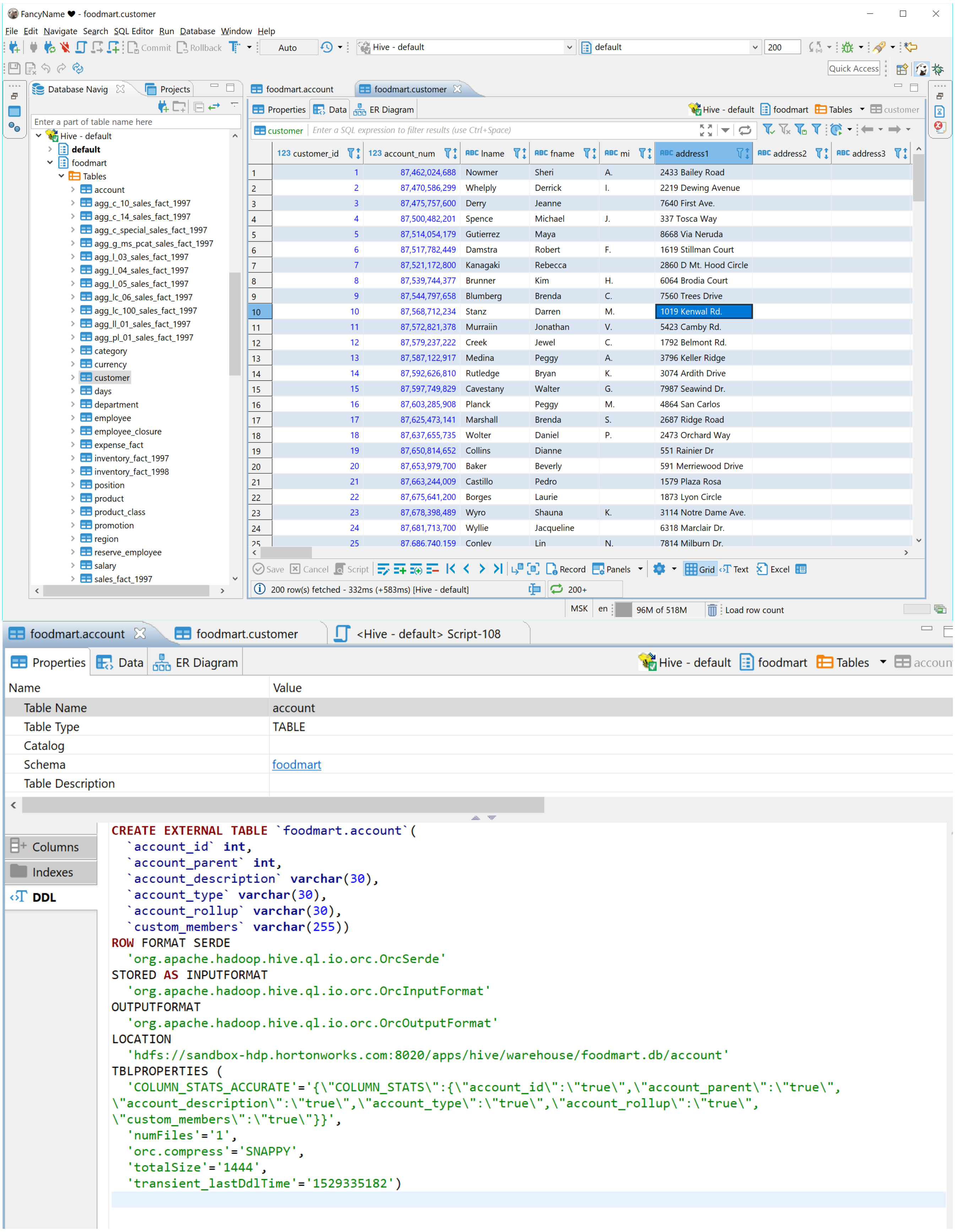
Password: ☑ Save password locally

Driver Name: Hadoop / Apache Hive

Server: Apache Hive 1.2.1000.2.6.5.0-292
Driver: org.apache.hive.jdbc.HiveDriver -1.-1

Connected (2349 ms)

Schema/data browser



	customer_id	account_num	lname	fname	mi	address1	address2	address3
1	1	87,462,024,688	Nowmer	Sheri	A.	2433 Bailey Road		
2	2	87,470,586,299	Whelply	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	Conley	Lin	N.	7814 Milburn Dr.		

200 row(s) fetched - 332ms (+583ms) [Hive - default]

