DBeaver user guide
Table of contents

General User Guide

Installation
Application Window Overview
Views
  Database Navigator
  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 XML and JSON
  Managing Data Formats
ER Diagrams
  Database Structure Diagrams
  Custom Diagrams
SQL Editor
  SQL Templates
  SQL Assist and Auto-Complete
  SQL Formatting
  SQL Execution
  Visual Query Builder
  Script Management
  Client Side Commands
Debug
  PostgreSQL Debugger
Filters
  Filter Database Objects
  Configure Filters
Search
  File Search
  DB Full-Text Search
  DB Metadata Search
Dashboards, DB monitoring
Projects
Bookmarks
Shortcuts

Database Management

Database Connections
  Create Connection
  Edit Connection
  Connect to Database
  Invalidate/Reconnect to Database
  Disconnect from Database
Connection Types
Transactions
  Auto and Manual Commit Modes
  Transaction Log
  Pending transactions
Database drivers
Mock Data generation
Data export/import
Data migration
Database compare

Customizing DBeaver
DBeaver extensions - Office, Debugger, SVG, SSH
Installing extensions - Themes, version control, etc

Databases
Generic
MySQL/MariaDB
PostgreSQL
Oracle
Exasol
DB2
Apache Hive/Spark/Impala
MongoDB
Cassandra
InfluxDB
Redis
Clickhouse

Troubleshooting
Command-Line
Reset UI settings
Reset workspace
Posting issues
Log files
Thread dump

Admin Guide
Managing connections
Managing drivers
Windows Silent Install

Development
Build from sources
Develop in Eclipse
Develop in IDEA
Features structure
Resources localization
Automatic Tests

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

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

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

ZIP Archive
When installing DBeaver manually, without using an installer:

1. Install Java 1.8.
2. Extract the contents of the archive.
   - NOTE: Do not unzip the archive over a previous DBeaver version. If you already have any version of DBeaver extracted in the same location - remove it before unzipping the new version.
   - NOTE: All configurations, scripts and other necessary data are stored in a separate location (usually in the user’s home directory) so the program deinstallation does not affect them.
3. Run the dbeaver executable.

Debian Package
To install DBeaver using a Debian package:

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

RPM Package
To install DBeaver using RPM package:

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

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

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

You can also use brew:

brew install Caskroom/cask/java
Application Window Overview

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

Menu Bar

By default, the menu bar contains the following menus:

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

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

Toolbar

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

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

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

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

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

**Workspace: Views and Editors**

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

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

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

There can be several views and editors simultaneously open in the workspace but only one of them can be active at a time.

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

**Changing Workspace Layout**

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

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

You can also swap locations of two views or editors.

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

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

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

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

**Maximizing, Minimizing and Restoring View and Editors**

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

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

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

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

When one view is maximized, other views and editors appear as shortcuts on the shortcut bar.
To restore a maximized view or editor to its latest docked position, double-click its title bar or click the Restore button in its upper-right corner.

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

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

To minimize a view, do one of the following:

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

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

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

To open a view:

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

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

Views provide their own toolbar and menu:

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

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

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

Database Navigator is the main view to work with the structure and content of databases. To open Database Navigator, on the Windows menu, click **Database Navigator**. For information on how to change the view layout, please see the Application Window Overview article.

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>🛠️</td>
<td>Driver Manager</td>
<td>Opens the Driver Manager window that allows creating, editing and deleting drivers for databases. See Database Drivers for information about managing database drivers.</td>
</tr>
<tr>
<td>🔒</td>
<td>New Connection</td>
<td>Opens the Create new connection wizard. See Create Connection for information about creating connections.</td>
</tr>
<tr>
<td>(empty)</td>
<td>Active Project</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>📃</td>
<td>New Folder</td>
<td>Opens a dialog box for creating a new folder</td>
</tr>
<tr>
<td>🍃</td>
<td>Collapse All</td>
<td>Collapses the tree to the root level</td>
</tr>
<tr>
<td>⚪️</td>
<td>Link with editor</td>
<td>Enabled when at least one editor is open, otherwise disabled - highlights the object in the tree that has its editor open</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 context menu items for all types of objects that may appear in the tree. Note that the presence or absence of context menu items for an object depends on the database and object types.

<table>
<thead>
<tr>
<th>Menu item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Menu item</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Open folder</td>
<td>Opens a folder in a separate view</td>
</tr>
<tr>
<td>Create new connections / New Folder</td>
<td>Opens the Create new connection wizard or 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>
</tbody>
</table>
| Delete                          | Deletes an object  
**WARNING!** The Delete menu item removes the object not only from the tree but from the database itself or the file system, and this action is irreversible. |
| Rename                          | Opens the Rename [object] dialog box                                                                                                                                                                   |
| Properties                      | Opens the Properties for [object] window which allows viewing and modifying the object’s properties                                           |
| Refresh                         | Depending on the object, refreshes the object itself, or its parent, or its subnodes – mostly used for refreshing tables and schemes         |
| Connect                         | Attempts to connect to the database                                                                                                                                                                    |
| Invalidate/Reconnect            | Checks the status of connection, if it is broken, attempts to reconnect                                                                                                                                |
| Disconnect                      | Disconnects from the database                                                                                                                                                                         |
| SQL Editor                      | Opens a new SQL editor for the connection                                                                                                                                                            |
| Recent SQL Editor               | Opens the most recently opened SQL editor                                                                                                                                                             |
| Edit Connection                 | Opens the Connection Configuration window that allows configuring connection settings                                                                                                                  |
| View [objects]                  | - For objects that DBeaver can create and delete, opens the object in a separate editor  
- For objects that DBeaver cannot create and delete, opens the object in a separate viewer                                                                 |
| Edit [object]                   | - For objects that DBeaver can create and delete, opens the object in a separate editor  
- For objects that DBeaver cannot create and delete, opens the object in a separate viewer                                                                 |
| Create new [object]             | Opens an editor in which you can specify properties and save the new object                                                                                                                             |
| Filter                          | Opens a submenu of one or more filtering options (depending on the object):  
- Hide [object]  
- Show only [object]  
- Configure [objects] filter  
- Toggle filter  
- Clear filter  
See Filters for information. |
| Copy Advanced Info              | Copies the full name of an object                                                                                                                                                                    |
| Read Data in SQL Console        | Opens an SQL console displaying the object’s data                                                                                                                                                     |
| Copy Advanced Info              | Copies the full name of an object                                                                                                                                                                    |
| Read Data in SQL Console        | Opens an SQL console displaying the object’s data                                                                                                                                                     |
| Compare                         | - Appears only if you select several objects of the same level  
- Opens the Compare objects wizard which guides you through the steps to generate a comparison report for the selected objects |
| Generate SQL                    | Opens a submenu on which you can select the type of SQL query to generate:  
- SELECT  
- INSERT  
- UPDATE  
- DELETE  
- MERGE  
- DDL  
Clicking one of the items (for example INSERT) generates a relevant query in a separate window. |
| Export Table Data               | Opens the Data Transfer wizard that helps you select a format and export table data                                                                                                                    |
| Import Table Data               | Opens a window with existing database connections in which you can select a table to import data from                                                                                                  |
| Tools                           | Opens a submenu that provides tools for database backup and restore, vacuum, etc.                                                                                                                     |

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

You might need to classify and group database connections into projects. Projects store objects related not to a particular database but to all database connections. These are usually files stored on the file system.

The Projects view displays all projects created in the system and provides tools to manage them. To open the Projects view, on the Window menu, click Projects (or use ALT+W+P shortcut).

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

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

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

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

<table>
<thead>
<tr>
<th>Icon</th>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🏕️</td>
<td>Create Project</td>
<td>Opens the Create Project wizard</td>
</tr>
<tr>
<td>🔄</td>
<td>Refresh Projects</td>
<td>Refreshes the projects tree to display changes caused by creating modifying or deleting projects</td>
</tr>
<tr>
<td>🌳</td>
<td>Collapse All</td>
<td>Collapses the tree to the root level</td>
</tr>
<tr>
<td>🛡️</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>

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

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

- The Set Active Project menu item (for a project root node) makes the project active, that is visible in the Database Navigator.
- The Link File (SQL Script) and Link Folder menu items allow creating links to files and folders in the file system.

For information about managing projects, please see Projects article.
Project Explorer

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

![Project Explorer](image)

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

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

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

![Query Manager toolbar](image)

- On the Window menu, click Show View -> Query Manager:

![Query Manager in DBeaver](image)

The Query Manager logs all queries together with their execution statistics (execution time, duration, number of fetched/updated rows, errors, etc.).

![Query Manager log](image)

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

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

Query Manager Properties

To manage the look of the Query Manager, filter the entries, and modify storage settings, click the Set query manager filter button (Filter icon) 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 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.
Properties Editor

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

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

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

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

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![search icon]</td>
<td>Search items</td>
<td>Displays a search field next to the button:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Type in the search combination - the content updates dynamically</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- To remove the filter, click the cross icon next to the search field</td>
</tr>
<tr>
<td>![filter icon]</td>
<td>Filter settings</td>
<td>Opens the Filtering window which allows setting a custom filter, see Configure Filters</td>
</tr>
<tr>
<td>![configure icon]</td>
<td>Configure columns</td>
<td>Opens the Configure columns dialog box in which you can select the columns to display or hide in the current view</td>
</tr>
<tr>
<td>![refresh icon]</td>
<td>Refresh the selected items</td>
<td>Depending on the database type, refreshes either the current item or its parent or the whole database object – reloading data from the database</td>
</tr>
<tr>
<td>![view icon]</td>
<td>View</td>
<td>Opens an editor/viewer for the item currently in focus</td>
</tr>
<tr>
<td>![create icon]</td>
<td>Create new [items]</td>
<td>Creates a new item of the same type as currently displayed in the open view, for example, a column</td>
</tr>
<tr>
<td>![delete icon]</td>
<td>Delete database object</td>
<td>Deletes the item currently in focus</td>
</tr>
<tr>
<td>![save icon]</td>
<td>Save</td>
<td>- Same as the Save button on the application main toolbar</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Opens the Persist Changes window that allows saving changes in the currently open sub-entity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NOTE: DBeaver recommends saving work after each change.</td>
</tr>
<tr>
<td>![revert icon]</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><img src="image" alt="Load form file" /></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><img src="image" alt="Save to file" /></td>
<td>Save to file</td>
<td>Allows saving the current SQL code to a file</td>
</tr>
<tr>
<td><img src="image" alt="Open in SQL console" /></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.
The Data editor appears:

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

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

To learn how many rows the data table contains, click the **Calculate total row count** button in the bottom toolbar. The number of rows appears in a status field next to the button: **8,715**.

