DBeaver user guide

Table of contents

General User Guide

Database Management

Enterprise Databases support

Customizing DBeaver

DBeaver Enterprise

Troubleshooting

Admin Guide

License management

Tutorials

Installation

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

Windows / MacOS Installer

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

NOTE:

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

ZIP Archive

When installing DBeaver manually, without using an installer:

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

Debian Package

To install DBeaver using a Debian package:

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

RPM Package

To install DBeaver using RPM package:

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

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

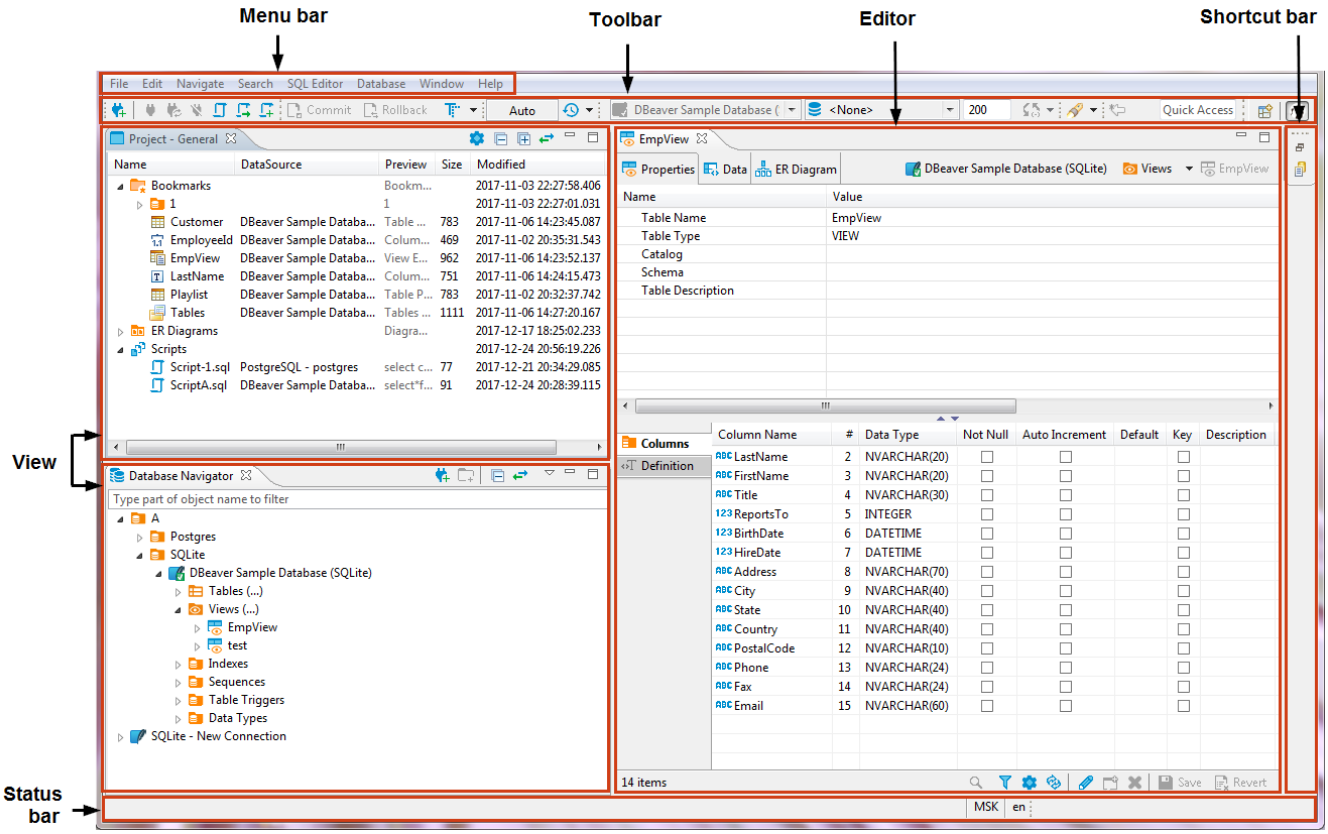
Automatic updates

This feature is available only in Windows and macOS.

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

Application Window Overview

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



Menu Bar

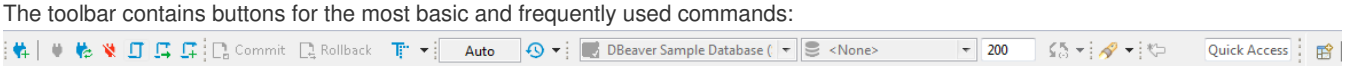
By default, the menu bar contains the following menus:

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

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

Toolbar

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



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

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

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

Shortcut Bar

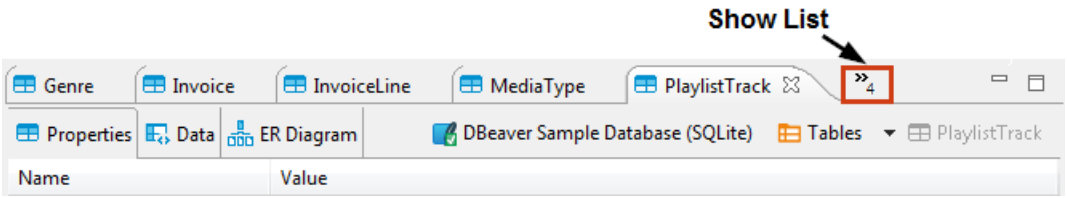
There are two shortcut bars - one on the left and one on the right side of the workspace zone. Shortcut bars host shortcuts of views and editors and appear if at least one view or editor is minimized, otherwise they are hidden.

Workspace: Views and Editors

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

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

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



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

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

Changing Workspace Layout

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

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

You can also swap locations of two views or editors.

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

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

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

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

Maximizing, Minimizing and Restoring View and Editors

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

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

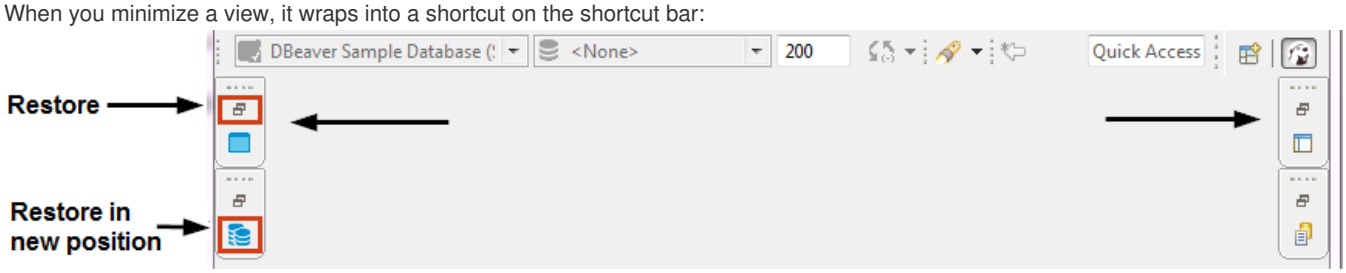
To maximize a view or editor to the size of the whole workspace, do one of the following:

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

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

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

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



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

To minimize a view, do one of the following:

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

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

Views

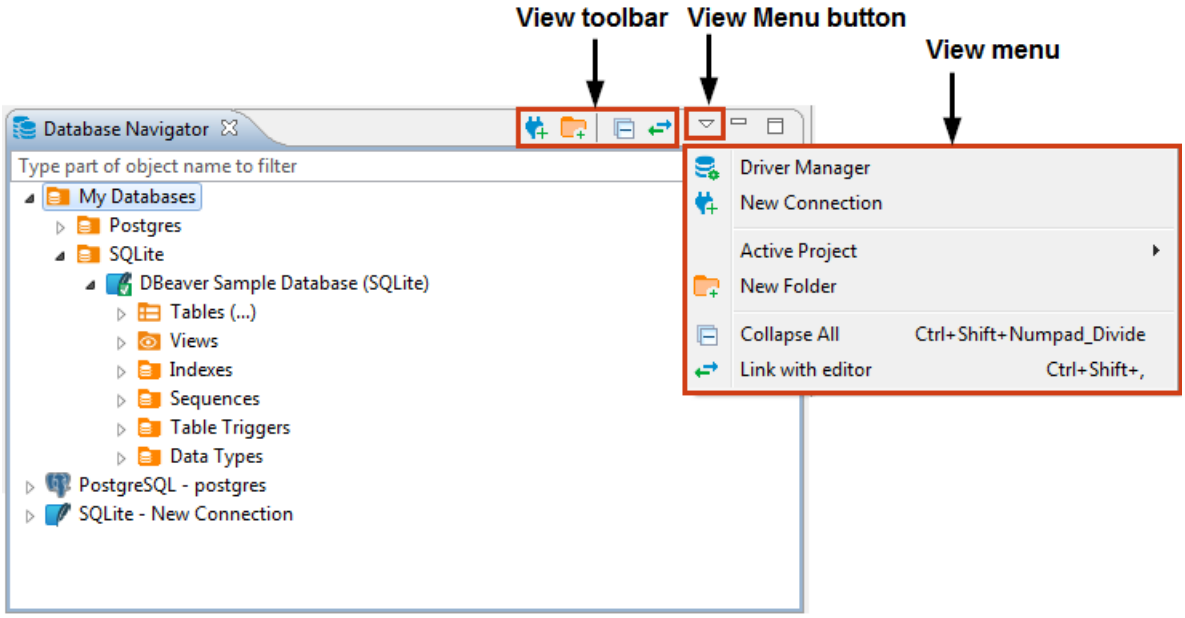
Views are windows within the workspace that provide presentations and ways to navigate the information. The main views in DBeaver are: Database Navigator, Projects and Project Explorer.

To open a view:

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

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

Views provide their own toolbar and menu:



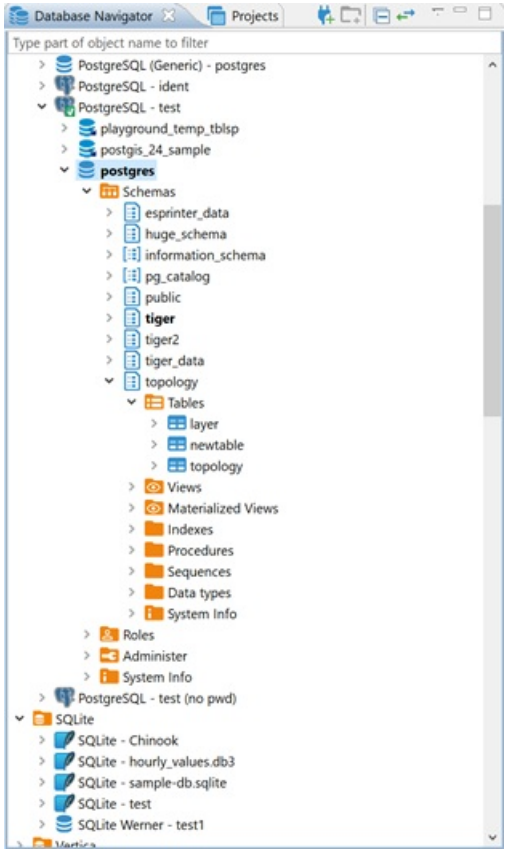
To open the view menu, click the View Menu button in the upper-right corner of the view's title bar, next to the Minimize button.

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:

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

The toolbar is located in the title bar of the window. Its buttons duplicate the menu items, except for the **Active Project**.

To open the context menu for an object, right-click the object in the tree. The following table summarizes the context menu items for all

types of objects that may appear in the tree. Note that the presence or absence of the context menu items for an object depends on the database and object types.

Menu item	Description
Open folder	Opens a folder in a separate view
Create new connections / **Create New Connection**	Opens the Create new connection wizard
New Folder	Opens a dialog box for creating a new folder
Copy	Copies an object to the clipboard
Paste	Inserts the copied object into a selected folder - most convenient for copy-pasting connections
Delete	Deletes an object **WARNING!** The Delete menu item removes the object not only from the tree but from the database itself or the file system, and this action is irreversible.
Rename	Opens the Rename [object] dialog box
Properties	Opens the Properties for [object] window which allows viewing and modifying the object`s properties
Refresh	Depending on the object, refreshes the object itself, or its parent, or its subnodes – mostly used for refreshing tables and schemes
Connect	Attempts to connect to the database
Invalidate/Reconnect	Checks the status of connection, if it is broken, attempts to reconnect
Disconnect	Disconnects from the database
SQL Editor	Opens a new SQL editor for the connection
Recent SQL Editor	Opens the most recently opened SQL editor
Edit Connection	Opens the Connection Configuration window that allows configuring connection settings
View [objects]	Opens the object in a separate viewer
Edit [object]	Opens the object in a separate editor
Create new [object]	Opens an editor in which you can specify properties and save the new object
Filter	Opens a submenu of one or more filtering options (depending on the object): - Hide [object] - Show only [object] - Configure [objects] filter - Toggle filter - Clear filter See Filters for information.
Copy Advanced Info	Copies the full name of an object
Read Data in SQL Console	Opens an SQL console displaying the object`s data
Compare	- Appears only if you select several objects of the same level - Opens the Compare objects wizard which guides you through the steps to generate a comparison report for the selected objects
Generate SQL	Opens a submenu on which you can select the type of SQL query to generate: - SELECT - INSERT - UPDATE - DELETE - MERGE - DDL Clicking one of the items (for example INSERT) generates a relevant query in a separate window.
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

Menu item	Description
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.

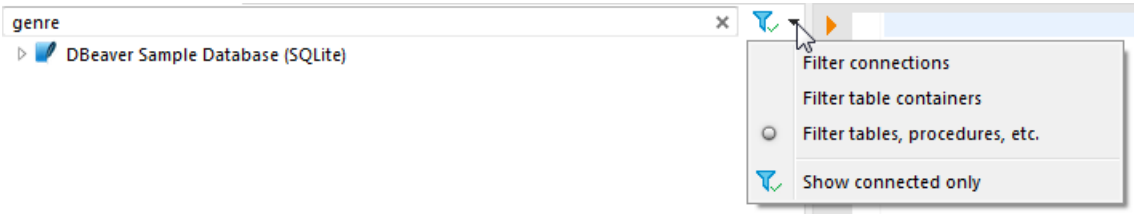
Filter Database Objects

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

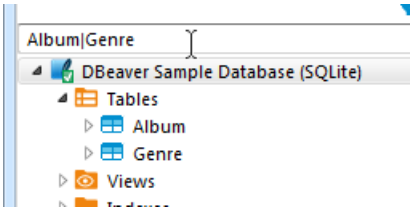
There are several ways in which you can filter objects. One of the ways is to filter objects by the names of tables and views using the filter field above the tree of objects:

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

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



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



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

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

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

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

Filter submenu item	Description
Hide N objects	Hides the selected objects while displaying the rest
Show only selected objects	Shows the selected objects while hiding the rest

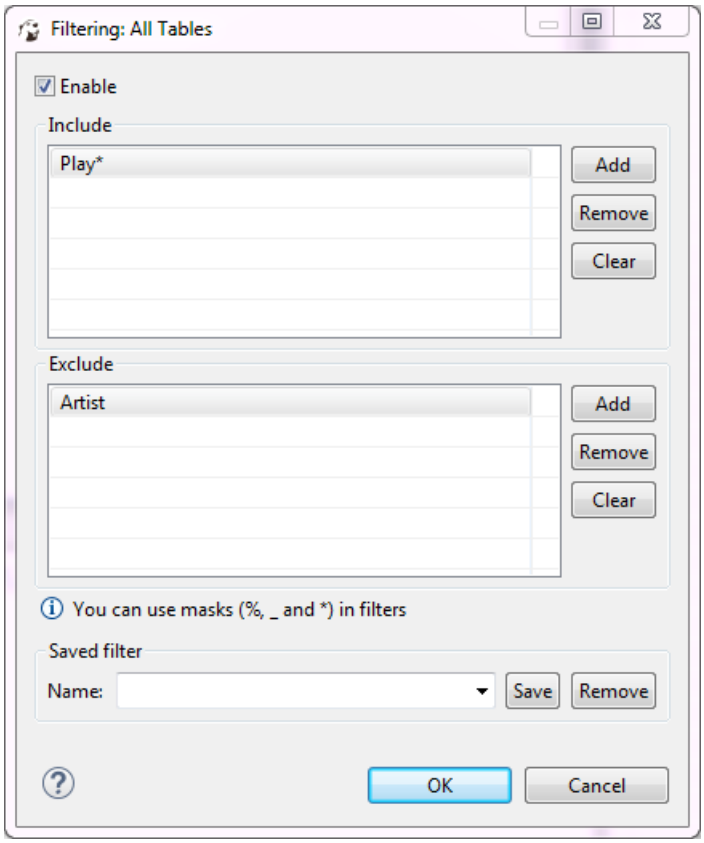
To reset such filters, right-click the parent (folder) node displaying the dots sign `(...)`, and then click **Filter -> Clear filter**.

Configure Filters

You can configure custom filters to filter database objects in the Database Navigator and Database Object Editor.

To configure a custom filter:

1. In the Database Navigator, right-click the object and on the context menu click **Filter -> Configure [objects] filter**. In the Database Object editor, in the toolbar of the **Properties** tab, click the Filter settings button (). The Filtering window opens.

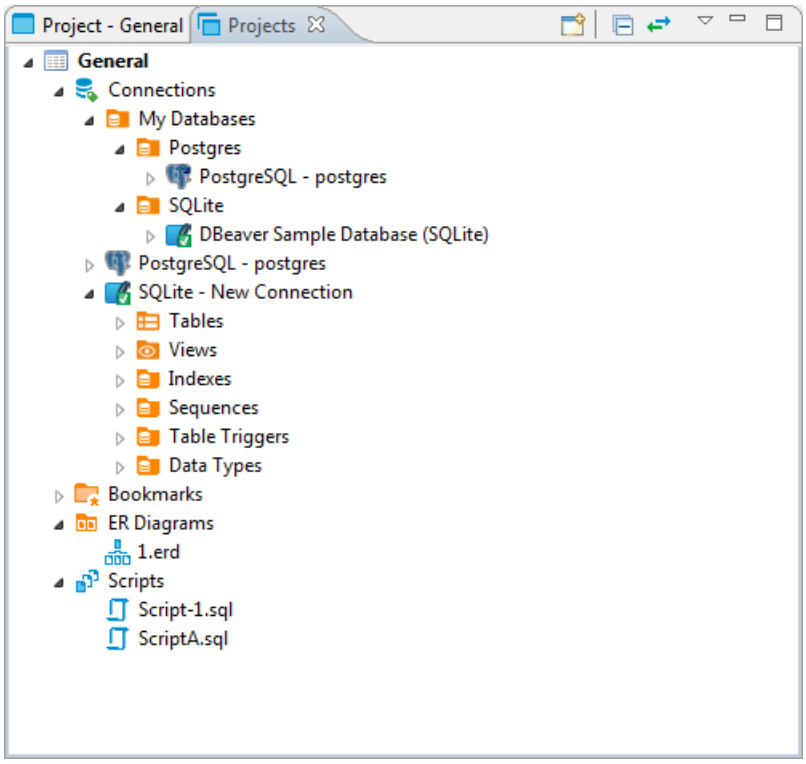


2. Select the **Enable** checkbox to activate the fields of the window.
3. If you want the filter to apply to all objects of a certain type, for example to all schemes, click **Show global filter**. Otherwise, the filter will apply only to the current object. NOTE: Once you apply the global filter, you cannot revert back to the local filer in the same window. To create a local filter, reopen the Filtering window, see Step 1.
4. For objects that you want to show, click **Add** next to the **Include** field and then, in the field itself, enter the name or combination of symbols to search. For objects that you want to hide, click **Add** next to the **Exclude** field and then, in the field itself, enter the name or combination of symbols to search. NOTE: You can use masks with `%` and `*` to replace one or more symbols and `_` to replace one symbol in the search combination.
5. To remove one filtering combination, click the combination in the field and then click **Remove**. To remove all combinations from either of the fields, click **Clear** next to the field.
6. Once you set all filtering criteria, you can save a filter to use for other objects. To save the filter, in the Saved filter area, in the **Name** field, enter the filter`s name and click **Save**.
7. You can also remove any of the saved filters. To remove a filter, in the **Name** drop-down list, click the filter name and then click **Remove**.
8. Click **OK** to apply the filtering criteria. Otherwise, click **Cancel**.