Name	Value
Table Name	account
Table Type	TABLE
Catalog	
Schema	foodmart
Table Description	

```
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.ql.io.orc.OrcSerde'
STORED AS INPUTFORMAT
  'org.apache.hadoop.hive.ql.io.orc.OrcInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.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

Hie doesn't support referential integrity so you won't see primary keys or foreign keys. ER diagrams also don't make much sense.

Cassandra

Note: Cassandra extension is available only in Enterprise-Edition version.

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 doesn't support any kind of referential integrity. There are no foreign keys, references, etc. Note that Cassandra has 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.

Executing CQL

CQL (Cassandra Query Language) is a kind of very simple SQL language dialect. It supports simple SELECT queries, DDL statements (like CREATE TABLE) and some other. http://cassandra.apache.org/doc/4.0/cql/ You can use standard DBeaver SQL editor to execute CQL queries.

ERD

Physical ERD (Entity Relation Diagram) doesn't make much sense for Cassandra as there are no any foreign keys. However you can make you own custom ERD and connect actual Cassandra table with each other using logical associations.

Clickhouse

Yandex Clickhouse

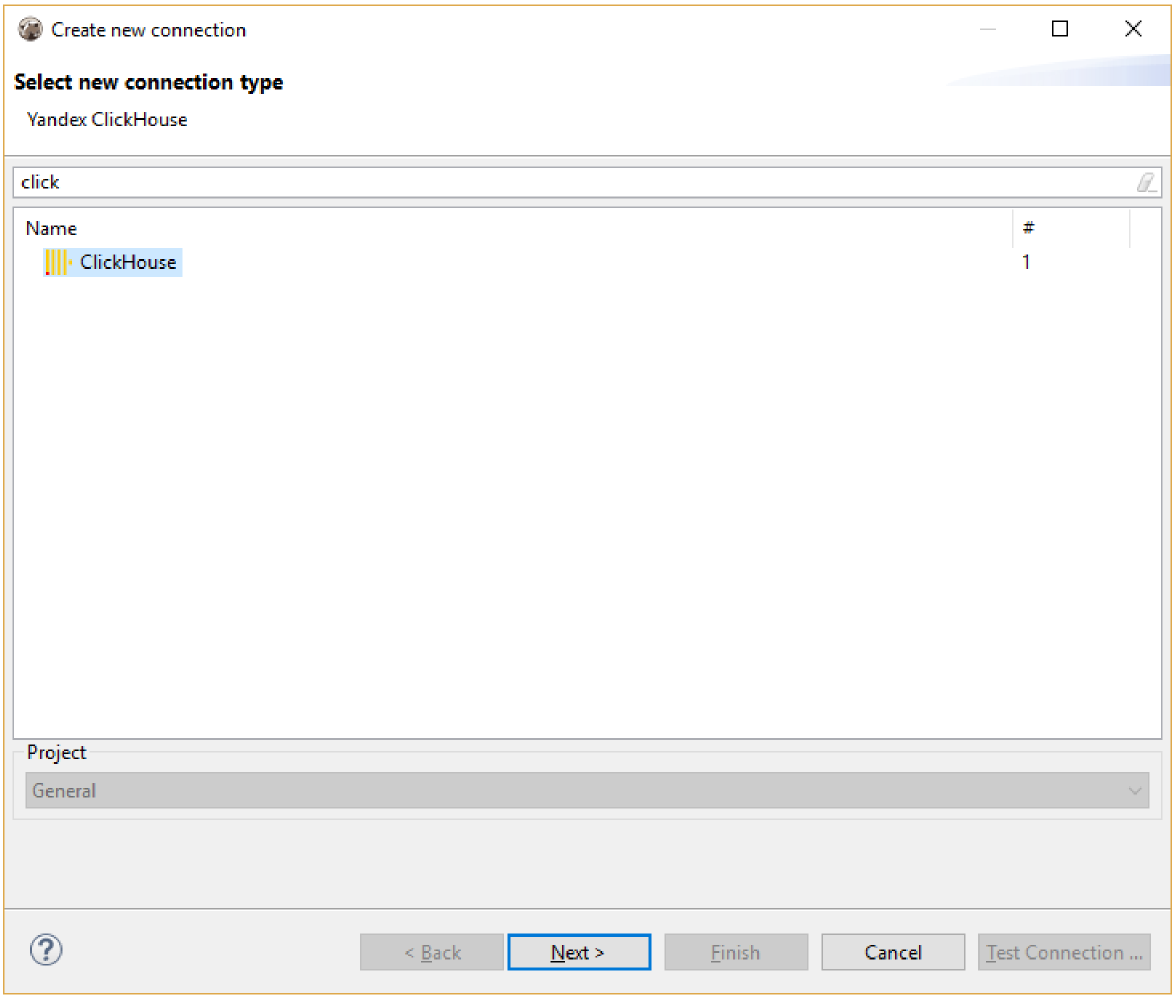