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

The top toolbar contains the following buttons:

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="Image" alt="Apply filter criteria" /></td>
<td><strong>Apply filter criteria</strong></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><img src="Image" alt="Remove all filters/orderings" /></td>
<td><strong>Remove all filters/orderings</strong></td>
<td>Removes all filters and orderings applied to the data</td>
</tr>
<tr>
<td><img src="Image" alt="Save filter settings for current object" /></td>
<td><strong>Save filter settings for current object</strong></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 Data Filters article</td>
</tr>
<tr>
<td><img src="Image" alt="Custom Filters" /></td>
<td><strong>Custom Filters</strong></td>
<td>Opens the Result Set Order/Filter Settings window, see Data Appearance article for more information</td>
</tr>
<tr>
<td><img src="Image" alt="Configure auto-refresh" /></td>
<td><strong>Configure auto-refresh</strong></td>
<td>Allows configuring data auto-refresh settings, see Data Refresh article for details</td>
</tr>
<tr>
<td><img src="Image" alt="Forward and backward - history navigation buttons" /></td>
<td><strong>Forward and backward - history navigation buttons</strong></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: <strong>Alt+Left</strong> (backward) and <strong>Alt+right</strong> (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><img src="Image" alt="Grid" /></td>
<td><strong>Grid</strong></td>
<td>Switches to grid view of data</td>
</tr>
<tr>
<td><img src="Image" alt="Text" /></td>
<td><strong>Text</strong></td>
<td>Switches to plain text view of data</td>
</tr>
<tr>
<td><img src="Image" alt="Chart" /></td>
<td><strong>Chart</strong></td>
<td>Switches to chart view. For more details on charts, see Managing Charts article.</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></td>
<td><strong>Save</strong></td>
<td>Saves all unsaved changes to the data such as adding, duplicating, deleting rows, inline editing of values, see Data Viewing and Editing article</td>
</tr>
<tr>
<td></td>
<td><strong>Cancel</strong></td>
<td>Discards all unsaved changes to the data</td>
</tr>
<tr>
<td></td>
<td><strong>Script</strong></td>
<td>Opens the Preview Changes window in which you can see changes that you have made to the data, see details in Data Viewing and Editing article</td>
</tr>
<tr>
<td></td>
<td><strong>Edit cell value in separate dialog/editor</strong></td>
<td>Opens the cell in focus for editing in a separate editor or dialog box, see details in Cell Editor section of Data Viewing and Editing article</td>
</tr>
<tr>
<td></td>
<td><strong>Add new row</strong></td>
<td>Adds a new empty row below the current row, see details in Adding, Copying and Deleting Rows section of Data Viewing and Editing article</td>
</tr>
<tr>
<td></td>
<td><strong>Duplicate current row</strong></td>
<td>Copies the current rows and pastes the copy below the current row, see details in Adding, Copying and Deleting Rows section of Data Viewing and Editing article</td>
</tr>
<tr>
<td></td>
<td><strong>Delete current row</strong></td>
<td>Colors the rows in focus in red to mark them for deletion, see details in Adding, Copying and Deleting Rows section of Data Viewing and Editing article</td>
</tr>
<tr>
<td></td>
<td><strong>Move to first row</strong></td>
<td>Moves the focus (highlighting) from the current to the first row of the table</td>
</tr>
<tr>
<td></td>
<td><strong>Move to previous row</strong></td>
<td>Moves the focus (highlighting) from the current to the previous row of the table</td>
</tr>
<tr>
<td></td>
<td><strong>Move to next row</strong></td>
<td>Moves the focus (highlighting) from the current to the next row of the table</td>
</tr>
<tr>
<td></td>
<td><strong>Move to last row</strong></td>
<td>Moves the focus (highlighting) from the current to the last row of the table</td>
</tr>
<tr>
<td></td>
<td><strong>Fetch next page of results</strong></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><strong>Fetch all rows</strong></td>
<td>Fetches the whole result set making it ready for display, see Scrolling Results Page section of Navigation article for more information</td>
</tr>
<tr>
<td></td>
<td><strong>Panels</strong></td>
<td>Opens panels on the right side of the Data Editor, see Panels for information</td>
</tr>
<tr>
<td></td>
<td><strong>Configure</strong></td>
<td>Opens a dropdown menu with settings</td>
</tr>
<tr>
<td></td>
<td><strong>JSON</strong></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><strong>XML</strong></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><strong>Generate Mock Data</strong></td>
<td>Opens the Mock Data Generator window, see TBD for details</td>
</tr>
<tr>
<td></td>
<td><strong>Rows count details</strong></td>
<td>Opens the Status details dialog box showing the timing details of fetching table rows</td>
</tr>
<tr>
<td></td>
<td><strong>Calculate total row count</strong></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>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>
</tbody>
</table>

Record - Same as pressing **Tab**
- Switches the positions of rows and columns so that columns appear as rows, and rows hide in one **Value** column, see details in the **Table vs. Record Views** section of Data Appearance article.
<table>
<thead>
<tr>
<th>Menu Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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 row red 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.</td>
</tr>
<tr>
<td></td>
<td>By default, DBeaver filters data by sending a request to the server (the Server-side results ordering checkbox selected). To filter data on the client side using DBeaver’s internal algorithm, clear the checkbox.</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: 4,715. 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:

- Maximum result-set size

Once you scroll to the last row of the current result portion, DBeaver fetches the next portion (next N rows). You can disable this behavior in preferences. You can also manually fetch the next portion of data equal to the maximum result set size. To do so, click the **Fetch next page of results** button (كتابة) 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:

- 1,000 rows fetched - 0ms

To see the details, click the details button in the status field.

To fetch the whole result set, click the **Fetch all rows** button (كتابة) 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:

- First row
- Next row
- Previous row
- Last row

To jump to a specific line, right-click anywhere in the table and click **Navigate -> Go to Line** on the context menu. Then in the Go to Row dialog box, enter the row number and click OK.

**History**

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

- Use the forward and backward buttons in the top toolbar: \[← \rightarrow\]
- Click Ctrl+Left or Ctrl+Right

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

**Navigate Foreign Keys / Referencing Tables**

You can navigate by foreign keys or to referencing tables – those that reference the current table. To open a referencing table, press Ctrl+1 or right-click the cell and click **Navigate->Referencing tables-[table name]**.
The referencing table opens in the same editor. To navigate back and forth between the initial and referencing tables, use the history navigation buttons (← →) in the top toolbar of the editor. In order to open referencing table in a new window use Ctrl+Shift+1 shortcut to show menu.

### Navigation Links

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

<table>
<thead>
<tr>
<th>#</th>
<th>ArtistId</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>António Carlos Jobim</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>#</th>
<th>ArtistId</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>António 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 and click the Navigate link icon (🔗) in the cell.
Data View and Format

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

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, see … below

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

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

Ordering Data in Columns

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

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

   The icon has three states:

   - Clicking once establishes ascending order ( )
   - Clicking a second time changes the order to descending ( )
   - Clicking a third time removes the ordering from the column ( )

   To order data by several columns, go column by column, setting the order with the Ordering icon, starting from the column by which you want to order data first.

2. Click the Custom Filters button ( ) in the top toolbar of the editor to open the Result Set Order/Filter Settings window (see above):
a) Next to the column by which you want to order data in the first turn, set the ascending or descending order using the same three-state principle as described above.
b) Set the ordering in other columns by which you want to sort data in the second, third, etc. turn. The Order column indicates the order in which the sorting will happen.

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

<table>
<thead>
<tr>
<th>Name</th>
<th>#</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlbumId</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ArtistId</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Column1</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

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 (ToLeft) in the same window.

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

Managing Display of Columns in Data Table

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

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.

   ![Checkboxes for column visibility](image)

2. Use the Show All (ToLeft) and Show None (ToLeft) buttons at the bottom of the window.

Sorting Columns in Data Table

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

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

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

3. In the Result Set Order/Filter Settings window, click the column to set focus to it and then move it using the navigation buttons: ( ⬆️ ⬇️)

Grid vs. Plain Text Views

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

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

Table vs. Record Views

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlbumId</td>
<td>22</td>
</tr>
<tr>
<td>Title</td>
<td>Sozinho Remic Ao Vivo</td>
</tr>
<tr>
<td>ArtistId</td>
<td>15</td>
</tr>
<tr>
<td>Column1</td>
<td>[NULL]</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 in the
To return back to the standard table view, click the Record button again.

**Rows Coloring**

In the data editor, you can color all rows that have the same value as a particular cell of a certain column. To do so, right-click the cell and click View/Format -> Color by {column name} on the context menu:

![View/Format -> Color by {column name}](image)

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

<table>
<thead>
<tr>
<th>AlbumId</th>
<th>Title</th>
<th>ArtistId</th>
<th>Column1</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Restless and Wilt</td>
<td>2</td>
<td>[NULL]</td>
</tr>
<tr>
<td>4</td>
<td>Let There Be Rock</td>
<td>1</td>
<td>engineq</td>
</tr>
<tr>
<td>5</td>
<td>Big Ones</td>
<td>3</td>
<td>[NULL]</td>
</tr>
</tbody>
</table>

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

**Coloring by Data Types**

Besides coloring rows by a value, you can colorize values in columns by data types. To do so, right-click any cell in the table and, on the context menu, click View/Format -> Colorize Data Types. Values in cells are colored in different colors according to preferences currently set:

![View/Format -> Colorize Data Types](image)

You can change the color preferences in the Preferences window: click Window -> Preferences on the main menu. Then, in the window, in the navigation pane on the left, expand General and then Appearance, and then click Colors and Fonts:
To remove coloring by data types, on the context menu, click View/Format -> Colorize Data Types again.

