Table of contents

General User Guide

Installation
Application Window Overview
Views
  Database Navigator
  Filter Database Objects
  Configure Filters
  Simple and Advanced View
Projects View
  Project Explorer
  Query Manager
  Background Tasks
Database Object Editor
  Properties Editor
Data Editor
  Navigation
  Data View and Format
  Data Filters
  Data Refresh
  Data Viewing and Editing
Panels
  Managing Charts
Data Search
SQL Generation
  Working with spatial/GIS data
  Working with XML and JSON
Managing Data Formats
  Virtual column expressions
SQL Editor
  SQL Templates
  SQL Assist and Auto-Complete
  SQL Formatting
  SQL Execution
Variables panel
Query Execution Plan
Visual Query Builder
Script Management
  Client Side Commands
Debug
  PostgreSQL Debugger
ER Diagrams
  Database Structure Diagrams
  Custom Diagrams
  Edit mode
Search
  File Search
  DB Full-Text Search
  DB Metadata Search
Schema compare
Data compare
Mock Data generation
Dashboards, DB monitoring
Projects
  Project security
  Team work (Git)
Database Management

Database Connections
- Create Connection
- Edit Connection
- Connect to Database
- Invalidate/Reconnect to Database
- Disconnect from Database
- Change current user password

Connection Types
- Transactions
  - Auto and Manual Commit Modes
  - Transaction Log
  - Pending transactions

Database drivers
- Tasks
  - Data export/import
  - Data migration
  - Data Import and Replace
  - Database backup/restore
  - Database maintenance tools
  - Task management
  - Task scheduler
  - Composite tasks

Cloud Explorer
- AWS
- AWS Credentials
- AWS SSO

Enterprise Databases support
- MongoDB
- Cassandra
- InfluxDB
- Redis
- AWS (Amazon Web Services)
  - DynamoDB
  - DocumentDB
  - Athena
  - Keyspaces
- GCP (Google Cloud Platform)
  - Bigtable
- Couchbase
- CouchDB
- Apache Hive/Spark/Impala
- Netezza
- SQLite Crypt
- Informix

Customizing DBeaer
- Changing interface language
- Installing extensions - Themes, version control, etc

DBeaer Enterprise
- Enterprise Edition
Troubleshooting

Command Line
Reset UI settings
Reset workspace
Posting issues
Log files
JDBC trace
Thread dump
Proxy configuration

Admin Guide

Managing connections
Managing drivers
Windows Silent Install

License management

License Administration
How to Import License
How to Reassign License
Installation

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

Windows / MacOS Installer

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

NOTE:

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

ZIP Archive

When installing DBeaver manually, without using an installer:

1. Extract the contents of the archive.
   - NOTE: Do not unzip the archive over a previous DBeaver version. If you already have any version of DBeaver extracted in the same location - remove it before unzipping the new version.
   - NOTE: All configurations, scripts and other necessary data are stored in a separate location (usually in the user’s home directory) so the program deinstallation does not affect them.

2. Run the `dbeaver` executable.

Debian Package

To install DBeaver using a Debian package:

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

2. Execute `dbeaver &`.

RPM Package

To install DBeaver using RPM package:

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

2. Execute `dbeaver &`.

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

Automatic updates

This feature is available only in Windows and macOS.

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

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 navigation 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 management of 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 the most basic and frequently used commands:

Some of the buttons are enabled (colored), others are disabled (grey). 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, go to the Window menu, click Appearance -> Hide Toolbar / Show Toolbar.

**Shortcut Bar**

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

**Workspace: Views 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 the 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 opened simultaneously 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 them to the latest docked position.

**Changing Workspace Layout**

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

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

You can also swap locations of two views or editors.

To dock a view to a position in the workspace, press and hold the 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 until it changes 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, 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:

View toolbar View Menu button View 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.

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, 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:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Menu item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon" alt="Driver Manager" /></td>
<td><strong>Driver Manager</strong></td>
<td>Opens the Driver Manager window that allows you to create, edit and delete drivers for databases. See Database Drivers for information about managing database drivers.</td>
</tr>
<tr>
<td><img src="icon" alt="New Connection" /></td>
<td><strong>New Connection</strong></td>
<td>Opens the Create new connection wizard. See Create Connection for information about creating connections.</td>
</tr>
<tr>
<td>(empty)</td>
<td><strong>Active Project</strong></td>
<td>Displays a submenu which allows you to choose a project. See Projects and Projects view for information about projects.</td>
</tr>
<tr>
<td><img src="icon" alt="New Folder" /></td>
<td><strong>New Folder</strong></td>
<td>Opens a dialog box for creating a new folder.</td>
</tr>
<tr>
<td><img src="icon" alt="Collapse All" /></td>
<td><strong>Collapse All</strong></td>
<td>Collapses the tree to the root level</td>
</tr>
<tr>
<td><img src="icon" alt="Link with editor" /></td>
<td><strong>Link with editor</strong></td>
<td>Synchronizes the active editor with the element in the database navigator</td>
</tr>
</tbody>
</table>

The toolbar is located in the title bar of the window. Its buttons duplicate the menu items, except for the **Active Project**.
To open the context menu for an object, right-click the object in the tree. The following table summarizes the context menu items for all types of objects that may appear in the tree. Note that the presence or absence of the context menu items for an object depends on the database and object types.

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open folder</td>
<td>Opens a folder in a separate view</td>
</tr>
<tr>
<td>Create new connections / Create New Connection</td>
<td>Opens the Create new connection wizard</td>
</tr>
<tr>
<td>New Folder</td>
<td>Opens a dialog box for creating a new folder</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies an object to the clipboard</td>
</tr>
<tr>
<td>Paste</td>
<td>Inserts the copied object into a selected folder - most convenient for copy-pasting connections</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes an object</td>
</tr>
<tr>
<td>WARNING!</td>
<td>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.</td>
</tr>
<tr>
<td>Rename</td>
<td>Opens the Rename [object] dialog box</td>
</tr>
<tr>
<td>Properties</td>
<td>Opens the Properties for [object] window which allows viewing and modifying the object’s properties</td>
</tr>
<tr>
<td>Refresh</td>
<td>Depending on the object, refreshes the object itself, or its parent, or its subnodes – mostly used for refreshing tables and schemes</td>
</tr>
<tr>
<td>Connect</td>
<td>Attempts to connect to the database</td>
</tr>
<tr>
<td>Invalidate/Reconnect</td>
<td>Checks the status of connection, if it is broken, attempts to reconnect</td>
</tr>
<tr>
<td>Disconnect</td>
<td>Disconnects from the database</td>
</tr>
<tr>
<td>SQL Editor</td>
<td>Opens a new SQL editor for the connection</td>
</tr>
<tr>
<td>Recent SQL Editor</td>
<td>Opens the most recently opened SQL editor</td>
</tr>
<tr>
<td>Edit Connection</td>
<td>Opens the Connection Configuration window that allows configuring connection settings</td>
</tr>
<tr>
<td>View [objects]</td>
<td>Opens the object in a separate viewer</td>
</tr>
<tr>
<td>Edit [object]</td>
<td>Opens the object in a separate editor</td>
</tr>
<tr>
<td>Create new [object]</td>
<td>Opens an editor in which you can specify properties and save the new object</td>
</tr>
<tr>
<td>Filter</td>
<td>Opens a submenu of one or more filtering options (depending on the object):</td>
</tr>
<tr>
<td></td>
<td>- Hide [object]</td>
</tr>
<tr>
<td></td>
<td>- Show only [object]</td>
</tr>
<tr>
<td></td>
<td>- Configure [objects] filter</td>
</tr>
<tr>
<td></td>
<td>- Toggle filter</td>
</tr>
<tr>
<td></td>
<td>- Clear filter</td>
</tr>
<tr>
<td></td>
<td>See Filters for information.</td>
</tr>
<tr>
<td>Copy Advanced Info</td>
<td>Copies the full name of an object</td>
</tr>
<tr>
<td>Read Data in SQL Console</td>
<td>Opens an SQL console displaying the object’s data</td>
</tr>
<tr>
<td>Compare</td>
<td>- Appears only if you select several objects of the same level</td>
</tr>
<tr>
<td></td>
<td>- Opens the Compare objects wizard which guides you through the steps to generate a comparison report for the selected objects</td>
</tr>
<tr>
<td>Generate SQL</td>
<td>Opens a submenu on which you can select the type of SQL query to generate:</td>
</tr>
<tr>
<td></td>
<td>- SELECT</td>
</tr>
<tr>
<td></td>
<td>- INSERT</td>
</tr>
<tr>
<td></td>
<td>- UPDATE</td>
</tr>
<tr>
<td></td>
<td>- DELETE</td>
</tr>
<tr>
<td></td>
<td>- MERGE</td>
</tr>
<tr>
<td></td>
<td>- DDL</td>
</tr>
<tr>
<td></td>
<td>Clicking one of the items (for example INSERT) generates a relevant query in a separate window.</td>
</tr>
<tr>
<td>Export Table Data</td>
<td>Opens the Data Transfer wizard that helps you select a format and export table data</td>
</tr>
<tr>
<td>Menu Item</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Import Table Data</td>
<td>Opens a window with existing database connections in which you can select a table to import data from</td>
</tr>
<tr>
<td>Tools</td>
<td>Opens a submenu that provides tools for database backup and restore, vacuum, etc.</td>
</tr>
</tbody>
</table>

For information on how to filter database objects in Database Navigator, please see the [Filter Database Objects](#) article.
Filter Database Objects

In the Database Navigator and Database Object Editor you can filter database objects to include or exclude some of them from the view. You can filter connections, schemas, tables, views, and procedures. A dots sign (…) next to the node’s name indicates that a filter is applied to its sub-nodes: Tables (…)

There are several ways in which you can filter objects. One of the ways is to filter objects by the names of tables and views using the filter field above the tree of objects:

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

You can select the types of filtering objects in the drop-down list on the right. And also use the filter only for active connections.

For multiple filtering use special symbols in the filter field - pipe ("|"), comma (","), or space between object names. You can also use an asterisk symbol to replace the part of the name.

Another way to filter objects is to use the Filter item on the context menu of a single object. To filter objects using the Filter menu, right-click the object, then click Filter on the context menu, and then click one of the items on the submenu:

<table>
<thead>
<tr>
<th>Filter submenu item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide ‘[object name]’</td>
<td>Hides the current object while displaying the other ones</td>
</tr>
<tr>
<td>Show only ‘[object name]’</td>
<td>Shows the current object while hiding the other ones</td>
</tr>
<tr>
<td>Toggle filter</td>
<td>Inverts the filtering – shows hidden objects and vice versa</td>
</tr>
<tr>
<td>Clear filter</td>
<td>Removes the filtering to display all objects</td>
</tr>
<tr>
<td>Configure [objects] filter</td>
<td>Appears only to the folder or parent nodes of database objects - like ‘Tables’, ‘Indexes’, etc. Allows the creation of a complex filter with multiple filtering criteria, see Configure Filters.</td>
</tr>
</tbody>
</table>

The third way of filtering is to use the Filter item on the context menu on several objects:

1. Select several objects of the same type using Ctrl or Shift keys.
2. Right-click the selection, then click Filter, and then choose one of the options on the submenu:

<table>
<thead>
<tr>
<th>Filter submenu item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide N objects</td>
<td>Hides the selected objects while displaying the rest</td>
</tr>
<tr>
<td>Show only selected objects</td>
<td>Shows the selected objects while hiding the rest</td>
</tr>
</tbody>
</table>
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 window](image)

2. Select the **Enable** checkbox to activate the fields of the window.

3. If you want the filter to apply to all objects of a certain type, for example to all schemes, click **Show global filter**. Otherwise, the filter will apply only to the current object. NOTE: Once you apply the global filter, you cannot revert back to the local filter in the same window. To create a local filter, reopen the Filtering window, see Step 1.

4. For objects that you want to show, click **Add** next to the **Include** field and then, in the field itself, enter the name or combination of symbols to search. For objects that you want to hide, click **Add** next to the **Exclude** field and then, in the field itself, enter the name or combination of symbols to search. NOTE: 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 saved filters. To remove a filter, in the **Name** drop-down list, click the filter name and then click **Remove**.

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

```
<table>
<thead>
<tr>
<th>Icon</th>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>Create Project</td>
<td>Opens the Create Project wizard</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Refresh Projects</td>
<td>Refreshes the projects tree to display changes caused by creating modifying or deleting projects</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Collapse All</td>
<td>Collapses the tree to the root level</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Link with editor</td>
<td>- Enabled when at least one editor is open, otherwise disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Highlights the object in the tree that has its editor open</td>
</tr>
</tbody>
</table>
```

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 the 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 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.
The Project Explorer view displays detailed contents of the currently active project. To open the Project Explorer, click `Window -> Project Explorer`.

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

The title of the Project Explorer includes the name of the project: Project – [Project name]. **General** is a project that initially exists in the system by default.

The Project Explorer displays the content of a project with metadata. The content includes: Bookmarks, ER Diagrams, and Scripts. The metadata appears in columns which you can hide or show.

The Project Explorer view provides a toolbar that contains the following buttons:

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Configure columns visibility</strong></td>
<td>Opens a dialog box in which you can select columns to display in the view</td>
</tr>
<tr>
<td></td>
<td><strong>Collapse All</strong></td>
<td>Collapses the tree to the root level</td>
</tr>
<tr>
<td></td>
<td><strong>Expand All</strong></td>
<td>Expands the tree nodes</td>
</tr>
<tr>
<td></td>
<td><strong>Link with editor</strong></td>
<td>- Enabled when at least one editor is open, otherwise disabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Highlights the object in the tree that has its editor open</td>
</tr>
</tbody>
</table>

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

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.):

You can modify the look of the Query Manager by filtering queries and setting the number of entries displayed per page, as well as specifying some storage settings, see the ‘Query Manager Properties’ section below.

To erase all entries from the Query Manager, click the Clear query manager log button ( ) in the view’s toolbar.
Query Manager Properties

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

- 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 by 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.

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:

- 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.
The Properties tab of the Database Object Editor provides you with the 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 the sub-entities appear in the side tabbed editors – to open such an editor, click the tabs on the left side of the area:

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 PostgreSQL) or SQL based views (DDL and Source):

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| ![Search items](image) | 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](image) | Filter settings | Opens the Filtering window which allows setting a custom filter, see Configure Filters |
| ![Configure columns](image) | 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](image) | 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](image) | View | Opens an editor/viewer for the item currently in focus |
| ![Create new [items]](image) | 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](image) | Delete database object | Deletes the item currently in focus |
| ![Save the current contents](image) | 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: DBBeaver recommends saving work after each change. |
<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Revert]</td>
<td>Revert to the last saved state</td>
<td>Reverts all changes made to the whole database object to the last saved state</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Load]</td>
<td>Load form file</td>
<td>- Allows selecting a file from the file system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Disabled if the SQL code is read-only</td>
</tr>
<tr>
<td>![Save]</td>
<td>Save to file</td>
<td>Allows saving the current SQL code to a file</td>
</tr>
<tr>
<td>![Open]</td>
<td>Open in SQL console</td>
<td>Opens the SQL code in an SQL Editor</td>
</tr>
</tbody>
</table>

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, which is only available for tables and views.
- As the Results tab when you run a custom SQL query in SQL Editor

The Data editor allows the viewing and data editing 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:

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: 

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

The top toolbar contains the following buttons:

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apply filter criteria</td>
<td>Applies filter criteria entered in the filter field above the data table, see Data Filters article for more information</td>
</tr>
<tr>
<td></td>
<td>Remove all filters/orderings</td>
<td>Removes all filters and orderings applied to the data</td>
</tr>
<tr>
<td></td>
<td>Save filter settings for current object</td>
<td>Saves the current filter settings for the database object to apply next time when you reopen it in the editor, see details in the Data Filters article</td>
</tr>
<tr>
<td></td>
<td>Custom Filters</td>
<td>Opens the Result Set Order/Filter Settings window, see Data Appearance article for more information</td>
</tr>
<tr>
<td></td>
<td>Configure auto-refresh</td>
<td>Allows configuring data auto-refresh settings, see Data Refresh article for details</td>
</tr>
<tr>
<td></td>
<td>Forward and backward - history navigation buttons</td>
<td>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).</td>
</tr>
</tbody>
</table>

The side bar contains the following tabs:

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description Chart_button</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Grid</td>
<td>Switches to grid view of data</td>
</tr>
</tbody>
</table>
The bottom toolbar provides the following buttons:

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

<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut</td>
<td>Cuts the content of the current cell or column to the clipboard</td>
</tr>
<tr>
<td>Menu Item</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Copy</td>
<td>Copies the content of the current cell or column to the clipboard</td>
</tr>
<tr>
<td>Advanced Copy</td>
<td>Opens advanced copy submenu that allows copying data with preset formatting parameters</td>
</tr>
<tr>
<td>Paste</td>
<td>Pastes the copied content to the cells in focus</td>
</tr>
<tr>
<td>Advanced Paste</td>
<td>Pastes several values delimited with a tabulation or line break</td>
</tr>
<tr>
<td>Delete</td>
<td>Deletes the row that has the cell in focus</td>
</tr>
<tr>
<td></td>
<td>NOTE: In fact, when users click <strong>Delete</strong>, the system only highlights the red row while the actual deletion happens when users click <strong>Save</strong>.</td>
</tr>
<tr>
<td>Edit cell</td>
<td>- For CLOB/BLOB data format, opens the contents of the cell in a new tab</td>
</tr>
<tr>
<td></td>
<td>- For all formats except CLOB/BLOB, opens a properties window for the cell</td>
</tr>
<tr>
<td>Inline edit</td>
<td>- Same as double-click on a cell</td>
</tr>
<tr>
<td></td>
<td>- Makes the cell editable</td>
</tr>
<tr>
<td>Set to NULL</td>
<td>Sets the value of selected cells to NULL</td>
</tr>
<tr>
<td>Hide column</td>
<td>Hides the column currently in focus, see the Managing Display of Columns in Data Table section further in this article</td>
</tr>
<tr>
<td>Save to file…</td>
<td>- Appears only for columns with BLOB/CLOB data</td>
</tr>
<tr>
<td></td>
<td>- Opens the standard Save As window that allows saving data contained in the cell to a file</td>
</tr>
<tr>
<td>Load from file…</td>
<td>- Appears only for columns with BLOB/CLOB data</td>
</tr>
<tr>
<td></td>
<td>- Opens a standard window for opening files</td>
</tr>
<tr>
<td>Order/Filter</td>
<td>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, DBBeaver filters data by sending a request to the server (the Server-side results ordering checkbox selected). To filter data on the client side using DBBeaver’s internal algorithm, clear the checkbox.</td>
</tr>
<tr>
<td>View/Format</td>
<td>Opens a submenu that provides tools for formatting and modifying the view of data, see Data View and Format</td>
</tr>
<tr>
<td>Navigate</td>
<td>Opens a submenu that helps users navigate throughout the data table, see Navigation</td>
</tr>
<tr>
<td>Layout</td>
<td>Changes the layout of data, see the Table vs. Record Views section of the Data View and Format article</td>
</tr>
<tr>
<td>Export Resultset</td>
<td>Opens the Data Transfer wizard that guides you through the steps to select a format and export data</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Generate SQL</td>
<td>Opens a submenu on which you can select the type of SQL query to generate</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refreshes the whole results set including all items that are not visible in the screen</td>
</tr>
</tbody>
</table>

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: 200.

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 this next portion of data equal to the maximum result set size. To do so, click the Fetch next page of results button (8) 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:

<table>
<thead>
<tr>
<th>Row(s) fetched</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0ms</td>
</tr>
</tbody>
</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 (18) 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 actions such as applying filters to data, opening reference tables and other tables via links. You can navigate among such tables and filtered views:

- Use the forward and backward buttons in the top toolbar: ← →
- Click Ctrl+Left or Ctrl+Right

Hovering over these buttons displays the names of the tables or filtered views saved in the history.

Navigate Foreign Keys / Referencing Tables

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

DBeaver user guide 28/285
The referencing table opens in the same editor. To navigate back and forth between the initial and referencing tables, use the history navigation buttons ( ⇙  ⇙ ) in the top toolbar of the editor. In order to open a referencing table in a new window use the Ctrl+Shift+1 shortcut to the show menu.

**Navigation Links**

In the data editor, you can navigate to linked tables – the ones that the current table references. To open a linked table, click the Navigate link icon in a cell that contains it:

```
<table>
<thead>
<tr>
<th>123 AlbumId</th>
<th>abc Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Face lift</td>
</tr>
<tr>
<td>8</td>
<td>Warner 25 Anos</td>
</tr>
<tr>
<td>9</td>
<td>Plays Metallica By Four Cel</td>
</tr>
</tbody>
</table>
```

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:

```
<table>
<thead>
<tr>
<th>ArtistId</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Antônio Carlos Jobim</td>
</tr>
</tbody>
</table>
```

NOTE: The table name in green above the table indicates which table is currently open in the editor.

To navigate back and forth between the initial and linked tables, use the history navigation buttons ( ⇙  ⇙ ) in the top toolbar of the editor.

You can open a linked table in a separate editor. To do so, simultaneously hold the Ctrl key (or Command key on macOS) and click the Navigate link icon (  ) in the cell.
Data View and Format

Grid vs. Plain Text Views

You can switch between two data presentations in SE version and four presentations in EE version. Pressing **CTRL+~** switches available presentations in turn.