ClickHouse is an open source column-oriented database management system capable of real time generation of analytical data reports using SQL queries.

DBeaver uses standard Clickhouse JDBC driver to communicate with Clickhouse servers. Driver is downloaded automatically once you establish connection with database server.

Connection setup

Connection initiation is very easy.

1. Select Clickhouse driver:



Create new connection

Select new connection type

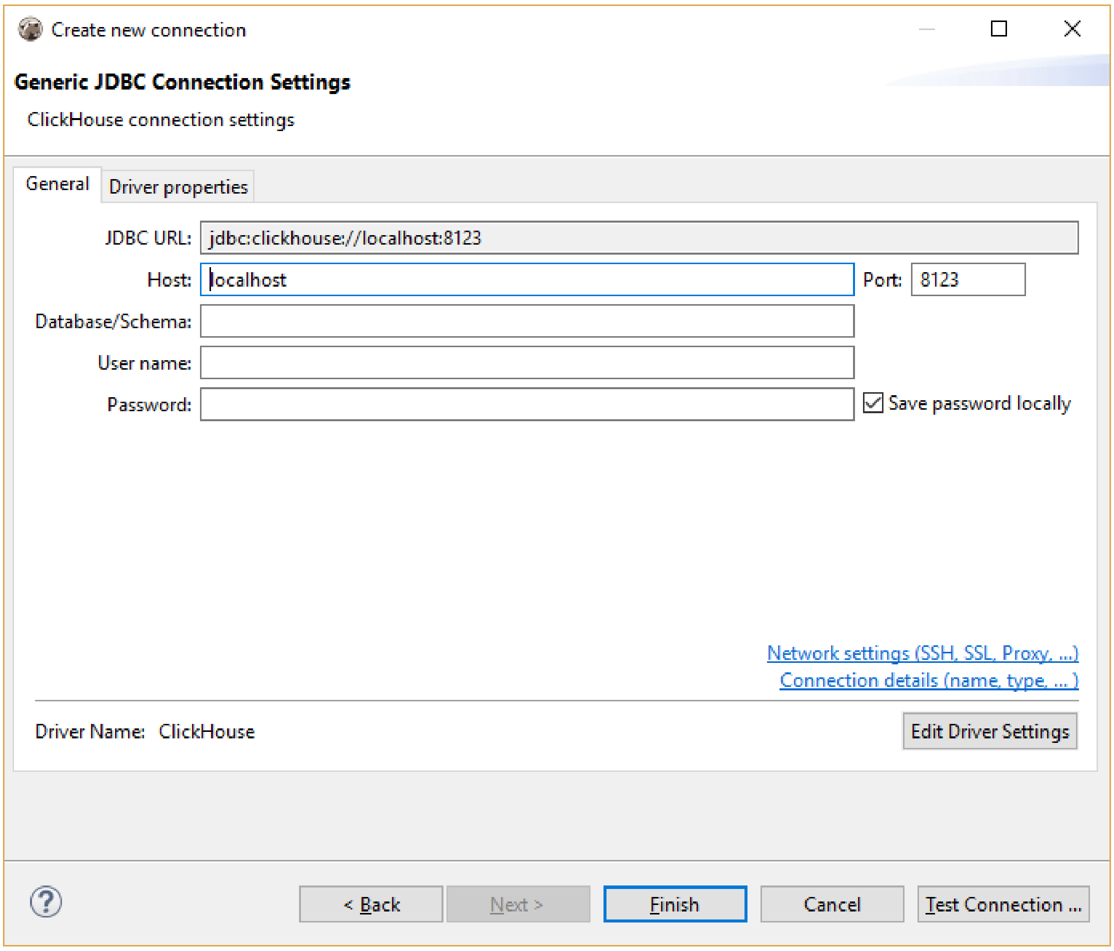
Yandex ClickHouse

click

Name	#
ClickHouse	1

Project: General

2. The only required connection parameter is host name.



Create new connection

Generic JDBC Connection Settings

ClickHouse connection settings

General | Driver properties

JDBC URL: jdbc:clickhouse://localhost:8123

Host: localhost   Port: 8123

Database/Schema:

User name:

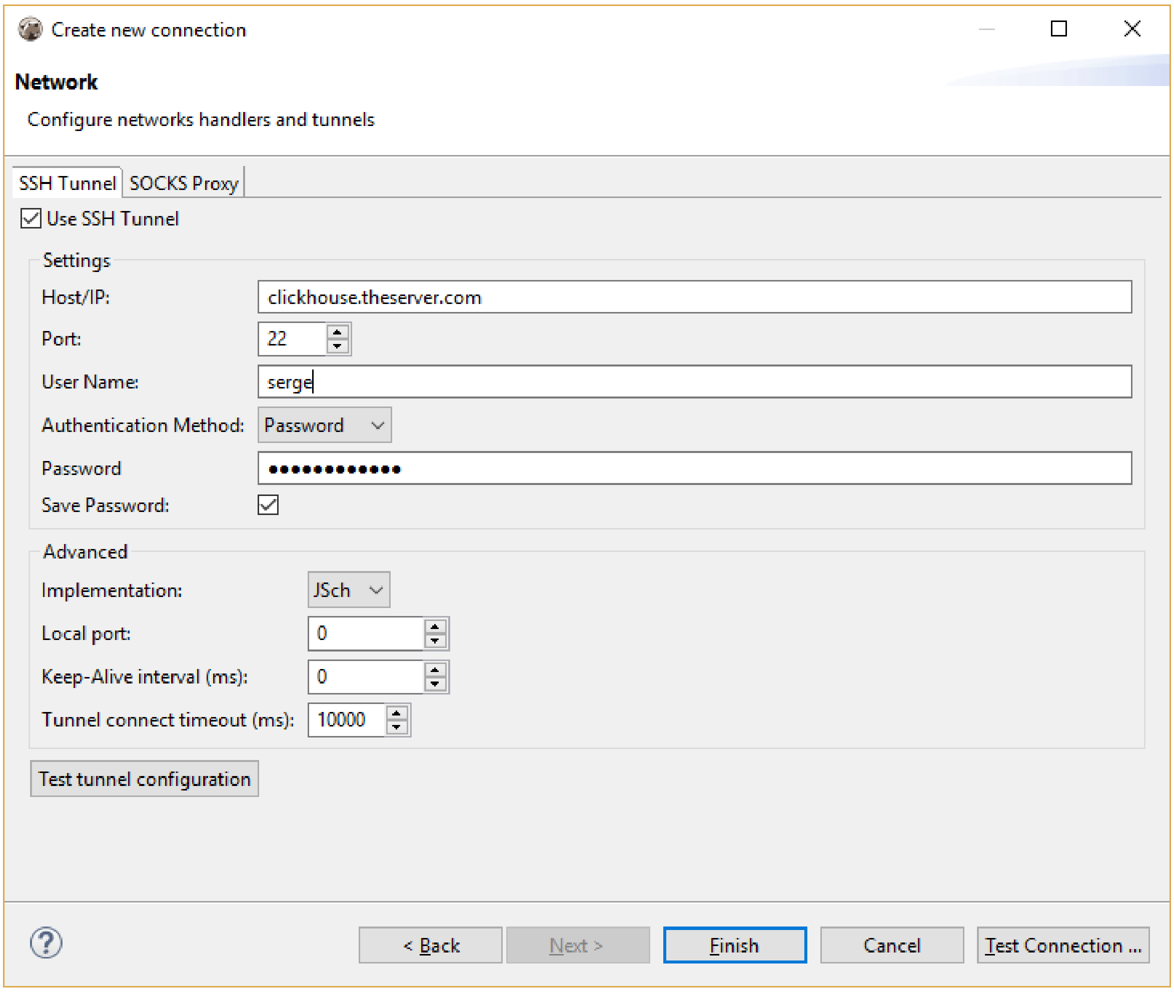
Password: ☑ Save password locally

Network settings (SSH, SSL, Proxy, ...)

Connection details (name, type, ...)

Driver Name: ClickHouse

3. You can configure SSH tunnel to access your server. In that case set database host name to `localhost` while real Clickhouse server host will be specified as SSH server.



Create new connection

Network

Configure networks handlers and tunnels

SSH Tunnel | SOCKS Proxy

☑ Use SSH Tunnel

Settings

Host/IP: clickhouse.theserver.com

Port: 22

User Name: serge

Authentication Method: Password

Password: ●●●●●●●●●●●●

Save Password: ☑