### Transforming Data Presentation

For string and numeric data types, DBeaver provides tools to transform the data presentation into a number of formats, such as URL and Binary for strings and Epoch Time, Number Radix, etc. for numbers. To change the data presentation in a certain column, right-click a cell in the column, then, on the context menu, click View/Format -> View as and then click the presentation type name:

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

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

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

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

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

One of the ways is to use the filter field above the table next to the top toolbar. To filter data, enter an SQL expression into the field and click the Apply filter criteria button (agination) 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 column value dialog box:

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

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

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

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

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

   ![Auto-refresh configuration 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 the two alternative options above, the system starts refreshing the data as scheduled and the Configure auto-refresh button changes to Stop auto-refresh button ( ). To stop the auto-refresh, click the Stop auto-refresh button or click the arrow next to it and click Stop on the auto-refresh menu.
Data Viewing and Editing

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

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

Inline Editing

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

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

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

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

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

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

### Adding, Copying and Deleting Rows

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

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

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

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

### Copying/Pasting Cells

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

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

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

### Defining Virtual Keys

To be able to persist column value changes, a table must have some unique key (primary key or unique index). Some databases (Oracle, DB2, PostgreSQL) support a special virtual unique column that DBBeaver 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, DBBeaver displays the following error message:
To use all columns as the virtual key, click **Use All Columns**. To create a custom key, click **Custom Unique Key**. Alternatively, to create a custom unique key, you can click the **Configure** button in the bottom toolbar and then click **Define virtual unique key** on the Configure menu. The Define virtual unique identifier window opens:

To define the key, select some of the columns or click **Select All** and then click **OK**. To remove a unique key from a table, click the **Configure** button in the bottom toolbar and then click **Clear virtual unique key**.
Panels

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

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

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

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

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

**Value Viewer**

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

<table>
<thead>
<tr>
<th>Button</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Content</td>
<td>Opens a menu with a set of options for content view change.</td>
</tr>
<tr>
<td>viewer</td>
<td>settings</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Save to file</td>
<td>Allows saving the content to a local file. NOTE: This button is only available for XML, JSON and Binary content.</td>
</tr>
<tr>
<td></td>
<td>Load from file</td>
<td>Allows uploading data from a local file. NOTE: This button is only available for XML, JSON and Binary content.</td>
</tr>
</tbody>
</table>
**Apply cell value**
Displays in the data table the changes made in the Value viewer. **NOTE:** This button does not save changes made to the database. To save the changes in the database, you need to use the **Save** button in the bottom toolbar of the **Data Editor**.

**Auto-apply value**
Enables automatic display of changes made in the Value viewer in the data table. When auto-saving is enabled, the changes appear in the data table at the same time when they are made in the Value viewer. **NOTE:** This button does not save changes made to the database. To save the changes in the database, you need to use the **Save** button in the bottom toolbar of the **Data Editor**.

---

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Label</th>
<th># Type</th>
<th>Catalog Name</th>
<th>Schema Name</th>
<th>Table Name</th>
<th>Max Length</th>
<th>Precision</th>
<th>Scale</th>
<th>JDBCType</th>
<th>Not Null</th>
<th>Auto</th>
</tr>
</thead>
<tbody>
<tr>
<td>film_id</td>
<td>film_id</td>
<td>0 serial</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>11</td>
<td>10</td>
<td>0</td>
<td>INTEGER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>title</td>
<td>title</td>
<td>1 varchar</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>235</td>
<td>235</td>
<td>0</td>
<td>VARCHAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>description</td>
<td>description</td>
<td>2 text</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>2,147,483,647,2,147,483,647</td>
<td>0</td>
<td>VARCHAR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>language_id</td>
<td>language_id</td>
<td>4 int2</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>SMALLINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>original_language_id</td>
<td>original_language_id</td>
<td>5 int2</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>SMALLINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rental_duration</td>
<td>rental_duration</td>
<td>6 int2</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>SMALLINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rental_rate</td>
<td>rental_rate</td>
<td>7 numeric</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>6</td>
<td>4</td>
<td>0</td>
<td>NUMERIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>length</td>
<td>length</td>
<td>8 int2</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>SMALLINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>replacement_cost</td>
<td>replacement_cost</td>
<td>9 numeric</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>NUMERIC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>rating</td>
<td>rating</td>
<td>10 mpaa_rating</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>2,147,483,647,2,147,483,647</td>
<td>0</td>
<td>OTHER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>last_update</td>
<td>last_update</td>
<td>11 timestamp</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>TIMESTAMP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>special_features</td>
<td>special_features</td>
<td>12 text</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>2,147,483,647,2,147,483,647</td>
<td>0</td>
<td>OTHER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fulltext</td>
<td>fulltext</td>
<td>13 text</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>2,147,483,647,2,147,483,647</td>
<td>0</td>
<td>OTHER</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>column1</td>
<td>column1</td>
<td>14 int2</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>SMALLINT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>column2</td>
<td>column2</td>
<td>15 boolean</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>BOOLEAN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>time_name</td>
<td>time_name</td>
<td>16 time</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>15</td>
<td>13</td>
<td>0</td>
<td>TIME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>date_name</td>
<td>date_name</td>
<td>17 date</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>15</td>
<td>13</td>
<td>0</td>
<td>DATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>date_time_name</td>
<td>date_time_name</td>
<td>18 timestamp</td>
<td>palgia</td>
<td>public</td>
<td>film</td>
<td>29</td>
<td>29</td>
<td>0</td>
<td>TIMESTAMP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

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

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

**Grouping Panel**

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

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

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 in the panel’s toolbar.

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

3. Click **OK**:

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

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

1. Click the **Edit grouping columns** button in the panel’s toolbar.

2. In the Grouping Configuration window, in the **Columns** area, click **Add**, then type the column name into the new row (you can use auto-complete options DBeaver provides), and then click **OK**.
For MySQL/MariaDB databases you can also add a column with an expression - the expression will be calculated in the resulting column:

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 (lesai) in the panel’s toolbar. Clicking the **Clear grouping** button (lesai) removes all results from the Grouping panel.
Managing Charts

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

Note: Analytical Charts present only in DBeaver Enterprise Edition

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

Creating Charts In SQL Editor

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

To build a bar chart in the SQL Editor, press the Charts button in the left vertical toolbar of the query results area.
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 sorting and applying various filters to its columns first. All the structural changes you make will then affect the chart you build, this way you can adjust the chart representation to the desired one.

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

A bar chart will be created.

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

Creating Charts In Grouping Panel

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

To build a chart in Grouping Panel, press the Charts button in the left vertical toolbar.
To edit chart settings select the Charts... option in the chart's context menu and the Chart Editor will appear.

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 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 opens:

You can also use the Find and Replace feature.

NOTE: The system searches only in already fetched rows.
SQL Generation

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

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 fully qualified names** checkbox.
To wrap the SQL query into one line, select the **Compact SQL** checkbox:
**Working with XML and JSON**

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

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

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

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

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 on database migration, for example.

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

In the Preferences dialog box go to Database -> Editors -> DataEditor -> 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 Result Sets -> Data Formatting 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:

### Data Type Format

The format of the following data types can be customized:

- Date
Date

The default value for this data type is yyyy-MM-dd.

Time

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

Timestamp

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

Numbers

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

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

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

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

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

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

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

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

To change the data type format, change the value displayed in the Pattern area, save the changes made by pressing the Apply button and observe the expected result in the Sample field.

Restoring Default Data Formats
To restore the default data format settings, press the **Restore Defaults** button.

To save changes made press **Apply**.
ER Diagrams

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

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

Both types of diagrams provide the same tools to adjust their view and structure. They can be printed and exported to image file formats.

Selection of Elements in Diagrams

You can use one of the two tools to select elements in diagrams:

- Select — supports both, single and multi-select modes. To select a single element (table, connection, entity inside a table) in a diagram, just click that element. To select multiple elements, similar to using the Marquee tool, click outside the first element and draw until all elements you need are in focus:

Structure Adjustment

NOTE: All changes to existing database schemas cannot be saved and are intended for exploration purposes only.

You can do the following structure changes in diagrams:

- Add new tables to a diagram by drag-n-dropping them onto the diagram field from the Database Navigator.
- Rearrange tables in the diagram by dragging them all over the space. You can select several tables and drag them to a new location.
- Auto-arrange tables into a compact view after manual rearrangements: click the Arrange Diagram button in the toolbar or on the context menu (right-click anywhere on the diagram tab).
- (Available for Custom Diagrams only) - connect tables with a connector: click the Show Palette button in the upper-left corner of the diagram tab and then, in the Palette panel, click Connection:
Now click the tables that you want to connect with each other in turn one by one, and then to stop the connection line double-clicking the last table.

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

### View Adjustment

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

- Enable/disable the diagram grid: Click Toggle Grid in the toolbar.
- Modify attributes visibility: Right-click the diagram and, on the context menu, click Show Attributes and then one of the options:
  - All - all attributes
  - Any keys - primary and foreign keys
  - Primary key - only primary keys
  - None - no attributes
- Modify attributes presentation: Right-click the diagram and, on the context menu, click View Styles and then one of the options:
  - Show Icons
  - Show Data Types
  - Show Nullability
  - Show Comments
  - Show Fully qualified names
- Change the color of entities/notes: Right-click the header of the entity or comment and then click Set color on the context menu. Then you can select the color and click OK.
- For elements located in front of/behind others, bring an element to front or send to back: Right-click the element and then click Bring to front / Send to back on the context menu.
- Zoom the diagram in/out: Click the Zoom In/Zoom Out buttons or choose the scaling value in the dropdown list in the toolbar:

### Refresh

To see changes made to the database schema by other users, you might need to refresh the diagram: click Refresh Diagram in the toolbar.