- To see the data in a grid view, similar to an Excel spreadsheet, click the Grid button (usahaan Grid on the bottom toolbar of the editor.
- To switch to the plain text view, click Text (usahaan Text) on the bottom toolbar.
- To switch to JSON view (available in EE version only for MongoDB documents and JSON tables), click JSON on the toolbar.
- To switch to XML view (available in EE version only for XML tables), click XML on the toolbar.

Table vs. Record Views

The table view is a standard table (Excel-like) in which columns are vertical and rows are horizontal. This view is the default one. If you click the Record button in the bottom toolbar of the editor (usahaan Record), or press Tab, or right-click a cell and then click Layout -> Record on the context menu, the rows and columns switch positions. The columns will appear as rows, and the rows will be hidden in one Value column which will show only one row of data. The column headers will shift from the top of the table to its left side:

```
<table>
<thead>
<tr>
<th>#</th>
<th>Album Id</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21</td>
<td>Samba De Lexington</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>Let There Be Rock</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Restless and Wild</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>Big Ones</td>
</tr>
</tbody>
</table>
```

The Record view is useful if the table contains a big number of columns. To navigate from row to row of data, use the navigation buttons on the bottom toolbar of the editor:

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

Rows Coloring

In the data editor, you can colour all rows that have the same value as a particular cell of a certain column. To do so, right-click the cell and click View/Format ⇒ Set the row colour for {column name = value} on the context menu:

Then choose the colour in the palette window that appears and click OK. The current row and all other rows that contain the same value change their colour to the one you have selected:

```
<table>
<thead>
<tr>
<th>#</th>
<th>Album Id</th>
<th>Title</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21</td>
<td>Samba De Lexington</td>
<td>NULL</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>Let There Be Rock</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Restless and Wild</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>Big Ones</td>
<td>NULL</td>
</tr>
</tbody>
</table>
```

To remove the coloring by a particular column, right-click the cell again and click View/Format ⇒ Clear colour for {column name = value} on the context menu.
By choosing View/Format ⇒ Row colors ... from the context menu, you can gain more precise control of coloring conditions:

Here, you can define multiple conditions for single column using rich set of predefined operators, change background color and define a range between two values.

Operators work as you may expect. Note that they're executed on the client-side, that means no extra queries are made in order to apply colors.

**Value range / Gradient**

Value range allows you to paint your rows with gradient that fades from first value to second value:

In this example we defined a range for column `AlbumId` that fades from `#80c6ff` to `#8000ff` between values `1` and `10`.

**Using regex**

You can use regular expressions for matching complex values. Otherwise, you can be artistic and, for example, paint rows with odd values in your favorite color:
Coloring by Data Types

Besides colouring rows by a value, you can colour the values in the columns by data types. To do so, right-click any cell in the table and, on the context menu, click View/Format -> Colourize Data Types. The values in the cells will be coloured in different colours according to the current colour preferences:

You can change the colour preferences in the Preferences window by: clicking Window -> Preferences on the main menu. In the window of the navigation pane on the left, expand User Interface and then Appearance, and then click Colours and Fonts:

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

Transforming Data Presentation

For string and numeric data types, DBeaver provides tools to transform the data presentation into a number of formats, such as URL and Binary for strings and Epoch Time, Number Radix, etc. for numbers. To change the data presentation in a certain column, right-click a cell in the column. Then, on the context menu, click View/Format -> Set {column name} format and click the presentation type name:
The Transformer settings window opens showing the value in the chosen format. Click OK to apply the change:

The values in the column appear in the new format.

NOTE: For URL format, the resulting cell provides a link to the URL in a browser window.

To roll back the changes to the default format, right-click any cell in the column, and on the context menu, click View/Format -> View as -> Default.

**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:

To do so, right-click a cell in the column and, on the context menu, click View/Format -> Visualize complex columns.

**Configuring Numeric and Time Data Formats**

You can specify the exact format of Time, Timestamp, Date, and Number data used in the currently open database or globally. To specify a format, right-click any cell in the table and, on the context menu, click View/Format -> Data formats. The Properties window opens displaying the Data Formats page:
To configure only the format for the current database, select the Data source "[Connection name]" settings checkbox. To configure the settings globally, to all databases that you have in DBeaver, click Global settings.

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

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

Configuring Boolean presentation

You can choose between two presentation modes:

1. Text
2. Icons
3. Align
4. Color
• Text-based
• Icon-based

Text-based presentation

This is the most customizable mode. You can:

1. Change labels under Label column. Otherwise, you can use presets available in Drop-down Menu.

2. Change alignment of value inside grid cell. Following variants are available: left, center, and right

3. Change color of value using color picker under Color column. You can reset color to match current theme's contrast color in Drop-down Menu ⇒ Colors ⇒ Use theme default color.

4. Change font style in Drop-down Menu ⇒ Styles. Following variants are available: normal, bold, and italic.

Icon-based presentation

This presentation only supports alignment changing.
Data Filters

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

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

You can apply ready-to-use SQL expressions or SQL expression templates via the context menu. To select a ready SQL expression or a template, press F11 or right-click the cell, then click Order/Filter on the context menu and then click one of the expressions.

The third way is to filter data by a cell value using the filter icon in the column header. To filter data this way, click the filter icon in the column header and then double-click the cell value in the Filter by the column value dialog box:

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

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

Advanced filters configuration

The main tool for managing the appearance of the data table is the Result Set Order/Filter Settings window.
To open this window, click the Custom Filters button (🔧) in the top toolbar of the editor or click the Configure button (🛠️) and then click Order/Filter on the dropdown menu.

The Result Set Order/Filter Settings window provides tools to:

- Order data inside columns
- Manage the display of columns in the table
- Manage the order of columns in the table
- Filter data in the table using an SQL expression

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

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

### Ordering Data in Columns

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

1. Click the ordering icon (↑↓) in the header of the column.

   ![Column Headers](image)

   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 [DBeaver user guide](https://wwwdbeaver.dev/doc/user-guide/)).
a) Next to the column by which you want to order data in the first turn, set the ascending or descending order using the same three-state principle as described above.

b) Set the ordering in other columns by which you want to sort the data in the second, third, etc. turn. The **Order** column indicates the order in which the sorting will happen.

**NOTE:** The number (#) column indicates the initial order of columns.

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 ( ![Reset] ) in the same window.

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

### Managing Display of Columns in Data Table

To hide a single column, right-click the column or any cell in it and click **View/Format -> Hide column** on the context menu. To unhide a hidden column, open the Result Set Order/Filter Settings window (see the image at the beginning of this article) and select the checkbox next to the column name, or click the Reset button ( ![Reset] ).

To display or hide columns in the data table, in the Result Set Order/Filter Settings window:

1. Select the checkboxes next to the columns that you want to see in the table and clear the checkboxes next to those that you want to hide.

2. Use the Show All ( ![Show All] ) and Show None ( ![Show None] ) buttons at the bottom of the window.

### Sorting Columns in Data Table

You can modify the order of columns in the data table in two ways:

1. Click the icon in the column header and drag-and-drop the column to a new position.

2. To sort the column alphabetically, in the Result Set Order/Filter Settings window (open by clicking the Custom Filters button ( ![Custom Filters] ) in the top toolbar of the editor), click the Sort button ( ![Sort] ).

3. In the Result Set Order/Filter Settings window, click the column to set the focus to it and then move it using the navigation buttons: ( ![Move up] , ![Move down] , ![Move to top] , ![Move to bottom] )
Data Refresh

Refreshing is necessary if the database contains changes made by other users working on it simultaneously with you, and you want to see them in your DBeaver window. To refresh data manually, right-click anywhere in the data table and click Refresh on the context menu or press F5.

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 dialog box]

   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:

   ![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 these 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 the steps described in this section, the changes will 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:

![Preview Changes](image)

**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 is now editable and 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 Cancel 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:
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 them 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 an image. It is convenient to open the 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 (撂) on the bottom toolbar. Use inline editing or open the 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 (撂) in the bottom toolbar. The duplicate rows appear below the rows in focus.

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

To save any such changes, click the **Save** button (撂) on the bottom toolbar. To discard the changes, click the **Cancel** button (撂) on the bottom toolbar.

### Copying/Pasting Cells

To copy the content of one or several cells to the clipboard in the 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 a 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 save column value changes, a table must have some unique key (primary key or unique index). Some databases (Oracle, DB2, PostgreSQL) support a special virtual unique column that DBeaver can use to save changes. In other cases, you can define a virtual key – a set of columns that forms a unique combination of values. When you try to save changes in a table without a unique key, DBeaver displays the following error message:

![Error message](image)

To use all columns as the virtual key, click **Use All Columns**. To create a custom key, click **Custom Unique Key**. Alternatively, to create a unique custom key, you can click the **Configure** button on the bottom toolbar and then click **Define virtual unique key** on the Configure menu. The Define virtual unique identifier window opens:

![Define virtual unique identifier](image)

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 data. The panels are handy if you work with complex types (structures, arrays), long text data, or BLOBs. Panels appear as tabs in an additional pane in the right hand side of the Data tab:

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

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

To open the panels, click Panels on the bottom toolbar. By default, the Value viewer panel opens. Alternatively, you can open the Value panel by pressing F7 on a cell. To open the other panels, click the down arrow next to the Panels button and click the name of the panel on the menu:
Panels will also open if you try to inline-edit a cell with a complex data type.

To close the panels, click the Panels button again or click the standard Close (cross) icon in the upper right corner of each panel. You can also show and hide panels by clicking the Configure button ( ) on the bottom toolbar and then Toggle result panels on the Configure dropdown menu.

**Value Viewer**

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

The toolbar of the Value panel contains the following buttons:

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Settings Icon]</td>
<td>Content viewer settings</td>
<td>Opens a menu with a set of options for content view change.</td>
</tr>
<tr>
<td>![File Icon]</td>
<td>Save to file...</td>
<td>Allows saving the content to a local file. <strong>NOTE:</strong> This button is only available for XML, JSON and Binary content.</td>
</tr>
<tr>
<td>Button</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>![Load from file]</td>
<td>Load from file...</td>
<td>Allows uploading data from a local file. <strong>NOTE:</strong> This button is only available for XML, JSON and Binary content.</td>
</tr>
<tr>
<td>![Apply cell value]</td>
<td>Apply cell value</td>
<td>Displays the data table changes in the Value viewer. <strong>NOTE:</strong> This button does not save changes made to the database. To save the changes in the database, you need to use the <strong>Save</strong> button on the bottom toolbar of the <strong>Data Editor</strong>.</td>
</tr>
<tr>
<td>![Auto-apply value]</td>
<td>Auto-apply value</td>
<td>Enables the automatic display of changes made in the Value viewer in the data table. When auto-saving is enabled, the changes appear in the data table at the same time when they are made in the Value viewer. <strong>NOTE:</strong> This button does not save changes made to the database. To save the changes in the database, you need to use the <strong>Save</strong> button on the bottom toolbar of the <strong>Data Editor</strong>.</td>
</tr>
</tbody>
</table>

### Metadata Panel

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

![Metadata panel example]

### Calc Panel

The Calc panel is useful for getting basic statistics across data in several columns and rows:
You can select several columns and rows in standard ways - by pressing and holding the left mouse button or by clicking cells while holding the Ctrl or Shift keys. The panel updates dynamically to show statistics for the selected data.

To see the data grouped by columns, click the Group by columns button ( ). To remove the grouping by columns and see the summary values for all columns, click the same button again.

By default, the panel applies and displays results for two functions – Count and Count Distinct. To add other functions, click the Add function button on the toolbar of the panel or right-click one of the rows in the Aggregate panel and click Add function on the context menu and then click the name of the function. The following functions are available:

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

To remove an individual function, click the function and then click Remove function on the toolbar of the panel, or right-click the function and click Remove function on the context menu. To remove all functions, click Reset on the toolbar or on the context menu.

You can copy the value of a particular function to the clipboard - right-click the row and click Copy Value on the context menu. You can also copy all functions with their values - right-click in the table and click Copy All on the context menu.

**Grouping Panel**

The Grouping panel provides tools to calculate statistics based on a table of a custom SQL query. It uses GROUP BY queries to extract unique values for COUNT (default), SUM, AVG, MIN, MAX and other analytics functions displaying the results in dedicated columns.

To obtain the grouping results for one or more columns of a data table, open the Grouping panel, then, in the results table, put the cursor onto the data type icon of the table header so that the cursor turns into a hand pointer ( ), and drag-n-drop the column(s) into the panel:
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.

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 on 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 the auto-complete options DBeaver provides.
   - You need to indicate the column name in brackets. COUNT is the only function that supports `*` instead of the 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 on 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 DBever 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:
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.

**References panel**

The references panel allows you to see all the related information for the chosen row from other connected tables. The information is presented in an additional data viewer window, filtered to show the information related to the currently selected row. If a table opened in data viewer has a foreign key referencing another table, or it is referenced with a foreign key by another table, all of those connected tables can be picked from a dropdown list.

When a table that is referenced by a foreign key in the current table is chosen, the information from the row corresponding to a referenced key will be shown, in this situation the record mode is enabled by default, but it can be turned off like in a normal data viewer.
When a table that references the current table is chosen, the references panel will show all the rows that refer a selected primary key in the current table.
Managing Charts

Note: This functionality is available only in the Enterprise Edition.

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

Note: Analytical Charts are only present in the 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 on the left vertical toolbar of the query results area.
A bar chart will be created.

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 first sorting and applying various filters to its columns. 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.

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**

An analytical tool such 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 axis X and axis Y source data by switching the columns in the
Charts Editor.

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

![Grouping Panel](image)

**Editing Chart Settings**

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

![Chart Editor](image)

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 supported:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVG</td>
<td>Average value</td>
</tr>
<tr>
<td>SUM</td>
<td>The sum of all values</td>
</tr>
<tr>
<td>FIRST</td>
<td>The first value</td>
</tr>
<tr>
<td>LAST</td>
<td>The last value</td>
</tr>
<tr>
<td>COUNT</td>
<td>Total count of all values</td>
</tr>
</tbody>
</table>

Copying to clipboard

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

Exporting Charts

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

Printing Charts

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

To search for data in the result set, press \texttt{CTRL+F}. The standard Find/Replace search dialog box will open:

![Find/Replace dialog box]

You can also use the Find and Replace feature.

\textbf{NOTE:} The system searches only in already fetched rows.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case sensitive</td>
<td>By default, the search is case insensitive.</td>
</tr>
<tr>
<td>Whole word</td>
<td>By default, the word specified in the search field can be found in the case when the word is part of another word. Enabling this setting will lead to a particular word search.</td>
</tr>
<tr>
<td>Regular expressions</td>
<td>Enable to use regular expressions in the search. In the search field, you can use \texttt{Ctrl+Space} for autocomplete regular expressions templates.</td>
</tr>
<tr>
<td>Wrap search</td>
<td>Enable this setting to find matches throughout the object from the beginning, and not only from the focus point to the end of the object.</td>
</tr>
<tr>
<td>Incremental</td>
<td>Matches are found incrementally as you type, if this setting enabled.</td>
</tr>
</tbody>
</table>
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:

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

To use table names in the format '[schema name].[table name]', select the Use the fully qualified names checkbox.

To wrap the SQL query into one line, select the Compact SQL checkbox:
`INSERT INTO public.film (film_id, title, description)`
Working with XML and JSON

DBeaver supports XML and JSON column types (in relational databases) by using standard JDBC interfaces. This feature was added in JDBC4 so you will need JDBC4 compliant driver for your database.

In the Data Editor, you can edit XML/JSON data right in the table cells. However, a huge amount of data may require a larger editor so you might want to save XML/JSON scripts to a local file or upload this type of data from a local file.

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

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

To auto-format XML/JSON script press Ctrl + Shift + F keyboard buttons.
Use Ctrl+S keyboard shortcut to save the changes made.

You can also edit XML/JSON content, save it locally and upload it from a local file with the help of Value panel toolbar.

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

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

To switch between the formats, press the Content viewer settings button and select the format.
Use **Word Wrap** feature that wraps the text within a screen.

Use **Auto Format** feature to automatically change the appearance of XML/JSON script (fix spaces around operators / commas, fix indentation, etc) and make it more readable.

To learn more about **Value** panel, see **Panels**.
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, for example, for database migration.

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

In the Preferences dialog box go to Editors -> Data Editor -> Data Formats.
Or, in the Database Navigator right-click a connection and select Edit Connection menu option.

In the right area of the opened Data formatting preferences dialog window go to Data editor -> Data Formats and select the Datasource settings check box in the left area to customize the data format settings.
Data Format Profiles

Data format profiles allow you to apply a set of data format preferences to the whole current project by one click.

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

**Changing Data Formats**

The following groups of data format settings can be adjusted:

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

Native Date/Time Mode

Select **Use native date/time format** check-box and the data format originally built-in to the datasource will be used.

You can change the format of the following data types:
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*.

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yyyy or y</td>
<td>Year of era (4 digits)</td>
</tr>
<tr>
<td>yy</td>
<td>Year of era (2 last digits)</td>
</tr>
<tr>
<td>YYYY</td>
<td>Week year</td>
</tr>
<tr>
<td>M</td>
<td>Month in year without leading zeros</td>
</tr>
<tr>
<td>MM</td>
<td>Month in year</td>
</tr>
<tr>
<td>MMM</td>
<td>Short month name in year</td>
</tr>
<tr>
<td>MMMM</td>
<td>Month name in year</td>
</tr>
<tr>
<td>D</td>
<td>Day in year</td>
</tr>
<tr>
<td>d</td>
<td>Day in month without leading zeros</td>
</tr>
<tr>
<td>dd</td>
<td>Day in month</td>
</tr>
<tr>
<td>E</td>
<td>Day name in week</td>
</tr>
<tr>
<td>G</td>
<td>Era designator</td>
</tr>
<tr>
<td>z</td>
<td>General time zone</td>
</tr>
<tr>
<td>Pattern</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>Z</td>
<td>RFC 822 time zone</td>
</tr>
</tbody>
</table>

**Time**

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

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>Hour in day (0-23) without leading zeros</td>
</tr>
<tr>
<td>h</td>
<td>Hour in day in am/pm (1-12) without leading zeros</td>
</tr>
<tr>
<td>HH</td>
<td>Hour in day (0-23)</td>
</tr>
<tr>
<td>hh</td>
<td>Hour in day in am/pm (1-12)</td>
</tr>
<tr>
<td>a</td>
<td>Am/pm marker</td>
</tr>
<tr>
<td>m</td>
<td>Minute in hour without leading zeros</td>
</tr>
<tr>
<td>mm</td>
<td>Minute in hour</td>
</tr>
<tr>
<td>s</td>
<td>Second in minute without leading zeros</td>
</tr>
<tr>
<td>ss</td>
<td>Second in minute</td>
</tr>
<tr>
<td>S</td>
<td>Millisecond</td>
</tr>
<tr>
<td>fffffff</td>
<td>Microseconds</td>
</tr>
</tbody>
</table>

**Timestamp**

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

[Link on Java date pattern documentation](#)

**Numbers**

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

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

- **Maximum integer digits** - Defines the maximum number of digits to the left of the decimal point.

- **Minimum integer digits** - Defines the minimum number of digits to the left of the decimal point.

- **Maximum fraction digits** - Defines the maximum number of digits to the right of the decimal point.

- **Minimum fraction digits** - Defines the minimum number of digits to the right of the decimal point.

- **Use data type scale for fraction digits** - Some numeric columns or parameters may have a predefined scale, that is the maximum number of digits to the right of the decimal point. Select this check-box if you want the predefined precision to be used.

- **Rounding mode** - Specifies a rounding behavior for numerical operations capable of discarding precision. Each rounding mode indicates how the least significant returned digit of a rounded result is to be calculated. To learn more, please refer to [Oracle documentation](#).

To change the data type format, change the value displayed in the **Pattern** area, save the changes made by pressing the **Apply** button and observe the expected result in the **Sample** field.
To restore the default data format settings, press the **Restore Defaults** button.

**Datasource Settings**

Press **Datasource settings** link to change data format settings for a particular datasource, then adjust the settings in the opened...
To save changes made press **Apply**.
Virtual column expressions

Expression language

You can use standard JavaScript-like expression language. DBeaver uses the Jexl engine to process expressions. Language reference and examples can be found here: http://commons.apache.org/proper/commons-jexl/reference/syntax.html

Column values

All columns' values in the current result set can be referred to by name. Expression column1 + column2 will produce the sum of two numeric columns or concatenation of two string columns column and column2.

Standard functions

Standard functions declared in namespaces. You can refer to the functions in the namespaces as variables - nsName.functionName(parameters).

math

You can access all math functions as math.function(parameters).
You can find all supported math functions here: https://docs.oracle.com/cd/E12839_01/apirefs.1111/e12048/functmath.htm

go

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wktPoint</td>
<td>(longitude, latitude)</td>
<td>Produces WKT (geometry) point out of two coordinates. Default SRID is 4326.</td>
</tr>
<tr>
<td>wktPoint</td>
<td>(longitude, latitude, srid)</td>
<td>Produces WKT (geometry) point out of two coordinates and SRID</td>
</tr>
</tbody>
</table>
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, 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). Unlike the console, it can save scripts and changes made to them.

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

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

The execution Log tab contains all queries executed in the current SQL editor:

You can open the 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:
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 pane and, on the context menu, click **Layout -> Horizontal**. To position both panels vertically, right-click anywhere in the script pane and, on the context menu, click **Layout -> Vertical**.