Projects View

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

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



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

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

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

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

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

Icon	Item	Description
	Create Project	Opens the Create Project wizard
	Refresh Projects	Refreshes the projects tree to display changes caused by creating modifying or deleting projects
	Collapse All	Collapses the tree to the root level
	Link with editor	- Enabled when at least one editor is open, otherwise disabled - Highlights the object in the tree that has its editor open

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

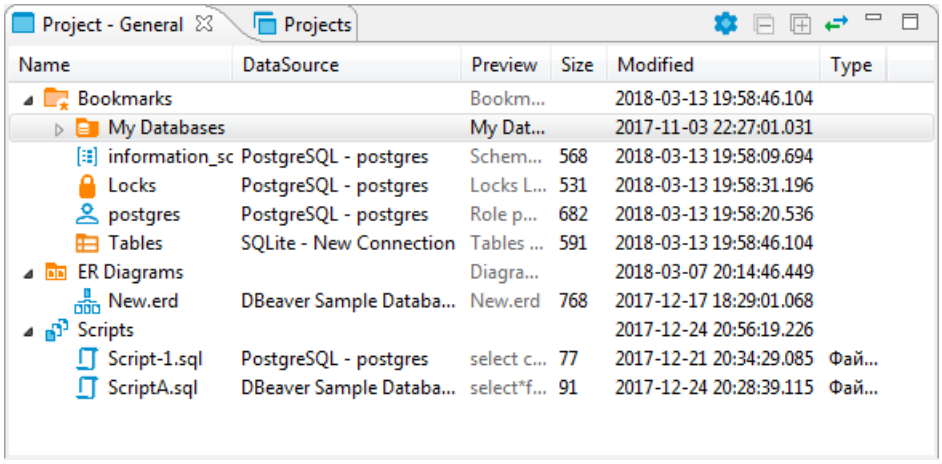
To open the context menu for an object in the tree, right-click the object. For information about context menu items of all objects under the **Connections** node of the tree, please see Database Navigator. The context menus of other nodes in the tree contain some basic items for copy-pasting, renaming,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

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

For information about managing projects, please see Projects article.

Project Explorer

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



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

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

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

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

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

To sort the metadata in the table by a certain column, click the column header.

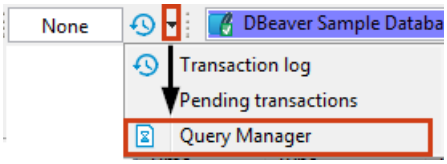
Query Manager

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

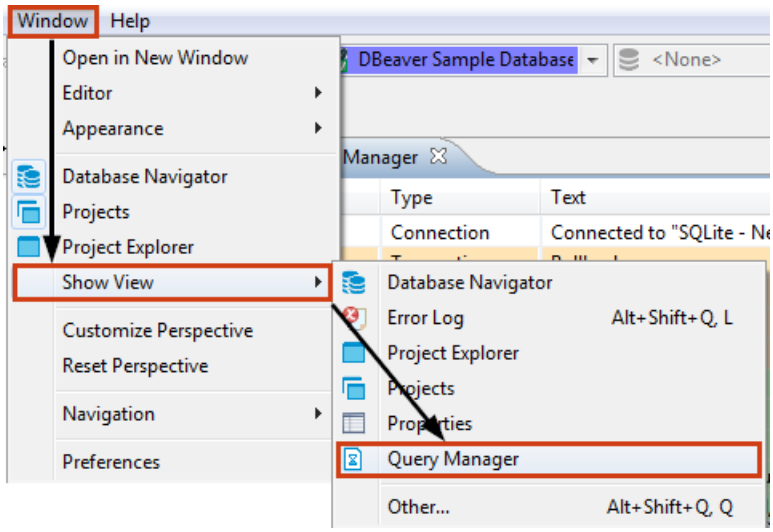
NOTE: The DBeaver EE version persists all executed queries in the internal database so its execution history is available after the program restarts.

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

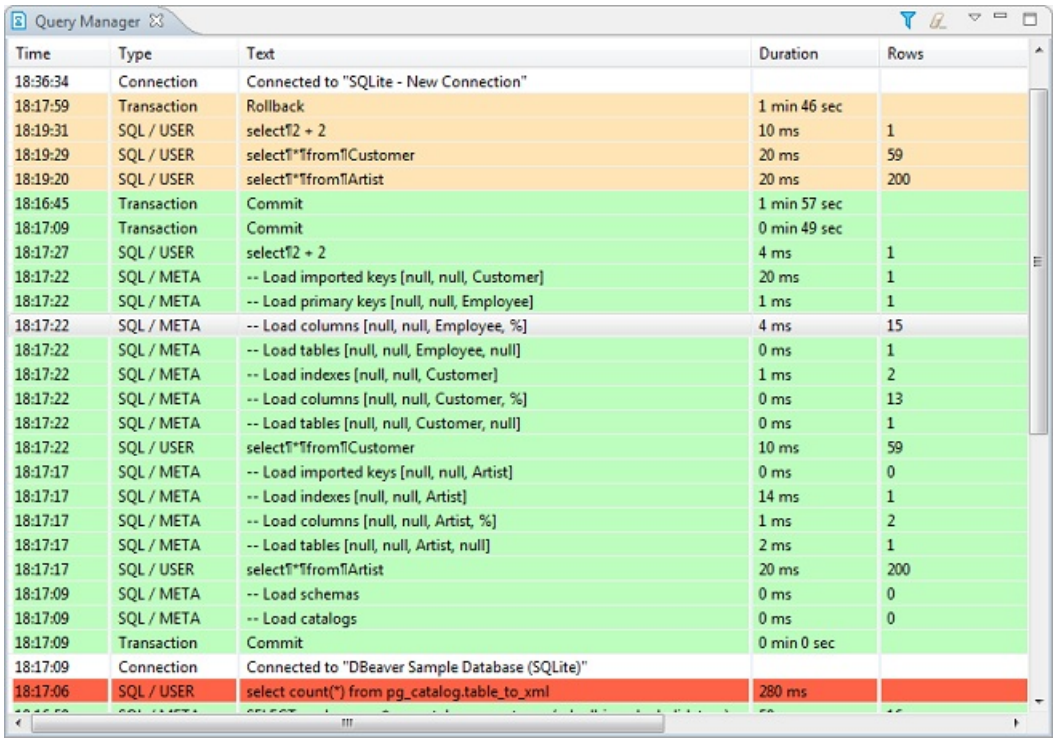
- Click the arrow next to the **Transaction Log** button in the toolbar and then click **Query Manager** on the dropdown menu:



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



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

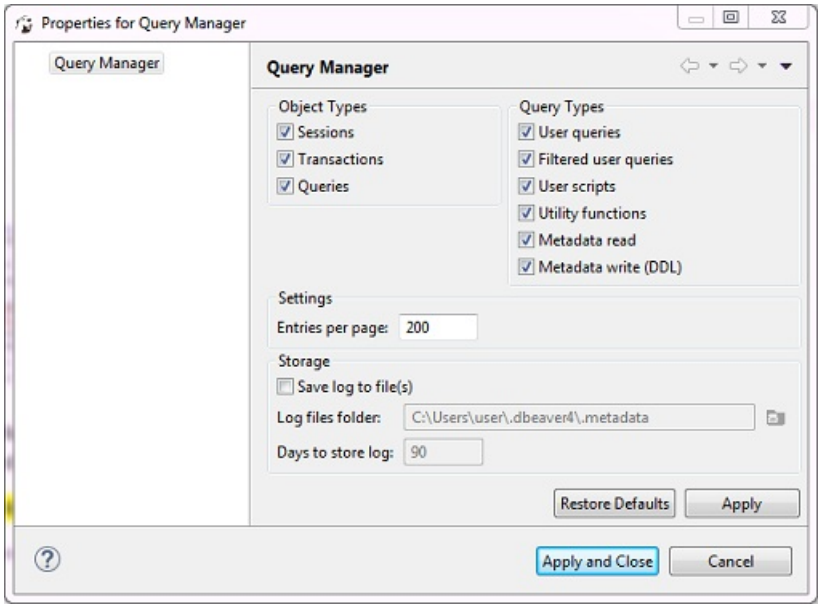


You can modify the look of the Query Manager by filtering queries and setting the number of entries displayed per page, as well as specifying some storage settings, see the 'Query Manager Properties' section below.

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

Query Manager Properties

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

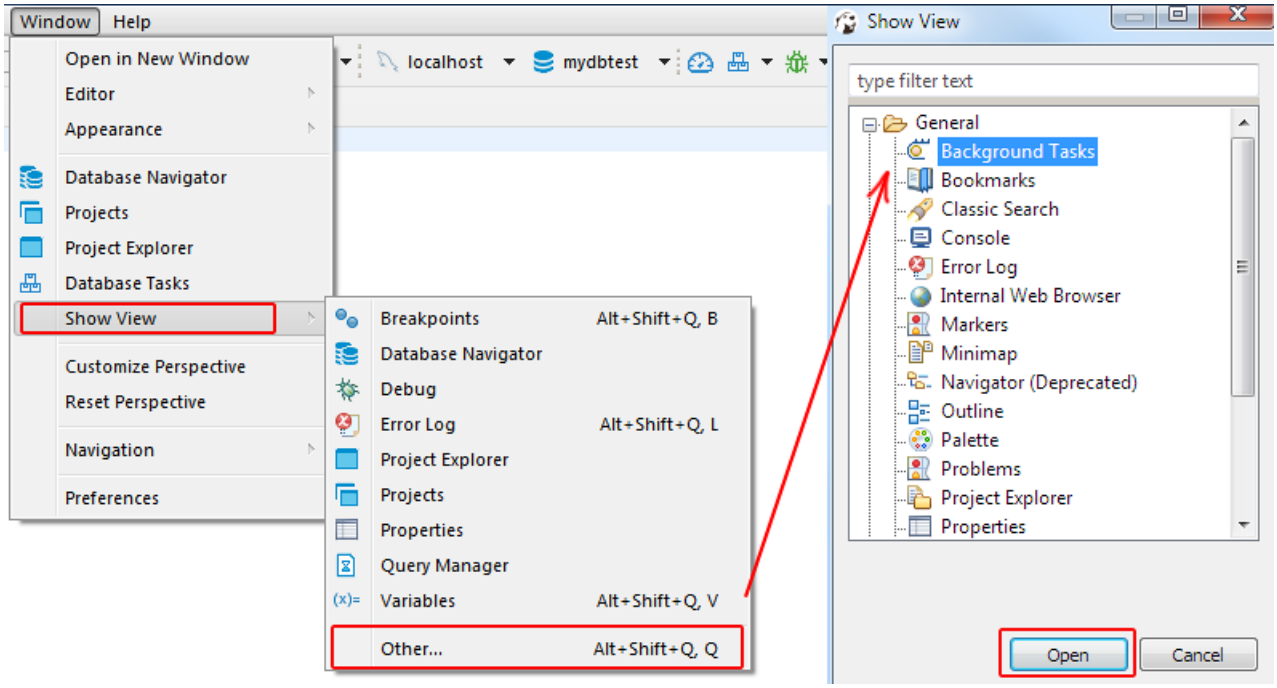


- To filter entries by object type, select or clear the checkboxes in the **Object Types** section. To filter entries by query type, select clear the checkboxes in the **Query Types** section.
- To change the number of entries displayed per page, enter the new number in the **Entries per page** field.
- To set DBeaver to save the query log in a file, select the **Save log to file(s)** checkbox and then specify the file location in the **Log files folder** field.

After you make all necessary changes to the settings, click **Apply** to apply the changes and keep the window open or click **Apply and Close** to apply the changes and close the window. To discard all changes and return to the previous state, click **Restore Defaults**.

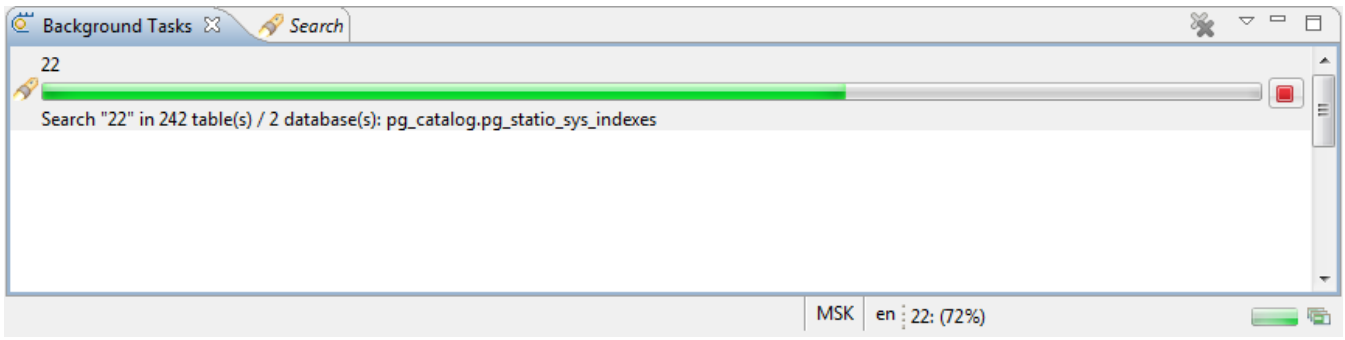
Background Tasks

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



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

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



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

Database Object Editor

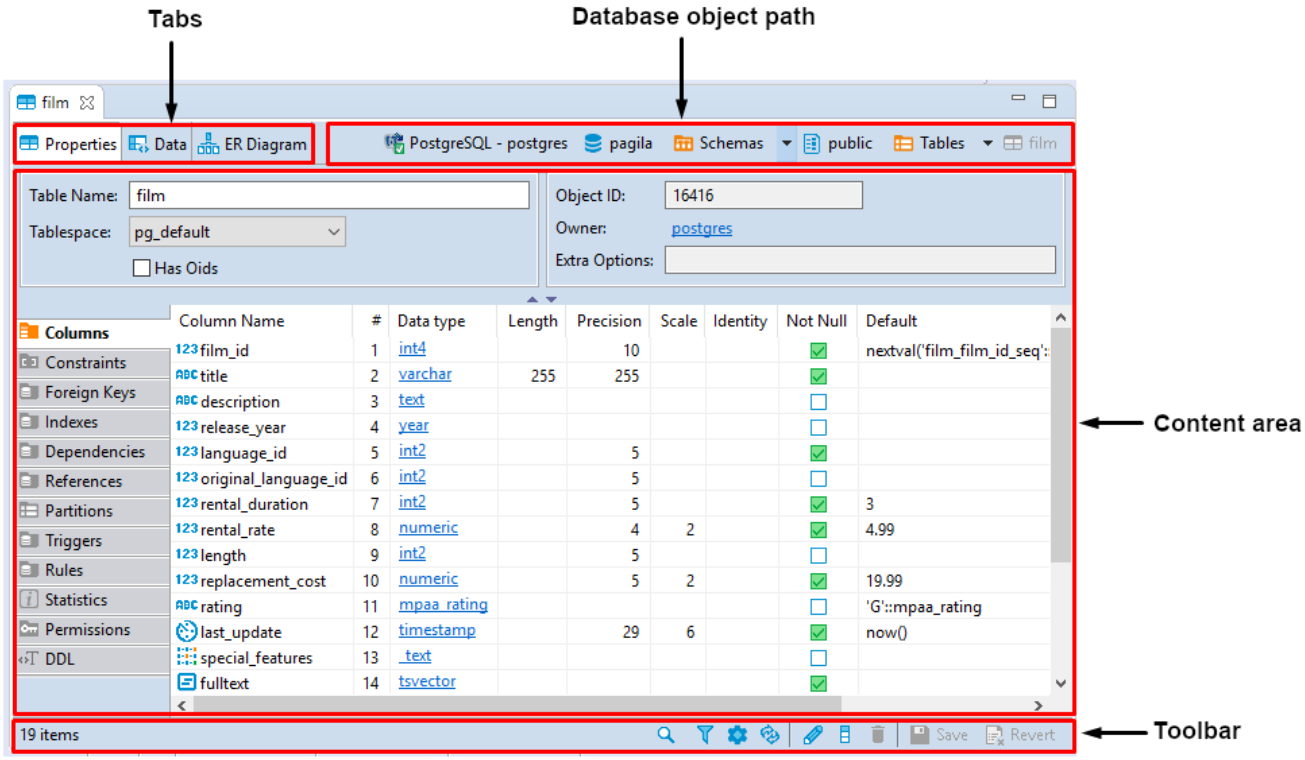
The Database object, or metadata editor is available for multiple database objects such as tables, views and schemas. To open the metadata editor for an object, in the Database Navigator or in the Projects view:

- Double-click the database object
- Click the database object and press `Enter` or `F4`

The editor has three tabs:

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

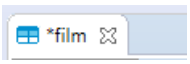
The tabs have the following common parts:



The object's path shows the chain of all its parent entities. The entities are clickable: clicking an entity in the path, depending on its nature, either shows its children or opens an editor or a settings window.

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

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



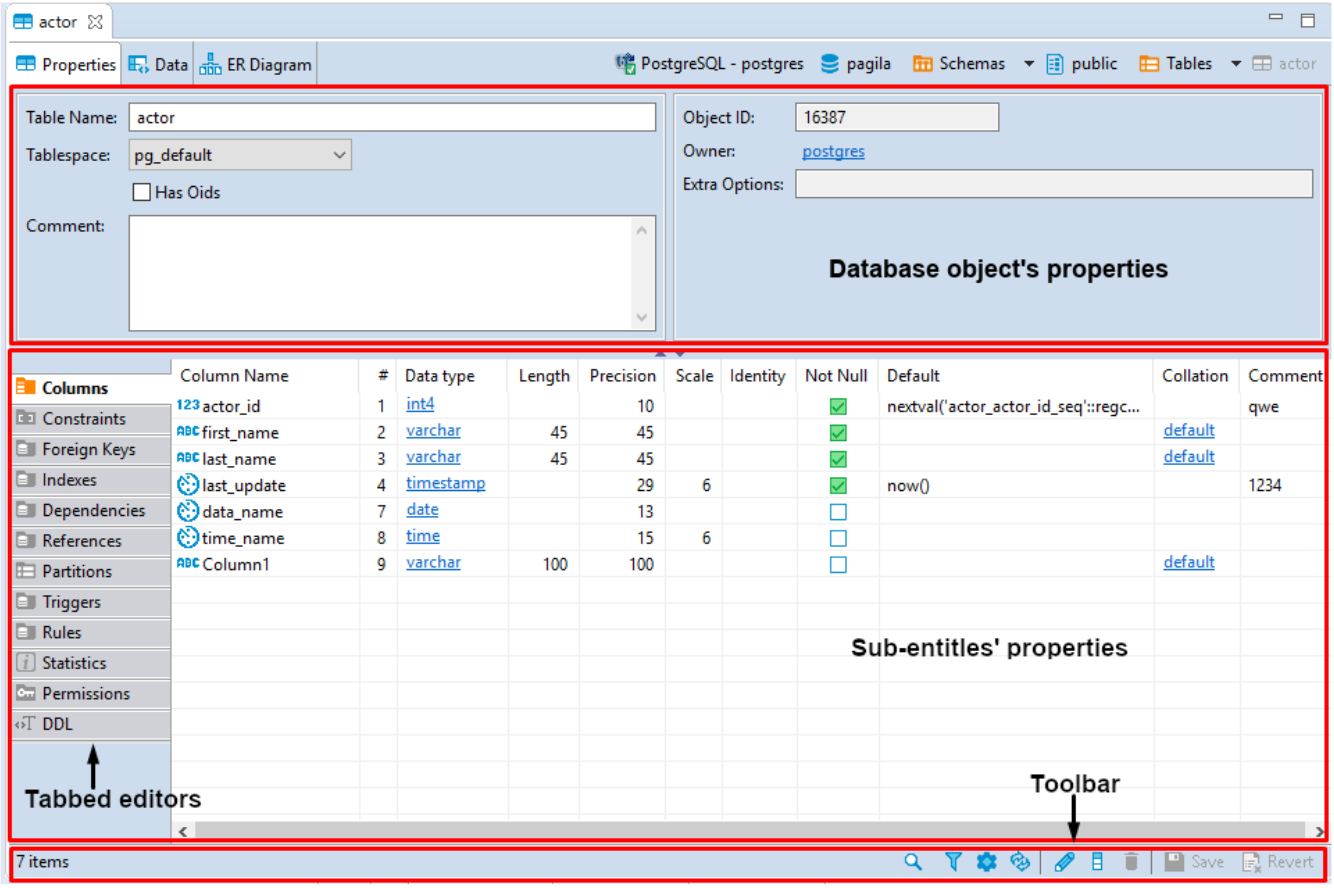
The Database Object editor supports the `Ctrl+Z` (undo) function.