### Notes

You can create notes only in Custom Diagrams. To create a note, click the Show Palette button in the upper-left corner of the diagram tab, then, in the Palette panel, click Note and then click anywhere in the diagram tab. Now you can double-click Note box to enter the note text:
Search in Diagram Entities

To search among entities of a diagram, click the Search items button ( ) in the toolbar, then type in the search combination. The entities that contain the search combination are highlighted in the diagram. To remove the filter, click the cross icon next to the search field.

Diagram Export

You can export (save) a diagram as an image (PNG, GIF, BMP formats) or as a file in GraphML format. To export a diagram, click Save diagram in external format ( ) in the toolbar.

Diagram Printing

To print a diagram, press CTRL+P or click Print Diagram ( ) in the toolbar.

Settings

To modify the diagram settings, click Configuration ( ) in the toolbar.
Database Structure Diagrams

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

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

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

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

You can create custom ER diagrams that can contain any tables, relations and notes. However, even custom diagrams may contain only real existing database entities (tables).

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

1. On the DBeaver main menu, click **File -> New**. Then in the new diagram wizard, click **DBeaver -> ER Diagram**, and then **Next**:

2. In the **Project Explorer** view, right-click the **ER Diagrams** node and then click **Create New ER Diagram** on the context menu.

In both cases, in the Diagram Create Wizard, specify the diagram name and (optionally) choose initial diagram contents (set of tables):
The new diagram appears in a separate editor. Now you can drag-n-drop any number of tables onto it. You can add tables from different connections and even different database type (for example, combine Oracle and MySQL tables in one and the same diagram).

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

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

Undo/redo functions are fully supported in diagram editing.
SQL Editor

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

- Click this connection in the Database Navigator view and press F3 or click SQL Editor -> SQL Editor on the main menu. Alternatively, you click SQL Editor on the context menu of this connection. DBeaver opens the Choose SQL script editor with saved SQL scripts linked to this connection. Click the SQL script to open it in a separate tab.

- Click Recent SQL Editor on the context menu for this connection or on the main menu (SQL Editor -> Recent SQL Editor). This opens the latest used SQL editor. You can also open the most recent SQL editor using Ctrl+Enter shortcut in the Database Navigator view.

- If you need to create a new SQL script, on the main menu, click SQL Editor -> New SQL Editor or press F3 and then click New Script in the Choose SQL Script window.

DBeaver uses SQL syntax highlighting which depends on the database associated with the script. Different databases have different sets of reserved keywords and system functions.

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

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

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

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

Results Panel

The results panel displays tabs with results in various formats. The tabs resulting from script execution represent instances of the Data Editor. You can create, edit and execute SQL scripts in the script panel and then see the results in the result tabs.

The results panel provides Output and Log views of results.

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

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

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 \texttt{CTRL+6} or right-click anywhere in the script pane and, on the context menu, click \texttt{Layout -> Toggle results panel}.
- To maximize the results panel, press \texttt{CTRL+Shift+6}, or double-click the results tab name, or right-click anywhere in the script panel and, on the context menu, click \texttt{Layout -> Maximize results panel}.
- To switch between the script panel and the results pane, press \texttt{Alt+6} or right-click anywhere in the script panel and, on the context menu, click \texttt{Layout -> Switch active panel}.

To position both panels horizontally, right-click anywhere in the script pane and, on the context menu, click \texttt{Layout -> Horizontal}.
To position both panels vertically, right-click anywhere in the script pane and, on the context menu, click \texttt{Layout -> Vertical}.

Hyperlinks

You can press and hold \texttt{Ctrl} and at the same time move the mouse over SQL text. If DBever 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:

\[
\text{select} \,* \text{ from } \text{Artist}; \\
\text{select} \,* \text{ from } \text{Customer}; \\
\text{select} \,2 + 2
\]

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 \texttt{Ctrl+9} or click the \texttt{Active datasource} box on DBeaver’s main toolbar:

![Select Data Source dialog box](image)

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

![Choose catalog/schema dialog box](image)

To change the active schema, press \texttt{Ctrl+0} or click the \texttt{Active Catalog/Schema} box in DBeaver’s main toolbar:

![Choose catalog/schema dialog box](image)

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 **Configure columns** button.

You can easily associate the SQL Editor with the connection that is currently in focus in the Database Navigator (the focus can be on any object of the connection - a table, a folder, etc.) - click the **Set connection from navigator** button in DBeaver’s main toolbar:

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

Templates allow you to insert frequently used SQL statements into an SQL script.

To see available templates, press Ctrl+Alt+SPACE or right-click the line in the script pane and click SQL Template on the context menu. A box appears with a list of available templates:

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

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

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

### Standard Eclipse templates:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>${cursor}</td>
<td>Specifies the cursor position when the template edit mode is left. This is useful when the cursor should jump to different place than to the end of the template upon leaving the template edit mode.</td>
</tr>
<tr>
<td>${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 potentially suitable objects in already loaded database metadata and in the database system tables.

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

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

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, 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.
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):

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

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

To execute the whole script, press \texttt{Alt+X} or click \textit{Execute \to Execute SQL Script} on the context menu or \textit{SQL Editor \to 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 \textit{Preferences} on the context menu).

To execute a script opening each query results in a separate tab, press \texttt{Ctrl+Alt+Shift+X} or click \textit{Execute \to Execute Statements In Separate Tabs} on the context menu or \textit{SQL Editor \to Execute Statements In Separate Tabs} on the main menu or in the main toolbar. This executes all queries in the script, but opens multiple result tabs. Each script query is executed in a separate thread (that is, all queries are executed simultaneously). NOTE: Be careful with this feature. If you execute a huge script with a large number of queries, it might cause unexpected problems.

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

![Bind parameter(s)](image)

Use Tab to switch. String values must be quoted. You can use expressions in values.

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

![Anonymous parameters setup](image)

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

### Execution Plan

If a database driver supports execution plan visualization, you can see the execution plan of the current query (under cursor) by pressing Ctrl+Shift+E or clicking Explain execution plan on the context menu or in the main toolbar. The execution plan command generates a tree of query execution as one of the result tabs and is convenient in estimating if the query/script is quick/optimal enough:
You can click the rows of the execution plan to see their details (statistics) in the panels below and to the right of the plan. To reevaluate the plan, click the **Reevaluate** button. To see the source script on which the plan is based, click the **View Source** button.

**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:

**Row Count**

If you want to know how many rows an SQL query will produce, you need to apply the Row Count feature – highlight and right-click the SQL text and then click **Execute -> Select row count** on the context menu:
Query Export

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

The Data transfer wizard opens. Go through its steps to complete the export of the query.
You can use special commands in SQL scripts. These commands are executed on DBeaver side, not on server-side.

### Client-side commands

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@set var = value</td>
<td>Sets default value for SQL parameter</td>
</tr>
<tr>
<td>@echo text</td>
<td>Prints string into server output viewer</td>
</tr>
<tr>
<td>@include file</td>
<td>Includes script file from file system</td>
</tr>
</tbody>
</table>

### Miscellaneous

- To select the current query row count, press Ctrl+Alt+Shift+C.
- To open the definition of the database object currently in focus (under cursor) in a viewer/editor, press F4.
Visual Query Builder

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

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

Opening Visual Query Builder

To open Visual Query Builder click the Open Query Builder button in the SQL Editor tool bar. The Visual Query Builder will appear on the right.

Creating Visual Query

Start creating a query from selecting query data source: drag-and-drop tables you want to work with from the Database Navigator pane into the Visual Query Builder area. All the connections existing between the tables will be shown automatically.
To create a new join between the tables press the left mouse button when the cursor is over the column of one table, holding the right mouse button pressed drag the cursor to the column of another table and release the right mouse button. The connection between the selected columns of the tables will be created visually and in the SQL script a new join will be added.

To remove a join between the tables click on it. The connection will be highlighted. Then, press Delete or use the Delete option in the context.
menu. The visual connection will be removed and the corresponding join will be automatically removed from the SQL script area.

To build a SELECT query you need to select columns in the tables you added. To select a column click the check-box next to its name - the column will be added to the Columns tab of the Query Settings Editor and SELECT query will be added to the SQL script area automatically.

**Adjusting Query Settings**

Visual Query Builder also allows setting query conditions and adjust representation of query results by means of Query Settings Editor.

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

Query Settings window contains five tabs described below.

**Columns**

![Columns tab](image)

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

To add a column, press Add button and a new instance will be added to the table. Click on the first cell in Column or Expression column and select a column from the list of available columns displayed in the dropdown list appeared.

To remove a column, click on the row containing its name and press the Remove button on the right.

To change the display order of columns in the result table use Move Up/Down buttons.

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

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

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

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

**Joins**

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

Joins cannot be added or removed by means of **Query Settings Editor**, however, the following join settings can be adjusted here:

• **Type** - defines the type of the join. Click the cell in the **Type** column - a drop down with available join types will be displayed. Select the required option from the list by clicking on it.
• **Alias** - defines a user friendly name of the join. To define this setting click on the cell in **Alias** column and input the name.

**Sorting**

In **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 be then adjusted to the one you need:

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

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

**Miscellaneous**

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

**Executing Visual Query**

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

Saving Scripts

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

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

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

To save a script to the file system, right-click the script, click File -> Save SQL script on the context menu and then select the folder in the file system.

You can also, click SQL Editor -> Save SQL script on the main menu:

Loading Scripts

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

Renaming Scripts

To rename a script, right-click anywhere in the script panel, click File -> Rename SQL Script on the context menu or press CTRL+F2:
Then enter the new name in the Rename SQL script dialog box and click **OK**:

![Rename SQL script dialog box](image)

**Reverting Changes**

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

**SQL Console**

In some cases you might want to execute a query and do not save it in a script. For example when you read table data using "Read data in SQL console" or open procedure/function source from DDL editor. SQL console doesn't has an associated .sql file. Its contents will be lost when you close it.
PostgreSQL Debugger

Prerequisites for Debugging

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

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

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

Installation

As the first step, you need to install the plugin_debugger extension in your PostgreSQL database. It is free and open source. You can download the source code [here](https://github.com/postgres-debgger/plugin_debugger). And you can find the installation README file [here](https://github.com/postgres-debgger/plugin_debugger/blob/master/README.md).