### Hyperlinks

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

```sql
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 datasourse** 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 \texttt{Ctrl+0} or click the \texttt{Active Catalog/Schema} box in DBeaver’s main toolbar:

The Choose catalog/schema dialog box opens. In the list of schemas, double-click the required schema:

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

To configure the set of columns to be visible for each schema in the dialog box, click the \texttt{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 \texttt{Set connection from navigator} button on 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 \texttt{Ctrl+Shift+}, or click the arrow next to the \texttt{Set connection from navigator} button in DBeaver’s main toolbar and then click \texttt{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 with a list of available templates appears:

To apply a template, in the SQL Editor, in the script pane:

- Type the template name and press Tab.
- Right-click the line where you want to insert a template expression, click SQL Template on the context menu, and then, in the list of templates, double-click the required template name. The template SQL statement appears in the script.

To edit/add/remove templates, click Configure ( november) in the bottom toolbar, then click Preferences -> SQL Editor -> Templates. For more information about managing templates, please visit Eclipse Website.

### Standard Eclipse templates:

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

### DBeaver-specific templates:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>${schema}</td>
<td>Takes the current schema name</td>
</tr>
<tr>
<td>${catalog}</td>
<td>Takes the catalog name</td>
</tr>
<tr>
<td>${table}</td>
<td>Takes the current table name (from the active catalog/schema)</td>
</tr>
<tr>
<td>${column}</td>
<td>Takes the column name (from the current table)</td>
</tr>
</tbody>
</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 objects in a database, by their names and/or descriptions.

When you start typing an SQL keyword in a statement, DBeaver offers auto-complete options as well. Another auto-complete function is that it searches for the last entered identifier - type the first letter and press Ctrl+Shift+Space.

You can also press Ctrl+Space after the asterisk in the query similar to SELECT * FROM tableName or similar to INSERT INTO tableName (*) (brackets are important) (you can use ()[]{} brackets) - the asterisk will be replaced with a list of all the table columns.
SQL Formatting

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

To format a script to upper or lower case, highlight the SQL text, then right-click it and click **Format -> To Upper Case / To Lower Case**, respectively, on the context menu.

To comment out an SQL line, press `Ctrl+/' or right-click the line and click **Format -> Toggle Line Comment** on the context menu.

To uncomment a commented line, manually remove the commenting syntax, or press the same button combination, or right-click the line and click the same item on the context menu.

To comment out a block of text, select the text, then press `Ctrl+Shift+/' or right-click it and click **Format -> Toggle Block Comment** on the context menu. To uncomment a commented block of text, you can either manually remove the commenting syntax or select the same block of text, right-click it and click the same item on the context menu or press the same button combination.

To trim spaces (leading and trailing) SQL text right-click the selected text or end of string and click **Format -> Trim spaces** on the context menu.

You can choose a part of the text or put the cursor at the end of the row you want to trim.
You will get the following result:

Or you can choose a part of the text from many lines

Then each of the selected rows will trim, and the final result will be like this:
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):**

- **DBeaver main menu:**

To execute a query under the cursor or selected text, press **Ctrl+Enter** or right-click the query and click **Execute -> Execute SQL Statement** on the context menu. You can do the same using the main toolbar or main menu: **SQL Editor -> Execute SQL Statement**. This executes the SQL query under the cursor or selected text and fills the results pane with the query results.

To execute a query under the cursor in a separate tab, press **CTRL+\** or right-click the query and click **Execute -> Execute SQL in new tab** on the context menu. The same can be done using the main toolbar or the main menu: **SQL Editor -> Execute SQL in new tab**. This executes the SQL query under the cursor or selected text and creates a new results tab.

To execute the whole script, press **Alt+X** or click **Execute -> Execute SQL Script** on the context menu or **SQL Editor -> Execute SQL Script** on the main menu or in the main toolbar. This executes all queries in the current editor (or selected queries) as a script. DBeaver parses queries one by one using a statement delimiter (";" by default) and executes them consecutively. You can configure the script execution behavior in the SQL editor preferences (Right-click the script and click **Preferences** on the context menu).

To execute 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.

**Result tabs**
A single query may generate several result sets represented by tabs. These tabs are linked to the query they are executed from.

- To close an individual tab, press `CTRL+Shift+\` or middle-click on a tab header.
- To close all tabs expect current, click Close all result tabs except this on the context menu of this tab.
- To close all tabs of desired query, click Close all result tabs of same query on the context menu of this tab.

**Naming**

Tabs are named in a form of `Results <A> (<B>)`, where:

- **A** is an index of query
- **B** is an index of result set of this query

**Pinning**

Tabs can be moved around by dragging them with a mouse and pinned using Pin tab on the context menu of desired tab. Pinned tabs are stacked on the left. They can be moved among other pinned tabs, but can't be mixed with unpinned tabs. Pinned tabs cannot be closed without being unpinned first, and cannot be overwritten by executing a query in it (by making this tab active).

**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:

![SQL Expression Evaluation](image)

**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:
It might be useful to export a query if you have a long-running query and you do not need to see its results in the results panel. You can directly export the current query results to a file/table by right-clicking the query and then clicking **Execute -> Export From Query** on the context menu:

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

**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:

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

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

**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.
Variables panel

You can see all of the currently assigned local variables for SQL Editor. To do so you need to click a "Show SQL variables" button in the SQL editor.

A new tab alongside Output and Execution log panels will be opened with a list of assigned variables. On this panel you can also show assigned parameters by clicking the corresponding button.

Manipulating variables

You can change values for current variable using variables tab. Simply click on a variable’s row and edit it's value in a window below. To delete or add a variable you can use a corresponding button. You can use these buttons instead of typing @set or @unset in the script.

Note: Adding and deleting works only for variables, but not parameters.

If you have a long list of variables, you can click a magnifying glass to initiate a search bar. Start typing either a variable’s name or it’s value to filter the list.

Moving a panel

Variables panel is always connected to output and execution log panels, but this group of tabs can be configured to be shown either at the right side of the editor or at the bottom alongside the results panel. By default the panels are shown at the right side. To change their location you need to either check or uncheck Show panels in result tabs in the context menu.
If a database driver supports the visualization of the execution plan, you can see the execution plan of the current query (under cursor) by pressing \texttt{Ctrl+Shift+E} or clicking \texttt{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:

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 \texttt{Reevaluate} button. To see the source script on which the plan is based, click the \texttt{View Source} button.

In DBeaver \texttt{Enterprise Edition} you can use an advanced (graph) visualization of the execution plan. This visualization shows the most expensive (cost-based) plan nodes. You can hide all irrelevant nodes (see node details), use horizontal or vertical plan layouts, export it to an image or save it as JSON to send to a colleague.
Visual Query Builder

Note: This feature is available in Lite, Enterprise, and Ultimate editions only.

Query Builder is a user-friendly visualization tool that will help you make sense of your complex database designs. It can be useful when you need to understand the various relationships between different tables. Also, it can be helpful for those who are not very familiar with SQL scripting or if you do not want to insert script commands manually. The tool creates SQL scripts automatically based on the visual schema you have created.

*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.

Creating Visual Query

Start creating a query by selecting a query data source: drag-and-drop tables you want to work with from the Database Navigator pane into the Visual Query Builder area. All the connections existing between the tables will be shown automatically.
To create a new join between the tables, press the left mouse button when the cursor is over the column of one table, holding the right mouse button drag the cursor to the column of another table and release the right mouse button. The connection between the selected columns of the tables will be created visually and in the SQL script a new join will be added.
To remove a join between the tables, click on it. The connection will be highlighted. Then, press **Delete** or use the **Delete** option in the context menu. The visual connection will be removed and the corresponding join will be automatically removed from the SQL script area.

To build a SELECT query you need to select columns in the tables you added. To select a column, click the 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 you to set the query conditions and adjust the representation of query results by means of **Query Settings Editor**.

To open **Query Settings Editor** use **Visual builder query settings** button in the vertical tool bar on the left.

**Query Settings** window contains five tabs described below.

### Columns

**Columns** tab of the **Query Settings Editor** contains all the columns you added by selecting column names in **Visual Builder** main window. In this tab you can add and remove columns using **Add** and **Remove** buttons correspondingly.

To add a column, press **Add** button and a new instance will be added to the table. Click on the first cell in **Column or Expression**...
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.

If you want a **grouping** condition to be added to your expression, you can click on the checkbox in the column row. The expression will update automatically. The other previously selected columns will become aggregate. If there are no other columns, then the expression `COUNT(*)` will be automatically added.

You can select other aggregation functions from the drop-down list. Or enter your own version in the cell.
When removing columns from the list, they will be removed from the grouping expression as well. When adding new columns to the list, it is added to the grouping expression.

**Conditions**

The **Conditions** tab is used for managing query conditional expressions.

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

- **Left Operand** setting defines the left operand of the conditional expression. To set the left operand, click the cell in the **Left Operand** column and a drop down list of all available columns will be displayed. Select a column you want to use as the left operand in your conditional expression or insert a digit.

- **Operation** setting defines the comparison rule between the left and the right operands of the conditional expression. To set a comparison rule, click the cell in the **Operation** column and select the rule you need from the drop down list which will appear.
Right Operand setting defines the right operand of the conditional expression. To set the right operand, click in the Right Operand column and a drop down list of all available columns will be displayed. Select a column you want to use as the left operand in your conditional expression or insert a digit.

To remove a conditional expression, click on the row containing the expression and press the Remove button on the right.

**Joins**

All the joins existing between the tables in Visual Query Builder main window are displayed in the **Joins** tab of Query Settings Editor.

Joins cannot be added or removed by means of Query Settings Editor, however, the following join settings can be adjusted here:
- **Type** - defines the type of the join. Click the cell in the **Type** column - a drop down with available join types will be displayed. Select the required option from the list by clicking on it.

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

**Sorting**

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

To add a new sorting condition press the **Add** button on the right and the default conditional expression ORDER BY will be added to the SQL script area automatically. This default conditional expression can then be adjusted to the one you need:

- Once a new condition is added, click the first cell in **Conditions or Expressions** column and a drop down list of all available columns will appear. Select the required column by clicking on its name.
In the **Order** column you can define whether the rows of the selected column should be sorted in ascending or descending order in the result table. To set the order, click the cell in **Order** column and select the required option from. The order by command will be added to the script.

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

**Miscellaneous**

In the **Miscellaneous** tab it is possible to:

- Enable or disable the automatic generation of aliases for tables by selecting the **Add table aliases** check-box.
- Disable auto-completion for table names by selecting the **Use fully qualified table names** check-box.
- Enable Autosave on SQL-editor switch by selecting the **Autosave on SQL-editor switch** check-box.
Executing Visual Query

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

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

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

Unsaved data is highlighted in color on the left side of the editor, in addition to having an asterisk in the name of the script.

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

**Renaming Scripts**

To rename a script, right-click anywhere in the script panel, click **File -> Rename SQL Script** on the context menu or press **CTRL+F2**:

Then enter the new name in the Rename SQL script dialog box and click **OK**:

**Reverting Changes**

If you want to revert all changes made to the current SQL script and return it to its initial state (reload from disk), right-click anywhere in the script panel and click **File -> Revert** on the context menu.

**Changing default scripts directory**

By default all of the scripts are saved to a "Scripts" folder located in your project inside the workspace directory. This can be changed by clicking the **Configure** button in **Project Explorer** view. There you can click on a folder’s name and pick any other folder inside the Project.
Adding external directory

You can also link an external directory to your project to either save your scripts into it, or to access scripts that were created outside of DBeaver.

To link an External directory right-click anywhere in the Project Explorer and pick Create -> Link Folder. There you can link any directory on your drive to a project. This will allow you to open any externally created scripts through Project explorer and to set this folder as default to save new scripts into.

SQL Console

In some cases you might want to execute a query and not save it in a script. For example, when you read table data using “Read data in SQL console” or open procedure/function source from DDL editor. SQL console does not have an associated .sql file. Its contents will be lost when you close it.
**Client Side Commands**

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

DBeaver supports the following commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Database</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@set var = value</td>
<td>All</td>
<td>Sets a script variable. You can use expressions as a value. Variables can be used as SQL queries input parameters. For more information see <a href="#">Dynamic parameter bindings</a>.</td>
</tr>
<tr>
<td>@unset var</td>
<td>All</td>
<td>Unsets a script variable.</td>
</tr>
<tr>
<td>@echo message</td>
<td>All</td>
<td>Prints message to output log. You can use a macro in a message (for example <code>${var}</code>).</td>
</tr>
</tbody>
</table>
| @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.                                                                                           |
To enable interactive debugging of PL/SQL procedures on a Postgres server, you need to use the `_plugin_debugger_`. The `_plugin_debugger_` is a typical interactive debugger delivered as an extension. It requires a shared library preload in Postgres to operate the `shared_preload_libraries` parameter in the settings. The debugger is developed and maintained by EnterpriseDB. Its source code is available for examination and improvement.

The debugger provides the required server API for debugging PL/SQL procedures with:

- Breakpoint management;
- Step-by-step tracing;
- Variable acquisition and management.

### Installation

#### PostgreSQL 12 on Ubuntu-based distros

If you happen to have a PostgreSQL 12 installed via `apt`, then the procedure is quite straightforward:

```bash
sudo apt install postgresql-12-pldebugger
sudo service postgresql restart
```

After that, run the following command in the database or databases that you wish to debug functions in:

```
CREATE EXTENSION pldbgapi;
```

#### Installation from source code

You can find the source code in this repository. Installation instructions are located in the README file.

#### Running debugger in the DBeaver interface

Open the source code of the function you want to debug. To toggle breakpoints, place the caret on the line you want the function to be stopped at and use a shortcut `Shift + Control + B`. Alternatively, you can toggle the breakpoint with your mouse by clicking on a ruler, as demonstrated in the screenshot below:
NB. You must only toggle the breakpoints when Show header option is not on.

Then you need to set up a debugging configuration. Locate the downward-facing arrow right to the bug icon, click on it, then

**Debug As ➞ Database Debug:**

The *Edit Configuration* dialog opens. Set up input values in the table *Function parameters*.
Click on _OK_ button, and you are ready to go!

The usual buttons essential for debugging such as *Step Over* and *Continue* are located here:

```
DECLARE
cmd text;
    retval bigint;
BEGIN
    cmd := SELECT COUNT(*) FROM quote_ident(tabname);
    EXECUTE cmd INTO retval;
    RETURN retval;
END;
```
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 the relations between them. DBeaver allows you to view the diagrams of existing tables and whole database schemas, see Database Structure Diagrams. DB also allows the creation of custom diagrams, see Custom Diagrams. 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 structural changes in the diagrams:

- Add new tables to a diagram by drag-n-dropping them onto the diagram field from the Database Navigator.
- Rearrange tables in the diagram by dragging them all over the space. You can select several tables and drag them to a new
location.

- Auto-arrange tables into a compact view after manual rearrangements: click the **Arrange Diagram** in the toolbar or on the context menu (right-click anywhere on the diagram tab).

- (Available for Custom Diagrams only) - connect tables with a connector: click the **Show Palette** button in the upper-left corner of the diagram tab and then, in the Palette panel, click **Connection**.

![Diagram showing tables connected with a connector.](image)

Now click the tables that you want to connect with each other in turn, one by one. To stop the connection line, double-click the last table.

- (Available for Custom Diagrams only) - removes tables and connections: right-click the table or connection and click **Delete** on the context menu or just click the table or connection and press `Delete`.

### View Adjustment

You can adjust the view of any diagram in the following ways:

- Enable/disable the diagram grid: Click **Toggle Grid** in the toolbar.

- Modify attributes visibility: Right-click the diagram and, on the context menu, click **Show Attributes** and then select one of the options:
  - **All** - all attributes
  - **Any keys** - primary and foreign keys
  - **Primary key** - only primary keys
  - **None** - no attributes

- Modify attributes presentation: Right-click the diagram and, on the context menu, click **View Styles** and then select one of the options:
  - **Show Icons**
  - **Show Data Types**
  - **Show Nullability**
  - **Show Comments**
  - **Show Fully qualified names**

- Change the color of the entities/notes: Right-click the header of the entity or comment and then click **Set color** on the context menu. Then you can select the color and click **OK**.

- For elements located in front of/behind others, bring an element to the front or send it to the back: Right-click the element and then click **Bring to front** / **Send to back** on the context menu.

- Zoom the diagram in/out: Click the **Zoom In** / **Zoom Out** buttons or choose the scaling value in the dropdown list in the toolbar:

### Refresh

To see changes made by others to the database schema, you might need to refresh the diagram: click **Refresh Diagram** in the toolbar.
Notes
You can create notes only in Custom Diagrams. To create a note, click the Show Palette button ( \( \text{\textbullet} \) ) in the upper-left corner of the diagram tab. Then, in the Palette panel, click Note and click anywhere in the diagram tab. Now you can double-click the Note box to enter the note text:

Search in Diagram Entities
To search among entities of a diagram, click the Search items button ( \( \text{\textbullet} \) ) 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 ( \( \text{\textbullet} \) ) in the toolbar.

Diagram Printing
To print a diagram, press \( \text{CTRL}+P \) or click Print Diagram ( \( \text{\textbullet} \) ) on the toolbar.

Settings
To modify the diagram settings, click Configuration ( \( \text{\textbullet} \) ) on 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).

The ER diagram for a table shows the table itself and its relations with other tables inside the schema. To view the ER diagram for a table or view, double-click the table or view in the Database Navigator and then, in the Database Object Editor, switch to the ER Diagram tab:

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

NOTE: Table and schema diagrams are read-only. You can rearrange the layout, drag-n-drop elements inside a diagram but you cannot save the changes state or delete/add anything. This is because the diagrams represent the actual state of databases.

Relationship Notation

Lines representing the relationship between tables can look different depending on the nature of the relationship. Please note that any line can have only one beginning and one end. Even for one-to-many and other relationships that imply otherwise. In those situations you will just see more than one line.
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Solid Line" /></td>
<td>The solid line means that the foreign key column is also a primary key in a referencing table</td>
</tr>
<tr>
<td><img src="image.png" alt="Dashed Line" /></td>
<td>The dashed line means that the foreign key column is not a primary key in a referencing table</td>
</tr>
<tr>
<td><img src="image.png" alt="Black Dot" /></td>
<td>The black dot represents the beginning of the line and is attached to a table that has a foreign key referencing another table</td>
</tr>
<tr>
<td><img src="image.png" alt="White Rectangle" /></td>
<td>The white rectangle represents the end of the line and is attached to a referenced table. It only appears at the end of the dashed line</td>
</tr>
</tbody>
</table>

If the relationship between two tables is other than one-to-one, you will see multiple lines that all start at the same one point and all merge at the end point.
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 (optional) choose the initial diagram contents (set of tables):
The new diagram appears in a separate editor. Now you can drag-n-drop any number of tables into it. You can add tables from different connections as well as from different database types (for example, combine Oracle and MySQL tables in one and the same diagram).

You can also add notes and custom relations (associations) using the ERD palette on the left side of diagram tab - see details in the ER Diagrams article. For example, to create a diagram similar to the one shown at the beginning of this article, you need to:

1. Add required tables and relationships between them and move them around to create a well-shaped structure (see Structure Adjustment section of the ER Diagrams article).
2. Add notes (see the Notes section of the ER Diagrams article).
3. Stretch the notes to cover the intended tables, then send the notes to the back, and then set a color to the tables and notes (see the View Adjustment section of the ER Diagrams article).

Undo/redo functions are fully supported in diagram editing.
Edit mode

Note: This feature is available in Enterprise and Ultimate editions only.

Edit mode for ER Diagrams is a special feature that lets you create database objects while using the visual presentation of ER Diagrams. It can be accessed on the ERD tab of any database object that supports it. Changes made to a diagram in edit mode will generate an SQL script that can be executed to persist all the changes made. You can enable Edit mode by either choosing it from the context menu or by clicking a button on the bottom toolbar.

Working in Edit mode

To create an object you need to right-click on the diagram and select an object that you want to create. Foreign keys are created similarly to virtual foreign keys in custom diagrams. Just drag a column from a table for which the foreign key is being added and drop it in the referenced table. This will open a window that lets you choose a unique key in the referenced column.

Saving changes to a database is performed by clicking a save button. This will show you a preview of a generated script that you then can execute or cancel. The revert button will cancel all the changes that you did to a diagram after the last save.
Search

DBeaver provides:

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

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

Please see the dedicated articles for information about searching for different types. This article describes common features of the three search types.

Search View

Search results for any of the search types appear in a separate Search view. The following image shows the Search view for the database full text search:

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

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Run the Current Search Again</td>
<td>Refreshes the search results</td>
</tr>
</tbody>
</table>
|                      | 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:
Search History

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

To open the results of a previous search query, do one of the following:

- Click the arrow next to the Show Previous Searches button in the toolbar and then click the query in the dropdown list:

- Click the Show Previous Searches button itself or the arrow next to it and then History on the dropdown menu to open the Previous Searches window. Then, in the window, click the query and then either click Open to open it in the active Search view or click Open in New to open it in a new view:

To remove one or more of the previous search queries:

1. Click the Show Previous Searches button in the toolbar or click the arrow next to it and then History on the dropdown menu. The Previous Searches window opens.

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

3. Click Remove.

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

To search file contents for a string, click the Search button on the main toolbar or the arrow next to the Search button and then **File Search** on the dropdown menu:

![Search window](image)

The Search window opens displaying the File Search tab:

![Search window](image)

You can apply a case sensitive search, search by regular expressions, search among particular file types (**File name patterns** field), and use the find and replace function.

After you click **Search**, the results appear in a **Search view**. The results represent a tree or list of files with the search combination highlighted:
The toolbar of the Search view for File search provides more tools in addition to those available for all search types:

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Show Next / Previous Match</td>
<td>Open the file in a separate viewer and move the highlight to the next/previous match, respectively</td>
</tr>
<tr>
<td></td>
<td>Remove Selected Matches</td>
<td>Removes selected row (row in focus) of the results</td>
</tr>
<tr>
<td></td>
<td>Remove All Matches</td>
<td>Removes all results in the view</td>
</tr>
<tr>
<td></td>
<td>Expand/Collapse All</td>
<td>Expand/collapse the tree of results</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Menu item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show as List</td>
<td>Presents the results in the form of list</td>
</tr>
<tr>
<td>Show as Tree</td>
<td>Presents the results in the form of tree</td>
</tr>
<tr>
<td>Filters</td>
<td>Opens Search Filters dialog box</td>
</tr>
<tr>
<td>Preferences</td>
<td>Opens the Preferences window on the Search page</td>
</tr>
</tbody>
</table>

Double-clicking a results row opens it in a separate viewer.
DB Full-Text Search

To do a full text search in the database, click the arrow next to the Search icon in the main toolbar and then click File Search on the dropdown menu:

Alternatively, you can click the Search button on the main menu and then click the DB Full-Text tab in the Search window:

Now you need to choose the database connection or database objects against which to run the search – expand the tree in the Databases field to the database connections level or further down and select the checkboxes next to the required connections or database objects.

NOTE: The Search button is enabled only when you select the right level of checkboxes – database connections or lower nodes.

You can apply a case-sensitive search, fast search and search in numbers and LOBs.

After you click Search, the results will open in Search view:

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

Now you need to choose the database connection against which to run the search. You only need to select the database connection(s) in the Objects Source field.

In the Object Types field, you can select the database objects among which DBeaver will run the metadata search – select or clear the checkboxes.

You can specify that the object name starts with, contains or is similar to the search combination (Name match field). You can also set the maximum number of results to display (Max results field) and apply Case-sensitive search.

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

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

Schema compare/migration

You can compare two schema/database structures and generate a report in the following formats:

- DDL script (series of create/alter/drop statements)
- Diff diagram (sort of ER diagram)
- Liquibase change log
- Liquibase change report (json, yaml or plaintext)

Selecting objects to compare

- Select the two objects (schemas, databases, or tables) you want to compare
- Open the context menu
- Open the sub-menu Compare/Migrate
- Click on Compare/Migrate Schema element

Compare settings

Re-validate that you have chosen the correct objects to compare. You can also specify the types of changes to be processed: create, drop, or alter. By default, all types of changes are enabled.
For comparisons, table containers should be used. Schemes - if the database supports the schemas. Databases - if the database supports catalogs and does not support the schemes. Datasources - if there is no support schemas or catalog support (you can find an example below in "Compare schemaless bases").

You can exclude the specific compared types of objects.
For example, you can do this if you do not want to see the sequences, views, or external keys in the final comparison result.
Click on Compare Schemas to generate a diff report.

By default, DDL diff is generated. It contains a series of creating, alter and/or drop statements that will modify the schema on the right side. Thus it will make it identical to the schema on the left side.

You can enable/disable particular changes in the tree on the left side of the diff page:

You can also switch to another diff report representation (diagram, json, yaml, plaintext).
If you want to get acquainted with the comparison logs, you first specify the logging level on the Preferences->Editors->Schema Compare preference page. Specify one of the logging levels and click on Apply. By default, the logging level is the OFF level. To get maximum information you can choose the DEBUG level.

After comparing operations, you can click on the Show log button. A log will be open in the Editor and the content of this log will depend on the logging level you choose in the settings. Log level changes from preferences will not be applied to the comparison wizard if it is already open in another window. Close and open the schema compare wizard in this case.
Some bases (like SQLite and Firebird) do not have catalogs and schemes that can be selected for comparison. In this case (and only for these databases), it is possible to compare the entire datasource entirely.

Databases supporting schema comparison

- MySQL/MariaDB
- Oracle
- PostgreSQL
- SQLServer
- Snowflake
- SQLite
- Firebird
- Redshift
- DB2
- Informix
- Derby
- Greenplum
- Netezza
- Cockroach
Using schema compare with Liquibase PRO key.

If you have a Liquibase PRO key, then you can use it with DBeaver. Steps you need:

- Find and open your dbeaver.ini file. It is located in the DBeaver root directory.
- Find -vmargs command
- Add a new line after this command: -Dliquibase.license.key=yourKey (example: -Dliquibase.license.key=ABwwGgQU...)
- Open DBeaver and "Schema compare" window. The key will be checked at this step.

You can also add the Liquibase Pro key via UI in Preferences->Editors->Schema Compare preference page. Use the Import Liquibase Pro Key button to open the Import key dialog.

You can add your key in the Liquibase Key text field manually, throw the Paste button, or use the Load button to download a file. You can check the license state with the Check Key State button. After pressing the button, you can see the result of the checking in the Messages field.
We advise you to restart the program after adding a key for more correct program work. Settings changes will not be applied to the comparison wizard if it is already open in another window. The key will be saved in the DBFaver settings. If you specified the key in the .ini file, as well as installed another key through the Import Key dialog, then the key from the .ini file will be in priority.

If the license key is valid, then the **Object types** dialog will be extended on PRO objects. (If PRO objects didn't appear in schema compare changelog - check your logs. Maybe license expired or key is invalid)
Data compare

Sometimes you need to compare data from two sources (tables) which have almost identical data with just a few differences. There may be plenty reasons to do so: quickly visualize and navigate through all the differences; copy different rows or individual values; export them using Data Transfer.

Preparing the tool

1. You will need to choose one or two desired tables in the Database Navigator - it may be the tables from the same databases or from different databases or even from different RDBMS (e.g. PostgreSQL and MySQL).

2. Then choose Database ⇒ Compare/Migrate ⇒ Data Compare from the menu and the Data Compare Wizard will appear. Here you can preview selected tables or choose other ones. Then you can navigate to the second page.

3. On the second page you should choose the columns that will be used as a unique key during the comparison (the amount of columns must be equal). If the keys chosen are wrong, it may lead to invalid results. By default, if the table has a unique key in it, it will be chosen automatically during the initial setup.

![Data Compare Wizard](image)

Without unique keys the rows cannot be compared properly, since there is no way to distinguish between individual rows.

4. After reaching the last page, you can tweak limits (e.g. you only care about the first rows) and exclude categories of the resulting rows (e.g. you are not interested in modified rows). Also, you can press the Save task button to save the configuration in Task to use it later, or Schedule it.

Viewing the results

After finishing the wizard, you can press the Start button to begin the actual comparison process - it may take some time depending on the databases you are comparing, the amount of data in them, and your network speed (*).

When the results are ready, DBeaver will play a beep sound, and the editor will open:
Here you can examine the differences, swap the panels using the **Swap Containers** button in the toolbar, or preview the summary including the statistics by using the **Show Summary** button in the toolbar too. Everything you can do in the regular data editor is possible to perform here - you can copy data or transfer it to another database, except modify the values.

* - Please note that the actual preview is only available when Data Compare is launched through the wizard - otherwise only the statistics will be written into the **Task Log**.

**Under the hood**

This section is under construction. It will include information about the engine's implementation details and more.
MockData generation

Note: since version 6.2 MockData generator extension is available only in Enterprise Edition.

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.

Disclaimer: The idea behind Mock Data is to generate mock data in a table but it should NOT TO BE USED IN PRODUCTION ENVIRONMENTS. Please make sure you have a backup of your database before running the Mock Data generation process.

The following are features of the DBeaver Mock Data generator:

- Works for all the RDBMS that are supported by DBeaver (DB2, MS SQL Server, MySQL, Oracle, PostgreSQL, SQLite, etc.)
- Generates data that matches your database schema:
  - Generated data matches the database column types.
  - All base data types are supported.
  - Constraints (PK, FK, multi-column FK, unique) are supported.
- Supports over 20 configurable data generators (constants, randoms, sequences, names, domains, addresses, prices, regex based, etc.)
- Automatically associates a column with a generator based on the column characteristics
- Saves or overwrites old database data
The following are mock data generators for data types with their configurable parameters:

- **Boolean**
  - Random
  - Sequence (initial, order)

- **Date**
  - Random (start, end)
  - Sequence (start, step, reverse)

- **Numeric**
  - Random
  - Sequence (start, step, reverse)
  - Advanced (min, max, precision, scale)
    - Price preset
    - Coordinate preset

- **String**
  - Text (template, min length, max length)
  - UUID
  - Address
  - City
  - Country
  - Domain
  - Email (gender, with surname, numeric suffix)
- Name (gender, with surname)
- Price (country, min, max)
- Regex based random (regex template)
  - Credit Card preset
  - Email preset
  - Gender preset
  - HEX Color preset
  - IP4 address preset
  - IP6 address preset
  - Phone Number preset
  - Postal Code preset
  - Price preset
- Template with parametrized directives for other generators:
  - address() - US postal address
  - city() - one of the world's largest cities
  - country() - country
  - domain() - one of the top Internet domains
  - email(gender,surname) - e-mail address (gender is ALL|FEMALE|MALE, surname is true|false)
  - name(gender,surname) - personal name (gender is ALL|FEMALE|MALE, surname is true|false)
  - random(minimum,maximum) - random integer
  - regex(pattern) - regex based value for the pattern
  - sequence(start,step) - sequence of integers
- NULL values
- FK - data from the referenced table according to the constraint
### Mock Data Generator

**Mock data generator configuration**

Set mock data generator settings

<table>
<thead>
<tr>
<th>Entity: public.mockdata</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
</tr>
</tbody>
</table>
| Remove Old Data: 
| Rows: 11 |

**Generators**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>column1</td>
<td>Template</td>
</tr>
<tr>
<td>column2</td>
<td>Numeric Random</td>
</tr>
<tr>
<td>column3</td>
<td>Date Random</td>
</tr>
<tr>
<td>column4</td>
<td>Boolean Random</td>
</tr>
<tr>
<td>column5</td>
<td>Numeric Random</td>
</tr>
</tbody>
</table>

Generators for the red highlighted attributes aren't enabled.

- Template string can contain the directives like: `@generator(parameter1, parameter2,...)`. If they are processed by the appropriate generators.
- The available directives:
  - `address()` - US postal address,
  - `city()` - one of the world largest cities,
  - `country()` - country,
  - `domain()` - one of the top Internet domains,
  - `email(name, 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, range|pattern - regex based value for the pattern, sequence(start, step) - sequence of integers. ```

---

### Mock Data Generator progress

Mock Data Generator progress log

Removing old data from the 'mocked1'.
Rows updated: 15
Duration: 76ms

Inserting mock data into the 'mocked1'.
Rows updated: 11
Duration: 160ms
These features are available in the DBEaver Enterprise Edition only.
Dashboards, DB monitoring

Dashboards tool allows DBAs and programmers to quickly identify performance, disk space issues, the number of connections and other important KPIs associated with a single database connection. To learn more about database connections, see Database Connections.

By default, DBeaver is delivered with a number of predefined sets of dashboards for such data bases as PostgreSQL, MySQL, Oracle and Exasol. Custom dashboards are also supported. To learn more about custom dashboards, see the Managing Dashboards section below.

Managing Dashboards Panel

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

Opening Dashboard Panel

To open the dashboards panel press the Open Dashboard button on the main toolbar. The default configuration of the dashboards panel for the current database connection will appear. To learn more about database connections, see Database Connections.

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

The following controls are available on the dashboards panel toolbar:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Settings</td>
<td>Opens dashboard's configuration.</td>
</tr>
<tr>
<td></td>
<td>Add dashboard</td>
<td>Adds dashboard to the dashboard panel.</td>
</tr>
<tr>
<td>Icon</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Remove dashboard</td>
<td>Removes dashboard from the dashboard panel.</td>
</tr>
<tr>
<td></td>
<td>Reset dashboards</td>
<td>Restarts dashboard's calculation.</td>
</tr>
</tbody>
</table>

### Adding Dashboards

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

![Add Dashboard](image)

**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.

![Right Click Add Dashboard](image)

### Removing Dashboards

To remove a dashboard from the dashboards panel, click on the dashboard you want to remove and press button Remove dashboard in the dashboards panel toolbar or select Remove dashboard option in the dashboard's context menu.
Resetting Dashboards

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

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

To reset a particular dashboard right-click on it and select Reset dashboards menu option or left click a dashboard and press Reset dashboards button on the dashboards panel's toolbar.

Changing Dashboard Representation

To adjust dashboard representation settings, right click on a dashboard and select the Settings menu option, then, in the opened dialog change the parameters you want.
The following dashboard representation parameters can be adjusted:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Defines a name of a dashboard.</td>
</tr>
<tr>
<td>Description</td>
<td>Defines dashboard's description. Use this field to make it easy to understand what kind of information the dashboard represents.</td>
</tr>
<tr>
<td>Update periods(ms)</td>
<td>Defines how often dashboard's rendering should be updated. The default value is 1000 ms.</td>
</tr>
<tr>
<td>Maximum items</td>
<td>Defines maximum number of fetched items. The default value is 300.</td>
</tr>
<tr>
<td>View</td>
<td>Defines visual representation of the dashboard. The following options are available: Bar, Pie, Time series.</td>
</tr>
<tr>
<td>Show legend</td>
<td>If this check-box is selected, the legend will be displayed on the dashboard.</td>
</tr>
<tr>
<td>Show grid</td>
<td>If this check-box is selected, the grid will be displayed on the dashboard.</td>
</tr>
<tr>
<td>Show domain axis</td>
<td>If this check-box is selected, the domain axis will be displayed on the dashboard.</td>
</tr>
<tr>
<td>Show range axis</td>
<td>If this check-box is selected, the range axis will be displayed on the dashboard.</td>
</tr>
</tbody>
</table>

**Adjusting Dashboard Configuration**

To adjust dashboard's configuration settings right-click on a dashboard, select the **Settings** menu option, then, in the opened dialog box press the **Configuration** menu option.
The following dashboard parameters can be configured:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Defines dashboard's ID. Make sure that ID has numeric values in it.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines dashboard's name.</td>
</tr>
<tr>
<td>Database</td>
<td>Defines the database driver. To learn more about database drivers, see Database Drivers.</td>
</tr>
<tr>
<td>Data type</td>
<td>Defines the data type. The following options are available: timerseries (the default option) and statistics. Select timeseries type if you want to track the actual value returned by the server. Select the statistics type if your dashboard will show historical data.</td>
</tr>
<tr>
<td>Calc type</td>
<td>Defines how the data should be calculated. The following options are available: value (the default option) and delta. Select value if you are interested in the current value. Select delta if you want to track the difference between the current value and the previous one. This may be very useful when you work with statistics data, for example.</td>
</tr>
<tr>
<td>Value type</td>
<td>Defines the value to be shown on the range domain. The following options are available: decimal (the default option), integer, percent, bytes. Choose the value type in accordance with your data, for example, memory usage is convenient to be tracked in KBytes.</td>
</tr>
<tr>
<td>Interval</td>
<td>Defines time interval to be shown on the domain axis. The following time intervals are available: millisecond (the default option), second, minute, hour, day, week, month, year.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Fetch type</td>
<td>Defines whether the query should fetch data from rows or columns.</td>
</tr>
<tr>
<td>Description</td>
<td>Defines the description of a dashboard. Use this field to make it easy to understand what kind of information the dashboard represents.</td>
</tr>
<tr>
<td>Queries</td>
<td>Defines an SQL query whose fetched data will be used to build the chart displayed on the dashboard.</td>
</tr>
<tr>
<td>Default view</td>
<td>Defines the default visual representation of a dashboard on the dashboard panel. The following options are available: Bar, Pie, Time series (the default option).</td>
</tr>
<tr>
<td>Update period</td>
<td>Defines how often the dashboard's rendering should be updated.</td>
</tr>
<tr>
<td>Maximum items</td>
<td>Defines the maximum number of items to be fetched for the dashboard.</td>
</tr>
</tbody>
</table>

**Note:** Predefined dashboards are read-only and cannot be re-configured, but you can copy them and use them templates to create new dashboards with any query and other settings. To learn about creating new dashboards, see the Managing Dashboards section.

### Setting Connection Preferences

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

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

### Detaching Dashboards

If you have several monitors and would like to place a dashboard into a separate screen, you can either detach the whole dashboards panel or a single dashboard, and drag-and-drop them to any place you want.

To detach the whole dashboard panel, right click on the dashboard's tab name and select the **Detach** menu option.

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

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

Copying Dashboards to Clipboard

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

Saving Dashboards

If you want to save a screenshot of a dashboard locally in PNG format, right click on it and select the Save as ... option in the context menu displayed.
Printing Dashboards

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

Zooming

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

- **Zoom In**
- **Zoom Out**
- **Zoom Reset**

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

Creating Dashboards

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

To create a dashboard from scratch:

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

To create a dashboard from a template:

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

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

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

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

**To delete a dashboard:**

1. Press the **Settings** button on the dashboards panel toolbar.
2. In the opened dialog box click **Manage...** dashboards.
3. In the **Manage dashboards** window select any of the custom dashboards from the list and click **Delete**.
Managing dashboards in DBeaver:

1. Open the 'Manage dashboards' window.
2. Select the dashboard you wish to delete.
3. Click on the 'Delete' button to remove the dashboard from the list.

Predefined dashboards are read-only, but you can copy them as needed.
Projects

The Projects view allows the creation of new projects as well as renaming and deleting projects that are not active. NOTE: You cannot rename or delete a project that is set as active.

Creating Project

To create a project, in the Projects view, in the toolbar, click Create Project. The Project Create Wizard opens.

1. In the Project screen, in the Project name field, specify the name of the project.
2. To keep the default location to store the project, leave the Use default location checkbox selected. If you want to change the location, clear the checkbox and enter the name of the new directory into the Location field or click Browse and select the directory in the folder tree.
3. Click Finish. The new project appears in the projects tree.

Deleting Project

To delete a project, in the Projects view, right-click its name in the tree and click Delete on the context menu. Two confirmation dialog boxes will appear one after another:

1. Delete object dialog box is to confirm the deletion of the project itself. Click Yes if you are sure you want to delete it. Otherwise, click No.
2. Delete project dialog box is to confirm the deletion of the project’s contents. These are the project configuration files and scripts stored in the file system. Click Yes if you want the contents to be deleted as well. To keep the contents, click No.

NOTE: If you have deleted a project and then re-create it with the same name, the new project picks up all the database connections of the deleted project.
Project security

Note: This functionality is available only in Enterprise-Edition.

DBeaver supports local storage for connection secure data. It includes:

- Database server user credentials
- SSH tunnel user credentials
- Proxy user credentials

By default, user names and passwords are stored in file credentials-config.json. This file is encrypted using the AES key. However the key is insecure because it is found in the DB open-sources and thus this file can be un-encrypted by 3rd party people using some 3rd party software.

In the DBeaver Enterprise, the security support is much safer because of its strong encryption.

Master password for local configuration

It is possible to set a master password for all projects in a local workspace. Go to Preferences->Database->Security and enable the option Use secure passwords storage. There are several password storage providers (you can see them on page General->Security->Secure Storage), DBeaver Enterprise Password Provider is the default one (in standalone DBeaver). It will ask you to specify master password. DBeaver doesn't store this password anywhere, it only encrypts user credentials in a special local storage. It is not possible to decrypt this password without a password (at least easily).

The side effect of this configuration is that you cannot share your connections (with password) between different users because user credentials are stored in a completely separate location and they are protected by a local user password.

Use Windows Integration password provider

You can disable the default password provider and enable a “Windows Integration” provider. This provider does not need a master password but it uses a randomly generated password stored in a local user secure storage (in Windows). This is easier (as you don't need to remember the master password) but less secure (anybody who has access to your Windows user account will have access to DBeaver's stored credentials).

Project password

You may specify a password for a project. It will encrypt all the project's configurations with this password. Also, you will be able to share your project settings with other users (you will need to pass the project password as well).

In order to enable a project password open the project properties. You can do this by:

- Clicking on main menu File->Project security
- Clicking on “Configure” icon in the project explorer view toolbar and switching to the Project Security tab
- Pressing ALT+Enter on a project element in Projects view and switch to Project Security tab
On the project security page click on the "Set Password" button to enable the project password. Click on Clear to disable it (you will need to enter a current project password to clear it).

"Encrypt configuration" option
Bookmarks

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

To create a bookmark:

1. In the Database Navigator or under Connections node of the Projects view, click the database object of interest to focus on it.
2. Press CTRL+ALT+SHIFT+D. The Bookmark Name dialog box appears.
3. In the Bookmark Name field, enter the bookmark name, then in the Bookmark folder field, click the folder, and then click OK:

   ![Bookmark Name dialog box]

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

This is a brief list of the most important DBeaver shortcuts. 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's main menu, context menu or editor toolbar (or from all of them). Use the context menu wherever it is possible - it usually shows all actions accessible at this moment.

### SQL Editor

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

### Data viewer

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

### Data editor

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shortcut</td>
<td>Action</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Enter</td>
<td>Activate inline editor</td>
</tr>
<tr>
<td>Shift+Enter</td>
<td>Open value editor dialog or separate value editor (for LOB values)</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete row</td>
</tr>
<tr>
<td>Alt+Delete</td>
<td>Delete row</td>
</tr>
<tr>
<td>Ctrl+Alt+Insert</td>
<td>Add new row</td>
</tr>
<tr>
<td>Ctrl+Alt+Insert</td>
<td>Copy current row</td>
</tr>
<tr>
<td>Escape</td>
<td>Cancel changes in current cell/row</td>
</tr>
</tbody>
</table>

**Database Navigator**

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2</td>
<td>Rename current element (if supported)</td>
</tr>
<tr>
<td>F4</td>
<td>Open editor of selected element(s)</td>
</tr>
<tr>
<td>F5</td>
<td>Refresh selected element(s)</td>
</tr>
<tr>
<td>Delete</td>
<td>Delete selected element(s) (if supported)</td>
</tr>
<tr>
<td>Ctrl+Alt+Shift+B</td>
<td>Add bookmark on selected element</td>
</tr>
<tr>
<td>Alt+Enter</td>
<td>Show properties of selected element</td>
</tr>
<tr>
<td>F3 Ctrl+[</td>
<td>Open SQL editor for current connection (**`). Shows script selector popup.</td>
</tr>
<tr>
<td>Ctrl+Enter</td>
<td>Open recent SQL editor for current connection (**`). Opens last modified script or creates a new script.</td>
</tr>
</tbody>
</table>

**Other**

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alt+~</td>
<td>Shows database tools context menu</td>
</tr>
<tr>
<td>Ctrl+0</td>
<td>Switch active schema/catalog (available if SQL/database editor is open)</td>
</tr>
<tr>
<td>Ctrl+Shift+C</td>
<td>Advanced copy. Works in different contexts and performs &quot;smart copy&quot; operation (usually with parameters).</td>
</tr>
<tr>
<td>Ctrl+Shift+V</td>
<td>Advanced paste. Same as &quot;smart copy&quot; but for &quot;paste&quot;.</td>
</tr>
</tbody>
</table>

**References**

* - Current query is the query under cursor or the selected text. Query is separated from other script queries by delimiter (; by default) or by empty lines.
** - Current script is a set of all queries in the current SQL file. If there is a text selection then only queries in this selection are processed. Queries are separated from each other with a delimiter (; by default).
*** - Current connection detected from active window and selection. If active (focused) window is SQL editor or database object editor then the current connection is the same as in this editor. If the active window is the database navigator then the active connection is the "owner" connection of the currently selected element. In other cases there is no current connection and DBeaver will ask you to choose the connection explicitly.
**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:

- ![icon](image) - not connected
- ![icon](image) - has network settings specified (such as SSH tunnel, etc.)
- ![icon](image) - connected
- ![icon](image) - 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:

  ![New Connection Wizard](image)

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

  ![New Database Connection](image)

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

  ![New](image)

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 displays the number of existing 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.

You can choose the Simple mode on this step. Simple mode gives simplified access to the database, which is basically with the ability to view data only in schemas and tables.

2. In the Connection Settings screen, on the General tab, set all primary connection settings:
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:
DBeaver supports following SSH authentication methods: user/password, public key authentication and agent authentication. Supported implementations for agent authentications are pageant and ssh-agent.

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

More information about SSH configuration can be found on SSH configuration page.

**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.).
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.
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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>${host}</td>
<td>Host name</td>
</tr>
<tr>
<td>${port}</td>
<td>Port number</td>
</tr>
<tr>
<td>${database}</td>
<td>Database name</td>
</tr>
<tr>
<td>${server}</td>
<td>Server name</td>
</tr>
<tr>
<td>${url}</td>
<td>Connection URL</td>
</tr>
<tr>
<td>${user}</td>
<td>User name</td>
</tr>
<tr>
<td>${password}</td>
<td>User password</td>
</tr>
</tbody>
</table>

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

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

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

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

Driver settings

- In development

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

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

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

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

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

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

To disconnect from all active connections, click Database -> Disconnect All on the main menu.
DBeaver has a possibility to change credentials for the current database user.

Databases, which support this feature:

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

Changing expired user password

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

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

How to change user password

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

1. Connect to the database.
2. Open the context menu by right-clicking on the connection in the Navigation tree.
3. Select a “Security” point and click on a “Change user password” point in a sub-menu.
4. When a new password input dialog opens enter the new password and confirm it.
5. Confirm the password changes. (This dialog will not appear if the entered password was incorrect).

The password has been changed.
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 the required user confirmation for SQL execution, DBeaver shows a confirmation message every time you attempt to execute a ‘transaction’ type of query (INSERT/DELETE/UPDATE, etc.):

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

To manage connection types for a database connection, in the Database Navigator or Projects view, click the connection to set the focus on it and then press F4 to open the connection properties window. Then, in the properties window, in the navigation pane on the left, click General to see the general settings. You can see Connection Type field among the settings:
There are three default connection types – Development, Test, and Production. You can change the connection type for your database connection as well as you can create a new connection type, edit or delete an existing one.

**Change Connection Type**

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

To change the connection type:

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

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

**Create Connection Type**

To create a connection type:

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

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

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

**Auto-Commit Mode**

Auto-commit mode is the default one for the Development and Test connection types, see [Connection Types](#). Auto-commit mode is on if you can see the auto-commit view of the mode selection button ( ) in the application toolbar. If you see the manual commit view ( ), then in order to switch to auto-commit mode, click the mode selection button – it changes to auto-commit. At the same time, this disables the two manual commit buttons in the toolbar: Commit and Rollback – these are available only in manual commit mode.

The statistics field next to the mode selection button always shows **Auto** in auto-commit mode:

Clicking the statistics field opens the Transaction Log.

**Manual Commit Mode**

Manual commit is intended to protect your database from inadvertent changes and that is why it is the default mode for Production connection type, see [Connection Types](#).

Manual commit mode is on if you can see the Manual commit view of the mode selection button ( ) in the application toolbar. If you see the auto-commit view ( ), then in order to switch to manual commit mode, click the auto-commit button – it changes to manual commit. At the same time, this enables the two manual commit buttons in the toolbar: Commit (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:

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

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

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

Clicking the statistics field opens the Transaction Log.

**Smart Commit Mode**

When smart commit is enabled and you are in auto-commit mode, which allows DBeaver to monitor your activity. When you try to execute any data modifying query (UPDATE, INSERT, DELETE, UPSERT, MERGE, etc) DBeaver will switch to manual commit mode before executing your query. Also if you edit any table data and save your changes - DBeaver will also switch to manual mode before the actual data modification.

If the option “Return to auto-commit on transaction end” is on, then DBeaver will switch back to auto-commit mode once you have executed Commit or Rollback command (using the main toolbar or the main menu actions).

Smart commit mode is very useful if you work mostly in the read-only mode. It does not lock tables when you perform SELECT queries. The transaction will only be started when you start to modify your data.

**Transaction Isolation Level**

For both, Auto and Manual commit modes, you can select the transaction isolation level. To do so, click the arrow next to the mode icon and then click the required option in the dropdown list:
<table>
<thead>
<tr>
<th>Switch to manual commit (Repeatable read)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read committed</td>
</tr>
<tr>
<td>Read uncommitted</td>
</tr>
<tr>
<td>Repeatable read</td>
</tr>
<tr>
<td>Serializable</td>
</tr>
</tbody>
</table>
The Transaction Log shows all transactions (queries of ‘transaction’ type such as INSERT/DELETE/UPDATE and others) made during the current DBeaver session. To open the Transaction Log, click the Transaction log button in the toolbar or the statistics field to the left of it.

The Transaction Log window shows transactions that are:

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

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

It might be useful to check your pending transactions because they might lock your database. To see your pending transactions, click the arrow next to the Transaction Log button in the toolbar and then click Pending Transactions on the dropdown menu:

The upper table of the Pending transactions window shows currently active connections and the number of their transactions. The bottom table shows query details of the connection that is currently in focus in the upper table:

When the Pending transactions window opens, the upper table shows only those connections that have pending transactions. If no connections have pending transactions, the table is empty. To see all connections that are currently active (connected), select the Show all connections checkbox.

You can commit or roll back transactions right from the Pending transactions window: in the upper table, click the row with the required uncommitted transactions and then click the Commit or Rollback button, depending on your purpose. If a transaction is committed/rolled back successfully, both buttons are disabled. If the operation is unsuccessful, the system displays an error message.

To see all previous transactions made during the current session, select the Show previous transactions checkbox. To see all queries including non-transactional ones, select the Show all queries checkbox. The green rows are committed transactions, the orange (or red) ones are rolled back, rows without a special color are non-transactional or pending transactions.
Database drivers

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

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

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

Obtaining JDBC driver

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

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

Adding driver configuration in DBeaver

Open driver manager dialog

You can open the driver manager from the main menu:
Add a new driver

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

Main parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

- DBeaver user guide 170/285
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver Name</td>
<td>Name of your driver. It can be any name you like</td>
</tr>
<tr>
<td>Driver Type</td>
<td>Driver provider. In 99% cases you will need a generic driver (JDBC provider)</td>
</tr>
<tr>
<td>Class Name</td>
<td>JDBC driver class name. You can get it from the documentation or find it in the jar files (see “Find Class” button description)</td>
</tr>
<tr>
<td>URL Template</td>
<td>Template of driver URL. You can leave it empty. But in this case you will be ready to set JDBC URL for each your connection. It is better to define a valid template, which will greatly simplify the connections creation. See “URL Templates” for a detailed description</td>
</tr>
<tr>
<td>Default Port</td>
<td>Default database port. You can get it from the documentation or leave it empty</td>
</tr>
<tr>
<td>Embedded</td>
<td>Enable it for server-less databases. This flag affects a few config options related to the network/connections management</td>
</tr>
<tr>
<td>No Authentication</td>
<td>This means that driver does not require authentication (no user/password fields will be shown)</td>
</tr>
<tr>
<td>Category</td>
<td>Driver category, deprecated</td>
</tr>
<tr>
<td>ID</td>
<td>Driver unique ID, ignore it</td>
</tr>
<tr>
<td>Description</td>
<td>Driver description, it is shown in some dialogs/wizards as a hint</td>
</tr>
</tbody>
</table>

**Libraries**

This is the list of jar files, binary libraries (dll or so) and any other files required by the driver. In most cases you only need the jar files. Click "Add File" to add a single jar file, "Add Folder" to add to the folder with Java classes/resources and "Add Artifact" to add the Maven artifact (see below).

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

**Maven artifacts**

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

**Saving driver, adding connection**

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

Now you can create connection.

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

---

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

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

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

Supported URL variables:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>{host}</td>
<td>Database server host name</td>
</tr>
<tr>
<td>{port}</td>
<td>Database server port number</td>
</tr>
<tr>
<td>{database}</td>
<td>Target database name</td>
</tr>
<tr>
<td>{server}</td>
<td>Target server name (rarely used)</td>
</tr>
<tr>
<td>{folder}</td>
<td>Folder path (on the local file system). Used for embedded drivers</td>
</tr>
<tr>
<td>{file}</td>
<td>File path (on the local file system). Used for embedded drivers</td>
</tr>
</tbody>
</table>

**Advanced settings**

For most drivers you do not need to change any advanced properties. But in some cases you can use this as driver tuning, e.g. for better performance or for structure fixing.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Driver supports indexes</td>
<td>Driver supports table indexes</td>
</tr>
<tr>
<td>Driver supports stored code</td>
<td>Whether this driver supports stored code (procedures, functions, packages, etc)</td>
</tr>
<tr>
<td>Driver supports references</td>
<td>Driver supports table references (foreign keys)</td>
</tr>
<tr>
<td>Driver supports SELECT count(*) clause</td>
<td>Driver supports SELECT count(*) clause</td>
</tr>
<tr>
<td>Driver supports views</td>
<td>Driver supports table views</td>
</tr>
<tr>
<td>Split procedures and functions</td>
<td>Show procedures and functions in different folders</td>
</tr>
<tr>
<td>Script delimiter</td>
<td>Literal for SQL queries separation in scripts</td>
</tr>
<tr>
<td>Script delimiter redefiner</td>
<td>SQL clause which redefines script delimiter value</td>
</tr>
<tr>
<td>Use script delimiter after query</td>
<td>Keep SQL script delimiter after each SQL query</td>
</tr>
<tr>
<td>Use script delimiter after SQL block</td>
<td>Keep SQL script delimiter after SQL script blocks (BEGIN/END)</td>
</tr>
<tr>
<td>String escape character</td>
<td>Character used to escape special symbols in strings</td>
</tr>
<tr>
<td>Meta model type</td>
<td>Type of metadata reading model - standard or indexed</td>
</tr>
<tr>
<td>All Objects Pattern</td>
<td>SQL pattern for all metadata objects</td>
</tr>
<tr>
<td>Omit catalog(s)</td>
<td>Do not read and use catalog (aka database) information</td>
</tr>
<tr>
<td>Omit single catalog</td>
<td>Hide catalog (database) if there is only one catalog on server</td>
</tr>
<tr>
<td>Omit schema(s)</td>
<td>Do not read and use schemas information</td>
</tr>
</tbody>
</table>
Omit single schema
Hide schema if there is only one schema on the server

Use schema filters
Use JDBC schema filters when the database does not support catalogs. Otherwise just read all database schemas and filter on client-side

Omit type cache
Do not use data types provided by driver

Shutdown parameter
Database shutdown URL parameter

Create database parameter
Database create URL parameter

Driver supports multiple results
Driver supports multiple results for a single query

Driver supports result set limit
Driver supports multiple result set limit (max rows)

Driver supports structure cache
Driver supports structure cache reading. Enables schema columns, keys, etc

Driver supports TRUNCATE operation
Driver supports TRUNCATE command. It is much faster than DELETE without criteria

### Queries (Custom driver queries)

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

### DDL (DDL generation options)

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

### Formatting (SQL values formats)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp format</td>
<td>Format pattern for timestamp columns</td>
</tr>
<tr>
<td>Date format</td>
<td>Format pattern for date columns</td>
</tr>
<tr>
<td>Time format</td>
<td>Format pattern for time columns</td>
</tr>
</tbody>
</table>

### Summary

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

Thank you.
Data export/import

You can perform data export/import or migration for database table(s). We will describe most typically used cases.

Exporting table data to CSV format

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

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

3. Set data extraction options (how the data will be read from the tables). This may affect the extraction's performance. And set export format option. They are specific to the data format you chose on step 2:
4. Set options for output files or clipboard. Note: Timestamp pattern is used here to target the file name pattern:

5. Review what you want to format and into which format you will export it. You can also save all your settings as a task in this step or change the task variables:
6. Press finish. See extraction progress. You can keep working with your database during the export process as the extraction will be performed in the background. Note: avoid changing data in tables you have selected to be exported while the exporting is in progress. In the end you will see status message:

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":

![Image of DBeaver Data Transfer window]

Variables ... Save task Open Tasks view
2. Choose import format (CSV);

3. Select input CSV file for each table you want to import and you can change the Importer settings (format specific) at this step:
4. Set CSV-to-table mappings. You need to set a column in the CSV file for each database table column. You can skip columns (the value will be set to NULL in the target table column). You can set constant values for the table column if there is no source column for it in the CSV.
5. Set options for loading data in the database. These options may affect the loading's performance:

![Data Transfer Settings](image)

About the replacing method option, you can read here.

6. Review which file(s) and to which table(s) you will import. You can also save all your settings as a task in this step:

![Data Transfer Confirm](image)

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

![Transfer data](image)

Related topic: Migrating table(s) data to another database table(s)
Data migration

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

To perform a 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, choose “Export results” in the results context menu).

**Step 2: Define data transfer target type**

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

**Step 3: Define data mapping**

For proper table mapping, the following options are available:
- **Target container** - defines a database or a scheme where the data will be transferred to. Press the button and choose the container.

- **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 the button.

- **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 the cells are marked with ![question mark](?), it means that in the target table there are no columns with matching names, otherwise the names will be filled in automatically.

You may also want to transform the values of some columns during the transfer. To do that, define column transformers by clicking on corresponding cells in the **Transform** column. You can choose one of three options:

- **Set to NULL.** All values in the corresponding column will be set to null.
- **Constant.** All values in the corresponding column will be set to the specified constant.
- **Expression.** This transformer uses expressions (namely, JEXL expressions) to calculate the cell's value. You can use basic arithmetic operations and column names to construct an expression.

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 button](New) and enter a new name in the opened dialog box.