Properties Editor

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

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

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



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

Button	Name	Description
	Search items	Displays a search field next to the button: - Type in the search combination - the content updates dynamically - To remove the filter, click the cross icon next to the search field
	Filter settings	Opens the Filtering window which allows setting a custom filter, see Configure Filters
	Configure columns	Opens the Configure columns dialog box in which you can select the columns to display or hide in the current view
	Refresh the selected items	Depending on the database type, refreshes either the current item or its parent or the whole database object – reloading data from the database
	View	Opens an editor/viewer for the item currently in focus
	Create new [items]	Creates a new item of the same type as currently displayed in the open view, for example, a column
	Delete database object	Deletes the item currently in focus
Save	Save the current contents	- Same as the **Save** button on the application main toolbar - Same as `Ctrl+S` - Opens the **Persist Changes** window that allows saving changes in the currently open sub-entity NOTE: DBeaver recommends saving work after each change.

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

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

SQL Script Editors

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

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

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

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

Data Editor

The Data editor appears:

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

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

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

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

The top toolbar contains the following buttons:

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

The side bar contains the following tabs:

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

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

The bottom toolbar provides the following buttons:

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

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

Menu Item	Description
Cut	Cuts the content of the current cell or column to the clipboard
Copy	Copies the content of the current cell or column to the clipboard
Advanced Copy	Opens advanced copy submenu that allows copying data with preset formatting parameters

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

For more information about using the Data Editor, please see the subsections of this article - open them via the contents tree on the right.

Navigation

Scrolling Results Page

If the result set has many rows, you can scroll the results page. To learn how many rows the data table contains, click the **Calculate total row count** button in the bottom toolbar. The number of rows appears in a status field next to the button: 8,715 . Alternatively, you can right-click a cell in the table and then click **Navigate -> Row Count** on the context menu.

By default, DBeaver limits the number of rows fetched to **200** (you can change this value in the main toolbar or in preferences). The maximum number of rows that DBeaver fetches to display in the Data tab is specified in the Maximum result-set size field in the main toolbar: ` 200 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 ( ) in the bottom toolbar or right-click the table and click **Navigate -> Fetch next page** on the context menu.

The number of rows fetched is visible in the status field under the data table:

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

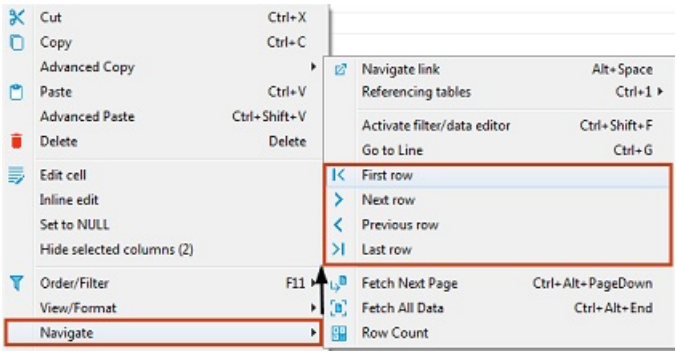
To fetch the whole result set, click the **Fetch all rows** button ( ) in the bottom toolbar or right-click the table and click **Navigate -> Fetch All Data** on the context menu.

NOTE: Be careful when fetching the whole result set. If it is huge, it might cause program hangup or out-of-memory errors.

You can navigate through the result set using standard shortcuts `Home`, `End`, `PgUp`, `PgDown`, `Ctrl+Home`, `Ctrl+End`.

Data Rows

To jump to the first or last row or move one row forward or backward, use the navigation buttons in the bottom toolbar or on the context menu:



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

History

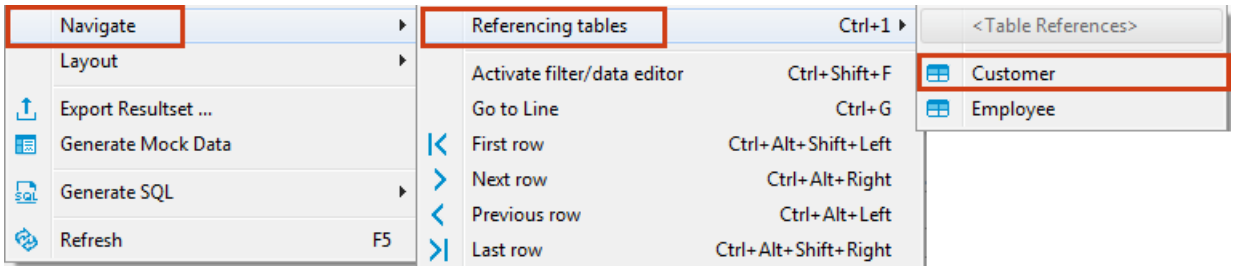
DBeaver remembers the history of actions such as applying filters to data, opening reference tables and other tables via links. You can navigate among such tables and filtered views:

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

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

Navigate Foreign Keys / Referencing Tables

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



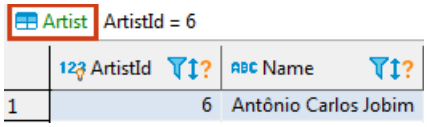
The referencing table opens in the same editor. To navigate back and forth between the initial and referencing tables, use the history navigation buttons ( ) in the top toolbar of the editor. In order to open a referencing table in a new window use the `Ctrl+Shift+1` shortcut to the show menu.

Navigation Links

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

	AlbumId	Title	ArtistId
7	7	Facelift	5
8	8	Warner 25 Anos	6
9	9	Plays Metallica By Four Cel	7

Another way is to right-click such a cell and click **Navigate -> Navigate link** on the context menu. The linked table opens in the same editor, filtered by the cell value:



Artist ArtistId = 6

	ArtistId	Name
1	6	Antônio Carlos Jobim

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

To navigate back and forth between the initial and linked tables, use the history navigation buttons ( ) in the top toolbar of the editor.
You can open a linked table in a separate editor. To do so, simultaneously hold the `Ctrl` key (or `⌘ command` key on macOS) and click the Navigate link icon ( ) in the cell.

Data View and Format

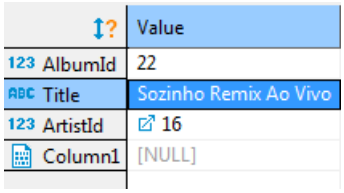
Grid vs. Plain Text Views

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

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

Table vs. Record Views

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

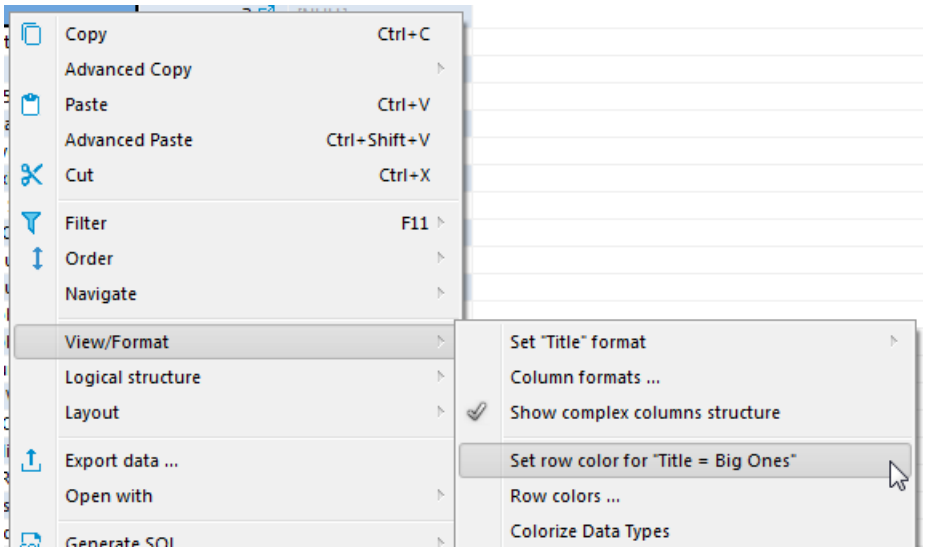


The Record view is useful if the table contains a big number of columns. To navigate from row to row of data, use the navigation buttons on the bottom toolbar of the editor:

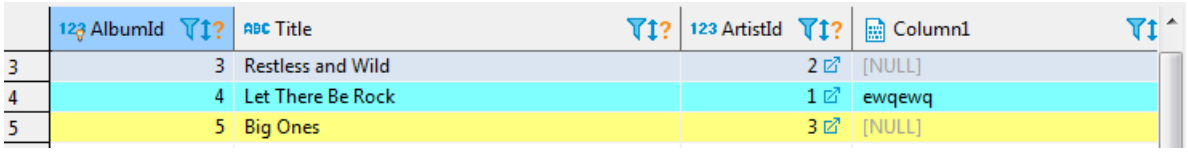
To return back to the standard table view, click the **Record** button again.

Rows Coloring

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

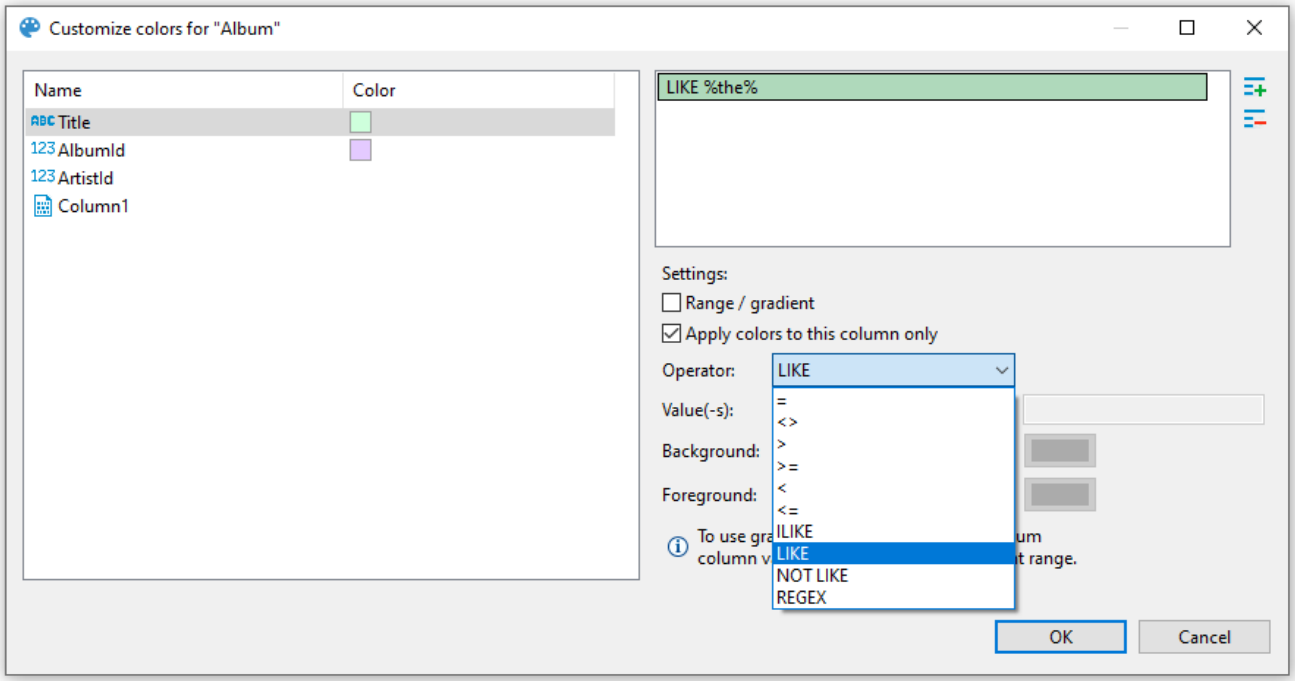


Then choose the colour in the palette window that appears and click **OK**. The current row and all other rows that contain the same value change their colour to the one you have selected:



To remove the coloring by a particular column, right-click the cell again and click `View/Format` ⇒ `Clear colour for {column name = value}` on the context menu.

By choosing `View/Format` ⇒ `Row colors ...` from the context menu, you can gain more precise control of coloring conditions:



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

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

Value range / Gradient

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

Title	AlbumId	ArtistId	Column1
For Those About To Rock W	1	1	PNG IHDR ¸Ó9...
Balls to the Wall	2	2	PNG IHDR n n I9±.
Restless and Wild	3	2	[NULL]
Let There Be Rock	4	1	ewqewq
Big Ones	5	3	[NULL]
Jagged Little Pill	6	4	[NULL]
Facelift	7	5	[NULL]
Warner 25 Anos	8	6	[NULL]
Plays Metallica By Four Cell	9	7	[NULL]
Audioslave	10	8	[NULL]

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

Using regex

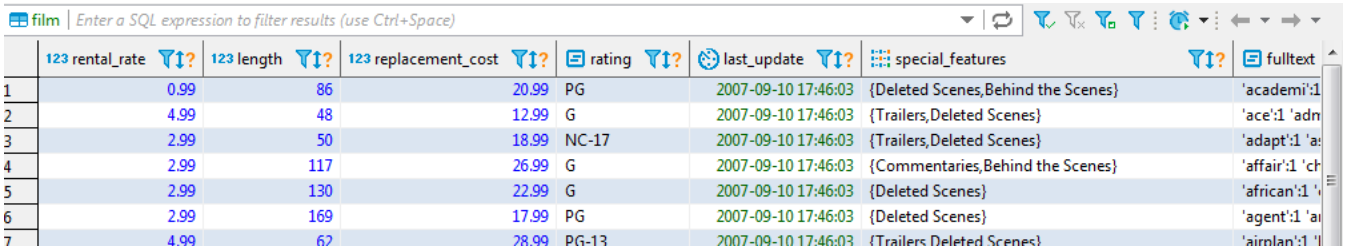
You can use regular expressions for matching complex values. Otherwise, you can be artistic and, for example, paint rows with odd values in your favorite color:

Title	AlbumId	ArtistId	Column1
For Those About To Rock W	1	1	PNG IHDR ¸Ó9...
Balls to the Wall	2	2	PNG IHDR n n I9±.
Restless and Wild	3	2	[NULL]
Let There Be Rock	4	1	ewqewq
Big Ones	5	3	[NULL]
Jagged Little Pill	6	4	[NULL]
Facelift	7	5	[NULL]
Warner 25 Anos	8	6	[NULL]
Plays Metallica By Four Cell	9	7	[NULL]
Audioslave	10	8	[NULL]

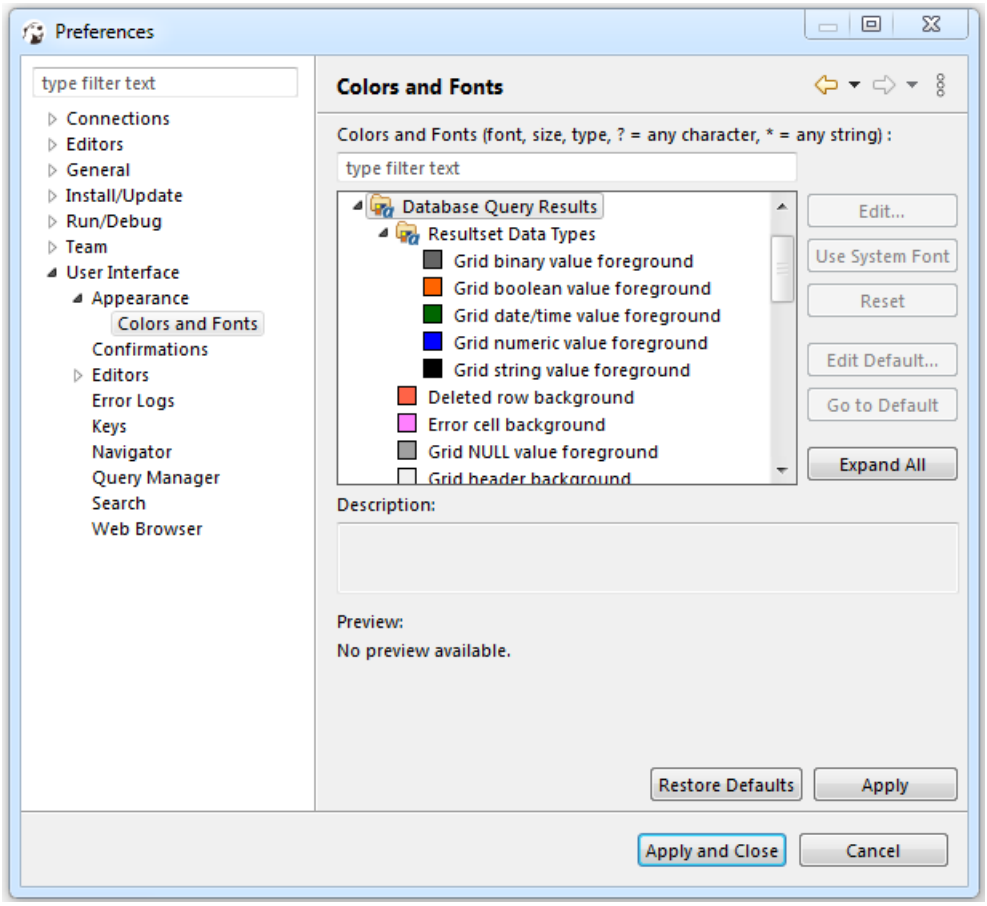
Snippet for coloring odd rows: `^\d*[13579]$`*; even rows:* `^\d*[02468]$`

Coloring by Data Types

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



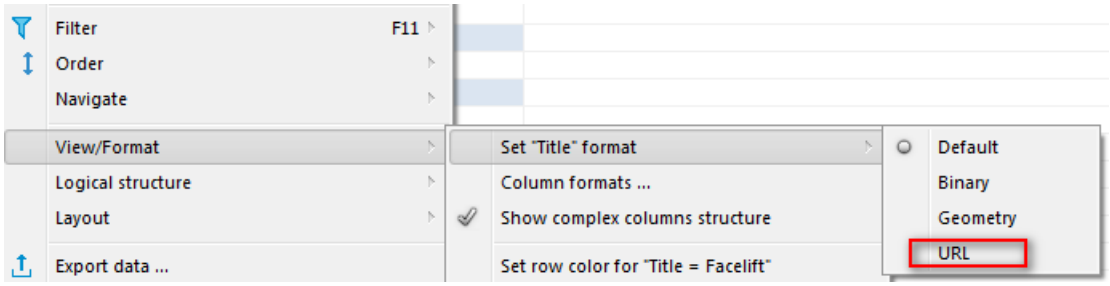
You can change the colour preferences in the Preferences window by: clicking **Window -> Preferences** on the main menu. In the window of the navigation pane on the left, expand **User Interface** and then **Appearance**, and then click **Colours and Fonts**:



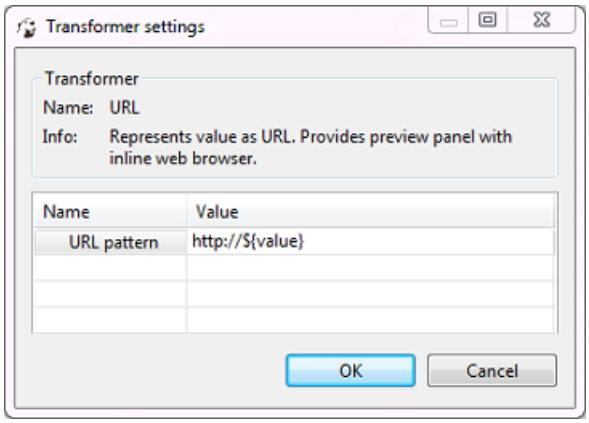
To remove the colouring by data types, on the context menu, click **View/Format -> Colourize Data Types** again.