To install the debugger plugin:

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

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

After installing the plugin on the database server, you need to install the debug plugin from either a marketplace or P2 repository, see this [article](https://www.postgresql.org/download/) for more information.

You can find a detailed DBbeaverCE Debug plugin installation guide from Jkiss P2 repository [here](https://www.postgresql.org/download/downloads/).

How to start debug with local breakpoint

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

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

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

1. The client creates a session and then establishes a connection with PROXY (receiving session). After that, the client creates a global breakpoint for the selected function, and then becomes blocked via the `pldbg_wait_for_target` API call waiting for the server process.

2. PROXY receives the global control point from the client and adds global control point to the shared memory, if the point of this type has not been added before.

3. The initiator creates a connection and runs the selected procedure.

4. On each executed line, the target process checks if the global breakpoint with corresponding conditions exists, and becomes blocked if the breakpoint exists and conditions are met.

5. The target process establishes a connection with PROXY and reports that the breakpoint is reached.

6. Afterwards, the process becomes blocked and waits for PROXY (client) commands. The process will be implementing commands unless the procedure finishes or an exception is thrown.
If you have successfully installed the debugger plugin, and the plugin is up and running, you can start Debug from procedure source page by right-clicking the procedure source text - a context menu appears:

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

For a **Global** session, you need to specify the target process PID filter (enter -1 value for any process):
After running a Global session, you need to wait for any (or specified PID) process to call your procedure.

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

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

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

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

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

After you click OK, the debug toolset (perspective) opens:
```sql
CREATE OR REPLACE FUNCTION public.get_count(tabname text)
RETURNS bigint
LANGUAGE plpgsql
STABLE
AS $function$
DECLARE
  cmd text;
BEGIN
  cmd := 'SELECT COUNT(*) FROM ' || quote_ident(tabname);
  EXECUTE cmd INTO retval;
  RETURN retval;
END;$function$
```
Filter Database Objects

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

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

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

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

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

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

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

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

<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 (猗). The Filtering window opens.

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

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

4. For objects that you want to show, click Add next to the Include field and then, in the field itself, enter the name or combination of symbols to search. For objects that you want to hide, click Add next to the Exclude field and then, in the field itself, enter the name or combination of symbols to search. NOTE: You can use masks with % and _ to replace one or more symbols and _ to replace one symbol in the search combination.

5. To remove one filtering combination, click the combination in the field and then click Remove. To remove all combinations from either of the fields, click Clear next to the field.

6. Once you set all filtering criteria, you can save a filter to use for other objects. To save the filter, in the Saved filter area, in the Name field, enter the filter’s name and click Save.

7. You can also remove any of the saves filters. To remove a filter, in the Name drop-down list, click the filter name and then click Remove.

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

DBeaver provides:

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

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

![Search View](image)

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

Search View

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

![Search View](image)

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><img src="image" alt="Run the Current Search Again" /></td>
<td>Run the Current Search Again</td>
<td>Repeats the search the results of which are displayed in the Search view</td>
</tr>
</tbody>
</table>
| ![Cancel Current Search](image)       | 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.
| ![Show Previous Searches](image)      | Show Previous Searches                   | - Inactive state (grey) indicates that the search is complete. The button in this state is non-actionable. |
| ![Pin the Search View](image)         | 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 (red)
- 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 (blue) appears in the bottom-right corner of the view next to the search progress bar. Clicking the button opens the Background Tasks view:
Search History

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

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

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

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

To remove one or more previous search queries:

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

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

3. Click **Remove**.

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

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

![File Search dropdown menu]

The Search window opens displaying the File Search tab:

![File Search window]

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

After you click Search, the results appear in a Search view. The results represent a tree or list of files with the search combination highlighted:

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

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

<table>
<thead>
<tr>
<th>Menu item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Show as List</strong></td>
<td>Presents the results in the form of list</td>
</tr>
<tr>
<td><strong>Show as Tree</strong></td>
<td>Presents the results in the form of tree</td>
</tr>
<tr>
<td><strong>Filters</strong></td>
<td>Opens Search Filters dialog box</td>
</tr>
<tr>
<td><strong>Preferences</strong></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 full text search in the database contents, click the arrow next to the Search icon in the main toolbar and then click File Search on the dropdown menu:

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

Now you need to choose the database connection or database objects against which to run the search – expand the tree in the Databases field to the database connections level or further down and select the checkboxes next to the required connections or database objects. NOTE: The Search button is enabled only when you select the right level of checkboxes – database connections or lower nodes.

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

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

Double-clicking a row in the Search view opens the respective object in a dedicated Database Object editor.
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 – expand the tree in the Objects Source field to the database connections level and click the required database connection.

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

You can specify if the metadata should start with or contain or be similar to the search combination (Name match field). You can also set the maximum number of results to display (Max results field) and apply Case-sensitive search.

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

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

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

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

Managing Dashboards Panel

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

Opening Dashboard Panel

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

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

The following controls are available in 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>Allows managing dashboards' configuration.</td>
</tr>
<tr>
<td>📈</td>
<td>Add dashboard</td>
<td>Allows to add dashboards to the dashboard panel.</td>
</tr>
<tr>
<td>🗑️</td>
<td>Remove dashboard</td>
<td>Allows to remove dashboards from the dashboard panel.</td>
</tr>
<tr>
<td>⚙️</td>
<td>Reset dashboards</td>
<td>Allows to restart dashboard calculation.</td>
</tr>
</tbody>
</table>

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

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

You can also add a dashboard by right-click in any place of the dashboards panel and then select the Add dashboard menu option.

Removing Dashboards

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

Resetting Dashboards

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

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

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

To adjust dashboard representation settings right click on a dashboard and select the **Settings** menu option, then, in the opened dialog change the parameters you want.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Defines a name of a dashboard.</td>
</tr>
<tr>
<td><strong>Description</strong></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><strong>Update periods(ms)</strong></td>
<td>Defines how often dashboard's rendering should be updated. The default value is 1000 ms.</td>
</tr>
<tr>
<td><strong>Maximum items</strong></td>
<td>Defines maximum number of fetched items. The default value is 300.</td>
</tr>
<tr>
<td><strong>View</strong></td>
<td>Defines visual representation of the dashboard. The following options are available: Bar, Pie, Time series.</td>
</tr>
<tr>
<td><strong>Show legend</strong></td>
<td>If this check-box is selected, the legend will be displayed on the dashboard.</td>
</tr>
<tr>
<td><strong>Show grid</strong></td>
<td>If this check-box is selected, the grid will be displayed on the dashboard.</td>
</tr>
<tr>
<td><strong>Show domain axis</strong></td>
<td>If this check-box is selected, the domain axis will be displayed on the dashboard.</td>
</tr>
<tr>
<td><strong>Show range axis</strong></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 <strong>Database Drivers</strong>.</td>
</tr>
<tr>
<td>Data type</td>
<td>Defines the data type. The following options are available: timeseries (the default option) and statistics. Select timeseries type if you want to track the actual value returned by the server. Select 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're interested in the current value. Select delta if you want to track the difference between the current value and the previous one. This may be very useful when you work with statistics data, for example.</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 convinient 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: milisecond (the default option), second, minute, hour, day, week, month, year.</td>
</tr>
<tr>
<td>Fetch type</td>
<td>Defines whether the query should fetch data from rows or columns.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</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(ms)</td>
<td>Defines how often the dashboard’s rendering should be updated.</td>
</tr>
<tr>
<td>Maximum items</td>
<td>Defines 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 as templates to create new dashboards with any query and other settings. To learn about creating new dashboards, see 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 database connection on the dashboard panel’s activation by pressing Settings button on the dashboards panel’s toolbar and then selecting the Connect on activation check-box.

**Detaching Dashboards**

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

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

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

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

Copying Dashboards to Clipboard

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

Saving Dashboards

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

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

### Zooming

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

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

### Managing Dashboards

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

### Creating Dashboards

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

**To create a dashboard from scratch:**

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

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

Editing Dashboards

If you need to change dashboard's name, ID or any other configurational setting, you can edit a dashboard.
Note: Only custom dashboards can be edited, predefined dashboards are read-only, but you can use them as templates and create a custom dashboard whose parameters will be editable. To learn how to create dashboards from templates, see Creating Dashboards.

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

Deleting Dashboards

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

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

To delete a dashboard:
1. Press Settings button in the dashboards panel toolbar.
2. In the opened dialog box click Manage... dashboards.
3. In the Manage dashboards window select any of the custom dashboards from the list and click Delete.
Projects

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

Creating Project

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

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

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

3. Click Finish. The new project appears in the projects tree.

Deleting Project

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

1. Delete object dialog box is to confirm the deletion of the project itself. Click Yes if you are sure you want to delete it. Otherwise, click No.

2. Delete project dialog box is to confirm the deletion of the project’s contents: these are the data stored in the file system, database connections are not affected. Click Yes if you want the contents to be deleted as well. To keep the contents, click No.

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

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

To create a bookmark:

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

2. Press CTRL+d. The Bookmark Name dialog box appears.

3. In the Bookmark Name field, enter the bookmark name, then in the Bookmark folder field, click the folder, and then click OK:

The bookmark appears in the selected folder of the related project.