```
<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Mapping</th>
<th>Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>column1</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>column2</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>column3</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>column4</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>column5</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>column6</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>column7</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>column8</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>column9</td>
<td>create</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>column10</td>
<td>create</td>
<td></td>
</tr>
</tbody>
</table>
```

* DEL - skip column(s)  SPACE - map column(s)
You can also choose a name for a target table from the drop-down list.

Or select a table from the existing tables in the target container by pressing the Browse button.
To define the 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 collapse the list, press .

If you want tables of the target container to be named like those of source, press the Auto assign button 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.

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

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

If you want to see a preview of the imported data, you can select the **Preview data** button.
The following keyboard shortcuts for easy navigation within the mapping table area of **Table mapping** screen are supported:

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up</td>
<td>Moves one row up.</td>
</tr>
<tr>
<td>Down</td>
<td>Moves one row down.</td>
</tr>
<tr>
<td>Right</td>
<td>Releases the list of source table columns.</td>
</tr>
<tr>
<td>Left</td>
<td>Swaps the list of source table columns.</td>
</tr>
<tr>
<td>Space</td>
<td>Auto-assigns the target.</td>
</tr>
<tr>
<td>Del</td>
<td>Sets mapping type to skip.</td>
</tr>
</tbody>
</table>

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:

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

Data load settings

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

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncate data load table before load</td>
<td>Select this check-box only if you want all the data be cleared from the target table. Be very careful with this option!</td>
</tr>
<tr>
<td>Replace method</td>
<td>Select this drop-down list if you want to change the import behavior in case of a duplicate primary key value. <a href="https://dbbeaver.io/docs/tutorials/data-import-replace">Data Import and Replace page</a></td>
</tr>
<tr>
<td>Open new connections</td>
<td>Use this option to speed up data transfer. If selected, a new connection will be opened and the data transfer will not interfere with other calls to the database where data is being transferred to.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use transactions</td>
<td>This option allows you to speed up the data transfer and to define the number of rows for each transaction by setting the <em>Commit after insert of</em> parameter.</td>
</tr>
<tr>
<td>Use multi-row insert</td>
<td>Use multi-row insert with extended values number for higher performance. Database-specific setting.</td>
</tr>
<tr>
<td>Skip bind values during insert</td>
<td>This option can drastically increase performance for some drivers like Redshift by skipping a process of binding values and setting them directly, but it opens up a vulnerability to SQL injections. Not recommended if you are not sure of imported file contents.</td>
</tr>
<tr>
<td>Disable batches</td>
<td>Select this check-box if you want to disable the use of batch imports. The import will be made row by row. Enabling this function will show all import errors, but make the import process slower.</td>
</tr>
<tr>
<td>Open table editor on finish</td>
<td>If selected, the table editor is to be opened when data transfer is finished.</td>
</tr>
<tr>
<td>Show finish message</td>
<td>If selected, a notification message will be shown when the transfer is finished.</td>
</tr>
</tbody>
</table>

### Step 5: Confirm

Check out the data transfer settings and press **Start** or save as **task**.
Sometimes there are situations when you want to ignore the current primary key value when importing into a table. Some databases have syntax constructs in addition to the INSERT INTO that may help.

The choice of the replacement method is in the import settings - in “Data load settings”.

By default, the selection is <None>, you can select other options from the drop-down list. The options available depend on the target database you are importing to.

The database can only support the replace method or the ignore method. In this case, the list of methods will consist of only one item except <None>. If the base does not support the replacement methods, or if we have not added an implementation yet, then the combo with the list will be disabled.

Further, you will find a list of databases supporting these methods and examples of syntax.

Let’s take a look at an example of how this works. We use a small, simple, slightly-modified Sakila (MySQL) table - sakila.language

```sql
CREATE TABLE language_insert (  
  language_id tinyint unsigned NOT NULL,  
  name char(20) NOT NULL,  
  last_update timestamp NOT NULL,  
  PRIMARY KEY (language_id)  
);
```
```sql
INSERT INTO sakila.language_insert (language_id, name, last_update) VALUES
(1, 'English', '2006-02-15 05:02:19.0'),
(2, 'Italian', '2006-02-15 05:02:19.0'),
(3, 'Japanese', '2006-02-15 05:02:19.0');
```

If we try to execute this request twice, we will get the following error: SQL Error [1062] [23000]: Duplicate entry '1' for key 'language_insert.PRIMARY' (This message may look different in other databases).

Let’s take a new .csv file with the following content and try to use the replace methods.