Transforming Data Presentation

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



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

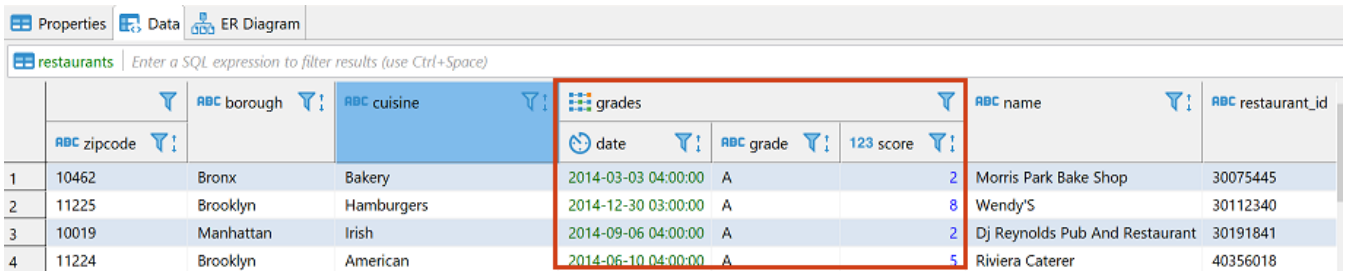


The values in the column appear in the new format.
NOTE: For URL format, the resulting cell provides a link to the URL in a browser window.

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

Structurizing Complex Data Types

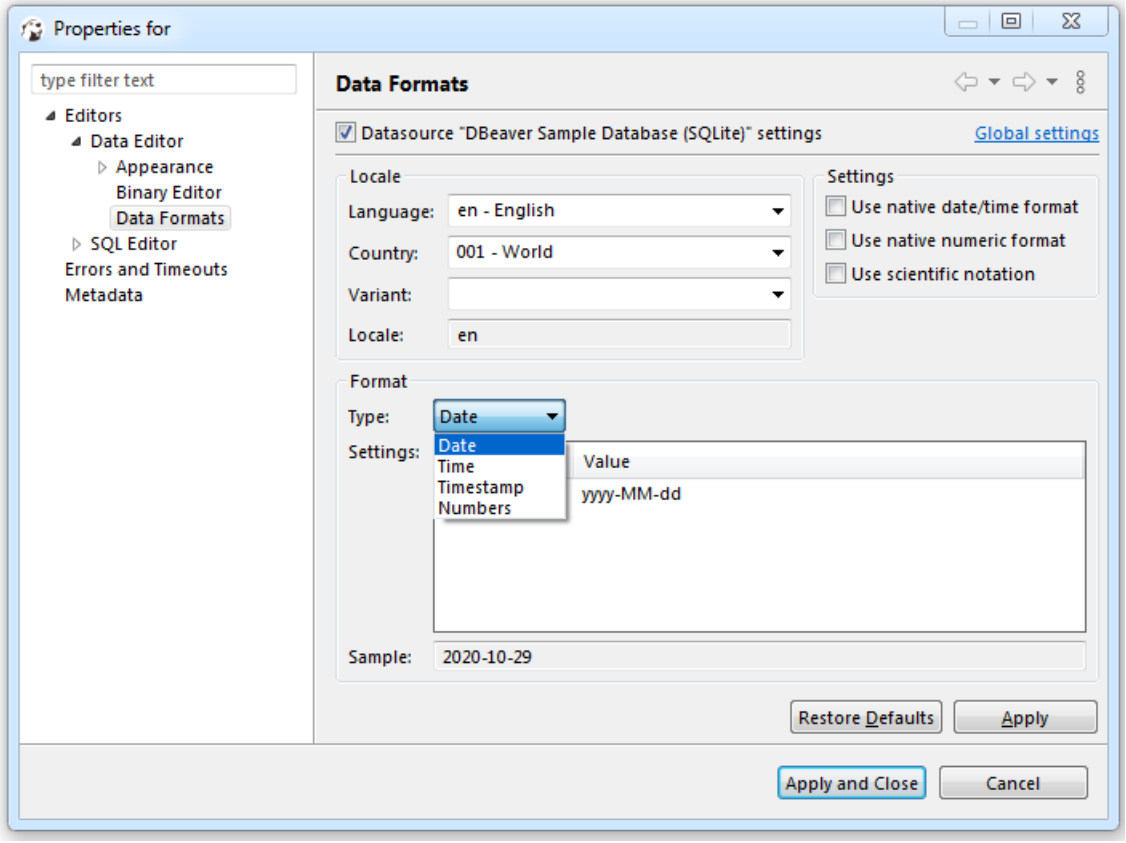
For complex data types (that themselves represent a structure), such as objects, structures and arrays, DBeaver provides a tool for breaking them into columns:



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

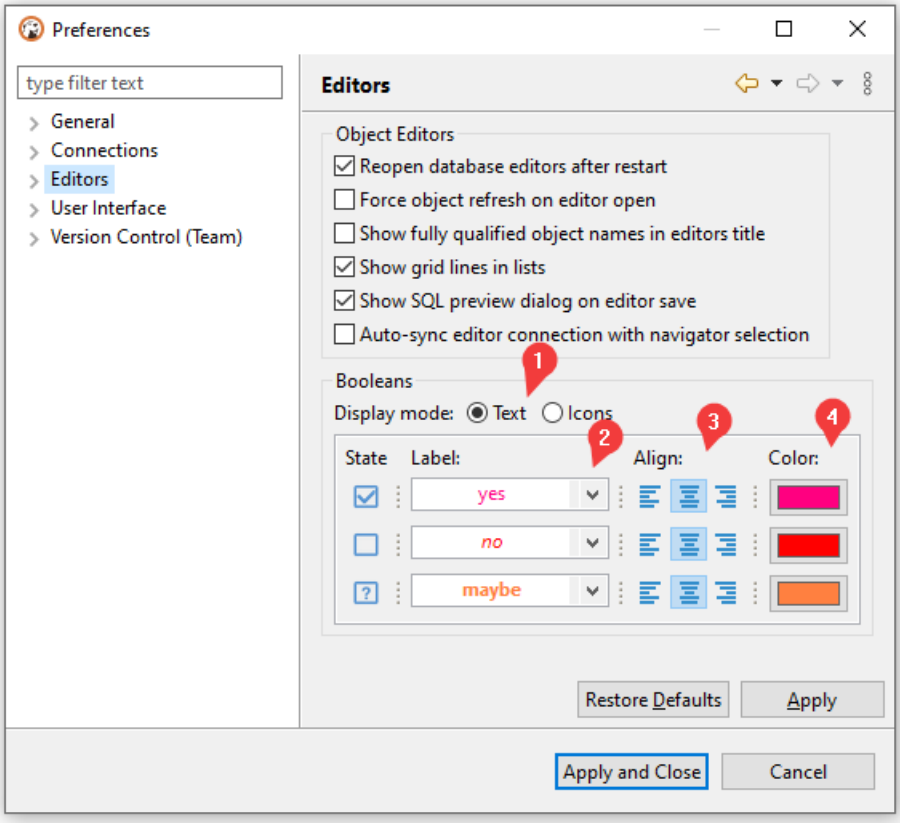
Configuring Numeric and Time Data Formats

You can specify the exact format of Time, Timestamp, Date, and Number data used in the currently open database or globally. To specify a format, right-click any cell in the table and, on the context menu, click **View/Format -> Data formats**. The Properties window opens displaying the **Data Formats** page:



To configure only the format for the current database, select the **Datasource "[Connection name]" settings** checkbox. To configure the settings globally, to all databases that you have in DBeaver, click **Global settings**.
You can specify the locale for the data format in the **Locale** area. In the **Type** dropdown list, click the name of the data type and in the **Settings** table, click the required format.
To apply the changes and make them visible in the table, click **Apply and Close** and then refresh the window (`F5`).

Configuring Boolean presentation



You can choose between two presentation modes[1]:

- Text-based

- Icon-based

Text-based presentation

This is the most customizable mode. You can:

1. Change **labels** under `Label` column.
   Otherwise, you can use presets available in `Drop-down Menu`[2]
2. Change **alignment** of value inside grid cell.[3]
   Following variants are available: **left**, **center**, and **right**
3. Change **color** of value using color picker under `Color` column[4].
   You can reset color to match current theme's contrast color in `Drop-down Menu`[2] ⇒ `Colors` ⇒ `Use theme default color`
4. Change font **style** in `Drop-down Menu`[2] ⇒ `Styles`.
   Following variants are available: normal, **bold**, and *italic*

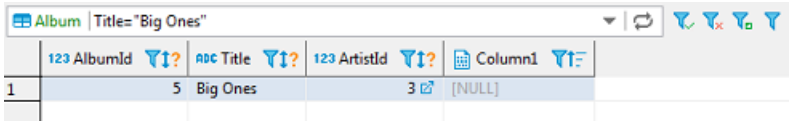
Icon-based presentation

This presentation only supports alignment changing.

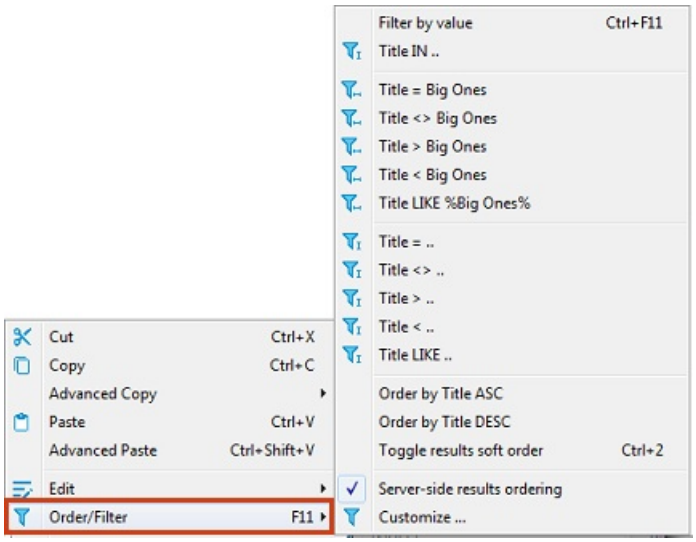
Data Filters

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

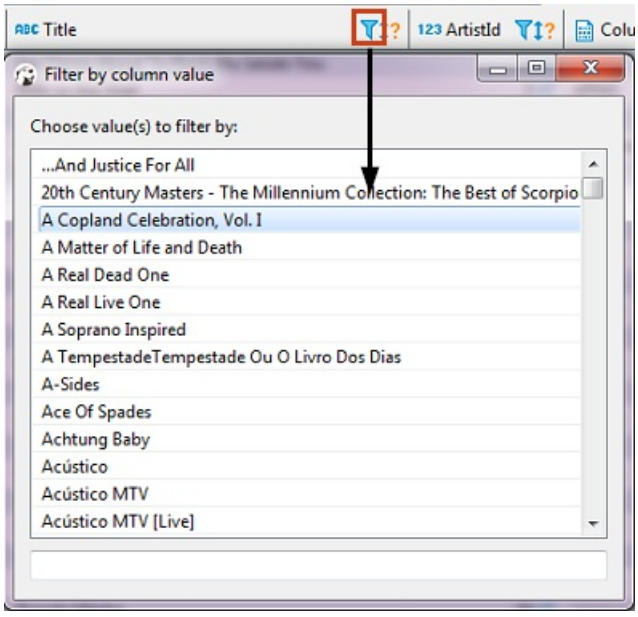
One of the ways is to use the filter field above the table next to the top toolbar. To filter data, enter an SQL expression into the field and click the Apply filter criteria button ( ) next to the field or press `Enter`.



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



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

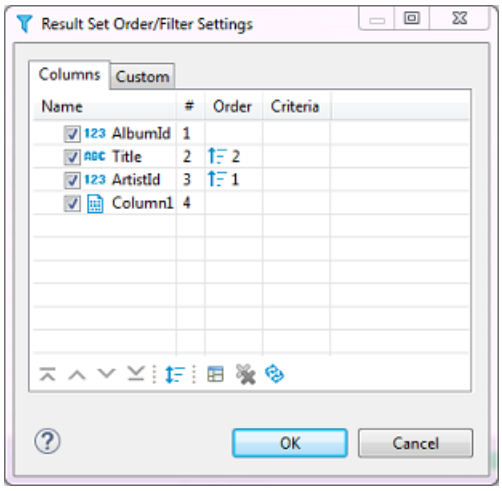


The data updates dynamically. To remove a filter, click the Remove All Filters/Orderings button ( ) in the top toolbar of the editor.

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

Advanced filters configuration

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

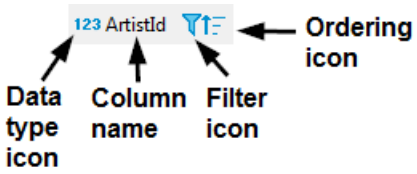


To open this window, click the Custom Filters button ( ) in the top toolbar of the editor or click the Configure button ( ) and then click **Order/Filter** on the dropdown menu.

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

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

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

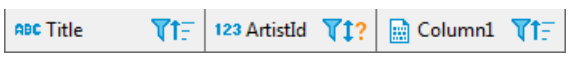


- By simply clicking the column name or column type, the icon highlights the whole column.
- You can click the column type icon and then drag and drop the column to a different position in the table.
- You can click the column name and then drag the cursor to the right or left to highlight multiple columns.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

Ordering Data in Columns

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

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

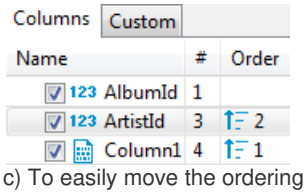
   The icon has three states:

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

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

2. Click the Custom Filters button ( ) in the top toolbar of the editor to open the Result Set Order/Filter Settings window (see above):

a) Next to the column by which you want to order data in the first turn, set the ascending or descending order using the same three-state principle as described above.
b) Set the ordering in other columns by which you want to sort the data in the second, third, etc. turn. The **Order** column indicates the order in which the sorting will happen.
NOTE: The number (**#**) column indicates the initial order of columns.



c) To easily move the ordering setting from column to column, you can use the Move up/down/to top/to bottom/ buttons:

To reset the data ordering to its initial state, click the Reset button ( ) in the same window.

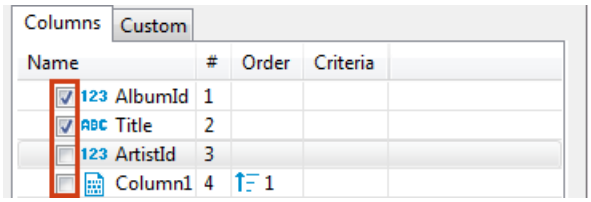
Also, to remove all ordering settings, click the Remove All Filters/Orderings button ( ) in the top toolbar of the Data Editor.

Managing Display of Columns in Data Table

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

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

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



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

Sorting Columns in Data Table

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

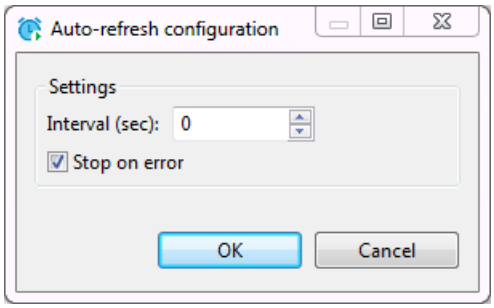
1. Click the icon in the column header and drag-and-drop the column to a new position.
2. To sort the column alphabetically, in the Result Set Order/Filter Settings window (open by clicking the Custom Filters button ( ) in the top toolbar of the editor), click the Sort button ( )
3. In the Result Set Order/Filter Settings window, click the column to set the focus to it and then move it using the navigation buttons: ( )

Data Refresh

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

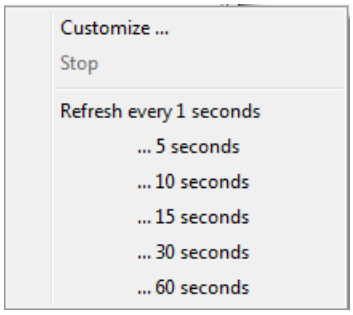
You can also schedule auto-refresh to happen on a regular basis. To auto-refresh the database on schedule:

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



   a) Set **Interval** in seconds.
   b) Select the **Stop on error** checkbox if you want the refresh to stop when it encounters an error or clear it, if the refresh should ignore errors.
   c) Click **OK**.

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



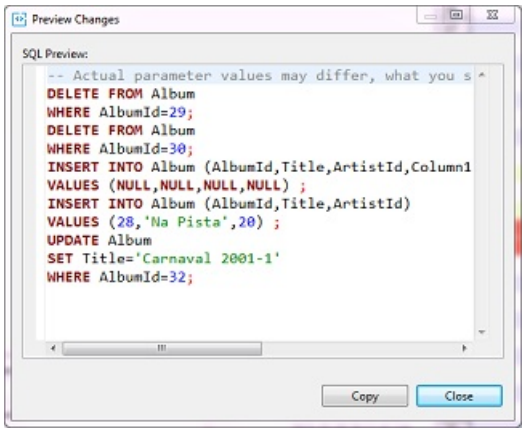
   On the menu, you can click one of the preset options or click **Customize** to open the Auto-refresh configuration dialog box, see option 1.

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

Data Viewing and Editing

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

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



Inline Editing

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

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

The cell is now editable and you can change its value.

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

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

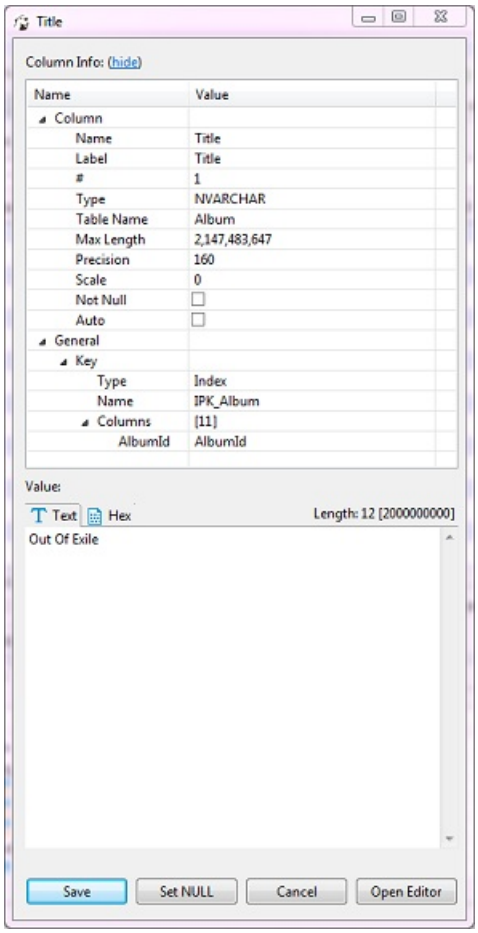
NOTE: Both the **Save** and **Cancel** buttons become editable only when you make changes in a cell and then jump to another cell.