To open an object using its bookmark, double-click the bookmark or right-click it and click Open Bookmark on the context menu. You can rename and delete bookmarks using the context menu as well.
**Shortcuts**

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

### SQL Editor

<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+T</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+X</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+F9</td>
<td>Switch active connection (for SQL script)</td>
</tr>
<tr>
<td>CTRL+Space</td>
<td>SQL completion proposals popup</td>
</tr>
<tr>
<td>CTRL+ALT+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+Up ALT+Down</td>
<td>Toggle single/multi line comment</td>
</tr>
<tr>
<td>CTRL+6 CTRL+SHIFT+6 ALT+6</td>
<td>Toggle editor/results panels (maximize/minimize/switch)</td>
</tr>
<tr>
<td>CTRL+SHIFT+X CTRL+SHIFT+Y</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+-</td>
<td>Switch presentation (grid, plain text, json ,etc)</td>
</tr>
<tr>
<td>CTRL+l</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>F7</td>
<td>Current column filters menu</td>
</tr>
<tr>
<td>CTRL+F11</td>
<td>Current column filter dictionary panel</td>
</tr>
<tr>
<td>F5</td>
<td>Toggle right panels on/off</td>
</tr>
<tr>
<td>F5</td>
<td>Refresh results (re-run query) have the same meaning as a regular F5 key.</td>
</tr>
</tbody>
</table>

### Data editor

<table>
<thead>
<tr>
<th>Shortcut</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<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 ALT+Delete</td>
<td>Delete row</td>
</tr>
<tr>
<td>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+D</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+I</td>
<td>Open SQL editor for current connection (***)</td>
</tr>
<tr>
<td>CTRL+F3 CTRL+I</td>
<td>Open new SQL editor for current connection (***). Always creates new script.</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 “smart copy” operation (usually with parameters).</td>
</tr>
<tr>
<td>CTRL+SHIFT+V</td>
<td>Advanced paste. Same as “smart copy” but for “paste”.</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 current connection is the same as in this editor. If active window is database navigator then active connection is “owner” connection of currently selected element. In other cases there is no current connection and DBeaver will ask you to choose connection explicitly.
Database Connections

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

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

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

Database connections might have the following states:

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

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

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

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

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

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

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

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

NOTE: The list of database drivers 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.
2. In the Connection Settings screen, on the General tab, set all primary connection settings:

![Connection Settings](image)

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

**Connection Details (name, type, etc.)**

You can also set the connection name, type and initial settings (such as bootstrap queries, transaction state, global filters, etc.).
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 configuration settings of a database connection, in the Database Navigator or in the Projects view, right-click the connection and click Edit Connection on the context menu. The Connection configuration window opens:

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

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

**Driver settings**

- In development

JDBC Time Zones
Connect to Database

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

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

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

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

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

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

![Database Navigator](image)

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

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

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

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

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

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

Connection types define how DBeaver behaves regarding:

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

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

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

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

To manage connection types for a database connection, in the Database Navigator or Projects view, click the connection to set focus to it and then press F4 to open the connection properties window. Then, in the properties window, in the navigation pane on the left, click General to see the general settings. You can see Connection Type field among the settings:
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 (Edit 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 focus to it and then click the delete button under the list.
3. Click Yes in the confirmation dialog box to confirm the deletion. Otherwise, click No.
4. Click Apply to apply the changes and keep the window open or click Apply and Close to apply the changes and close the window.
Auto and Manual Commit Modes

DBeaver supports two modes for committing changes to the database:

- **Auto-commit** transfers all changes that you make immediately to the database.
- **Manual commit** requires your confirmation before committing a change to the database or rolling it back.

Though available in many cases, the two modes are actionable only in SQL Editor. See the next sections for details of using the modes.

To switch between the modes, use the mode selection button that appears in one of the two views:

**Auto-Commit Mode**

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

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

Clicking the statistics field opens the [Transaction Log](#).

**Manual Commit Mode**

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

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

In manual commit mode, when you execute SQL statements (Ctrl+Enter), the number of database modifying statements that pend commitment to the database appears in the statistics field next to the mode selection button: `▼` 2.

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

![Statistics Field](#)

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

If no modifying statements have been made, the statistics field shows None: `▼` None.

Clicking the statistics field opens the [Transaction Log](#).

**Transaction Isolation Level**

For both, Auto and Manual commit modes, you can select the transaction isolation level. To do so, click the arrow next to the mode icon and then click the required option in the dropdown list:
**Transaction Log**

Transaction Log shows all transactions (queries of 'transaction' type such as INSERT/DELETE/UPDATE and others) made during the current DBeaver session. To open the Transaction Log, click the **Transaction log** button in the toolbar or the statistics field to the left of it.

![Transaction Log window](image)

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.
Database drivers

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

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

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

Obtaining JDBC driver

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

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

Adding driver configuration in DBeaver

Open driver manager dialog

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

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

Main parameters

<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 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 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. This is better to define a valid template, this will greatly simplify connections creation. See “URL Templates” for the detailed description</td>
</tr>
<tr>
<td>Default Port</td>
<td>Default database port. You can get it from 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 network/connections management</td>
</tr>
<tr>
<td>No Authentication</td>
<td>This means that driver doesn't 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 on 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 driver. In most cases you need only jar files. Click “Add File” to add single jar file, “Add Folder” to add folder with Java classes/resources and “Add Artifact” to add Maven artifact (see below).

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

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

**Saving driver, adding connection**

After you finished configuring your driver just press Ok button. Now you can create connection.

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

![Connection settings](image)

**URL Templates**

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

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

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

Supported URL variables:

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

<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>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 single schema</td>
<td>Hide schema if there is only one schema on server</td>
</tr>
<tr>
<td>Omit type cache</td>
<td>Do not use data types provided by driver</td>
</tr>
<tr>
<td>Shutdown parameter</td>
<td>Database shutdown URL parameter</td>
</tr>
<tr>
<td>Create database parameter</td>
<td>Database create URL parameter</td>
</tr>
<tr>
<td>Driver supports multiple results</td>
<td>Driver supports multiple results for a single query</td>
</tr>
<tr>
<td>Driver supports result set limit</td>
<td>Driver supports multiple result set limit (max rows)</td>
</tr>
<tr>
<td>Driver supports structure cache</td>
<td>Driver supports structure cache reading. Enables schema columns, keys, etc</td>
</tr>
</tbody>
</table>
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>Use legacy SQL dialect for DDL</td>
<td>Use legacy SQL dialect for DDL</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 good and you think that it makes sense to have this driver configuration in standard DBeaver, please send your configuration to us. Just create a feature request issue on GitHub and copy/paste driver description to the ticket (in any suitable form).

Thank you :)
Data export/import

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

Exporting table data to CSV format

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

2. Choose export format. DBeaver support many different output formats including CSV, HTML, XLSX, etc:
3. Set data extraction options (how data will be read from tables). This may affect extraction performance:
4. Set export format option. They are specific to the data format you chose on step 2:
5. Set options for output files or clipboard:
6. Review what and to what format you will export:
7. Press finish. See extraction progress. Actual data extraction will be performed in background, you can keep working with your database during export process. Note: avoid changing data in tables you selected for export while export is running. In the end you will see status message:

![Data transfer](image)

**Data transfer completed (385ms)**

**Importing data from CSV format**

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

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

![Importing data](image)
2. Choose import format (CSV):
3. Select input CSV file for each table you want to import:
4. Set CSV-to-table mappings. You need to set some column in CSV file for each database table column. You can skip some column at all (in target table column value will be set to NULL). You can set some constant value for table column if there is no source column for it in CSV.
5. Set options for loading data in database. These options may affect performance:

Data load settings
Configuration of table data load

Data load
- [ ] Truncate target table before load

Performance
- [ ] Open new connection(s)
- [ ] Use transactions
Commit after insert of: 10000

General
- [ ] Open table editor on finish
- [ ] Show finish message

6. Review what file(s) and to what table(s) you will import:
7. Press finish. See extraction progress. Actual data loading will be performed in background, you can keep working with your database during export process. Note: avoid changing data in tables you selected for import while import is running. In the end you will see status:

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

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

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

**Step 1: Define the data source**

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

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

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

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

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

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

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

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

Or select from the list of tables already existing in the target container by pressing the **Browse** button ![Browse button](Browse ...)
To define mapping setting for a column in a target table, release the list of source table columns by pressing \( \text{Browse} \) next to the table's name in the \textbf{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 \( \text{Auto assign} \).

If you want tables of the target container to be named like those of source, press the \textbf{Auto assign} button and the \textbf{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 \textbf{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 source column will be transferred to. To change the name, click the cell and enter a new name.
- **Target type** - this column contains the list of data types that will be assigned to the columns in the target table.

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

- **Mapping** - this column contains the list of actions to be applied to the data on data transfer.

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

You can also view the SQL script that will be run on data transfer by pressing the DDL button.
The following keyboard shortcuts for easy navigation within the mapping table area of the 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 Single query option if your data load is not too big. Select By segments option if you need to migrate a solid amount of data. When this options is selected you can set the Segment size value, that is to define a number of rows to be transferred in each segment.</td>
</tr>
<tr>
<td>Open new connections</td>
<td>If selected, a new connection will be opened and data transfer will not interfere with other calls to the database whose data is being transferred.</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>Open new connections</td>
<td>Use this option to speed up data transfer. If selected, a new connection will be opened and data transfer will not interfere with other calls to the database where data is being transferred to.</td>
</tr>
<tr>
<td>Use transactions</td>
<td>This option allows to speed up data transfer and to define the number of rows for each transaction by setting Commit after insert of parameter.</td>
</tr>
<tr>
<td>Open table editor on finish</td>
<td>If selected, the table editor to be opened when data tanfer is over.</td>
</tr>
<tr>
<td>Show finish message</td>
<td>If selected, a notification message will be shown when transfer is over.</td>
</tr>
</tbody>
</table>

Step 5: Confirm
Check out data transfer settings and press Finish.
DBeaver extensions - Office, Debugger, SVG, SSH

You can install optional extensions (plugins) in DBeaver.

Install Process

Extension installation in CE version:

1. Main menu Help -> Install New Software

2. Paste extension P2 repository URL into Work with field and press Enter

3. Check items you wish to install


IMPORTANT: installation won’t work if DBeaver is installed in write-protected folder (like Program Files, /Applications, /usr/). To proceed you need to run DBeaver from some non write-protected folder or run it as Administrator/root.

Extension installation in EE

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

Extensions

Office integration

This extension supports data export in Office formats (XLS). It works for all platforms and all popular Offices - MS Office, Open Office, Libre Office, MacOS X. 
It is included in EE version.

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

Debugger support

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


SVG format support

This extension supports ERD export in SVG (vector) format.
It is included in EE version.