```
"language_id","name","last_update"
1,Spanish,"2020-04-20 05:02:19.0"
2,Russian,"2020-05-20 05:02:19.0"
3,Belgian,"2020-06-20 05:02:19.0"
4,Mandarin,"2006-02-15 05:02:19.0"
5,French,"2006-02-15 05:02:19.0"
6,German,"2006-02-15 05:02:19.0"
```

If we set the "INSERT IGNORE" method in the settings, the result of the insert will look like this:

There will be no insertion errors, the first three lines will not change, and the fourth to sixth lines will be added to the table.

If we set the "REPLACE INTO" method in the settings, the result of the insert will look like this:

There will be no insertion errors, the first three lines will be replaced and the fourth to sixth lines will be added to the table.

**Which databases support replace/insert methods?**

**MySQL**

**INSERT IGNORE** and **REPLACE INTO**

Insert examples:

"INSERT IGNORE"
**SQLite**

**Documentation**

"INSERT OR IGNORE" and "INSERT OR REPLACE"

Insert examples:

"INSERT OR IGNORE"

```
INSERT OR IGNORE INTO language_insert(language_id, name, last_update)
VALUES (1, 'English', '2006-02-15 05:02:19.0');
```

"INSERT OR REPLACE"

```
INSERT OR REPLACE INTO language_insert(language_id, name, last_update)
VALUES (1, 'English', '2006-02-15 05:02:19.0');
```

**PostgreSQL**

Available for PostgreSQL version 9.5.

"ON CONFLICT DO NOTHING" and "ON CONFLICT DO UPDATE SET"

Insert examples:

"ON CONFLICT DO NOTHING"

```
INSERT INTO language_insert(language_id, name, last_update)
VALUES (1, 'English', '2006-02-15 05:02:19.0') ON CONFLICT DO NOTHING;
```

"ON CONFLICT DO UPDATE SET"

```
INSERT INTO language_insert(language_id, name, last_update)
VALUES (1, 'English', '2006-02-15 05:02:19.0')
ON CONFLICT (language_id)
DO UPDATE SET (language_id, name, last_update) = (EXCLUDED.language_id, EXCLUDED.name, EXCLUDED.last_update);
```

**FireBird**

Available for FireBird version 2.1.

"UPDATE OR INSERT INTO"

Insert examples:

"UPDATE OR INSERT INTO"

```
UPDATE OR INSERT INTO language_insert(language_id, name, last_update)
VALUES (1, 'English', '2006-02-15 05:02:19.0');
```

**Oracle**

Available for Oracle version 11.2.

"INSERT IGNORE ROW INDEX"

Insert examples:

"INSERT IGNORE ROW INDEX"
/* + IGNORE_ROW_ON_DUPKEY_INDEX (LANGUAGE_INSERT, LANGUAGE_INSERT_PK) */

INSERT INTO LANGUAGE_INSERT (LANGUAGE_ID, NAME, LAST_UPDATE) VALUES (1, 'English', TIMESTAMP '2006-02-15 05:02:19.0');
Database backup/restore

Database Backup/restore

DBBeaver supports native database backup/restore functions for the following databases:

- PostgreSQL
- MySQL

The native backup restore differs from the standard DBBeaver data transfer feature. It uses database native dump formats and it may work much faster as it uses special utilities for the direct high-performance database access.

These functions can be accessed from the Context Menu's Tools or the Main Menu's Database->Tools.

Native client configuration

In order to execute native backup/restore tools you need to configure the database native client. The native client is a set of binaries (different for different OSes) which will be executed by DBBeaver to process an actual backup/restore. The native client configuration can be done in driver editor dialog or directly from the backup/restore wizard. Just click on the Client ... button in the button bar:

To configure a new client location, choose the Browse ... item and add a new client in the following dialog:

Database dump object selector

You can choose what schemas/tables you want to backup/dump:
You can pass a set of additional dump/restore parameters to the native tool. The particular set of configuration options depends on the database type.

Database native tool configuration

You can pass a set of additional dump/restore parameters to the native tool. The particular set of configuration options depends on the database type.
Task management

Creating tasks

The task is a saved configuration of a database tool. It can be started from the task management view or from the menu by a single click. You can create tasks for frequently used tools. Also, tasks can be scheduled for regular execution.

Create a task from tool configuration

You can save the tool configuration into a task and run your task later with a single click. For example, you can start the Data Transfer wizard and configure the data export from several tables in the SQLite database into CSV files:

Click on the Save configuration as task button and fill the task properties:

Now click on the Open Tasks view link to open the task list:
You can configure the Database Tasks View to see more or fewer View columns. Right-click inside the Database Tasks tab and choose the Configure columns button from the menu.

Also, you can create a new task from the main menu Database -> Tasks -> Create new task...
Editing/running tasks

DBeaver user guide
From the task view you can add, edit, remove and execute saved tasks. You can use the context menu or view tools for that:

By clicking on **Edit** or by double-clicking on a task you can open the tasks edit wizard. In this wizard, you can change the task settings as well (use button **Back**) as the actual tool configuration. You can change the set of input objects for data transfer or any export configuration. After changing the task settings, click on the **Save task** button (it is on the last page of the task configuration wizard).
Create a task from task management view

You can create a task from scratch using the tasks view. Open tasks view and click on the Create new task button in the View toolbar or in the context menu. In the task wizard, you can choose the task category, task type, task folder, and name. On the next wizard pages, actual tool configuration pages will be shown (they depend on the chosen task type).

Tasks folders

For better structuring, you can store your tasks in the folders. Create folder can also be from the context menu Database Tasks View.

You can choose a project for the new task folder and add a new unique name.

You can change the task folder for an existing task in the task Edit dialog (use button Back). Or you can drag your task to another task folder in the View.
The task folder as a task can be deleted with the button Delete.

**Scheduling tasks**

You can schedule tasks for later/regular execution. See the Task Scheduler article.
Task scheduler

Note: This functionality is available in Enterprise and Ultimate editions only.

DBeaver can schedule an execution for regular tasks. DBeaver supports Windows Task Scheduler on Windows and cron on macOS and GNU/Linux. In addition, you can manually configure schedulers using command line.

Scheduling tasks from the Tasks view

Windows

You can open the tasks view from the main toolbar:

![Database Tasks menu]

or from the main menu Window. Select a task that you want to schedule in the tasks view and open the context menu:

![Scheduler context menu]

The scheduler configuration dialog will be opened. You can configure task frequency, recurrence period, and start time there:
To schedule the task, click on the Schedule button. If everything is configured correctly, you will see the confirmation dialog:

If anything goes wrong, you will see an error message dialog. Error details can be viewed in the Error Log view.

You can change the scheduler settings at any moment by choosing Edit scheduled task command from the context menu. You can also cancel the schedule by clicking on Remove schedule.

**macOS or GNU/Linux**

You first need to open the tasks view. There are three ways to do that:

1. Database -> Tasks -> Database Tasks
2. Window -> Database Tasks

3. Click on 'Show View (Database Tasks)' icon

Select a task you want to schedule in the tasks view. To open the scheduler dialog, either:

1. Open the context menu with right-click -> Scheduler -> Schedule task
The scheduler dialog will be opened. It has a lot of similarities with the corresponding dialog in Windows, but unfortunately, there are fewer settings on macOS and GNU/Linux due to the limitations of cron. For instance, when configuring an hourly task, you can only choose the minute at which the task is to be executed. In the example below, the task executes at 1:42 PM, 2:42 PM, 3:42 PM, and so on:

There is also no start date option, and, in case of minutely tasks, no start time either. The scheduler will execute the task at the specified time, but there are no guarantees about when the execution will start. It is also worth pointing out that even though you can specify the seconds in the start time selector, they will be ignored. Even though we try to be compliant with as many cron implementations as possible, most cron implementations do not support this type of granularity.

On macOS 10.15 or newer versions, when scheduling a task for the first time you will be prompted with something like this:

Click 'Yes' to proceed. The reason for that prompt is that the cron settings (crontabs) are considered to be system settings by macOS, and DBeaver will not be able to change them without your permission.

After that, if everything is configured correctly you will see the confirmation message. Just like in Windows, you can change the scheduler settings at any moment by choosing the 'Edit scheduled task' command from the context menu, or cancel the schedule by clicking on 'Remove schedule'.

See schedule details

Windows
You can see and change the scheduled task details in the Windows Task Scheduler. Click on the Open scheduler settings command in the task view context menu:

All DBeaver tasks are located in a folder called DBeaver.

macOS or GNU/Linux

You can take a look at the crontab DBeaver uses to schedule tasks in cron by clicking the 'Open scheduler settings' command in the task view context menu. You can also do it in the terminal by using the command crontab -l. Although you can also edit the crontab by using crontab -e, we strongly do not recommend it.

Monitoring for task execution (any OS)

You can look through the task execution logs on the right side of the tasks view. By double-clicking on a task run item you can see the full log with all details, errors, and warnings:
DBeaver keeps the task run logs in the workspace directory, subfolder .metadata/task-stats.

**Running tasks from the command line (any OS)**

The task scheduler uses the DBeaver command line interface to perform task executions. Command line parameter `-runTask TASK_ID` launches saved task executions (immediately). TASK_ID has the form `@projectName:taskName`. You can omit the project name part if you have only one project in your workspace. In Windows, you can use `dbeaver-cli` executable to run tasks. Please note that if you use `dbeaver` executable (for any reason), you will need to add the command line parameter `-nosplash` to avoid a splash screen appearance.

**Troubleshooting**

**Windows scheduler overview**

There are two implementations of Windows scheduler present:

1. CLI-based (Legacy): uses `schtasks.exe` to communicate with the scheduler; sensitive to locale-dependent data, such as unicode names, date-time format.
2. COM-based (New): uses COM API to communicate with the scheduler; more flexible and provides more features than CLI version.

COM-based implementation is used by default starting from the 21.1 version of DBeaver EE.

**Windows Task Scheduler: COM exception**

Non-legacy scheduler only

If you encounter an error in Windows which contains the following text: `com.sun.jna.platform.win32.COM.COMException`, do the following:

1. Open the file `dbeaver.ini` in the directory with your DBeaver installation
2. Place the line `-Ddbeaver.scheduler.windows.legacy=true` below the `-vmargs` line.

**Windows Task Scheduler: incorrect date format**

Legacy scheduler only

If you encounter an error in Windows which looks like this: `ERROR: Invalid Start Date [Date should be in %some_format% format].`, do the following:

1. Open the file `dbeaver.ini` in the directory with your DBeaver installation
2. Place the line `-Ddbeaver.scheduler.windows.dateFormat=%some_format%` (where `%some_format%` is a format from the error message) below the `-vmargs` line.

This flag is available starting from the 7.3.4 EA version of DBeaver Enterprise and might be removed in the future.

**macOS 10.15+: Unable to read or write to crontab**

When scheduling tasks on macOS 10.15 or newer versions, the OS will prompt you to elevate DBeaver’s permissions to administer your computer. If you do not grant these permissions, DBeaver will fail to schedule your tasks with an error `Unable to read or write to crontab`. To bypass this, simply restart DBeaver and try to schedule the task again. You will be prompted to elevate the permissions again. If you were never prompted to do that in the first place, you can grant `Full disk access` permissions in the macOS settings. Here is how to do that:

1. Open `System Preferences`.
2. Click on `Security & Privacy`.
3. Choose the `Privacy` tab.
4. Choose the `Full Disk Access` folder.
5. Unlock the preferences lock to the bottom if it is locked.
6. Click the + button.
7. Select DBeaverEE in the file picker that opens.
8. Click Open.
9. Close the lock.
Composite tasks

Note: This functionality is available only in the Enterprise Edition.

As the name suggests, the composite task is a type of task that consists of other tasks. Just like the other type of tasks, the composite tasks can be scheduled via Task Scheduler. Let's take a look at what they can offer.

Creating a composite task

The first thing we need to open is the Create a task dialog. You can do it in multiple ways:

- From the context menu in the database navigator -> Tools -> Create new task... -> Composite task
- By clicking Database -> Tasks -> Create new task... -> Composite task
- From the context menu in the Database Tasks view.

Choose Composite task, enter the task name, description (optional), and hit Next.

You will be presented with the following dialog:

![Composite task settings dialog]

Setting up a composite task

When creating a composite task, you need to specify which tasks the composite task consists of.

This can be done:

1. By adding an existing task. To do that, click the button with the plus sign

2. By creating a new task and adding it simultaneously. To do that, click the button below the aforementioned button with the plus sign

3. By drag-and-dropping a task from the Database tasks panel.

As a side note, you can add a composite task to your new composite task.

You can edit tasks in the same dialog, delete a task from a composite task, and change the execution order.

There is also a very important checkbox, Ignore task error. The tasks from the composite task are executed in the order they appear in the settings dialog. Executing a task from a composite task might produce an error that will block the next tasks from proceeding. The Ignore task error checkbox can be used to bypass this behavior.
Cloud Explorer

Overview

Cloud Explorer provides a deep integration with classic cloud service providers such as Amazon, Google and Azure.

Note: Cloud Explorer is supported only in DBeaver Ultimate Edition. Version 21.0 supports only AWS (Amazon Cloud Services) cloud.

It allows users to configure cloud access once and then easily browse, connect and manage all cloud databases with just a few clicks. There is no need to configure each database connection manually, all database endpoint information reads directly from the cloud provider. Authentication is managed in a centralized mode - you use your cloud account to get access to the cloud databases.

Cloud configuration

Before you begin to work with cloud explorer you need to configure your cloud provider access. Configuration includes access credentials, availability zones which will be used to search databases and some other cloud-specific settings. Cloud configuration is different for each cloud service provider.

Configuring AWS cloud

Explorer

Once you configure the cloud configuration you can open the Cloud Explorer dialog and start adding database connections. In the top drop-down of explorer dialog you can select the active cloud configuration or click "Edit" to change the cloud configuration.

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

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

You can drag-and-drop cloud databases directly to database navigator view or projects view. You can also check any number of databases in the Cloud Explorer using the checkbox control on the left side of the Cloud Explorer tree, and then click on the "Add to Project" button in the bottom right corner.

Database cloud information

You can always see your cloud database configuration in a special tab in the connection settings dialog. This information depends on both cloud and database type. You can also click on the external link to open your database configuration in the cloud provider web console.
### Connection settings

#### Oracle connection settings

<table>
<thead>
<tr>
<th>Cloud:</th>
<th>AWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region:</td>
<td>eu-central-1</td>
</tr>
<tr>
<td>ARN:</td>
<td>ami:aws:dk:eu-central-1 db:ora-database-1</td>
</tr>
<tr>
<td>Identifier:</td>
<td>ora-database-1</td>
</tr>
<tr>
<td>Engine:</td>
<td>oracle-se2</td>
</tr>
<tr>
<td>Engine Version:</td>
<td>19.10.0.0.0.2020-10.0ur-2020-10.0r1</td>
</tr>
<tr>
<td>Instance status:</td>
<td>available</td>
</tr>
<tr>
<td>Instance class:</td>
<td>db.t3.small</td>
</tr>
</tbody>
</table>

[![Test Connection...](image)](image)
AWS

AWS (Amazon Web Services) Cloud Explorer

Cloud configuration

![AWS Cloud Explorer Configuration Screen]

Supported cloud databases

![AWS Cloud Explorer Screen showing supported cloud databases]

DBeaver user guide 213/285
AWS Credentials

DBeaver is integrated with AWS IAM authentication. Thus it provides the possibility to authenticate in AWS in order to access your cloud databases.

AWS IAM has endless ways to authorize and authenticate users. DBeaver supports all basic ones.

Default credentials

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

1. Java system properties
2. Environment variables
3. Web identity token from AWS STS
4. The shared credentials and config files
5. Amazon ECS container credentials
6. Amazon EC2 instance profile credentials
7. Amazon SSO credentials

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

Please read the AWS credentials documentation for a detailed explanation.

Access keys

It is the most simple way to authenticate. You only need to enter the IAM user access key and secret key. You can save them locally or (more securely) enter them every time you connect to a database.

Official AWS instructions: Managing access keys for IAM users

AWS Profiles

Similar to default credentials but you can also choose which credentials profile you want to use.

The official AWS instructions can be found at credentials config files.

Single Sign On

If your AWS account has configured SSO portal then you can use a web-based SSO authorization. SSO support can be enabled for Default and Profile-based AWS authorization types.

You need to turn on the "Enable SSO" option.
AWS SSO

Using AWS Single Sign On in DBeaver

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

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

AWS CLI

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

AWS CLI installation

AWS CLI version 2.2 is recommended.

AWS SSO configuration

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

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

Connection configuration

In the DBeaver database connection dialog you need to:

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

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

DBeaver EE supports MongoDB schema browser, data viewer, SQL and JavaScript queries execution. It also supports various administrative tools (like server sessions manager).

DBeaver uses MongoDB Java driver 3.8.0 to operate with a server. It supports MongoDB servers from 2.x to 4.x.

**Connecting to MongoDB Server**

You can connect directly to a server or use SSH tunneling or SOCKS proxy.

You can specify server address as a host/port/database configuration or you can enter the target database URL with all necessary parameters:

![MongoDB Connection Settings](image)

- **Address**: localhost
- **Port**: 27017
- **Database**: local
- **Replica Set**: Repl1
- **User**: admin
- **Password**: ********

You can use variables in connection parameters.
Connection 'MongoDB - Atlas' configuration

MongoDB connection settings

- Connection settings
  - Driver properties
  - SSH
  - Proxy
  - SSL

Address:
- Manual
- URL

URL: `mongodb://@freecluster-asidfewdggp.mongodb.net/test?retryWrites=true`

Do not put user name/password in the URL.
URL must look like `mongodb://[parameters] or mongodb-srv://[parameters]`

Credentials:
- Mechanism: SCRAM-SHA-1
- User: user
- Source: admin
- Password: **************

You can use variables in connection parameters.

Driver name: MongoDB

Test Connection... OK Cancel

Connection 'MongoDB - Atlas' configuration

MongoDB connection settings

- Connection settings
  - General
  - Metadata
  - Errors and timeouts
  - Data viewer
  - SQL Editor

Main Driver properties SSH Proxy SSL

Name Connection
- Value
  - Connection Timeout
  - Max Connection idle Time
  - Max Connection Life Time
  - Max Wait Time
  - Server Select Timeout: 15,000
  - Single node mode
  - Socket Keep Alive
  - Socket Timeout
  - Heartbeat
    - Heartbeat Connect Timeout
    - Heartbeat Frequency
    - Heartbeat Socket Timeout
  - Miscellaneous
    - Use client-side JavaScript
    - User Properties

Advanced driver properties

Driver documentation

Test Connection... OK Cancel
Browsing Mongo collections

You can view/edit MongoDB collections content as standard relational tables (grid/plain text presentations) or as JSON documents. The presentation can be switched in the Results Viewer toolbar.
In a grid, DBeaver will try to unify all documents in some particular collection (as they have the same structure/the same set of properties).
```json
{
    "id": 2,
    "Address": "1121 Loja Avenue",
    "City": "San Bernadino",
    "Country": "United States",
    "District": "California",
    "First Name": "PATRICIA",
    "Last Name": "JOHNSON",
    "Phone": "838635286849",
    "Rentals": [
        {
            "Film Title": "DOORS PRESIDENT",
            "Payments": [
                {
                    "Amount": 4.98999973118164,
                    "Payment Date": "2005-05-27 00:09:24.0",
                    "Payment Id": 33
                }
            ],
            "Rental Date": "2005-05-27 08:18:24.0",
            "Return Date": "2005-05-28 04:18:24.0",
            "filmid": 243,
            "rentalId": 20,
            "staffId": 2
        },
        {
            "Film Title": "BLACKOUT PRIVATE",
        }
    ]
}
```
<table>
<thead>
<tr>
<th>District</th>
<th>First Name</th>
<th>Last Name</th>
<th>Phone</th>
<th>Film Title</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nagasaki</td>
<td>MARY</td>
<td>SMITH</td>
<td>23303584239</td>
<td>MUSKETEERS WAIT</td>
<td>5,980999771</td>
</tr>
<tr>
<td>California</td>
<td>PATRICIA</td>
<td>JOHNSON</td>
<td>338835636643</td>
<td>DOORS PRESIDENT</td>
<td>4,980999771</td>
</tr>
<tr>
<td>Attika</td>
<td>LINDA</td>
<td>WILLIAMS</td>
<td>448477198468</td>
<td>RINGS HEARTBREAKERS</td>
<td>1,990000009</td>
</tr>
<tr>
<td>Mandalay</td>
<td>BARBARA</td>
<td>JONES</td>
<td>70581403527</td>
<td>BEDAZZLED MARRIED</td>
<td>0,990000009</td>
</tr>
<tr>
<td>Texas</td>
<td>JENNIFER</td>
<td>DAVIS</td>
<td>860456265434</td>
<td>BLUES INSTINCT</td>
<td>2,990000009</td>
</tr>
<tr>
<td>Nantou</td>
<td>ELIZABETH</td>
<td>BROWN</td>
<td>10653648674</td>
<td>BETRAYED REAR</td>
<td>6,980999771</td>
</tr>
<tr>
<td>Central Serbia</td>
<td>MARIA</td>
<td>MILLER</td>
<td>716571220873</td>
<td>RIDGEMONT SUBMARINE</td>
<td>5,980999771</td>
</tr>
<tr>
<td>Hamilton</td>
<td>SUSAN</td>
<td>WILSON</td>
<td>65728228570</td>
<td>HIGH ENCINO</td>
<td>5,980999771</td>
</tr>
<tr>
<td>Masqat</td>
<td>MARGARET</td>
<td>MOORE</td>
<td>38063732649</td>
<td>CRUELTY UNFORGIVEN</td>
<td>0,990000009</td>
</tr>
<tr>
<td>Kanagawa</td>
<td>LISA</td>
<td>ANDERSON</td>
<td>635297277345</td>
<td>BOOGIE AMELIE</td>
<td>6,980999771</td>
</tr>
<tr>
<td>Esfahan</td>
<td>DOROTHY</td>
<td>TAYLOR</td>
<td>648056363185</td>
<td>SNOWMAN ROLLERCOASTER</td>
<td>4,980999771</td>
</tr>
<tr>
<td>Haryana</td>
<td>NANCY</td>
<td>THOMAS</td>
<td>465687807014</td>
<td>FAMILY SWEET</td>
<td>4,980999771</td>
</tr>
<tr>
<td>Dimanaye</td>
<td>KAREN</td>
<td>JACKSON</td>
<td>695479627358</td>
<td>SOUTH WAIT</td>
<td>2,990000009</td>
</tr>
<tr>
<td>California</td>
<td>BETTY</td>
<td>WHITE</td>
<td>517338314335</td>
<td>OUTBREAK DIVINE</td>
<td>0,990000009</td>
</tr>
<tr>
<td>England</td>
<td>SANDRA</td>
<td>MARTIN</td>
<td>94932333339</td>
<td>FEUD FROGMEN</td>
<td>3,990000009</td>
</tr>
<tr>
<td>Madhy Pradesh</td>
<td>HELEN</td>
<td>HARRIS</td>
<td>99091107354</td>
<td>HOTEL HAPPINESS</td>
<td>5,980999771</td>
</tr>
<tr>
<td>Kalmykia</td>
<td>DONNA</td>
<td>THOMPSON</td>
<td>40775241688</td>
<td>TIES HUNGER</td>
<td>4,980999771</td>
</tr>
<tr>
<td>Kaduna</td>
<td>CAROL</td>
<td>GARCIA</td>
<td>74779159460</td>
<td>WIFE TURN</td>
<td>4,980999771</td>
</tr>
<tr>
<td>Maharashtra</td>
<td>NITH ISHANT</td>
<td>PARTHER</td>
<td>72736972608</td>
<td>ROLLING</td>
<td>6,980999771</td>
</tr>
</tbody>
</table>
JS statements can be executed in the SQL editor as usual. DBeaver supports all JS queries for MongoDB 2 and 3 as well as a subset of the `mongo` shell queries.

The following example creates a user in the current database.