Cell Editor

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

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

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



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

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

Adding, Copying and Deleting Rows

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

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

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

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

Copying/Pasting Cells

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

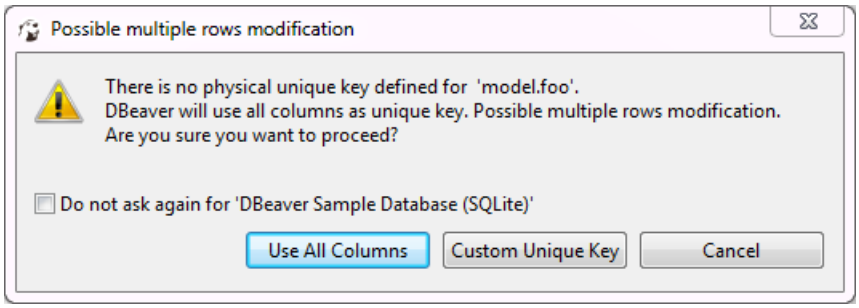
DBeaver provides the advanced copy option that allows configuring additional copy settings (copy with column names/row numbers, configure delimiter and choose value format). To copy cells with additional settings, press `Ctrl+Shift+C` or right click the cell(s) and click **Advanced Copy** on the context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

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

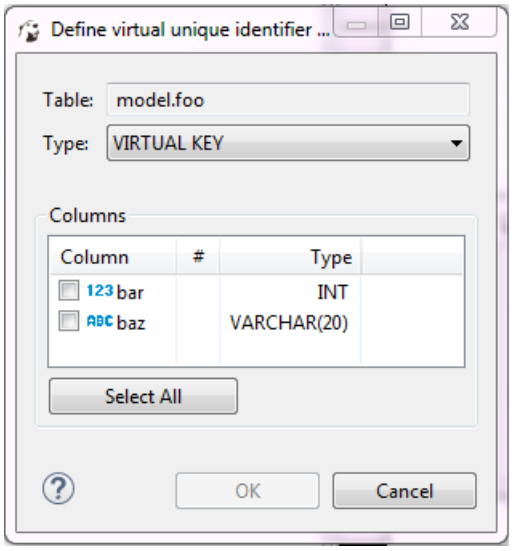
Defining Virtual Keys

To be able to save column value changes, a table must have some unique key (primary key or unique index). Some databases (Oracle,

DB2, PostgreSQL) support a special virtual unique column that DBeaver can use to save changes. In other cases, you can define a virtual key – a set of columns that forms a unique combination of values. When you try to save changes in a table without a unique key, DBeaver displays the following error message:



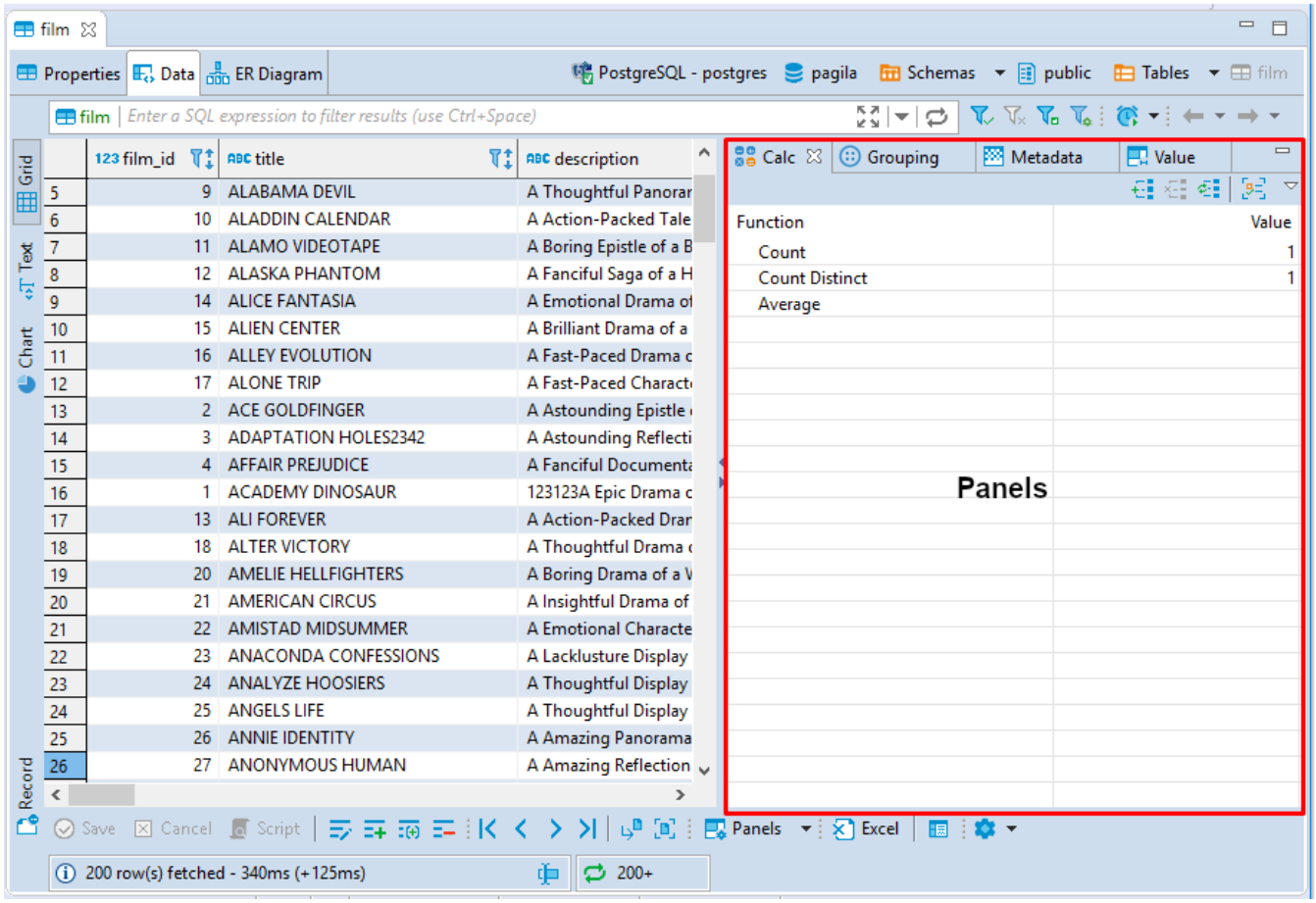
To use all columns as the virtual key, click **Use All Columns**. To create a custom key, click **Custom Unique Key**. Alternatively, to create a unique custom key, you can click the **Configure** ( ) button on the bottom toolbar and then click **Define virtual unique key** on the Configure menu. The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



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

Panels

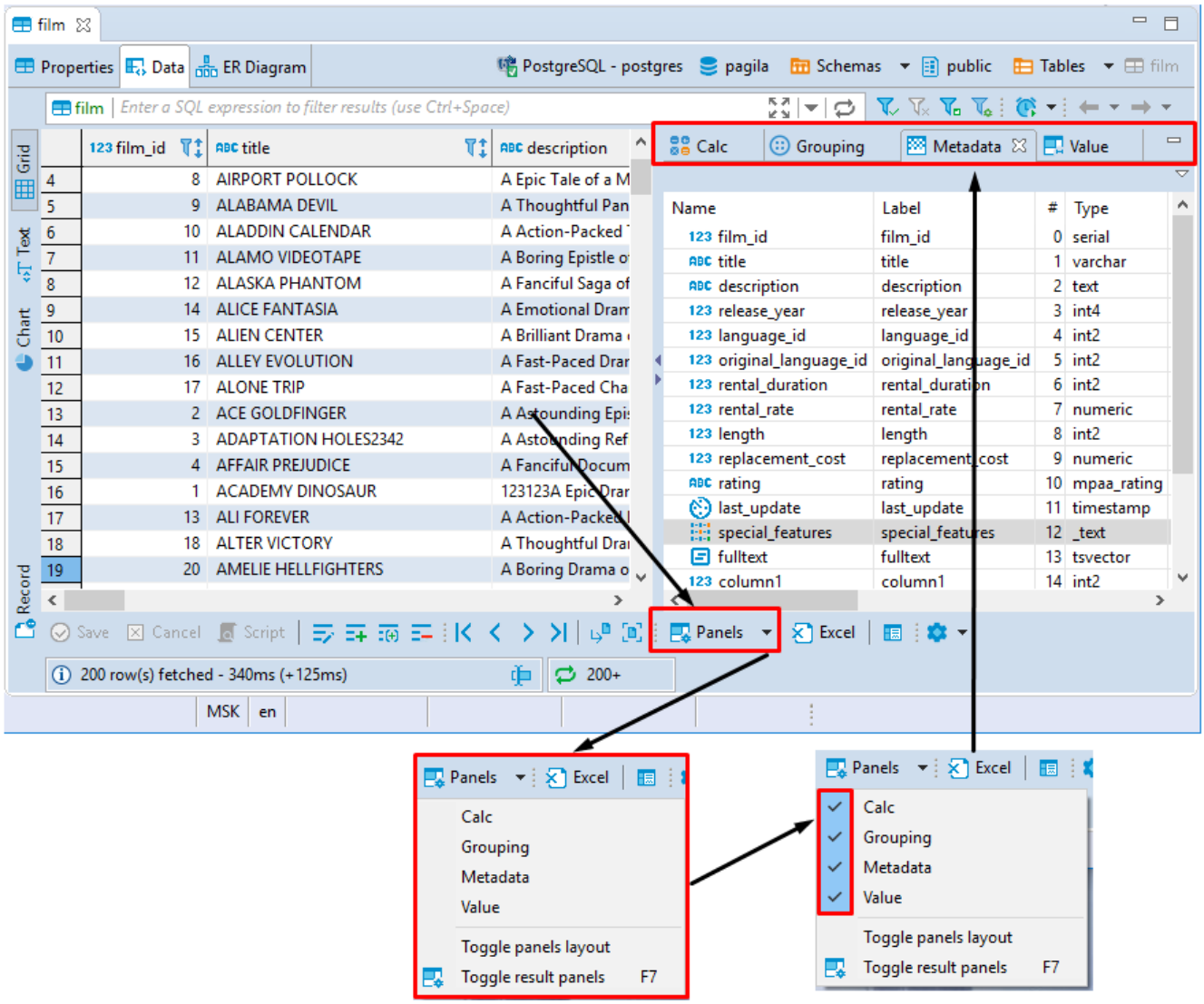
Panels provide additional space in the Data editor in which you can manipulate data. The panels are handy if you work with complex types (structures, arrays), long text data, or BLOBs. Panels appear as tabs in an additional pane in the right hand side of the Data tab:



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

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

To open the panels, click **Panels** on the bottom toolbar. By default, the Value viewer panel opens. Alternatively, you can open the Value panel by pressing `F7` on a cell. To open the other panels, click the down arrow next to the **Panels** button and click the name of the panel on the menu:

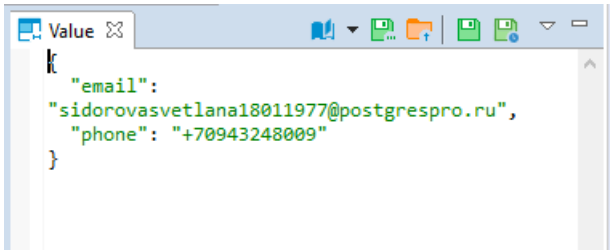


Panels will also open if you try to inline-edit a cell with a complex data type.

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

Value Viewer

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



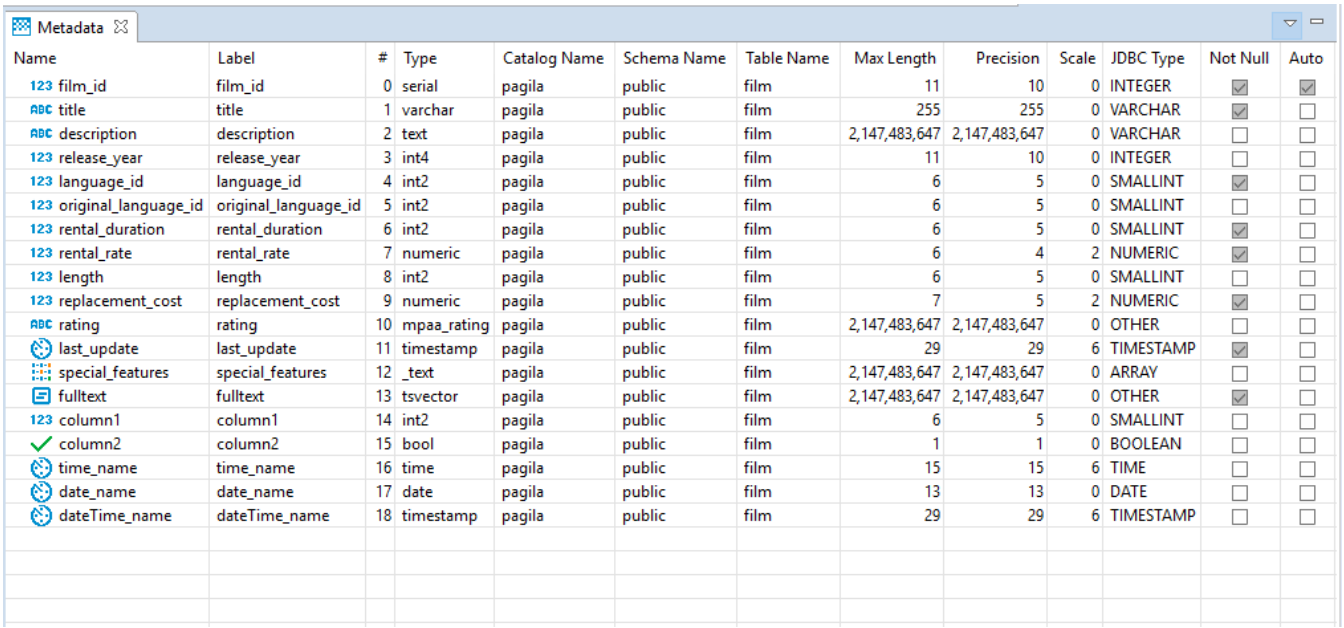
The toolbar of the Value panel contains the following buttons:

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

Button	Name	Description
	**Load from file...**	Allows uploading data from a local file. **NOTE**: This button is only available for XML,JSON and Binary content.
	**Apply cell value**	Displays the data table changes 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 on the bottom toolbar of the Data Editor..
	**Auto-apply value**	Enables the automatic display of changes made in the Value viewer in the data table. When auto-saving is enabled, the changes appear in the data table at the same time when they are made in the Value viewer.**NOTE**: This button does not save changes made to the database. To save the changes in the database, you need to use the **Save** button on the bottom toolbar of the Data Editor.

Metadata Panel

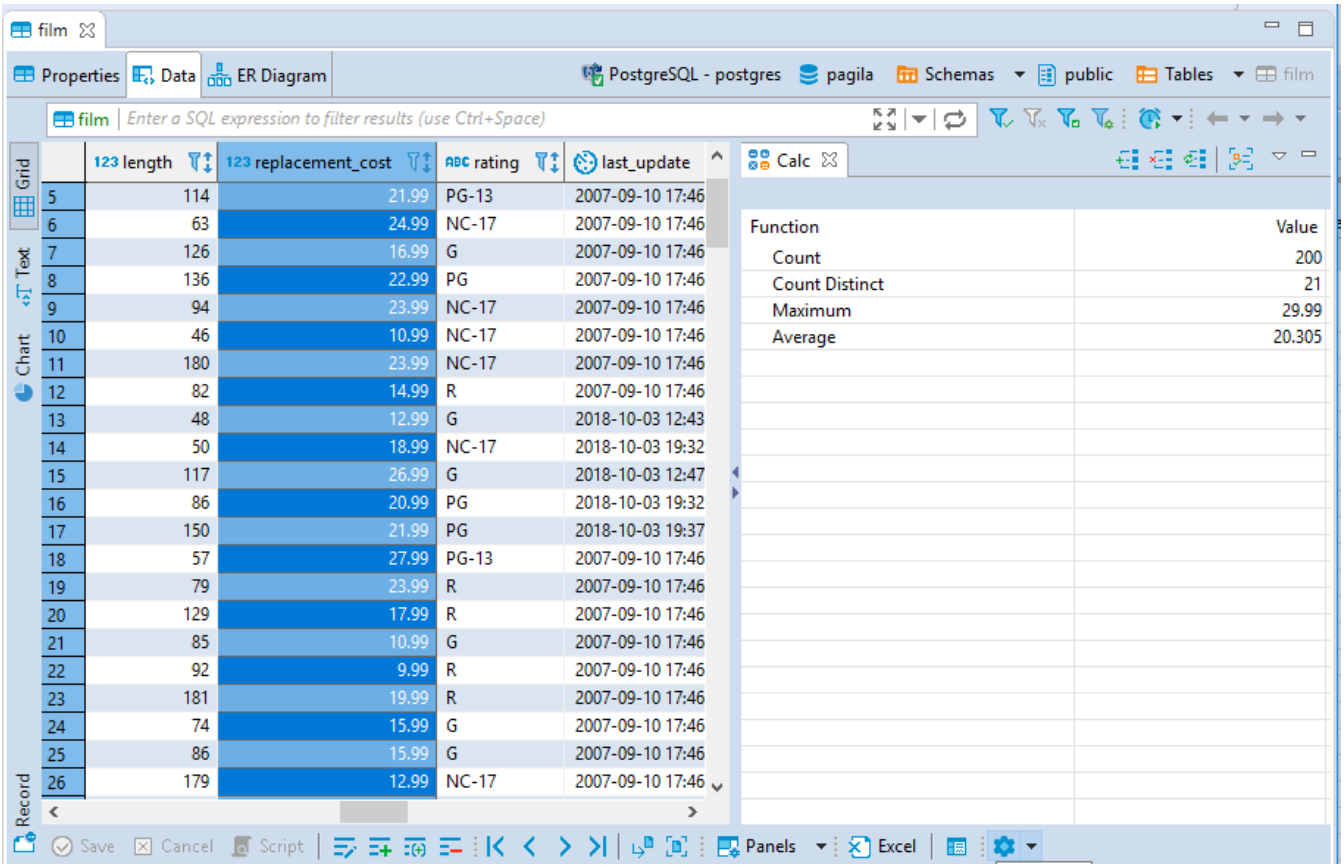
The Metadata panel displays metadata for each cell in the row containing the cell currently in focus. You can just view the metadata.