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

Advanced SSH tunnel libraries

Needed to support more key formats (e.g. ed25519).
P2 repository URL: https://dbeaver.io/update/sshj/latest/
Git integration

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

![Install dialog with available software](image)

Select installation options and proceed with installation as per the user guide.
Installing extensions - Themes, version control, etc

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

Popular extensions for DBEaver

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

DBEaver-specific extensions

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

Install Process

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

Extension installation in CE version:

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

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

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

7. Click Next->Finish. Installation will take some time. Restart DBeaver.
Apache Hive/Spark/Impala

Apache Hive

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

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

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

Connection setup

![Connection setup image]

DBeaver user guide
Generic JDBC Connection Settings

Hadoop / Apache Hive connection settings

General

Driver properties

JDBC URL: jdbc:hive2://hive.theserver.com:10000/foodmart
Host: hive.theserver.com
Port: 10000
Database/Schema: foodmart
User name: 
Password: 

Save password locally

Driver Name: Hadoop / Apache Hive

Edit Driver Settings

Network settings (SSH, SSL, Proxy, ...)
Connection details (name, type, ...)

Finish
Cancel
Test Connection...

Success

Server: Apache Hive 1.2.1000.2.6.5.0-292
Driver: org.apache.hive.jdbc.HiveDriver-1.1.1

Connected (2349 ms)

OK

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

Limitations

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

Yandex Clickhouse

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

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

Connection setup

Connection initiation is very easy.

1. Select Clickhouse driver:

   ![Select new connection type](image)

2. The only required connection parameter is host name.
3. You can configure SSH tunnel to access your server. In that case set database host name to `localhost` while real Clickhouse server host will be specified as SSH server.

4. Test connection:
Schema/data browser

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

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

**Limitations**

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

Command line parameters

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

**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>-f c:\some-path\some-file.sql</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>-reuseWorkspace</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></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 version upgrade.</td>
<td></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 (1.8): https://docs.oracle.com/javase/8/docs/technotes/tools/windows/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>Name</td>
<td>Value</td>
<td>Example</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</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>savePassword</td>
<td>Do not ask use password on connect</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-16a88e815bd-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 existing connection configuration will be used. name or id parameter must be specified.</td>
</tr>
</tbody>
</table>
Reset UI settings

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

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

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

To reset settings:

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

Sometimes (especially after multiple DBeaver versions upgrade) workspace become messy. Some keyboard shortcuts may stop working, toolbars layout may be broken, etc, etc.

To reset all UI settings (this includes menus, shortcuts, view and toolbar layouts):

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

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

That's it.
Posting issues

A few tips.

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

- Be clear about what your problem is: what was the expected outcome, what happened instead? Detail how someone else can recreate the problem.

- If you posting a bug report check “Error Log” view. If there are any errors related to your bug then post a complete stacktrace. Sometimes there are no errors in Error Log - then try to find them in log files.

- If your issue is related to a database data or metadata management - check the Query Manager view. It contains information about all queries DBeaver executes (explicitly or implicitly). To see more detailed information you can configure Query Manager in Preferences.

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

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

Error Log view

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

- Default workspace location is in the user home subfolder `DBeaverData`.
- On Windows you usually can find it in `%APPDATA%\DBeaverData\workspace6\.metadata`.
- On Linux just type `cd $XDG_DATA_HOME/DBeaverData/workspace6/.metadata`.
- On MacOS:
  - Type `open ~/Library/DBeaverData/workspace6/.metadata` in terminal or `~/Library/DBeaverData/workspace6/.metadata` in Spotlight Search.
  - To view hidden folders press `Cmd+Shift+` in the folder view.

Two standard log files:

- `<workspace-path>/.metadata/.log` - all warnings and errors which happens during normal work
- `<workspace-path>/.metadata/dbeaver-debug.log` - the same as `.log` plus debug information

In special cases log files can be written in other directories. Special case is an emergency situation when DBeaver can't start and there is no workspace. Two typical places to find emergency logs:

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

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

Old (before DBeaver 6.1.3) workspace location

- Windows: `C:\Users\YourName\dbeaver4`.
- Linux: `~/dbeaver`.
Thread dump

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

Mac and Linux

Run the following on your terminal:

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

Windows

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

```bash
jstack <PID> > thread-dump.txt
```

in Command Prompt.

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

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

Provide predefined connections

DBeaver keeps information about project connections in file \texttt{.dbeaver-data-sources.xml}. You can find it in project folder (by default all projects reside in workspace, default workspace location is \texttt{%HOMEPATH%/\texttt{.dbeaver4}} (or \texttt{${HOME}/.dbeaver4})). Default project folder is \texttt{%HOMEPATH%/\texttt{.dbeaver4/General}}.

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

Importing connections from CSV/XML

You can import connection from CSV or XML files.

CSV file must have a header row (first line of file) with column names (see list of supported columns below). XML file should contain top-level element and a set of nested elements. Connections config must be specified in attributes of nested elements. Attribute names are the same as CSV column names.

<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 specify just URL or set host/port/etc setting. User name/password are options.

Sample CSV

\begin{verbatim}
name,host,port,server,database,url,provider,username,password,type
Postgre Import XML 1,localhost,5432,,postgres,jdbc:postgresql://localhost:5432/postgres,postgres,postgres,postgres,dev
Postgre Import XML 2,localhost,5432,,postgres2,jdbc:postgresql://localhost:5432/postgres2,postgres2,postgres2,prod
\end{verbatim}

Sample XML

\begin{verbatim}
<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="postgres2" url="jdbc:postgresql://localhost:5432/postgres2" user="postgres2" password="postgres2" type="prod"/>
</connections>
\end{verbatim}

Secure connections from editing

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

- Generate MD5 hash of your password. You can do it from command line using Linux utility \texttt{md5sum} (\texttt{md5sum \textless \textgreater your password}) or you can do it online - just google "MD5 hash online".
- Add attribute \texttt{lockPassword} in connection descriptor (in \texttt{.dbeaver-data-sources.xml} file in \texttt{<dataSource>} tag. So it will look like this:
Now if user will try to change connection settings he/she will be asked for password
Managing drivers

Configure drivers with pre-installed jars

You can customize drivers configuration in `${HOME}/.dbeaver4/.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.jar" custom="true"/>
```

Also in drivers.xml you can use 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 (<code>${HOME}/.dbeaver-drivers</code> by default)</td>
</tr>
<tr>
<td>dbeaver_home</td>
<td>DBeaver installation folder</td>
</tr>
<tr>
<td>home</td>
<td>User home folder</td>
</tr>
<tr>
<td>workspace</td>
<td>DBeaver workspace path (<code>${HOME}/.dbeaver4</code> by default)</td>
</tr>
</tbody>
</table>

For instance:

```xml
<library type="jar" path="${dbeaver_home}/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="42.2.3"/>
      <file id="org.postgresql:postgresql" version="42.2.3" path="${drivers_home}/maven/maven-central/org.postgresql/postgresql-42.2.3.jar"/>
    </driver>
    <library type="jar" path="maven:/net.postgis:postgis-jdbc:RELEASE" custom="false" version="2.2.1"/>
    <file id="net.postgis:postgis-jdbc" version="2.2.1" path="${drivers_home}/maven/maven-central/net.postgis/postgis-jdbc-2.2.1.jar"/>
    <library type="jar" path="maven:/net.postgis:postgis-jdbc-jtsparser:RELEASE" custom="false" version="2.2.1"/>
    <file id="net.postgis:postgis-jdbc-jtsparser" version="2.2.1" path="${drivers_home}/maven/maven-central/net.postgis/postgis-jdbc-jtsparser-2.2.1.jar"/>
  </provider>
</drivers>
```
Windows Silent Install

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

Parameters

Command line parameters supported by DBeaver installer:

<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 /allusers parameter current user must have administrator permissions.

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's 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>

**Code**

**Meaning**
Build from sources

Build DBeaver from sources

Prerequisites:

1. Java (JDK) 8+.
2. Apache Maven 3+
3. Internet access

Build