```javascript
db.createUser({
  user: 'testuser',
  pwd: 'test',
  roles: []
})
```

This example returns all documents in the collection 'test_col':

```javascript
db.test_col.find().toArray()
```

Note: the script will be executed in the current database.
You cannot set an explicit database name in your query.
The current database can be changed on the SQL Editor toolbar or on the Database Navigator.

**Executing SQL**

You can use standard SQL statements (SELECT, INSERT, UPDATE, DELETE) to manipulate Mongo data.

SELECT queries support WHERE, ORDER BY, GROUP BY, JOIN and HAVING clauses.
SELECT * FROM test_col
WHERE propName.subProp='value'

UPDATE FROM test_col
SET propName.val1=123
WHERE propName.subProp='value'

Conditions

SELECT queries with WHERE support AND, OR, <=, >=, = and != operators:

SELECT * FROM Employees
WHERE (Country = 'CA' OR Country = 'RU') AND Age > 20;

Please note that AND has higher precedence than OR and will evaluate first, so you need to surround it with parentheses.

Nested fields

Nested JSON fields can be divided by dot. If your field contains any special characters (e.g. spaces, dashes, etc.), you must enclose it with double quotes. For example:

SELECT title FROM movies WHERE info."imdb-details".rating > 6

Working with dates

If you need to operate with dates then you must specify them in an ISO format. It is possible in both the JavaScript and SQL dialect:

```javascript
db.dates.insert({
  value: new Date('2016-05-18T16:00:00Z'),
  value: new Date('2017-05-18T16:00:00Z'),
  value: new Date('2018-05-18T16:00:00Z'),
  value: new Date('2019-05-18T16:00:00Z'),
  value: new Date('2020-05-18T16:00:00Z')
})
```

Querying data in JavaScript:

```javascript
db.dates.find({
  value: { $gte: new Date('2018-05-18T16:00:00Z') }
}).toArray()
```

Querying data in the SQL dialect (ISO and UNIX timestamp, in milliseconds):

```sql
SELECT value FROM dates
WHERE value > ISODate('2018-05-18T16:00:00.000Z')
ORDER BY value DESC
```

```sql
SELECT value FROM dates
WHERE value > ISODate(1526659200000)
ORDER BY value DESC
```

Working with object IDs

When you need to find a document by ID, you must use the function ObjectId:

```sql
SELECT * FROM documents
WHERE _id = ObjectId('5f9c458018e3c69d0adc0fbd')
ORDER BY value DESC
```

Working with JOINs

Currently, SQL dialect for MongoDB supports LEFT JOIN and INNER JOIN:
SELECT ar.Name as Artist,
al.Title as Album,
SUM(tr.Milliseconds) as Duration
FROM Track tr
INNER JOIN Album al ON tr.AlbumId = al.AlbumId
INNER JOIN Artist ar ON al.ArtistId = ar.ArtistId
GROUP BY Artist, Album
ORDER BY Duration DESC

The only limitation is that you have to specify aliases for both source and target tables in a particular order:

SELECT *
FROM <source> <source-alias>
INNER JOIN <target> <target-alias> ON <source-alias>.column = <target-alias>.column

Note that executing the following script will not result in a merged document, but it will result in separate documents for Track and Album:

SELECT *
FROM Track tr
INNER JOIN Album al ON tr.AlbumId = al.AlbumId
Cassandra

Overview

DBeaver EE supports Cassandra schema browser, data viewer and CQL queries execution. It also supports various administrative tools.

Connecting to Cassandra cluster

You can connect directly to a server or use SSH tunneling or SOCKS proxy. DBeaver uses the DataStax Java driver to operate with a server. It supports Cassandra servers 2.x, 3.x or higher.
You can browse, view, edit and filter Cassandra tables the same way as with regular (relational) tables. However, being a distributed key-value database, Cassandra does not support any kind of referential integrity. There are no foreign keys, references, etc.

You should note that Cassandra has a very advanced (comparing to relational databases) data type system. Each column may be a collection, map, or set of values (with very big number of values). In some cases this makes browsing data in the "Grid" mode inconvenient.

**Browsing Cassandra tables**

You can browse, view, edit and filter Cassandra tables the same way as with regular (relational) tables. However, being a distributed key-value database, Cassandra does not support any kind of referential integrity. There are no foreign keys, references, etc.

You should note that Cassandra has a very advanced (comparing to relational databases) data type system. Each column may be a collection, map, or set of values (with very big number of values). In some cases this makes browsing data in the "Grid" mode inconvenient.
CQL

Cassandra Query Language is a very simple SQL language dialect. It supports simple SELECT queries, DDL statements (like CREATE TABLE) and some other query types.

You can use the standard DBeaver SQL editor to execute CQL queries. DBeaver supports Cassandra query execution, results scrolling, data export/import, mock data generation and other features. Data viewer (of individual tables or custom CQL query results) query tracing is supported.

Executing CQL

CQL Cassandra Query Language is a very simple SQL language dialect. It supports simple SELECT queries, DDL statements (like CREATE TABLE) and some other query types.

You can use the standard DBeaver SQL editor to execute CQL queries. DBeaver supports Cassandra query execution, results scrolling, data export/import, mock data generation and other features. Data viewer (of individual tables or custom CQL query results) query tracing is supported.
Physical ERD (Entity Relation Diagram) does not make much sense for Cassandra as there are no foreign keys. However, you can make your own custom ERD and connect an actual Cassandra table with each other using logical associations.

**ERD**

Physical ERD (Entity Relation Diagram) does not make much sense for Cassandra as there are no foreign keys. However, you can make your own custom ERD and connect an actual Cassandra table with each other using logical associations.
InfluxDB

Overview

DBeaver EE supports InfluxDB schema browser, data viewer and InfluxQL queries execution. DBeaver uses InfluxDB Java driver 2.12 to operate with the server over HTTP/HTTPS (standard InfluxDB protocol). It supports InfluxDB servers of any version (in the moment of writing).

Connecting to Influx Server

You can connect directly to a server or use SSH tunneling or SOCKS proxy.
Browsing InfluxDB schema

InfluxDB is a TimeSeries database, it does not support tables, foreign keys and other relational entities.

DBeaver does not support data insert/update in InfluxDB. Database is basically a read-only state for DBeaver. You can browse schema and view/analyse data. While data itself is loaded by various sensors/data collectors in real time. Instead of tables InfluxDB has measurements. Instead of columns it has fields and tags.
Executing InfluxQL

InfluxQL is a query language similar to SQL.
DBeaver fully supports all InfluxQL statements. Query results are presented as grid or as graphs:
Redis

Overview

DBeaver EE supports Redis key browser, key value viewer and Redis commands shell. DBeaver uses Jedis driver 2.9.0 to operate with Redis server. It supports Redis servers of any version.

Connecting to Redis Server

You can connect directly to a server or use SSH tunneling or SOCKS proxy.
You can view/edit Redis keys as a plain list. However, the Redis database usually contains a lot of keys (millions or even billions) and using list presentation is not convenient (or it is not possible). DBEaver supports a hierarchy presentation of keys. Internally Redis does not support hierarchies but application level key names may be divided into groups using a character (e.g., comma, dash or colon). DBEaver uses this pattern to show hierarchy. Group separator can be configured in connection properties.

Key browser may be convenient in some cases but in the case of big databases it is very difficult to find your key in the navigator, so the SQL editor should be used instead. Redis commands is the most flexible way to operate with keys.
Redis does not support SQL or any other query language. Instead, it supports build-in commands and LUA scripts.

Redis commands can be executed in the same way as in a Redis command line shell: 

```
COMMAND ARG1 ARG2 ... ARGN
```

In order to execute a command, run it using CTRL+Enter or ALT+X. All standard DBeaver SQL editor shortcuts work for Redis as well.

In order to execute a LUA script, surround it with `{}` brackets and run it as a single statement. If the script contains empty lines or special characters, select the script text before the execution.

```lua
{ return {1,2,{3,'Hello World!'}} }
```

### Executing Redis commands

Redis is a powerful key-value store that is widely used in various applications. It supports a wide range of operations that are often used in caching and data processing. Redis commands can be executed in the same way as in a Redis command line shell.

In order to execute a Redis command, you can run it using CTRL+Enter or ALT+X. All standard DBeaver SQL editor shortcuts work for Redis as well.

In order to execute a LUA script, surround it with `{}` brackets and run it as a single statement. If the script contains empty lines or special characters, select the script text before the execution.
DynamoDB

Supported features:

- Table data view
- Table data edit in document (json) mode
- Data filters
- SQL queries execution
- JSON queries execution
- Data export and import

DynamoDB connection

DBeaver supports AWS Cloud and Standalone versions of DynamoDB. For standalone server you need to enter endpoint (http or https URL). For cloud server you must enter the AWS region. DynamoDB exists in all available regions in your AWS account but the tables are different.

AWS Access Key and Secret Key are used for authentication. For 3rd-party account access you must specify the 3rd party account ID (12-digits number) and the 3rd party role name. This role will be used for permission management. You account must be added to the whitelist in the 3rd party account.

Press "Test Connection" to validate your connection settings.

Database navigation

DynamoDB has a simple metadata structure. Basically, you can only access Table and Global tables. Table has primary attributes (a kind of primary key) and indexes. DynamoDB is a document-oriented database. Each table may have its own set of attributes and sub-attributes.
### Viewing table data

You can open table editor and see the table data. You may need to switch to the "Data" tab. DBeaver converts DynamoDB documents into a table format by default, but you can switch to another data representation.

You can use data filters in order to find documents.

### Viewing data in JSON document format

You view, search and edit JSON documents. Double-click on a document to activate the editor.
DBeaver supports simple SQL dialect for DynamoDB. You can use the WHERE clause in the same fashion as in regular SQL in order to find or filter documents.

You can also use JSON requests syntax to query documents. See Amazon DynamoDB query reference.
Exporting and importing data

You can export data from a DynamoDB table in different file formats (CSV, XLSX, XML, JSON, etc.) or export data directly to another table.
**DocumentDB**

AWS DocumentDB is based on the MongoDB engine. It has several minor differences in the query processing and network configuration. However, most features which work for MongoDB will work for DocumentDB as well. Please refer to the MongoDB article.

**Connections**

AWS restricts direct access to DocumentDB clusters from outside of the cloud (region). So you can connect to it directly (using a cluster host name) only when DBeaver is deployed on the EC2 instance.

In other cases you will need to use the SSH tunnel through a proxy machine to access DocumentDB instance. Please read the AWS Documentation about proxy configurations: [https://docs.aws.amazon.com/documentdb/latest/developerguide/connect-from-outside-a-vpc.html](https://docs.aws.amazon.com/documentdb/latest/developerguide/connect-from-outside-a-vpc.html)

In DBeaver you can use the SSH tab on the connection settings page. Just enter proxy host, user name and specify a private key file (it is provided by AWS as a keypair).

**Queries**

DBeaver processes DocDB SQL queries exactly like in MongoDB. It supports SELECT, UPDATE, INSERT and DELETE queries. SELECT queries support WHERE, ORDER BY, GROUP BY and HAVING clauses.

DocumentDB restricts the `eval` function so all JavaScript queries will be parsed on the client's side and then evaluated at a DocDB cluster one by one. Most JS functions work exactly like in Mongo Shell.
Keyspaces

Overview

AWS Keyspaces is a key-value database based on Apache Cassandra.

DBeaver EE supports the Keyspaces schema browser, data viewer and CQL queries execution. It also supports various administrative tools.

Connecting to Keyspaces

AWS Keyspaces uses AWS IAM authentication.
You need to specify your IAM credentials and AWS region. There is no specific endpoint - there is only one cluster per account per AWS region.
There is also no need to configure SSH or SSL - DBeaver uses default AWS settings to access the Keyspaces cluster.

Browsing Keyspaces tables

You can browse, view, edit and filter Cassandra tables the same way as with regular (relational) tables. However, being a distributed key-value database, Keyspaces does not support any kind of referential integrity. There are no foreign keys, references, etc.
You should note that Cassandra has a very advanced (comparing to relational databases) data type system. Each column may be a collection, map, or set of values (with a very big number of values). In some cases this makes browsing data in the "Grid" mode inconvenient.
CQL Cassandra Query Language is a very simple kind of SQL language dialect.
It supports simple SELECT queries, DDL statements (like CREATE TABLE) and some other query types.

You can use the standard DBaever SQL editor to execute CQL queries. DBaever supports Cassandra query execution, results scrolling, data export/import, mock data generation and other features.

**ERD**

Physical ERD (Entity Relation Diagram) does not make much sense for Keyspaces as there are no foreign keys. However, you can make you own custom ERD and connect Keyspaces tables with each other using logical associations.
Bigtable

Supported features:

- Table data view
- Table data edit in document (json) mode
- Data filters
- SQL queries execution
- Data export and import

Bigtable connection
Couchbase

Connecting to Couchbase server

Couchbase client uses multiple ports to connect to a cluster (8091-8096, 9140, etc). Some of these ports are dynamic (i.e. depend on server settings) and cannot be overwritten. It makes SSH tunnelling impossible. Thus, if you work with a remote Couchbase deployed behind a firewall, you will need to setup a VPN connection or SOCKS proxy.
Couchbase is a document-oriented database. It means that all documents may have different structures. You can view/edit buckets content, such as standard relational tables (grid/plain text presentations) or JSON documents.

Viewing and editing Couchbase tables

Couchbase is a document-oriented database. It means that all documents may have different structures. You can view/edit buckets content, such as standard relational tables (grid/plain text presentations) or JSON documents.
Couchbase uses N1QL language for queries. It is very similar to the standard SQL language.

### Executing Couchbase queries

Couchbase uses N1QL language for queries. It is very similar to the standard SQL language.
SELECT country FROM `travel-sample` WHERE name = "Excel Airways";
Apache Hive

Hive is a Hadoop-based storage system. Hive uses a special SQL dialect (HiveQL) to operate with data and metadata. Generally, it is quite similar to SQL.

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

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

Connection setup

![Connection setup screenshot]

Apache Hive

Hive JDBC

Project

General

< Back  Next >  Finish  Cancel  Test Connection...
Schema/data browser
Limitations

Hive does not support referential integrity so you will not see primary keys or foreign keys. ER diagrams also do no make much sense.
Changing interface language

Changing interface language in Preferences

Go to Preferences->User Interface:

Select your language in the drop-down list and click the "Apply and Close" button.

If DBeaver is installed in a read-only directory, the automatic language change is not possible. In this case, try to edit the configuration file (see below).

Changing interface language in configuration file

Locate the `dbeaver.ini` file. It is in the same directory where DBeaver is installed.

Open `dbeaver.ini` in a text editor and add the following lines before the line `-vmargs`