Advanced

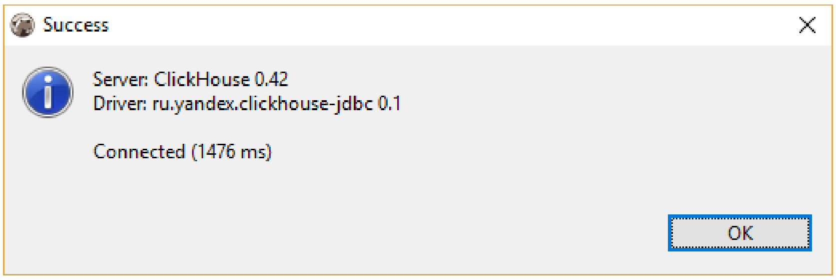
Implementation: JSch

Local port: 0

Keep-Alive interval (ms): 0

Tunnel connect timeout (ms): 10000

4. Test connection:

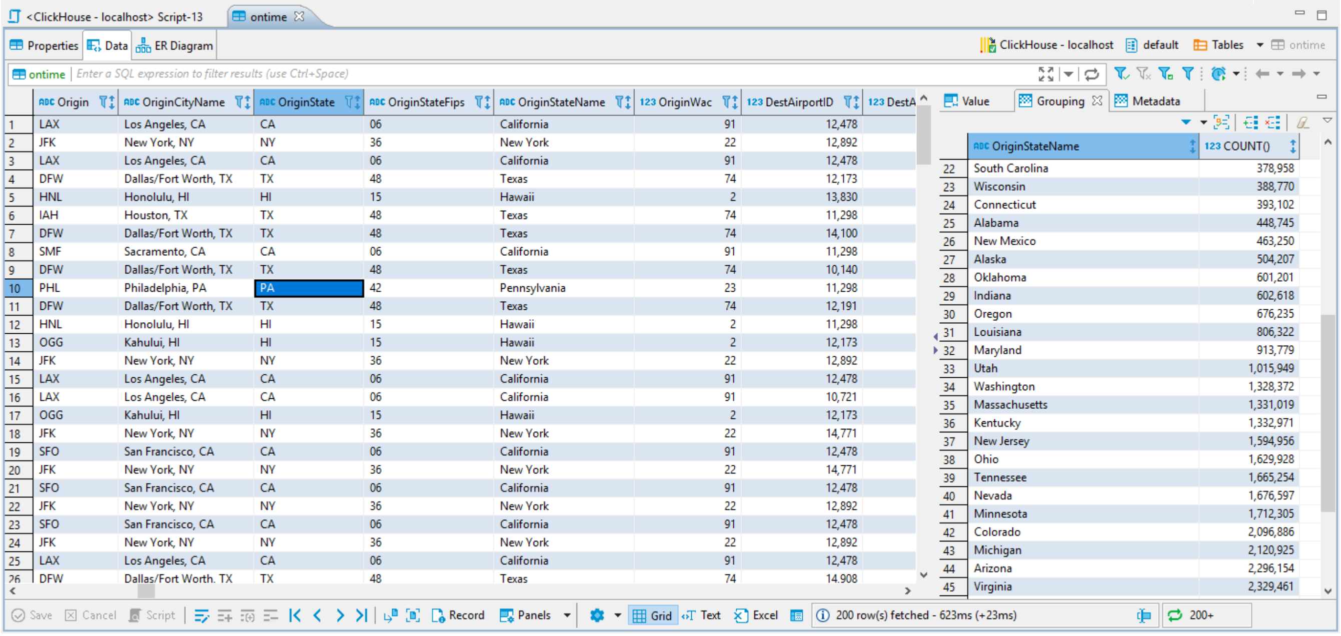


Server: ClickHouse 0.42
Driver: ru.yandex.clickhouse-jdbc 0.1

Connected (1476 ms)

Schema/data browser

You can browse/edit, analyse data in Clickhouse tables:

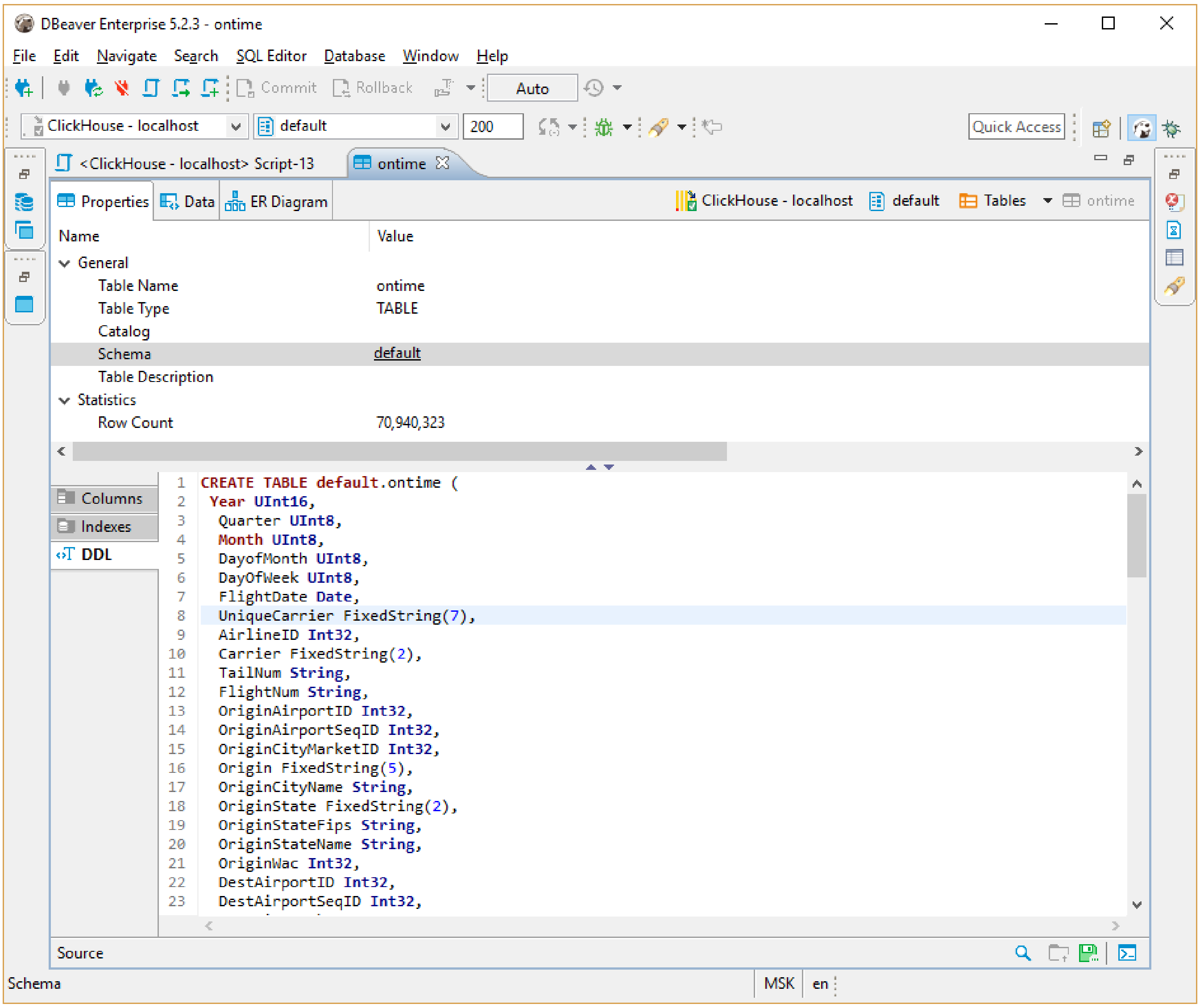


	Origin	OriginCityName	OriginState	OriginStateFips	OriginStateName	OriginWac	DestAirportID	DestA
1	LAX	Los Angeles, CA	CA	06	California	91	12,478	
2	JFK	New York, NY	NY	36	New York	22	12,892	
3	LAX	Los Angeles, CA	CA	06	California	91	12,478	
4	DFW	Dallas/Fort Worth, TX	TX	48	Texas	74	12,173	
5	HNL	Honolulu, HI	HI	15	Hawaii	2	13,830	
6	IAH	Houston, TX	TX	48	Texas	74	11,298	
7	DFW	Dallas/Fort Worth, TX	TX	48	Texas	74	14,100	
8	SMF	Sacramento, CA	CA	06	California	91	11,298	
9	DFW	Dallas/Fort Worth, TX	TX	48	Texas	74	10,140	
10	PHL	Philadelphia, PA	PA	42	Pennsylvania	23	11,298	
11	DFW	Dallas/Fort Worth, TX	TX	48	Texas	74	12,191	
12	HNL	Honolulu, HI	HI	15	Hawaii	2	11,298	
13	OGG	Kahului, HI	HI	15	Hawaii	2	12,173	
14	JFK	New York, NY	NY	36	New York	22	12,892	
15	LAX	Los Angeles, CA	CA	06	California	91	12,478	
16	LAX	Los Angeles, CA	CA	06	California	91	10,721	
17	OGG	Kahului, HI	HI	15	Hawaii	2	12,173	
18	JFK	New York, NY	NY	36	New York	22	14,771	
19	SFO	San Francisco, CA	CA	06	California	91	12,478	