```
git clone https://github.com/dbeaver/dbeaver.git dbeaver
cd dbeaver
mvn package
```

Binaries are in product/standalone/target/products
DBeaver is an Eclipse RCP application.
It consists of a set of Eclipse plugins, features and products.

Opening workspace in Eclipse

You will need to setup "Eclipse for RCP and RAP developers". Currently default Eclipse version is Eclipse 2018-09. Build may fail on newer or older versions. Then you will need to install a few additional Eclipse plugins which DBeaver depends on (and which are not included in the standard Eclipse distribution):

1. Open main menu Help -> Install New Software
2. Copy http://dbeaver.io/eclipse-repo/ into "Work with" field and press enter
3. Check all plugins and finish the installation process:

![Install](image)

4. Do the same for http://eclipse-color-theme.github.com/update
5. Restart Eclipse
7. In Eclipse run Import wizard

8. Choose "General->Existing Maven Projects".
9. Select DBeaver checkout directory. Check all plugins and features Eclipse will find and finish the wizard.
10. Eclipse will perform project configuration process. This may take a few minutes.

11. If Eclipse will ask about additional plugin install – just agree:
12. Restart Eclipse. Eclipse will build project automatically or you can press Ctrl+B.

13. Switch to Hierarchical projects presentations.
14. If some modules build will fail (they will be marked with red cross) - update Maven project
Running and Debugging in Eclipse

1. Import DBeaver launch configuration using Eclipse Import wizard:
2. Click on Debug icon in the main toolbar->Debug configurations.

3. Choose DBeaver.product and click Debug button
If the product won’t start with the error "No application id has been found" try this workaround: https://stackoverflow.com/questions/13575695/eclipse-rcp-no-application-id-has-been-found.

What you need to do is go to your Run Configurations (Run -> Run Configurations). Select the Plug-ins tab and then hit "Add Required Plug-ins". Apply, then Run, and it should work now.

That’s it.

**Sources structure:**

- **docs** - some documentations (mostly outdated)
- **features** - feature descriptors. Doesn’t contain any source code. Used to structurize product plugins/dependencies.
- **bundles** - very base plugins (like utils)
- **plugins** - main sources location
  - **org.jkiss.dbeaver.model** - model API and base classes. Doesn’t contain any UI dependencies, just pure data model.
  - **org.jkiss.dbeaver.core** - main DBeaver module. Most of base UI classes are here.
  - **org.jkiss.dbeaver.core.application** - relatively small module which configures standalone DBeaver application.
  - **org.jkiss.dbeaver.core.eclipse** - main Eclipse plugin. Adds some extra menus/views to standard Eclipse IDE.
  - **org.jkiss.dbeaver.ext.*** - DBeaver extensions
- **product** - final products (standalone and Eclipse plugin) configuration.

**Notes:**

Develop in IDEA

Develop and debug DBeaver in IntelliJ IDEA

Well, that’s a tricky part.

Intro

Why

DBeaver is based on Eclipse Platform and it seems logical to develop it in Eclipse IDE. However, many people find that even latest Eclipse Java IDE isn’t that good and convenient as IntelliJ IDEA.

I personally also prefer to develop in IDEA, mostly because of historical reasons but also because of many really helpful features in it.

Anyhow, here we will try to open DBeaver project in IDEA, write code, compile, run and eventually debug (as Java debugger really rocks in IDEA).

How

It is quite easy to open the project, see module structure and even compile.

But running/debugging is the most tricky part because it requires Eclipse runtime workspace and only Eclipse IDE can create one (IDEA doesn’t support this and nobody can blame JetBrains for that).

So, we’ll still need Eclipse to configure workspace for debugger. But once you will do this you will no longer need to run Eclipse (until plugins structure or versions will change).

Let’s start!

Instructions

Opening DBeaver in IDEA

1. Create some folder where you will clone DBeaver repositories. Note: all repositories must be located in the same folder. For example
   
   `dbeaver-ce`

2. `cd dbeaver-ce`

3. `git clone https://github.com/dbeaver/dbeaver.git`

4. `git clone https://github.com/dbeaver/dbeaver-idea-project.git`

5. Optional: `git clone https://github.com/dbeaver/dbeaver.wiki.git` (skip it if you don't need WIKI)

6. Start IDEA->Open. select folder `dbeaver-ce/dbeaver-idea-project`

7. Now you can see DBeaver modules structure:
You can edit code but you can't compile because there are no Eclipse dependencies configured.

Prepare Eclipse IDE and workspace

1. Perform all steps described in Develop in Eclipse article. Use just cloned dbeaver-ce/dbeaver as sources root.
2. Eventually you should be able to compile and run DBeaver from Eclipse
3. Remember where you Eclipse IDE is located (let's say eclipse-ide-path)

Configuring dependencies and compile

1. Now we need to add Eclipse dependencies in IDEA project configuration.
2. Open File->Project structure ...
3. Make sure you have proper project Java SDK configured (Java 8+)
4. Go to Global libraries
5. Create new library ECLIPSE (name and case are important)
6. Click "Add" button. Select folder `eclipse-ide-path/plugins`.

7. Recent IDEA should add all Eclipse jars as dependencies. But in some IDEA versions you will also need to add some other folders manually:

8. Eventually you see something like this:

9. Now you should be able to compile the project. Just hit **CTRL+F9** and wait.

### Running and debugging

DBeaver is Eclipse RCP and must be started using Eclipse Equinox Launcher bundle. This is a special jar file which resides in Eclipse plugins folder and we need to add it to dependencies directly in order to be able to debug.

1. Add launcher config as jar dependency for module `dbeaver`. Locate the jar `org.eclipse.equinox.launcher_*.jar` in the `eclipse-ide-path/plugins/` folder. Like this:

2. Open debug configuration "Debug DBeaver Community" and change path in parameters `-configuration` and `-dev`. Replace path before `.metadata` with path to your Eclipse workspace. Eventually it should look like this this:
3. Click ok and run debugger. DBeaver must start. Now you can enjoy debugging in IDEA ;)

Problems

Most likely something will go wrong as these instructions are quite long. Don't be afraid, rechecks configuration, try again. IF nothing helps - don't hesitate to ask in tickets or by email.
Resources localization

Localization (i18n + l10n)

DBeaver uses standard properties-based i18n model. All translatable resources reside in *.properties file. Each plugin (bundle) has its own set of resources. Almost all plugins have at least `bundle.properties` resource. Bigger plugins have additional resources in src folder. See full list of property files below.

bundle.properties contains original string in English language. All translated resources are placed in bundle_XX.properties files where XX is two-letter language code.

Environment setup

- Clone DBeaver repository
  - Install GitHub Desktop
    - In the top menu select File -> Clone Repository
    - In the appeared popup window select the URL tab and paste DBeaver URL: https://github.com/dbeaver/dbeaver.git
  - Press Clone

Localizing tools

Eclipse IDE

- Install Eclipse (any version, any package)
- Install ResourceBundle Editor plugin.
- Main menu -> File -> Import... -> General -> Existing Projects into Workspace -> Browse
- Choose your DBeaver clone directory and import all projects

- Open some properties file (e.g. bundle.properties - see below) in ResourceBundle editor:

  ![ResourceBundle Editor](image)

  - Edit properties:
IntelliJ IDEA Community

DBeaaver user guide
Push your changes

- Open GitHub Desktop
- At the top select Current branch and press New Branch button
- Create a new branch with a name related to l10n
- Commit to the new branch
- Publish branch/push (right button at the top)
- Fetch origin (right button at the top)
- In the main menu select Branch -> Create Pull Request
- On the opened GitHub page add some meaningful text for your pull request name and press Create pull request button
- Return to GitHub Desktop and switch to devel branch

or

Create a Pull Request with your changes (in branch develop) https://help.github.com/articles/creating-a-pull-request-from-a-fork/

Properties

<table>
<thead>
<tr>
<th>Module</th>
<th>Purpose</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core</td>
<td>Commands, properties</td>
<td>plugins/org.jkiss.dbeaver.core/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>Core</td>
<td>Messages, UI strings</td>
<td>plugins/org.jkiss.dbeaver.core/src/org/jkiss/dbeaver/core/CoreResources.properties</td>
</tr>
<tr>
<td>API</td>
<td>Properties</td>
<td>plugins/org.jkiss.dbeaver.model/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>API</td>
<td>Messages</td>
<td>plugins/org.jkiss.dbeaver.model/src/org/jkiss/dbeaver/model/messages/ModelResources.properties</td>
</tr>
<tr>
<td>UI</td>
<td>Properties</td>
<td>plugins/org.jkiss.dbeaver.ui/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>UI</td>
<td>Messages</td>
<td>plugins/org.jkiss.dbeaver.ui/src/org/jkiss/dbeaver/bundle/UIMessages.properties</td>
</tr>
<tr>
<td>Application (Standalone)</td>
<td>Commands, properties</td>
<td>plugins/org.jkiss.dbeaver.core.application/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>Application (Eclipse)</td>
<td>Commands, properties</td>
<td>plugins/org.jkiss.dbeaver.core.eclipse/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>Data Transfer</td>
<td>Commands, properties</td>
<td>plugins/org.jkiss.dbeaver.ext.data.transfer/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>Data Transfer</td>
<td>Messages</td>
<td>plugins/org.jkiss.dbeaver.ext.data.transfer/src/org/jkiss/dbeaver/tools/transfer/internal/DTMessages.properties</td>
</tr>
<tr>
<td>ERD</td>
<td>Commands, properties</td>
<td>plugins/org.jkiss.dbeaver.ext.erd/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>ERD</td>
<td>Messages</td>
<td>plugins/org.jkiss.dbeaver.ext.erd/src/org/jkiss/dbeaver/ext/erd/ERDResources.properties</td>
</tr>
<tr>
<td>SSH</td>
<td>Commands, properties</td>
<td>plugins/org.jkiss.dbeaver.net.ssh/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>SSH</td>
<td>Messages</td>
<td>plugins/org.jkiss.dbeaver.net.ssh.ui/src/org/jkiss/dbeaver/ui/net/ssh/SSHUIMessages.properties</td>
</tr>
<tr>
<td>Generic driver</td>
<td>Properties</td>
<td>plugins/org.jkiss.dbeaver.ext.generic/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>Generic driver</td>
<td>Messages</td>
<td>plugins/org.jkiss.dbeaver.ext.generic/src/org/jkiss/dbeaver/ext/generic/GenericResources.properties</td>
</tr>
<tr>
<td>MySQL</td>
<td>Properties</td>
<td>plugins/org.jkiss.dbeaver.ext.mysql/OSGI-INF/l10n/bundle.properties</td>
</tr>
<tr>
<td>MySQL</td>
<td>Messages</td>
<td>plugins/org.jkiss.dbeaver.ext.mysql/src/org/jkiss/dbeaver/ext/mysql/MySQLResources.properties</td>
</tr>
<tr>
<td>..</td>
<td>..</td>
<td>The same for Oracle (ext.oracle), DB2 (ext.db2), Exasol (ext.exasol), PostgreSQL (ext.postgresql) and SQL Server (ext.mssql)</td>
</tr>
</tbody>
</table>

Testing you changes

Once you have changed resource you might want to test your changes. You can:
Run DBever directly from Eclipse workspace
Build DBever from source and run executable

Changing interface language

You can change language of UI in DBever:

- Preferences->Database->Language
- In dbever.ini before line \vmargs add following lines

```
-nl
XX
```

where XX is two-letter language code (zh, de, ru, etc).
Automatic Tests

DBeaver Auto-Tests

SWT Bot install instructions

SWT Bot test plugin development instructions

SWT Bot tests run
Enterprise Edition

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

- All popular JDBC drivers are included in EE distribution so you won't need to download/configure them separately.
- Support of NoSQL databases:
  - Apache Cassandra
  - MongoDB
  - Redis
  - InfluxDB
  - WMI (Windows Management Instrumentation)
- Additional EE plugins:
  - Office formats support (XLS) for data export
  - Visual Query Builder
  - Analytical charts rendering
  - Persistent query database. Search in query history
  - JSON format viewer/editor (in MongoDB and relational databases)
  - Eclipse Marketplace (provides easily installation of any additional Eclipse plugins)
  - Advanced Mock data generators
  - Version control support
- 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!