Name	Label	#	Type	Catalog Name	Schema Name	Table Name	Max Length	Precision	Scale	JDBC Type	Not Null	Auto
film_id	film_id	0	serial	pagila	public	film	11	10	0	INTEGER	☑	☑
title	title	1	varchar	pagila	public	film	255	255	0	VARCHAR	☑	☐
description	description	2	text	pagila	public	film	2,147,483,647	2,147,483,647	0	VARCHAR	☐	☐
release_year	release_year	3	int4	pagila	public	film	11	10	0	INTEGER	☐	☐
language_id	language_id	4	int2	pagila	public	film	6	5	0	SMALLINT	☑	☐
original_language_id	original_language_id	5	int2	pagila	public	film	6	5	0	SMALLINT	☐	☐
rental_duration	rental_duration	6	int2	pagila	public	film	6	5	0	SMALLINT	☑	☐
rental_rate	rental_rate	7	numeric	pagila	public	film	6	4	2	NUMERIC	☑	☐
length	length	8	int2	pagila	public	film	6	5	0	SMALLINT	☐	☐
replacement_cost	replacement_cost	9	numeric	pagila	public	film	7	5	2	NUMERIC	☑	☐
rating	rating	10	mpaa_rating	pagila	public	film	2,147,483,647	2,147,483,647	0	OTHER	☐	☐
last_update	last_update	11	timestamp	pagila	public	film	29	29	6	TIMESTAMP	☑	☐
special_features	special_features	12	_text	pagila	public	film	2,147,483,647	2,147,483,647	0	ARRAY	☐	☐
fulltext	fulltext	13	tsvector	pagila	public	film	2,147,483,647	2,147,483,647	0	OTHER	☑	☐
column1	column1	14	int2	pagila	public	film	6	5	0	SMALLINT	☐	☐
column2	column2	15	bool	pagila	public	film	1	1	0	BOOLEAN	☐	☐
time_name	time_name	16	time	pagila	public	film	15	15	6	TIME	☐	☐
date_name	date_name	17	date	pagila	public	film	13	13	0	DATE	☐	☐
dateTime_name	dateTime_name	18	timestamp	pagila	public	film	29	29	6	TIMESTAMP	☐	☐

Calc Panel

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



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

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

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

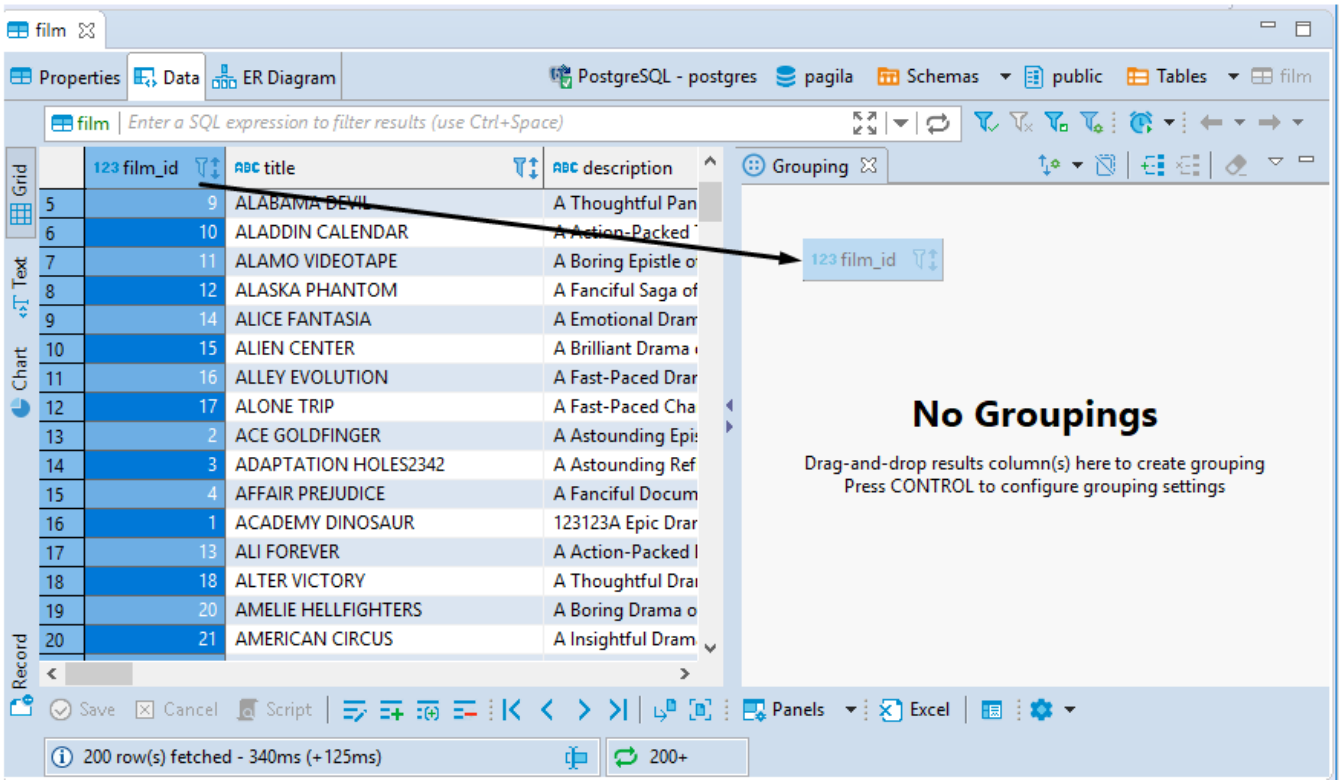
- Sum
- Average
- Minimum
- Maximum
- Median
- Mode

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

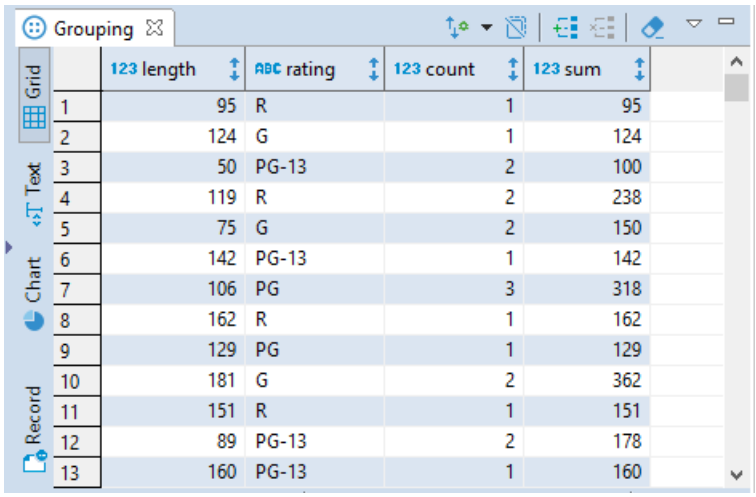
You can copy the value of a particular function to the clipboard - right-click the row and click **Copy Value** on the context menu.
You can also copy all functions with their values - right-click in the table and click **Copy All** on the context menu.

Grouping Panel

The Grouping panel provides tools to calculate statistics based on a table of a custom SQL query. It uses GROUP BY queries to extract unique values for COUNT (default), SUM, AVG, MIN, MAX and other analytics functions displaying the results in dedicated columns.
To obtain the grouping results for one or more columns of a data table, open the Grouping panel, then, in the results table, put the cursor onto the data type icon of the table header so that the cursor turns into a hand pointer ( ), and drag-n-drop the column(s) into the panel:

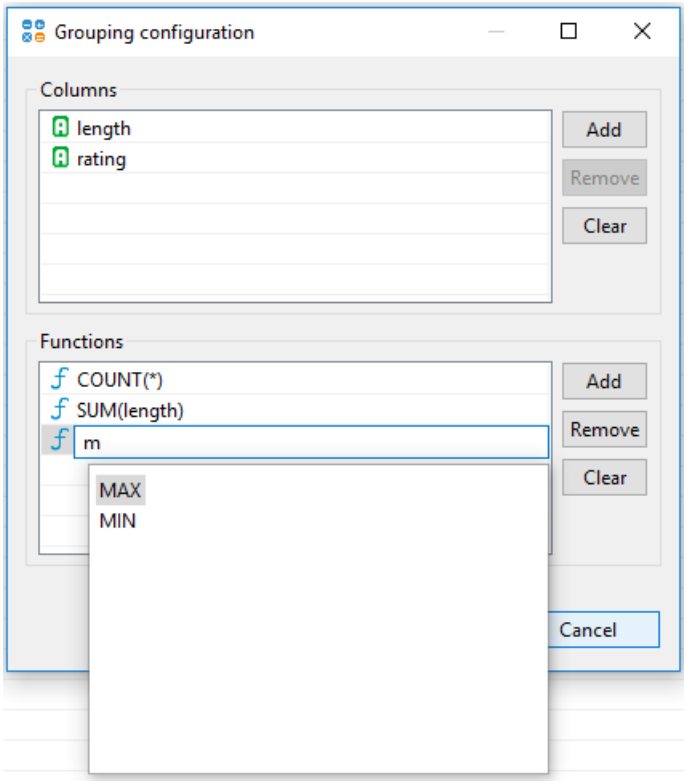


If you add several columns to the panel, DBeaver groups data in the order in which the columns go and calculates statistics based on the grouping.



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

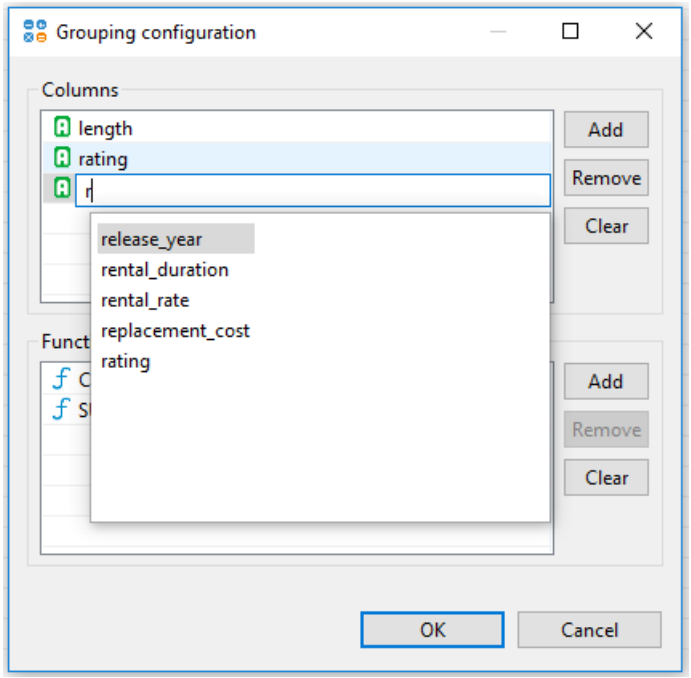
1. Click the **Edit grouping columns** button on the panel`s toolbar.
2. In the Grouping Configuration window, in the **Functions** area, click **Add**, then type the function into the new row:
    - You can use the auto-complete options DBeaver provides.
    - You need to indicate the column name in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



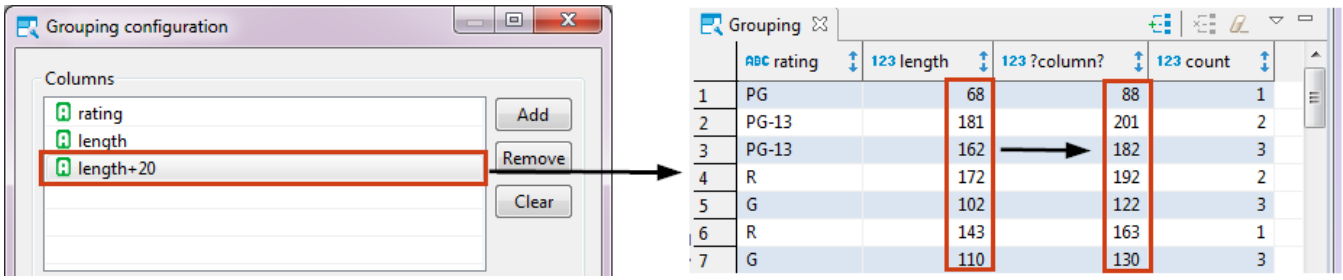
To remove a function, in the same Grouping Configuration window, click the function and click **Remove** and then **OK**. To remove all functions, click **Clear** and then **OK**.

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

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



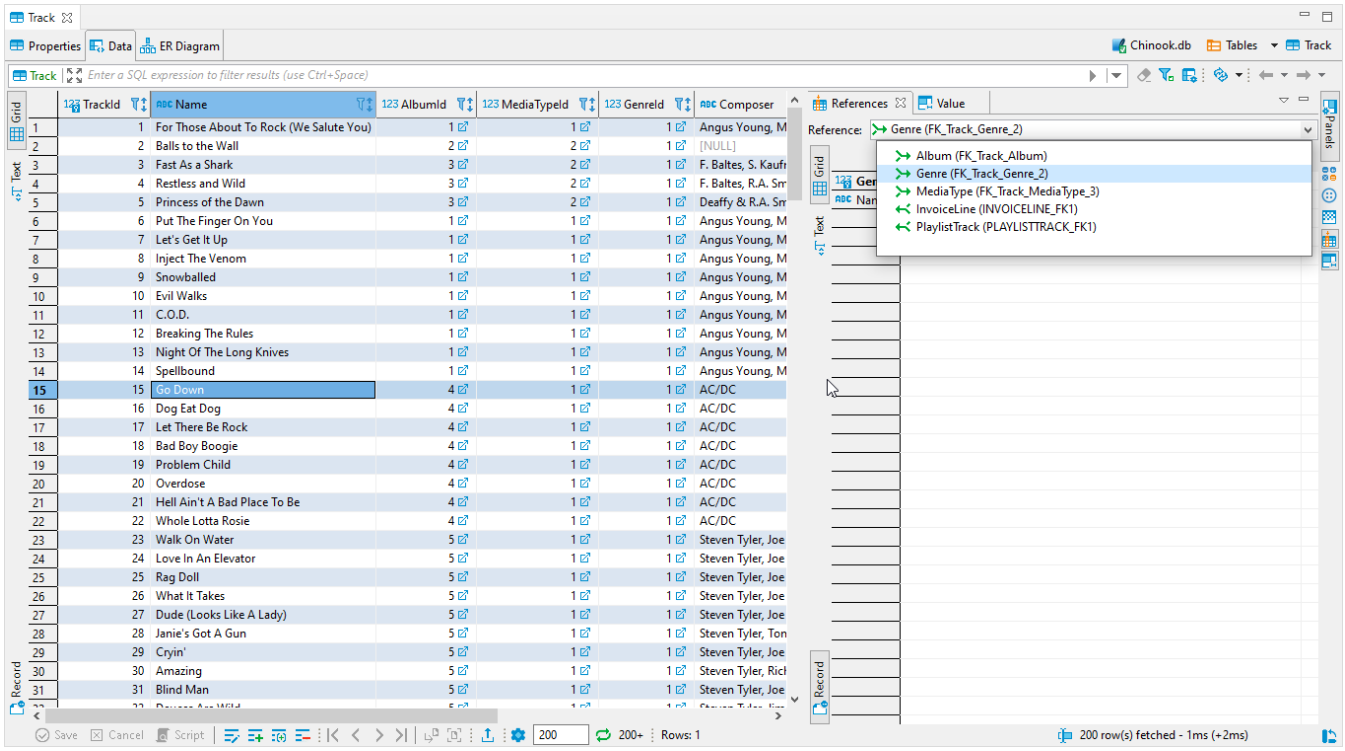
For MySQL/MariaDB databases you can also add a column with an expression - the expression will be calculated in the resulting column:



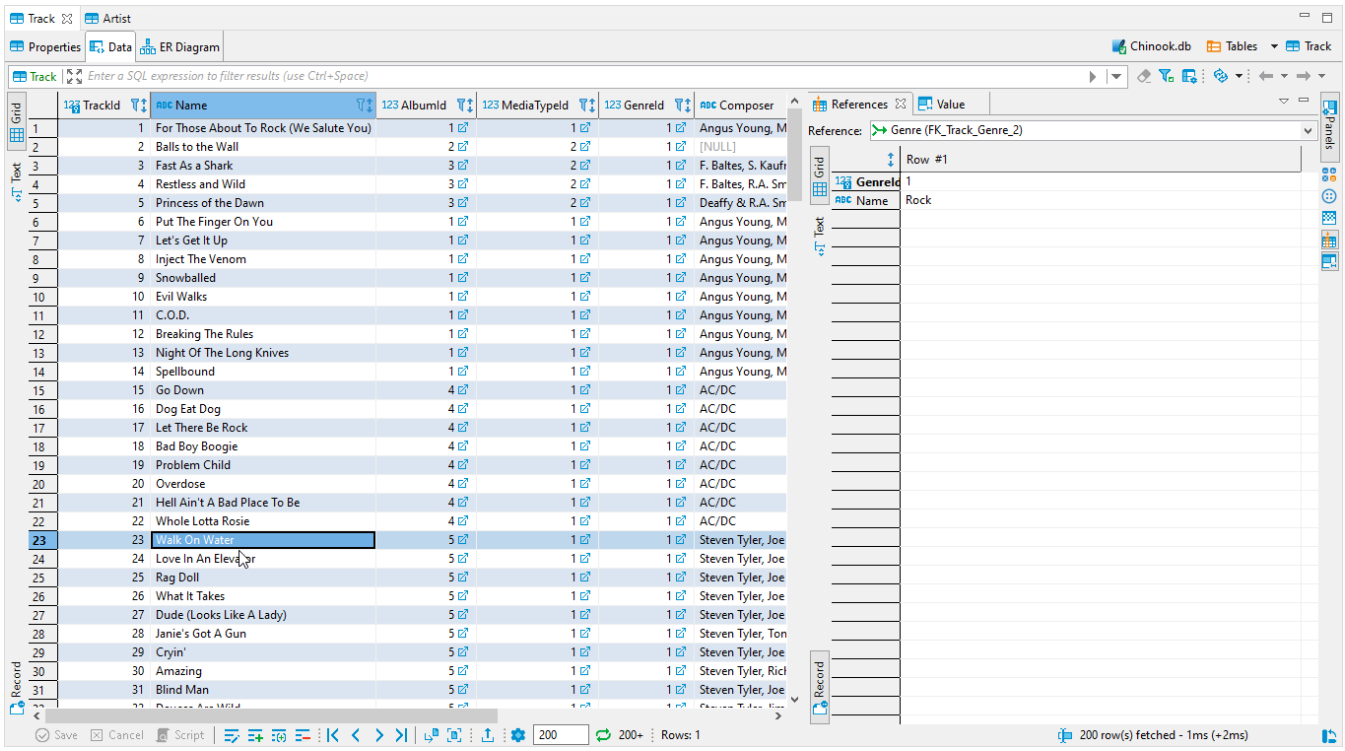
To remove a column, in the Grouping Configuration window, in the **Columns** area, click the column name, then **Remove** and **OK**. To remove all columns, click **Clear** and **OK**.
Another way to remove a column is to click the column and then the **Remove grouping column** button ( ) in the panel`s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

References panel

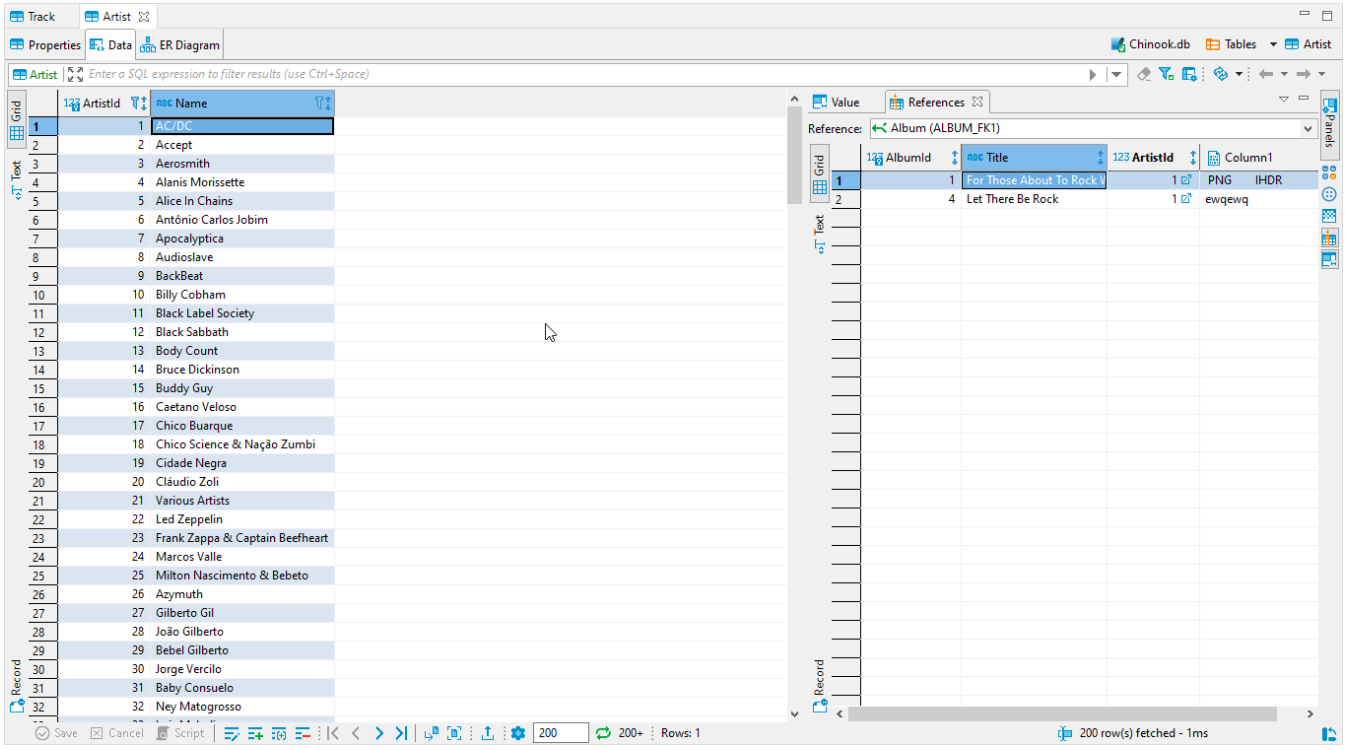
The references panel allows you to see all the related information for the chosen row from other connected tables. The information is presented in an additional data viewer window, filtered to show the information related to the currently selected row. If a table opened in data viewer has a foreign key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



When a table that is referenced by a foreign key in the current table is chosen, the information from the row corresponding to a referenced key will be shown, in this situation the record mode is enabled by default, but it can be turned off like in a normal data viewer.