20	JFK	New York, NY	NY	36	New York	22	14,771	
21	SFO	San Francisco, CA	CA	06	California	91	12,478	
22	JFK	New York, NY	NY	36	New York	22	12,892	
23	SFO	San Francisco, CA	CA	06	California	91	12,478	
24	JFK	New York, NY	NY	36	New York	22	12,892	
25	LAX	Los Angeles, CA	CA	06	California	91	12,478	
26	DFW	Dallas/Fort Worth, TX	TX	48	Texas	74	14,908	

	OriginStateName	COUNT()
22	South Carolina	378,958
23	Wisconsin	388,770
24	Connecticut	393,102
25	Alabama	448,745
26	New Mexico	463,250
27	Alaska	504,207
28	Oklahoma	601,201
29	Indiana	602,618
30	Oregon	676,235
31	Louisiana	806,322
32	Maryland	913,779
33	Utah	1,015,949
34	Washington	1,328,372
35	Massachusetts	1,331,019
36	Kentucky	1,332,971
37	New Jersey	1,594,956
38	Ohio	1,629,928
39	Tennessee	1,665,254
40	Nevada	1,676,597
41	Minnesota	1,712,305
42	Colorado	2,096,886
43	Michigan	2,120,925
44	Arizona	2,296,154
45	Virginia	2,329,461

200 row(s) fetched - 623ms (+23ms)

DBeaver supports native Clickhouse table DDLs:


DBeaver Enterprise 5.2.3 - ontime
File Edit Navigate Search SQL Editor Database Window Help
Commit Rollback Auto
ClickHouse - localhost default 200
Quick Access
<ClickHouse - localhost> Script-13
ontime
Properties Data ER Diagram
ClickHouse - localhost default Tables ontime
Columns
Indexes
DDL
Source
Schema
MSK en

Name	Value
General	
Table Name	ontime
Table Type	TABLE
Catalog	
Schema	default
Table Description	
Statistics	
Row Count	70,940,323