```
[lang]
xx
```

where XX is two-letter language code:

<table>
<thead>
<tr>
<th>Language</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>en</td>
</tr>
<tr>
<td>Chinese</td>
<td>zh</td>
</tr>
<tr>
<td>French</td>
<td>fr</td>
</tr>
<tr>
<td>Italian</td>
<td>it</td>
</tr>
<tr>
<td>Japanese</td>
<td>jp</td>
</tr>
<tr>
<td>German</td>
<td>de</td>
</tr>
<tr>
<td>Korean</td>
<td>ko</td>
</tr>
<tr>
<td>Portuguese (BR)</td>
<td>pt_BR</td>
</tr>
<tr>
<td>Russian</td>
<td>ru</td>
</tr>
<tr>
<td>Language</td>
<td>Code</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>Spanish</td>
<td>es</td>
</tr>
</tbody>
</table>
Installing extensions - Themes, version control, etc

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

**DBeaver-specific extensions**

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

**Popular 3rd party extensions for Eclipse and DBeaver**

- Darkest Dark theme - the best Dark theme for DBeaver
- Eclipse Color Theme - if for some reason you do not like the Darkest Dark theme, you can use this one
- Subversion support - Subversion integration
- Embedded Shell - Allows you to run shell commands directly from DBeaver
- Editor vertical indents - Adds vertical indents to all text editors

**Install Process**

In DBeaver EE you can use drag-n-drop from the Marketplace web site (see button ![Install](https://www.eclipse.org/dbeaver/) in the DBeaver main window. This will launch the Marketplace installation wizard automatically. In the DBeaver Community or other DBeaver-based products which do not include marketplace clients, you can use the following instructions:

**Extension installation in CE version:**

1. Copy URL of extension update site:
2. In the DBeaver main menu open `Help -> Install New Software`

3. Paste update site URL into `Work with` field and press `Enter`

4. Check items you wish to install (in most cases just all items)
5. Click Next. You may need to accept the extension license before installing.
6. Some extensions may contain unsigned bundles. Only install such extensions if you really trust the author.

7. Click Next->Finish. The installation will take some time. Restart DBeaver.
Enterprise Edition

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

- All popular JDBC drivers are included in the EE distribution so you will not need to download/configure them separately.
- Support of NoSQL databases:
  - Apache Cassandra
  - MongoDB
  - Redis
  - InfluxDB
  - Couchbase
  - CouchDB
  - Google BigTable
  - AWS DynamoDB
  - AWS Keyspaces
  - WMI (Windows Management Instrumentation)
- Additional EE plugins:
  - Office formats support (XLS) for data import
  - Visual Query Builder
  - Schema/table compare, diff DDL generation
  - Data compare
  - Analytical charts rendering
  - Persistent query database. Search in query history
  - Eclipse Marketplace (provides easily installation of any additional Eclipse plugins)
  - 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!
## Command Line

### Command line parameters

Command line parameters might be passed directly to dbeaver.exe executable. In Windows you can also use `dbeaver-cli.exe` executable (it does not spawn a new window so you can see the output messages). Also, you can add parameters in the `dbeaver.ini` configuration file - in the beginning of the file and each parameter on its own line.

#### DBEaver control

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-help</td>
<td>Prints help message</td>
<td></td>
</tr>
<tr>
<td>-stop</td>
<td>Quits DBeaver</td>
<td></td>
</tr>
<tr>
<td>-dump</td>
<td>Prints DBeaver thread dump</td>
<td></td>
</tr>
<tr>
<td>-f</td>
<td>Opens file in DBeaver UI</td>
<td><code>-f c:\some-path\some-file.sql</code></td>
</tr>
<tr>
<td>-con</td>
<td>Opens database connection in DBeaver UI</td>
<td>See connection parameters table</td>
</tr>
<tr>
<td>-closeTabs</td>
<td>Closes all open editor tabs</td>
<td></td>
</tr>
<tr>
<td>-disconnectAll</td>
<td>Closes all open connections</td>
<td></td>
</tr>
<tr>
<td>-renewWorkspace</td>
<td>Force reuse of single workspace by multiple DBeaver instances</td>
<td></td>
</tr>
<tr>
<td>-newInstance</td>
<td>Force new DBeaver instance creation (do not try to reuse already running one)</td>
<td><code>-newInstance</code></td>
</tr>
<tr>
<td>-var</td>
<td>Custom variables for runTask. You can change existing variables in the task. You cannot add new task variables with this parameter. You can add several parameters at once to the command line, each starting with &quot;-var&quot;. Used right before -runTask. Template: &quot;-var variableName=variableValue&quot;</td>
<td><code>-var film=sakila.film</code> <code>-var actor=sakila.actor</code> <code>-runTask &quot;exportFromSakila&quot;</code> EE version only.</td>
</tr>
<tr>
<td>-runTask</td>
<td>Executes specified task</td>
<td><code>-runTask *@projectName:taskName*</code>. EE version only. See task scheduler.</td>
</tr>
<tr>
<td>-license</td>
<td>Path to the EE license file</td>
<td><code>-license */etc/licenses/dbeaver.txt*</code>. EE version only.</td>
</tr>
</tbody>
</table>

#### System parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-nl</td>
<td>Locale</td>
<td>en_US</td>
</tr>
<tr>
<td>-data</td>
<td>Workspace path</td>
<td>c:\ProgramData\MyWorkspace</td>
</tr>
<tr>
<td>-nosplash</td>
<td>Omits splash screen</td>
<td>true</td>
</tr>
<tr>
<td>-clean</td>
<td>Clears all Eclipse caches. Use it if DBeaver fails to start after a version upgrade.</td>
<td>true</td>
</tr>
<tr>
<td>-vmargs</td>
<td>VM parameters</td>
<td>See VM arguments table</td>
</tr>
</tbody>
</table>

#### VM arguments

You can pass any advanced Java parameters supported by your local JVM (Oracle, OpenJDK, IBM, etc). Parameters supported by Oracle JVM (11): [https://docs.oracle.com/en/java/javase/11/tools/java.html](https://docs.oracle.com/en/java/javase/11/tools/java.html)

Parameters supported by all JVMs:
<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-Xms</td>
<td>Sets initial memory available for DBeaver</td>
<td>-Xms1000m</td>
</tr>
<tr>
<td>-Xmx</td>
<td>Sets maximum memory available for DBeaver</td>
<td>-Xmx4000m</td>
</tr>
</tbody>
</table>

**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"`

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Connection name</td>
<td>Test connection</td>
</tr>
<tr>
<td>driver</td>
<td>Driver name or ID</td>
<td>driver=sqlite, driver=mysql, etc</td>
</tr>
<tr>
<td>url</td>
<td>Connection URL. Optional (JDBC URL may be constructed by driver from other parameters)</td>
<td>url=jdbc:sqlite:C:\db\SQLite\Chinook.db</td>
</tr>
<tr>
<td>host</td>
<td>Database host name (optional)</td>
<td>host=localhost</td>
</tr>
<tr>
<td>port</td>
<td>Database port number (optional)</td>
<td>port=1534</td>
</tr>
<tr>
<td>server</td>
<td>Database server name (optional)</td>
<td>server=myserver</td>
</tr>
<tr>
<td>database</td>
<td>Database name or path (optional)</td>
<td>database=db-name</td>
</tr>
<tr>
<td>user</td>
<td>User name (optional)</td>
<td>user=root</td>
</tr>
<tr>
<td>password</td>
<td>User password (optional)</td>
<td>password=mysecret</td>
</tr>
<tr>
<td>auth</td>
<td>Authentication model ID. See Auth models</td>
<td>auth=postgres_pgpas</td>
</tr>
<tr>
<td>authProp.propName</td>
<td>Custom authentication parameters (depends on driver and auth model)</td>
<td>authProp.oracle.net.wallet_location=C:/temp/ora-wallet</td>
</tr>
<tr>
<td>savePassword</td>
<td>Do not ask user for a password on connection</td>
<td>savePassword=true</td>
</tr>
<tr>
<td>showSystemObjects</td>
<td>Show/hide system schemas, tables ,etc</td>
<td>showSystemObjects=true</td>
</tr>
<tr>
<td>showUtilityObjects</td>
<td>Show/hide utility schemas, tables ,etc</td>
<td>showUtilityObjects=true</td>
</tr>
<tr>
<td>folder</td>
<td>Put new connection in a folder</td>
<td>folder=FolderName</td>
</tr>
<tr>
<td>autoCommit</td>
<td>Sets connection auto commit flag (default value depends on driver)</td>
<td>autoCommit=true</td>
</tr>
<tr>
<td>prop.propName</td>
<td>Advanced connection parameters (depend on driver)</td>
<td>prop.connectTimeout=30</td>
</tr>
<tr>
<td>id</td>
<td>Connection id</td>
<td>oracle_thin-16a88e813bd-70598e648cedd28c (useful in conjunction with create=false)</td>
</tr>
<tr>
<td>connect</td>
<td>Connect to this database</td>
<td>connect=false</td>
</tr>
<tr>
<td>openConsole</td>
<td>Open SQL console for this database (sets connect to true)</td>
<td>openConsole=true</td>
</tr>
<tr>
<td>create</td>
<td>Create new connection</td>
<td>create=false (true by default). If set to false then an existing connection configuration will be used. The name or id parameter must be specified.</td>
</tr>
</tbody>
</table>
Reset UI settings

After multiple version and/or upgrades/incorrect shutdowns, DBeaver UI may become corrupted. You could experience glitches such as extra toolbar elements might appear, menu items might go missing, keyboard shortcuts and localization strings could be broken, etc.

To reset DBeaver UI just delete the 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 the 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) the workspace can become messy. Some keyboard shortcuts may stop working, toolbar layouts 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 does not help then you can try to remove the .metadata/.plugins/org.eclipse.core.resources folder.

If that does not help then remove the .metadata folder. It will erase all your UI settings (but all connections, settings and scripts will remain as is).

That is 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 the open and closed issues to see if what you are running into has already been addressed.

- Be clear about what your problem is: what was the expected outcome, what happened instead? Detail how someone else can recreate the problem.

- If you posting a bug report check "Error Log" view. If there are any errors related to your bug then post a complete stacktrace. Sometimes there are no errors in Error Log - if so, try to find them in log files.

- If your issue is related to database data or metadata management - check the Query Manager view. It contains information about all queries DBeaver executes (explicitly or implicitly). To see more detailed information you can configure Query Manager in Preferences.

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

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

Error Log view

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

![Error Log view](image)

Log files

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

- In Windows open Explorer and paste path `%APPDATA%\DBeaverData\workspace6\.metadata`.
- In Linux just type `cd $XDG_DATA_HOME/DBeaverData/workspace6/.metadata`.
- In MacOS open path `~/Library/DBeaverData/workspace6/.metadata` in Finder.
  - To view hidden folders press `Cmd+Shift+.` in the folder view.

Two standard log files:

- `workspace/workspace6/.metadata/.log` - all warnings and errors which happen during normal work
- `workspace/workspace6/.metadata/dbeaver-debug.log` - the same as `.log` plus debug information

In special cases log files can be written in other directories. A special case is an emergency situation when DBeaver cannot start and there is no workspace. Two typical places to find emergency logs:

- `<install-path>/configuration`
- `~/.eclipse/org.jkiss.dbeaver.product_<dbeaver-version>`

If you are reporting an error, please attach the applicable part of the log - not the complete file.
Logs are very useful. Many errors cannot be reproduced and fixed without a full error stacktrace (all the details).

Java fatal logs

On the rare occasion that the DBeaver process dies, it does not leave any valuable logs. This is caused by a Java VM crash.
JVM creates a fatal log file for each crash (log file `hs_err_PID.log`). This log usually resides in the same directory where the DBeaver launcher is (e.g. `dbeaver.exe`).
But in some cases it is a write-protected directory and the log file will be created in other folder.
Instructions on how to find the Java fatal log file: [https://docs.oracle.com/javase/9/troubleshoot/fatal-error-log.htm](https://docs.oracle.com/javase/9/troubleshoot/fatal-error-log.htm)
In some cases, custom JDBC drivers work incorrectly in DBeaver - they show the wrong metadata like table columns, constraints or foreign keys.
It usually happens because the driver is not compliant with the JDBC API specification and DBeaver cannot correctly interpret the metadata provided by the driver.

To understand what is going on inside the driver, you can enable JDBC tracing:

1. Find `dbeaver.ini` file (it is located in the same folder where DBeaver is installed)
2. Add line `-Ddbeaver.jdbc.trace=true` in the end of `dbeaver.ini`
3. Restart DBeaver
4. Connect to your database and browse the metadata in the database navigator/object editors.
5. In DBeaver Workspace go to `.metadata` folder
6. File `jdbc-api-trace.log` contains all JDBC API invocations and all queries with results.

Analyzing contents of `jdbc-api-trace.log` you can understand what is wrong with the metadata. Attach the piece of the trace file in the GitHub ticket if you think that something is wrong on DBeaver's side.

WARNING: disable JDBC tracing in your regular work. Enable it only for debugging. The trace generation decreases application performance and may produce huge log files.
Thread dump

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

**Mac and Linux**

Run the following on your terminal:

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

**Windows**

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

```bash
jstack <PID> > thread-dump.txt
```

in the Command Prompt.

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

External resources access

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

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

If you are behind some corporate firewall which restricts access to external internet resources then it may become a real problem. Sometimes corporate firewalls allow access to external resources using a web browser but restricts this for all other applications.

How to configure a proxy for drivers download

You need to ask your network administrator about proxy parameters. Then go to Preferences->Connections->Drivers.

![Proxy configuration](image)

You can enter a proxy host/port and (optionally) user/password here. It will be used for drivers download only. Drivers are usually downloaded from maven.org web site. You may also ask your network admin to add [maven.org](http://maven.org) to the list of allowed external domains.

How to configure network for license activation

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

Go to Preferences->General->Network Connections:
Switch to Manual or Native proxy (native proxy settings will use an active web browser proxy configuration).
Note: in order to activate/update a license DBeaver only needs to access the website `dbeaver.com`. You may ask your network administrator to add `dbeaver.com` to the white list.

How to configure a proxy for external databases access

You can configure proxy settings for an individual connection. You may set the proxy settings manually or use the active OS/web browser settings:
Managing connections

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

Provide predefined connections

DBeaver keeps connections information in the project folder. By default, all projects reside in the workspace. The default project folder is workspace/workspace6/General.

DBeaver 6.1.3+

DBeaver keeps information about project connections in the .dbeaver/data-sources.json file. All secured information (user name, password, secret keys, etc) is stored in the encrypted file, .dbeaver/credentials-config.json.

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

DBeaver < 6.1.3

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

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

Importing connections from CSV/XML

You can import a connection from CSV or XML files.

The CSV file must have a header row (first line of file) with column names (see list of supported columns below). The XML file should contain a top-level element and a set of nested elements. The connections config must be specified in the attributes of the nested elements. Attribute names are the same as the CSV column names.

Supported names:

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Connection name</td>
</tr>
<tr>
<td>url</td>
<td>JDBC URL</td>
</tr>
<tr>
<td>host</td>
<td>Database server host name</td>
</tr>
<tr>
<td>port</td>
<td>Database server port</td>
</tr>
<tr>
<td>database</td>
<td>Database/schema name</td>
</tr>
<tr>
<td>user</td>
<td>User name</td>
</tr>
<tr>
<td>password</td>
<td>User password</td>
</tr>
</tbody>
</table>

You can only specify the URL or the host/port/etc setting. User name/password are optional.

Sample CSV

<table>
<thead>
<tr>
<th>name, host, port, server, database, url, user, password, type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Postgre Import XML 1,localhost,5432,postgres,jdbc:postgresql://localhost:5432/postgres,postgres,postgres,dev</td>
</tr>
<tr>
<td>Postgre Import XML 2,localhost,5432,postgres2,jdbc:postgresql://localhost:5432/postgres2,postgres2,postgres2,prod</td>
</tr>
</tbody>
</table>

Sample XML

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

It is possible to set the connection settings as read-only (protected by password)

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

- Add field `lockPassword` in the connection descriptor (in `.dbeaver/data-sources.json` in `connections` element. So it will look like this:

```json
"postgres-jdbc-161537836e8-3e0957d039955715": {
  "provider": "postgresql",
  "driver": "postgres-jdbc",
  "name": "PostgreSQL - postgres",
  "save-password": true,
  "show-system-objects": true,
  "read-only": false,
  "folder": "PG",
  "lockPassword": "2ba81a47c5512d9e23c435c1f29373cb"
...}
```

- If the user will try to change connection settings now, he/she will be asked for a password.
Managing drivers

Configure drivers with pre-installed jars

You can customize drivers configuration in the workspace/.metadata/.plugins/org.jkiss.dbeaver.core/drivers.xml file. If you have some pre-installed jar files you can reference them in drivers.xml. Example:

```xml
<library type="jar" path="absolute-jar-folder-path\driver.jar" custom="true"/>
```

Also in drivers.xml you can use the following variables to specify relative paths:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>drivers_home</td>
<td>Standard DBeaver drivers location - ($workspace/drivers by default)</td>
</tr>
<tr>
<td>dbeaver_home</td>
<td>DBeaver installation folder</td>
</tr>
<tr>
<td>home</td>
<td>User home folder</td>
</tr>
<tr>
<td>workspace</td>
<td>DBeaver workspace path</td>
</tr>
</tbody>
</table>

For instance:

```xml
<library type="jar" path="${workspace}\drivers\my-driver.jar" custom="true"/>
```

Full drivers.xml example:

```xml
<?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="42.2.20">
        <file id="org.postgresql:postgresql" version="42.2.20" path="${drivers_home}/maven/maven-central/org.postgresql/postgresql-42.2.20.jar"/>
      </library>
      <library type="jar" path="maven:/net.postgis:postgis-jdbc:RELEASE" custom="false" version="2.5.0">
        <file id="net.postgis:postgis-jdbc" version="2.5.0" path="${drivers_home}/maven/maven-central/net.postgis/postgis-jdbc-2.5.0.jar"/>
      </library>
      <library type="jar" path="maven:/net.postgis:postgis-jdbc-jtsparser:RELEASE" custom="false" version="2.5.0">
        <file id="net.postgis:postgis-jdbc-jtsparser" version="2.5.0" path="${drivers_home}/maven/maven-central/net.postgis/postgis-jdbc-jtsparser-2.5.0.jar"/>
      </library>
    </driver>
  </provider>
</drivers>
```

Provide predefined drivers configuration

In some cases you may need to provide a driver's configuration or driver jar files for a number of DBeaver installations automatically. This can be done by adding a special parameter in the $DBEAVER.INI file: `-Dbeaver.drivers.configuration-file=c:\some\path\dbeaver-drivers-config.xml`

This file has the same structure as drivers.xml file (see above) and it will be loaded before drivers.xml.

You can specify partial driver configuration. For example if you need to configure only the jar path then it may look like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<drivers>
  <provider id="generic">
    <driver id="netezza">
      <library type="lib" path="X:jdbc-drivers\netezza-jdbc.jar"/>
    </driver>
  </provider>
</drivers>
```
Windows Silent Install

It is possible to install DBeaver in silent mode using the 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 the [https://github.com/Drizin/NsisMultiUser](https://github.com/Drizin/NsisMultiUser) team.

Parameters

Command line parameters supported by DBeaver installer:

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

In order to install with the /allusers parameter the current user must have the administrator's permission.

Installer return codes (decimal):

<table>
<thead>
<tr>
<th>Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>normal execution (no error)</td>
</tr>
<tr>
<td>1</td>
<td>(un)installation aborted by user (Cancel button)</td>
</tr>
<tr>
<td>2</td>
<td>(un)installation aborted by script</td>
</tr>
<tr>
<td>666660</td>
<td>invalid command-line parameters</td>
</tr>
<tr>
<td>666661</td>
<td>elevation is not allowed by defines</td>
</tr>
<tr>
<td>666662</td>
<td>uninstaller detected there is no installed version</td>
</tr>
<tr>
<td>666663</td>
<td>executing uninstaller from the installer failed</td>
</tr>
<tr>
<td>666666</td>
<td>cannot start elevated instance</td>
</tr>
<tr>
<td>other</td>
<td>Windows error code when trying to start elevated instance</td>
</tr>
</tbody>
</table>

DBeaver user guide
License Administration

Note: This functionality is available only in Enterprise Edition.

Manual license import

DBeaver EE asks the user to import the license file if the file cannot be found locally. For individual users this is the most simple and convenient way to import the product license.

![License Manager](image)

License management automation

There are several ways to automate the license management process. It makes sense for a multi-user environment.

Put the license file to the predefined locations

While DBeaver is starting up, it will look for a license file in the following locations:

- **Windows**
  - \%HOMEPATH\%\dbeaver-ee-license.dat
  - \%APPDATA\%\DBeaverData\workspace6\metadata\dbeaver-ee-license.dat

- **MacOS X**
  - ~/\dbeaver-ee-license.dat
  - ~/Library/DBeaverData/workspace6/.metadata/.dbeaver-ee-license.dat

- **Linux**
  - ~/.dbeaver-ee-license.dat
  - $XDG_DATA_HOME/DBeaverData/workspace6/.metadata/.dbeaver-ee-license.dat

Passing license file through command line

You can add the command line parameter `license <license-path>` to the DBeaver EE shortcut. Also you can add this parameter to `dbeaver.ini` config file.

Command line reference.
How to Import License

- Import from email
- Import from the personal account
- Insert the License key to License Manager
- Import of Subscription license
- Import of License extension
- License Manager

To start using DBeaver EE you can

- request a Trial license for 2 weeks;
- request an Academic license if you are a student or a teacher;
- buy a Subscription license, Standard DBeaver EE license or DBeaver EE license extension.

After purchasing the DBeaver EE license or getting the Trial/Academic license, you will receive a License text by email. It will also be available in your personal account on our site. This License text will contain your License ID e.g. DB-821MPZFO-ZA8W, the start date and license owner’s name and company name. It is very important to import your License correctly.

Import from email

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

Hello,

Thank you for your interest to DBeaver.

We are glad to see you among users of the enterprise version. Please, find your license below:

```
-- DBeaver EE LICENSE - DB-1Q1M03DB-35K5
-- Starts at Tue Mar 02 07:46:42 UTC 2021 to DBeaver Corp. / Joe Test //
S0czwxdker6ftfH45jkPRtt2PFL/8Wwq+qA5A0ymn/E/np1cGNSv5q175HqAptTUKx7kX8flolq
csn+jamxcujjuasbrX5Sgf8W0Fw8K8LJ+jZSP7v/0cc3f5j0LtcEHK/ap6evz0hj7gN9i6n2ZqG
hJafv9i3f1LmCkWaw18w1ynJ6RANN+sDTaj5tba6w6tt70ha3u7Huf+i8Sv/wj3EJARC3
AO1orctCUVljjE82EalCmysx3Wmz88nPqP9O1oljE+B5XNIgjLxwsw3P5f7psGy8Yko+b1WfFL
q4eq/af0e9vF11qmg9Q32t2GxMQ/kDzXE/Fxiea==
```

Enjoy your database research with DBeaver and good luck!

Regards,

DBeaver team

Sometimes an email client can corrupt the formatting of the License Key that can cause an error.
Therefore, you need to import your License Key from your personal account on our site https://dbeaver.com/.

**Import from the personal account**

Firstly, you need to Sign in.

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

To open the License details and copy the License Key press the View button. You can see the License details where you can find your license status, type, maintenance period, and end support date. You can also reassign the license to another user.
At the bottom of the page you can find the License Key required to start using DBeaver EE. There are two options how to copy your License Key from the personal account:

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

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

.txt file with your License Key will be downloaded to your download folder. The file name is License ID, e.g. DB-821MPZFO-ZA8W.
Then you need to insert the copied License Key to License Manager in Dbeaver EE.

**Insert the License Key to License Manager**

To start using DBeaver EE with your License Key you need to open License Manager in DBeaver EE: Help → DBeaver License Info

The License information window can look different depending on whether you already have a valid license or not.
Then click **License Manager** where you can find all your imported licenses and information about them.

Next press the **Import** button to paste your License Key.

If you copied the License Key to the clipboard, press the **Paste** button and then **Import**.
If you downloaded the .txt file with the License Key, press the **Load** button and select the file from the Downloads. The License Key will be pasted automatically.

Then press the **Import** button and your license will be added to the License Manager.

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

**Import of Subscription license**

Subscription license requires internet access on the workstation for the first activation and each prolongation.

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

*Invalid subscription*
Can’t find the subscription information for license ‘DB-821MPZFO-ZA8W’.

Check your internet connection and/or firewall settings and restart application.

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

**License extension**

The standard DBeaver EE license is a perpetual license with a limited period of support (1 year or 2 years).

After the end of the selected support period you can continue to use DBeaver Enterprise Edition without support and updates or buy a license extension or a new license.

If you buy the DBeaver EE license extension and Dbeaver has internet access, the license in DBeaver EE will be updated automatically. Otherwise, you have to import the license key from your personal account once again.

**License Manager**

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

- License ID e.g. DB-821MPZFO-ZA8W;
- License type: Trial/Academic/Subscription/Standard;
- Version;
- License owner’s name and company name;
- License owner’s email;
- Start time is the date the license was received;
- End time is the date the license expires (standard perpetual licenses do not have this)
- Number of users: single user or multiuser for group licenses;
- Support period is the period you have access to the internal support system on the site and the possibility to download new product versions;
- State: valid or expired.
How to Reassign License

After purchasing a bunch of DBeaver EE licenses, you have to assign each license to an end user.

If an employee subsequently is leaving the company or the team that is using DBeaver, the license admin may need to reassign the license to another employee.

You can reassign the license to another user in your personal account.

Firstly, you need to Sign in.

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

Select which license you need to reassign and press the VIEW button near its license ID. You can see the License details where you can find your license status, type, maintenance period, and end support date.
Also, there is the license end user field that contains the license end user’s details. To reassign the license, you need to press the CHANGE END USER button near this field.

A pop-up opens, and you can enter the new license end user's data: email; first and last name; and company. After filling in the form, you need to press the SAVE button and the license owner will be changed. The license key that contains the license end user’s name and company name will be changed too.
The license will remain in your personal account. The end user's email will be in the end user column on the Licenses tab.

The license will remain in your personal account. The end user's email will be in the end user column on the Licenses tab.

Also, the new license end user will be able to find the license in the personal account and import it. There will be no CHANGE END USER button because it is only the license customer who can assign a license to the end user. If the new license end user has not been signed up on our website, the account will be created automatically. The new user will receive a welcome email with a link to set up a password.