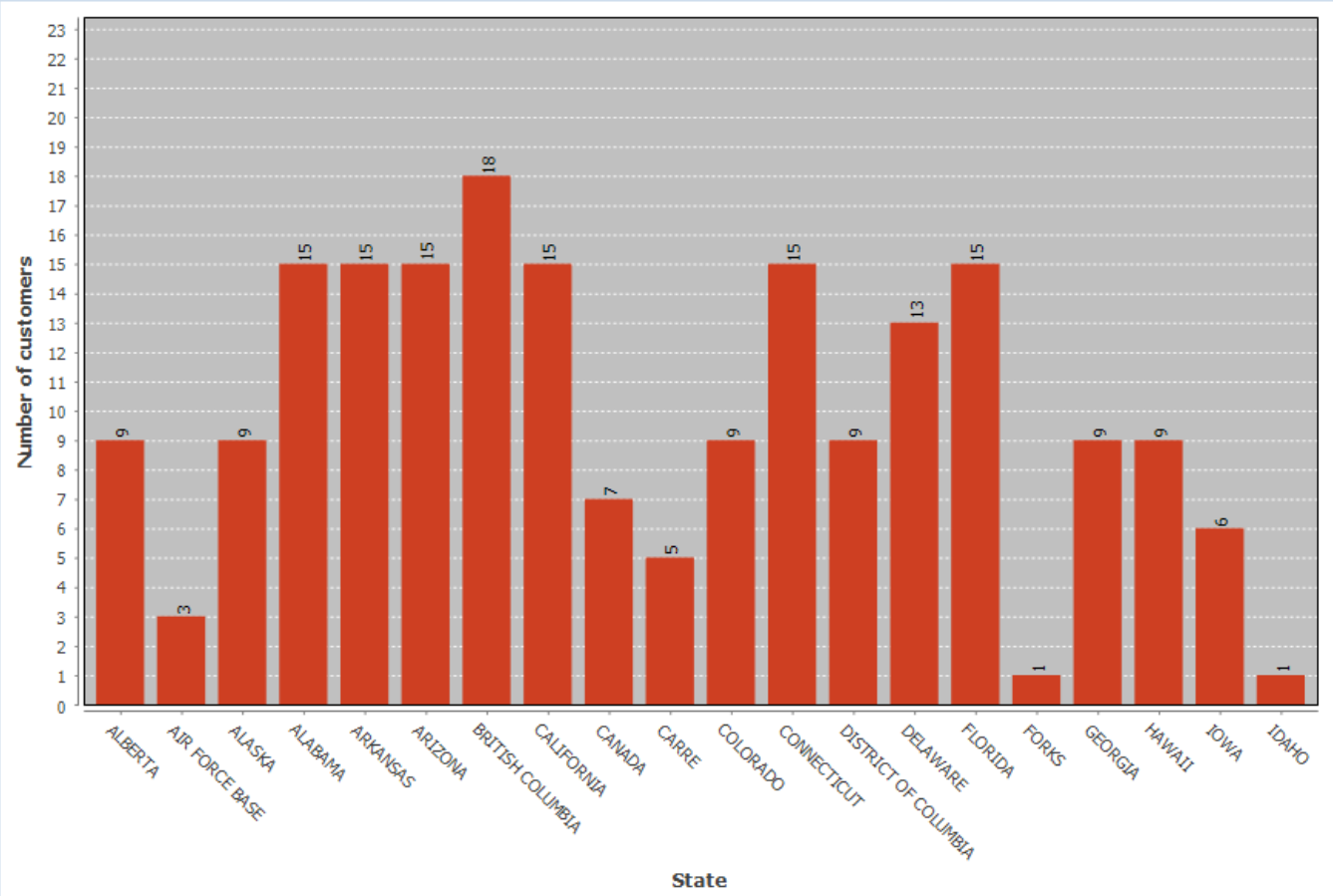
When a table that references the current table is chosen, the references panel will show all the rows that refer a selected primary key in the current table.



Managing Charts

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

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

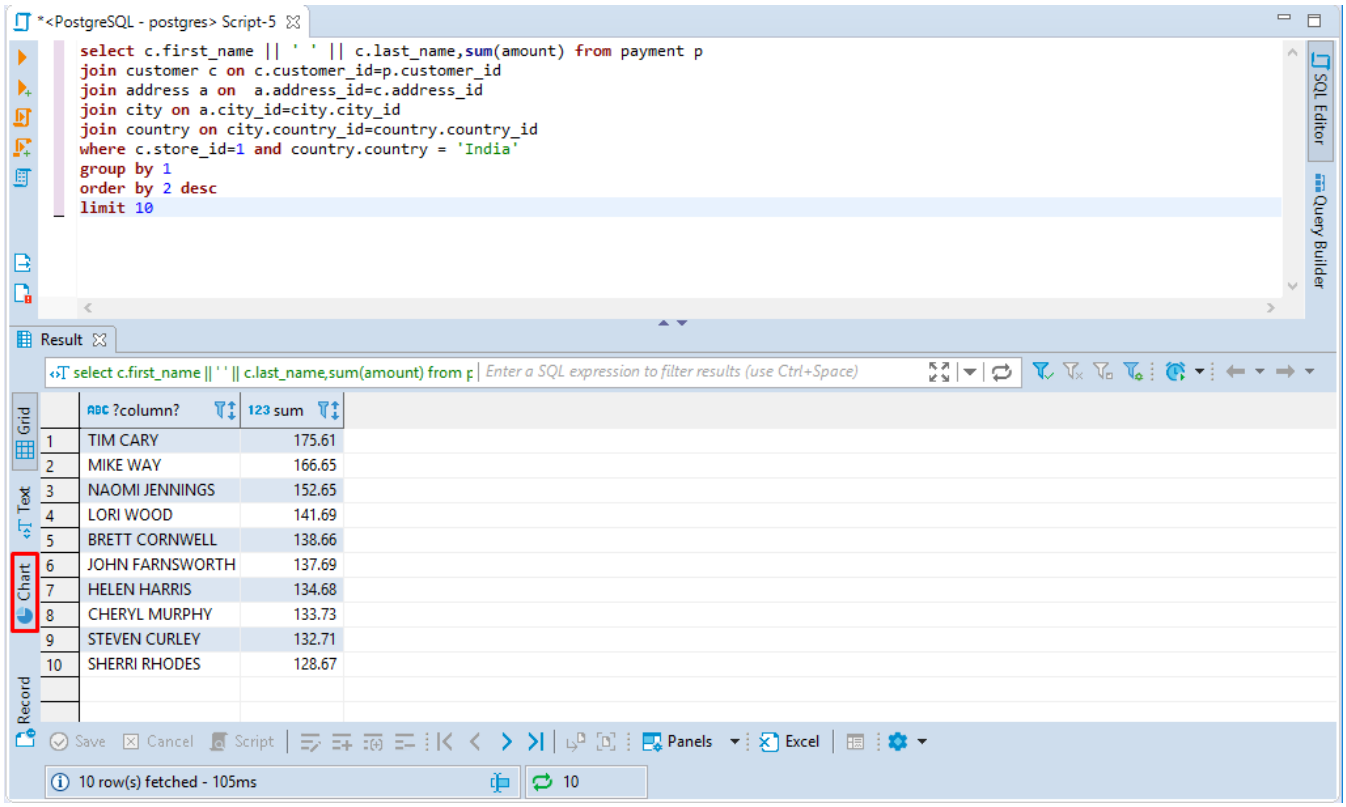


You can easily visualize your data by creating a chart bar both in **SQL Editor** and **Data Editor**.

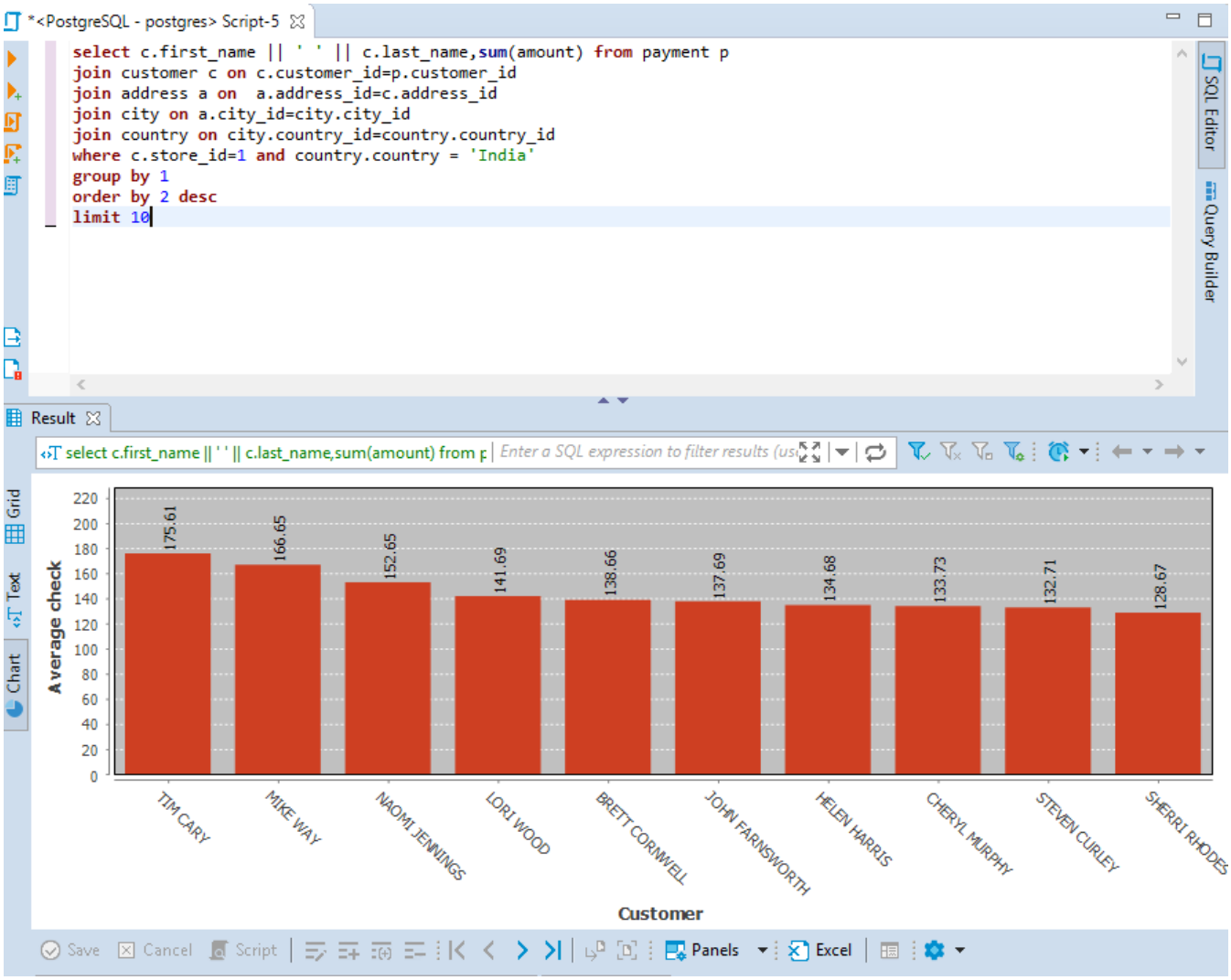
Creating Charts In SQL Editor

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

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



A bar chart will be created.

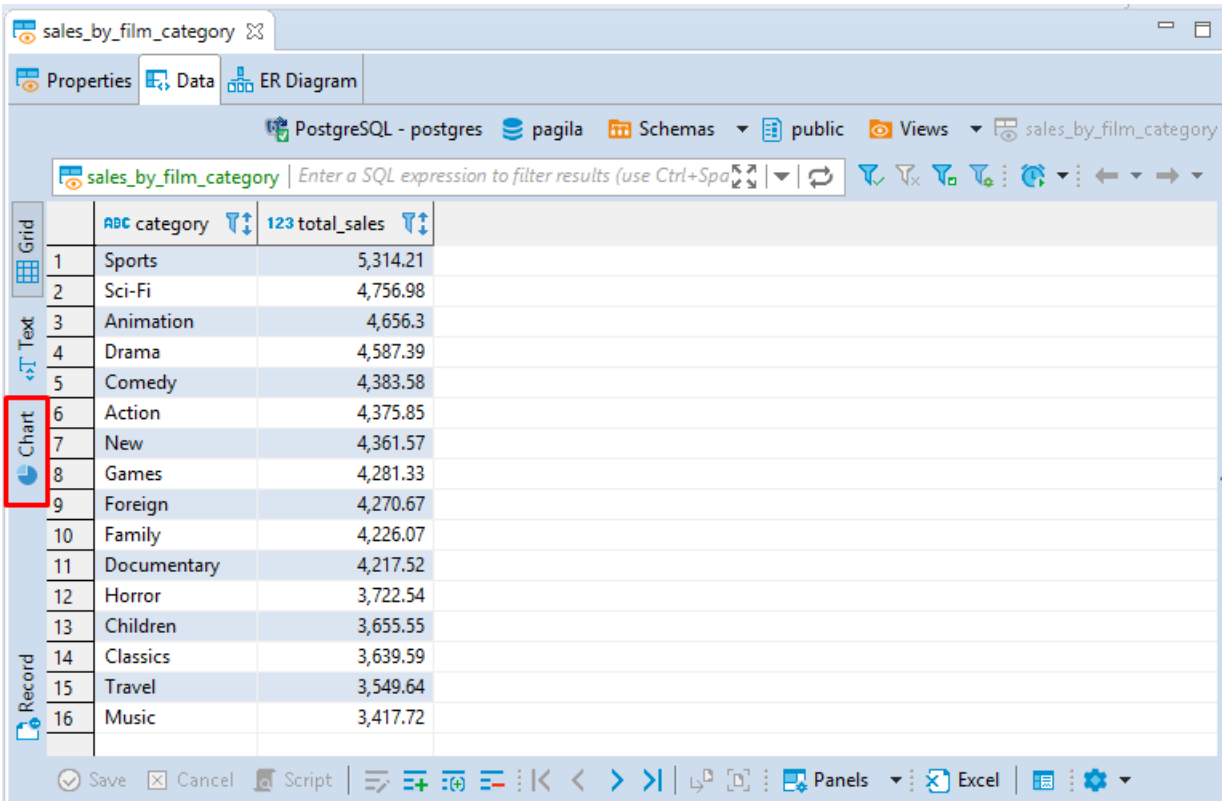


Creating Charts In Data Editor

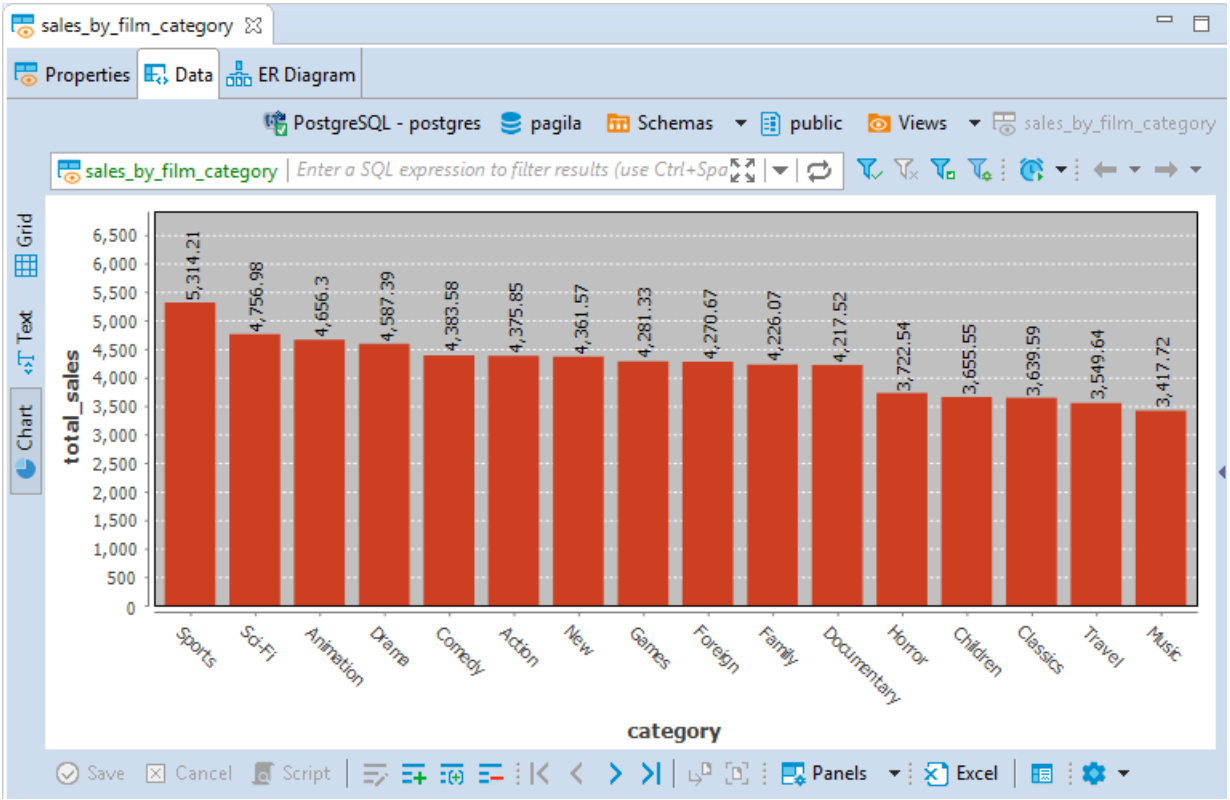
Charts can be very helpful for visualizing structured analytical data stored as **Views**, for example.