```
CREATE TABLE default.ontime (
 Year UInt16,
  Quarter UInt8,
  Month UInt8,
  DayofMonth UInt8,
  DayOfWeek UInt8,
  FlightDate Date,
  UniqueCarrier FixedString(7),
  AirlineID Int32,
  Carrier FixedString(2),
  TailNum String,
  FlightNum String,
  OriginAirportID Int32,
  OriginAirportSeqID Int32,
  OriginCityMarketID Int32,
  Origin FixedString(5),
  OriginCityName String,
  OriginState FixedString(2),
  OriginStateFips String,
  OriginStateName String,
  OriginWac Int32,
  DestAirportID Int32,
  DestAirportSeqID Int32,
```

Clickhouse extension support most of standard DBeaver features (SQL editor, data view/edit, data transfer, mock data generation, etc).

Limitations

Clickhouse doesn't support referential integrity so you won't see foreign keys. ER diagrams are also doesn't make much sense.

Command-Line

Command line parameters

Command line parameters might be passed directly to dbeaver[.exe] executable.
On Windows you also can use `dbeaver-cli.exe` executable (it doesn't spawn new window so you can see output messages).

Also you can add parameters in the `dbeaver.ini` configuration file (in the beginning, each parameter on its own line).

DBeaver control

Name	Value	Example
-help	Prints help message	
-stop	Quits DBeaver	
-dump	Prints DBeaver thread dump	
-f	Opens file in DBeaver UI	`-f c:\some-path\some-file.sql`
-con	Opens database connection in DBeaver UI	See connection parameters table
-closeTabs	Closes all open editor tabs	
-disconnectAll	Closes all open connections	
-reuseWorkspace	Force reuse of single workspace by multiple DBeaver instances	
-newInstance	Force new DBeaver instance creation (do not try to reuse already running one)	

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 version upgrade.	
-vmargs	VM parameters	See VM arguments table

VM arguments

You can pass any advanced Java parameters supported by your local JVM (Oracle, OpenJDK, IBM, etc).
Parameters supported by Oracle JVM (1.8): https://docs.oracle.com/javase/8/docs/technotes/tools/windows/java.html

Parameters supported by all JVMs:

Name	Value	Example
-Xms	Sets initial memory available for DBeaver	`-Xmx1000m`
-Xmx	Sets maximum memory available for DBeaver	`-Xmx4000m`

Connection parameters

All connection parameters must be supplied as a single command line argument, parameters are divided by pipe (`|`). Parameter name and value are divided by `=`.
Example: -con driver=sqlite|database=C:\db\SQLite\Chinook.db|name=SQLiteChin|openConsole=true|folder=SQLite

Name	Value	Example

Name	Value	Example
name	Connection name	`Test connection`
driver	Driver name or ID	`driver=sqlite` , `driver=mysql` , etc
url	Connection URL. Optional (JDBC URL may be constructed by driver from other parameters)	`url=jdbc:sqlite:C:\db\SQLite\Chinook.db`
host	Database host name (optional)	`host=localhost`
port	Database port number (optional)	`port=1534`
server	Database server name (optional)	`server=myserver`
database	Database name or path (optional)	`database=db-name`
user	User name (optional)	`user=root`
password	User password (optional)	`password=mysecret`
savePassword	Do not ask use password on connect	`savePassword=true`
showSystemObjects	Show/hide system schemas, tables ,etc	`showSystemObjects=true`
showUtilityObjects	Show/hide utility schemas, tables ,etc	`showUtilityObjects=true`
folder	Put new connection in a folder	`folder=FolderName`
autoCommit	Sets connection auto commit flag (default value depends on driver)	`autoCommit=true`
prop.propName	Advanced connection parameters (depend on driver)	`prop.connectTimeout=30`
id	Connection id	`oracle_thin-16a88e815bd-70598e648cedd28c` (useful in conjunction with `create=false`)
connect	Connect to this database	`connect=false`
openConsole	Open SQL console for this database (sets `connect` to true)	`openConsole=true`
create	Create new connection	`create=false` (true by default). If set to false then existing connection configuration will be used. name or id parameter must be specified.

Reset UI settings

Sometimes, usually after multiple version and /or upgrades/incorrect shutdowns DBeaver UI may become corrupted. Extra toolbar elements, missing menu items, broken keyboard shortcuts, broken localization strings and other glitches may happen.

To reset DBeaver UI just delete file file `workbench.xmi` in DBeaver workspace/.metadata. By default workbench.xmi file locations is:

- Windows: `%APPDATA%\DBeaverData\workspace6\.metadata\.plugins\org.eclipse.e4.workbench\workbench.xmi`
- MacOS: `~/Library/DBeaverData/workspace6/.metadata/.plugins/org.eclipse.e4.workbench/workbench.xmi`
- Linux: `$XDG_DATA_HOME/DBeaverData/workspace6/.metadata/.plugins/org.eclipse.e4.workbench/workbench.xmi`

To reset settings:

1. Close DBeaver
2. Delete workbench.xmi from Explorer/Finder or open terminal and run `del` (Windows) or `rm` (Linux/MacOS) followed by workbench.xmi path.
3. Start DBeaver

Reset workspace

Sometimes (especially after multiple DBeaver versions upgrade) workspace become messy.
Some keyboard shortcuts may stop working, toolbars layout may be broken, etc, etc.
To reset all UI settings (this includes menus, shortcuts, view and toolbar layouts):

1. Shutdown DBeaver
2. Go to the default workspace folder `.metadata\.plugins\org.eclipse.e4.workbench\`
    - Windows: `Win+R`, enter `%APPDATA%\DBeaverData\workspace6\.metadata\.plugins\org.eclipse.e4.workbench\`
    - MacOS: `open ~/Library/DBeaverData/workspace6/.metadata/.plugins/org.eclipse.e4.workbench/`
    - Linux: `cd $XDG_DATA_HOME/DBeaverData/workspace6/.metadata/.plugins/org.eclipse.e4.workbench/`
3. Delete file `workbench.xmi`
4. Start DBeaver

If that doesn't help then you can try to remove `.metadata` folder (see location above). This will erase all your UI settings and SQL scripts configurations (but all connections and scripts will remain). Do it only if nothing else helps!

That's it.

Posting issues

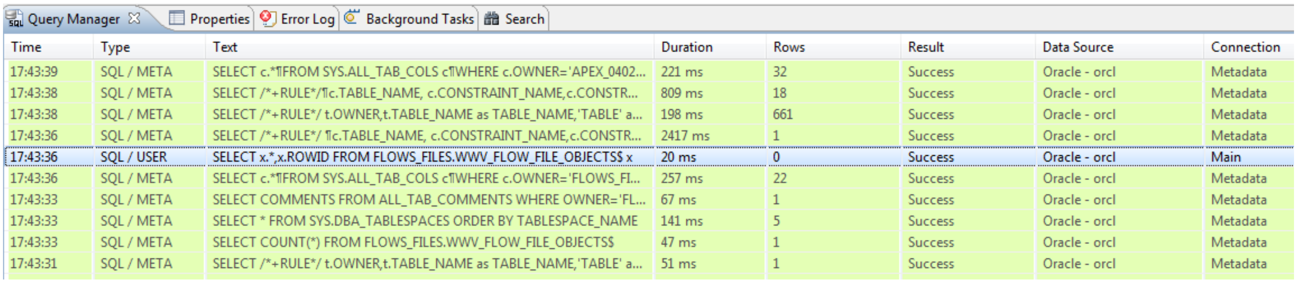
A few tips.

- Check existing issues for your issue (including closed ones). Duplicating an issue is slower for both parties so search through open and closed issues to see if what you’re running into has been addressed already.

- 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 - then try to find them in log files.

- If your issue is related to a 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 c¶WHERE c.OWNER='APEX_0402...	221 ms	32	Success	Oracle - orcl	Metadata
17:43:38	SQL / META	SELECT /*+RULE*/¶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*/ ¶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 c¶WHERE c.OWNER='FLOWS_FI...	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

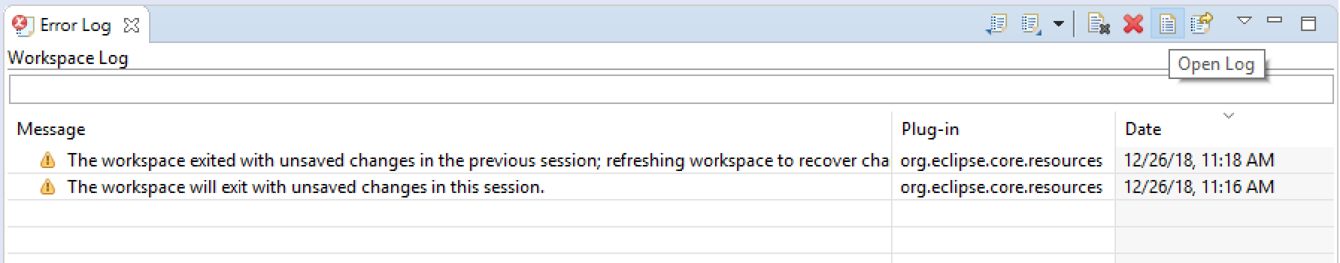
- Depending on the nature of your bug report provide information about:
  - Operating system
  - Window manager (for Linux)
  - Database (name and version)
  - Database driver (name and version)

- Do not write issue type in the issue title (like Feature Request:, Bug: etc). We'll review your issue and assign a corresponding label.

Log files

Error Log view

There is Error Log view (main menu Window->Show View->Error Log) which contains all errors occurred during DBeaver runtime.
You can double click on warning/error in the log viewer and see error stacktrace. Please attache it to the bug report.
Also you can open full log (all error messages) if you need:



Workspace Log

Message	Plug-in	Date
⚠ The workspace exited with unsaved changes in the previous session; refreshing workspace to recover cha	org.eclipse.core.resources	12/26/18, 11:18 AM
⚠ The workspace will exit with unsaved changes in this session.	org.eclipse.core.resources	12/26/18, 11:16 AM

Log files

DBeaver writes different log files. Most of them are Eclipse logs.
Usually log files reside in the workspace/workspace6/.metadata .

- On Windows open Explorer and paste path `%APPDATA%\DBeaverData\workspace6\.metadata`.
- On Linux just type `cd $XDG_DATA_HOME/DBeaverData/workspace6/.metadata`
- On 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 happens 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. Special case is an emergency situation when DBeaver can't start and there is no workspace. Two typical places to find emergency logs:

- `<install-path>/configuration`
- `${HOME}/.eclipse/org.jkiss.dbeaver.product_<dbeaver-version>`

If you are reporting about some error please attach logs (not complete file but valuable part of it).
Logs are very useful, big number of errors can't be reproduced/fixed without full error stacktrace.

Thread dump

Sometimes (due to some bug) DBeaver UI hangs, freezes or works incorrectly. Usually it is impossible to find the reason of such problem without thread dump. Thread dump is the information about internal execution state of Java program. To get thread dump:

Mac and Linux

Run the following on your terminal:

```
jstack $(ps aux | grep -m1 dbeaver | awk '{print $2}') > thread-dump.txt
```

Windows

Just open task manager (CTRL+Escape), find DBeaver in the process list and copy process ID value. On Windows 8+ you need to switch to "Details" tab. Run

```
jstack <PID> > thead-dump.txt
```

in Command Prompt.

Now you can attach thread-dump.txt to the GitHub issue.

Managing connections

This guide describes how to manage/secure DBeaver database connections. It is designed for System administrators. Regular users should check this guide.

Provide predefined connections

DBeaver keeps connections information in project folder. By default all projects reside in workspace.
Default project folder is workspace\workspace6\General.

DBeaver 6.1.3+

DBeaver keeps information about project connections in file `.dbeaver/data-sources.json`.
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 project folder matching `.dbeaver/data-sources*.json` pattern will be loaded on startup. So you can create a file, say, `.dbeaver/data-sources-2.json` in the project folder and DBeaver will see it.

DBeaver < 6.1.3

DBeaver keeps information about project connections in file `dbeaver-data-sources.xml`.

DBeaver can load multiple connection files. Any files in project folder matching `.dbeaver-data-sources*.xml` pattern will be loaded on startup. So you can create a file, say, `.dbeaver-data-sources-2.xml` in the project folder and DBeaver will see it.

Importing connections from CSV/XML

You can import connection from CSV or XML files.

CSV file must have a header row (first line of file) with column names (see list of supported columns below). XML file should contain top-level element and a set of nested elements. Connections config must be specified in attributes of nested elements. Attribute names are the same as 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 specify just URL or set host/port/etc setting.
User name/password are options.

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,dev
Postgre Import XML 2,localhost,5432,,postgres2,jdbc:postgresql://localhost:5432/postgres2,postgres2,postgres2,prod
```

Sample XML

```
<connections>
    <connection name="Postgre Import XML 1" host="localhost" port="5432" server="" database="postgres" url="jdbc:postgresq
l://localhost:5432/postgres" user="postgres" password="postgres" type="dev"/>
    <connection name="Postgre Import XML 2" host="localhost" port="5432" server="" database="postgres" url="jdbc:postgresq
l://localhost:5432/postgres2" user="postgres2" password="postgres2" type="prod"/>
</connections>
```

Secure connections from editing

It is possible to make connection settings read-only (protected by password)

- Generate MD5 hash of your password. You can do it from command line using Linux utility md5sum (`md5sum <<<"your password"`) or you can do it online - just google "MD5 hash online".
- Add field `lockPassword` in 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"
...
}
```

- Now if user will try to change connection settings he/she will be asked for password

Managing drivers

Configure drivers with pre-installed jars

You can customize drivers configuration in 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 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" url="jdbc:postgresql://{host}[:{port}]/[{database}]" port="5432" description="PostgreSQL standard driver">
            <library type="jar" path="maven:/org.postgresql:postgresql:RELEASE" custom="false" version="42.2.3">
                <file id="org.postgresql:postgresql" version="42.2.3" path="${drivers_home}/maven/maven-central/org.postgresql/postgresql-42.2.3.jar"/>
            </library>
            <library type="jar" path="maven:/net.postgis:postgis-jdbc:RELEASE" custom="false" version="2.2.1">
                <file id="net.postgis:postgis-jdbc" version="2.2.1" path="${drivers_home}/maven/maven-central/net.postgis/postgis-jdbc-2.2.1.jar"/>
            </library>
            <library type="jar" path="maven:/net.postgis:postgis-jdbc-jtsparser:RELEASE" custom="false" version="2.2.1">
                <file id="net.postgis:postgis-jdbc-jtsparser" version="2.2.1" path="${drivers_home}/maven/maven-central/net.postgis/postgis-jdbc-jtsparser-2.2.1.jar"/>
            </library>
        </driver>
    </provider>

</drivers>
```

Windows Silent Install

It is possible to install DBeaver in silent mode using Windows Installer command line parameters.
This might be very useful for mass install automation (SSCM and other similar systems).
Installer was improved in DBeaver 5.3.3, special thanks to https://github.com/Drizin/NsisMultiUser team.

Parameters

Command line parameters supported by DBeaver installer:

Parameter	Description
/S	silent mode, requires /allusers or /currentuser, case-sensitive
/D=path	(installer only) set install directory, must be last parameter, without quotes, case-sensitive
/allusers	(un)install for all users, case-insensitive
/currentuser	(un)install for current user only, case-insensitive
/uninstall	(installer only) run uninstaller, requires /allusers or /currentuser, case-insensitive

In order to install with /allusers parameter current user must have administrator permissions.

Installer return codes (decimal):

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's 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"

Enterprise Edition

DBeaver Enterprise Edition (EE) is a commercial version of DBeaver CE. EE version includes all features of CE version plus:

- All popular JDBC drivers are included in EE distribution so you won't need to download/configure them separately.
- Support of NoSQL databases:
  - Apache Cassandra
  - MongoDB
  - Redis
  - InfluxDB
  - Google BigTable
  - AWS DynamoDB
  - WMI (Windows Management Instrumentation)
- Additional EE plugins:
  - Office formats support (XLS) for data export
  - Visual Query Builder
  - Schema/table compare, diff DDL generation
  - Analytical charts rendering
  - Persistent query database. Search in query history
  - Eclipse Marketplace (provides easily installation of any additional Eclipse plugins)
  - Advanced Mock data generators
  - Version control support
  - Automatic proxy configuration (PAC)
  - A possibility for us to support and develop the Community version, add new features faster, provide better support and much more.

Even if you do not need enterprise features you can purchase a license as a form of donation. Thank you!