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

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



A bar chart will be created.

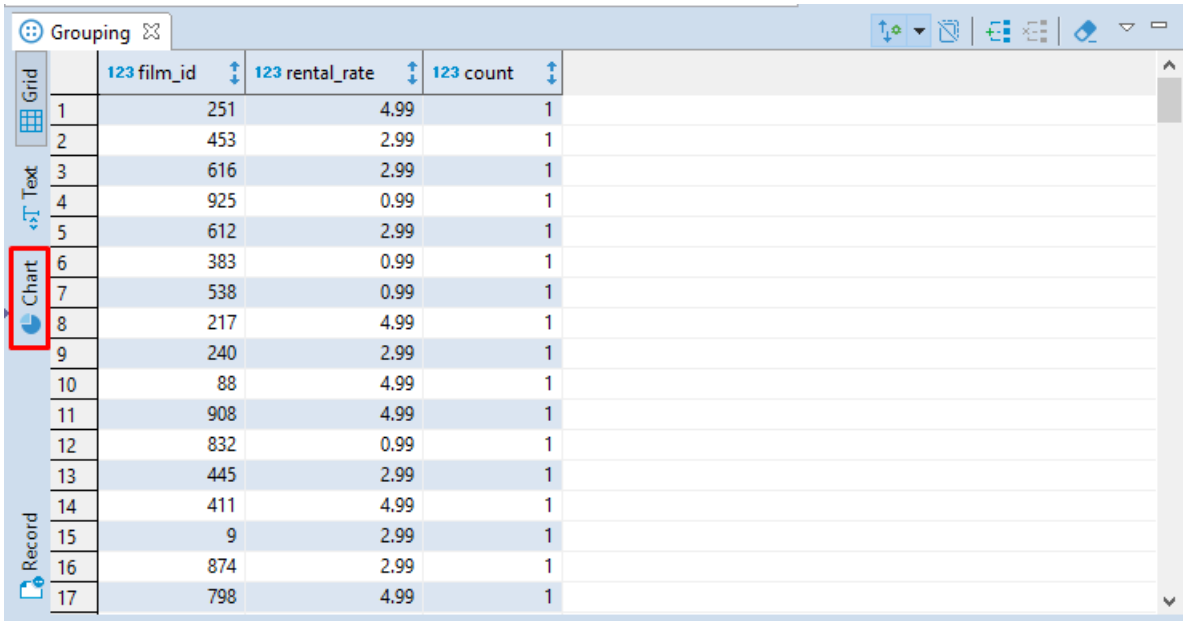


Note, that by default, the data for axis Y is taken from the first column of the table containing numeric values.

Creating Charts In Grouping Panel

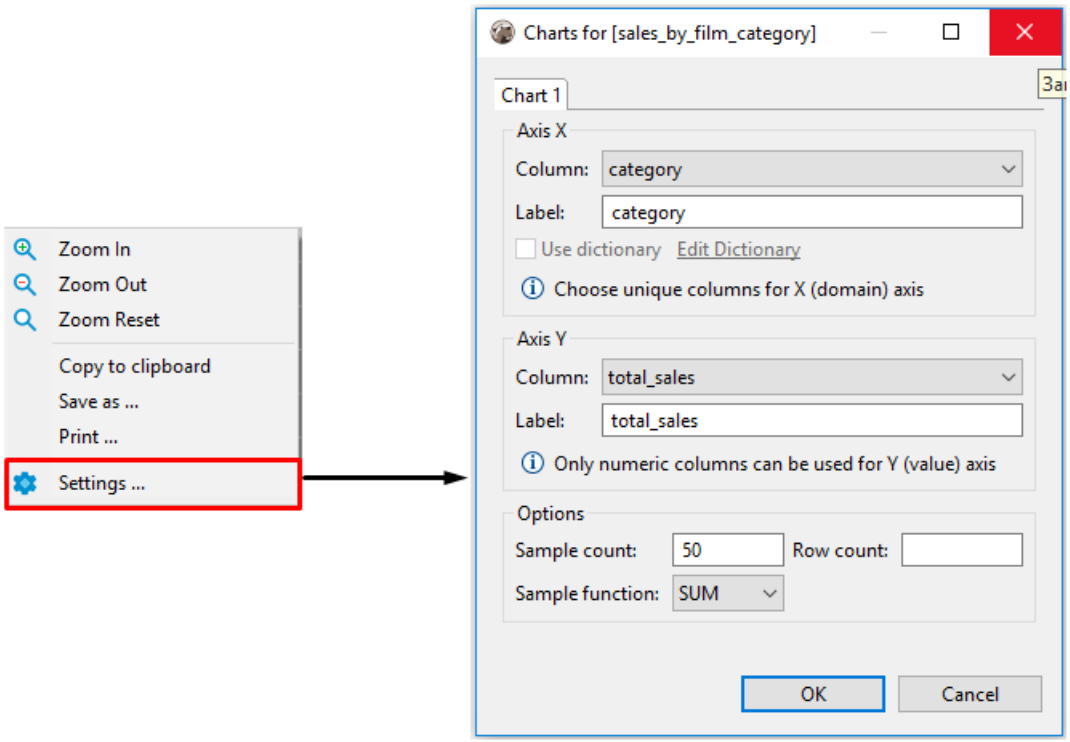
An analytical tool such as the Grouping Panel also supports the **Charts** feature. In a chart built for a table containing the grouping results for one or more columns of a data table, you can easily change axis X and axis Y source data by switching the columns in the **Charts Editor**.

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



Editing Chart Settings

To edit the 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:

Name	Description
AVG	Average value
SUM	The sum of all values
FIRST	The first value
LAST	The last value
COUNT	Total count of all values

Copying to clipboard

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

Exporting Charts

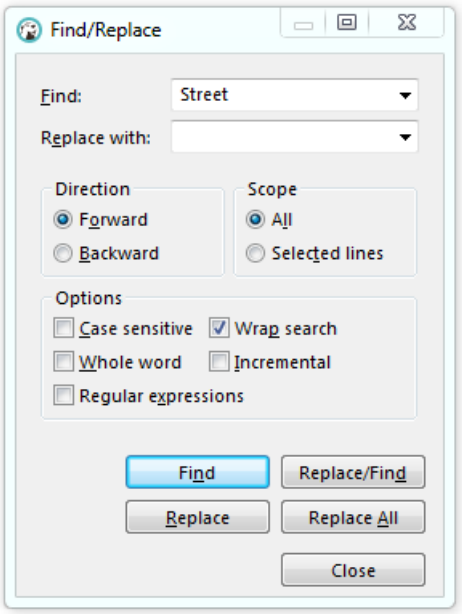
You can export a chart into PNG format by selecting the **Save as...** option in the chart's context menu.

Printing Charts

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

Data Search

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



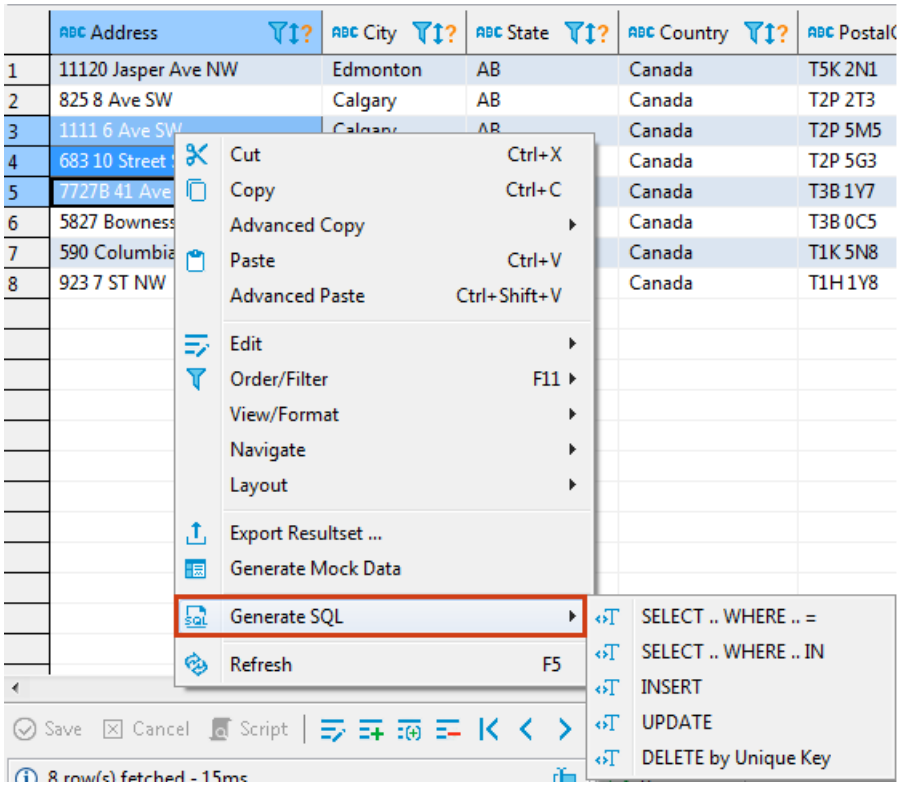
You can also use the Find and Replace feature.

NOTE: The system searches only in already fetched rows.

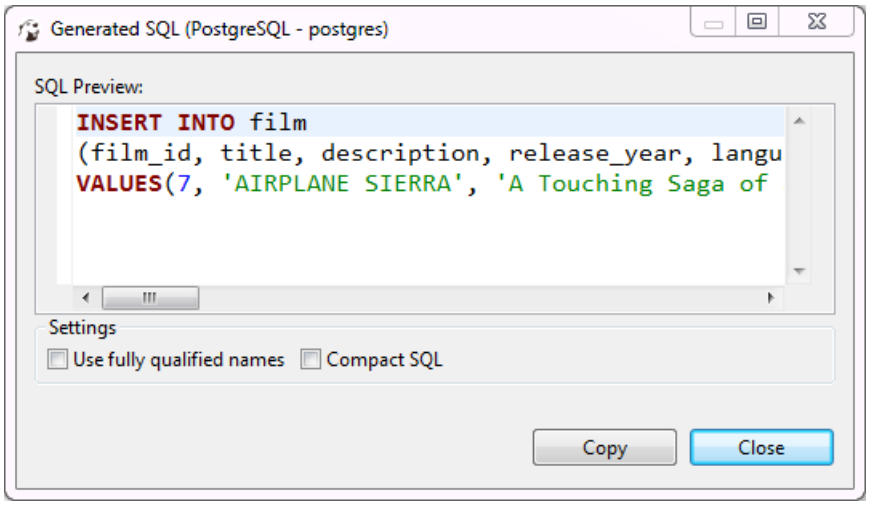
Setting	Description
Case sensitive	By default, the search is case insensitive.
Whole word	By default, the word specified in the search field can be found in the case when the word is part of another word. Enabling this setting will lead to a particular word search.
Regular expressions	Enable to use regular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

SQL Generation

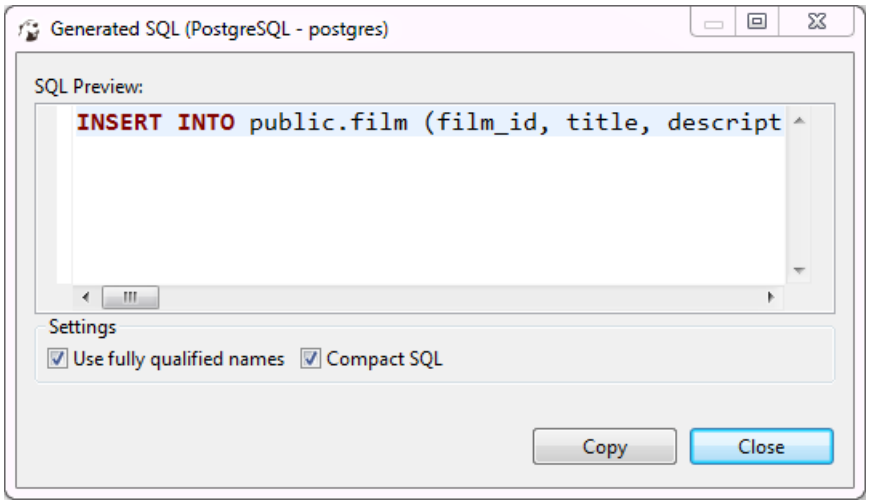
You can generate SQL statements (SELECT/INSERT/UPDATE/DELETE) based on selected rows. To generate SQL, right-click the selected rows and click **Generate SQL** and then one of the SQL commands on the context menu:



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



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


Generated SQL (PostgreSQL - postgres)
SQL Preview:
INSERT INTO public.film (film_id, title, descript
Settings
Use fully qualified names
Compact SQL
Copy
Close

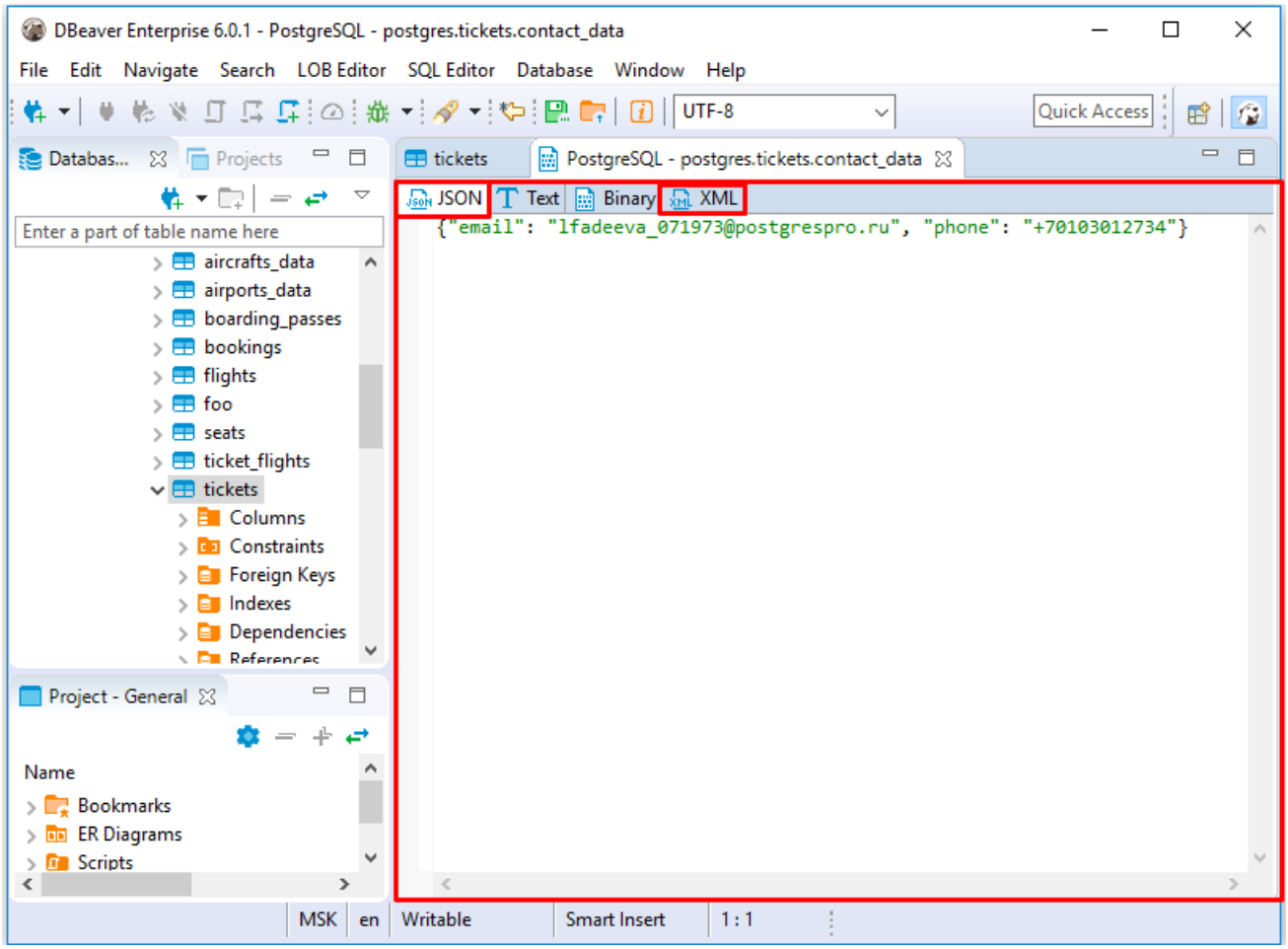
Working with XML and JSON

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

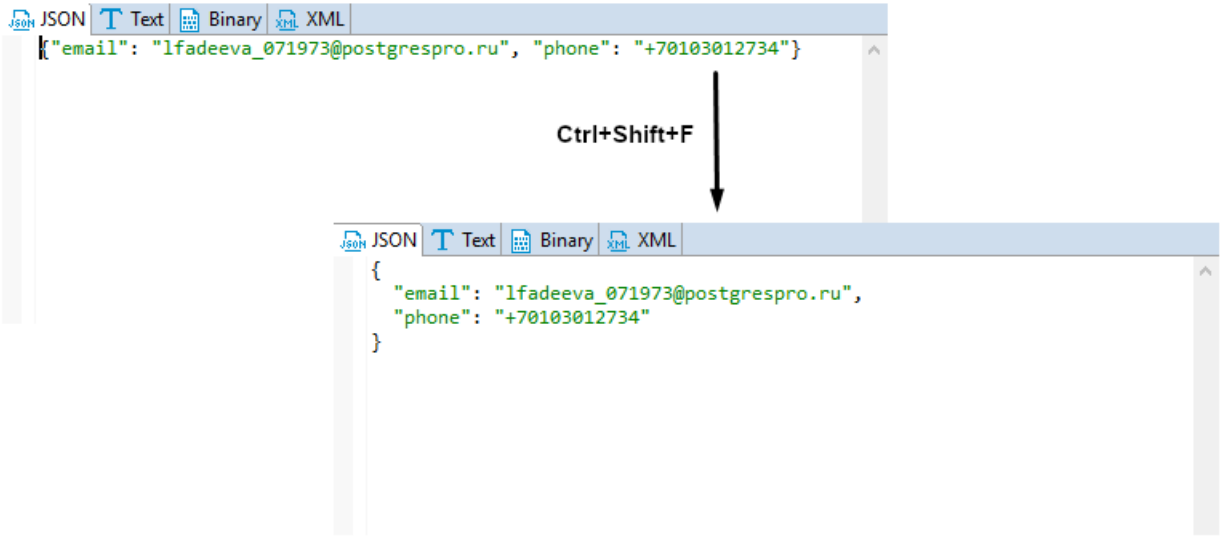
In the Data Editor, you can edit XML/JSON data right in the table cells/ However, a huge amount of data may require a larger editor so you might want to save XML/JSON scripts to a local file or upload this type of data from a local file.

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

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

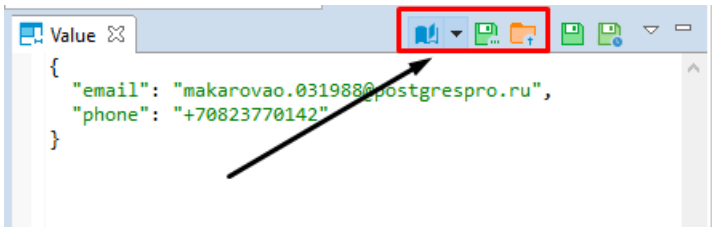


To auto-format XML/JSON script press `Ctrl`+`Shift`+`F` keyboard buttons.



Use `Ctrl`+`S` keyboard shortcut to save the changes made.

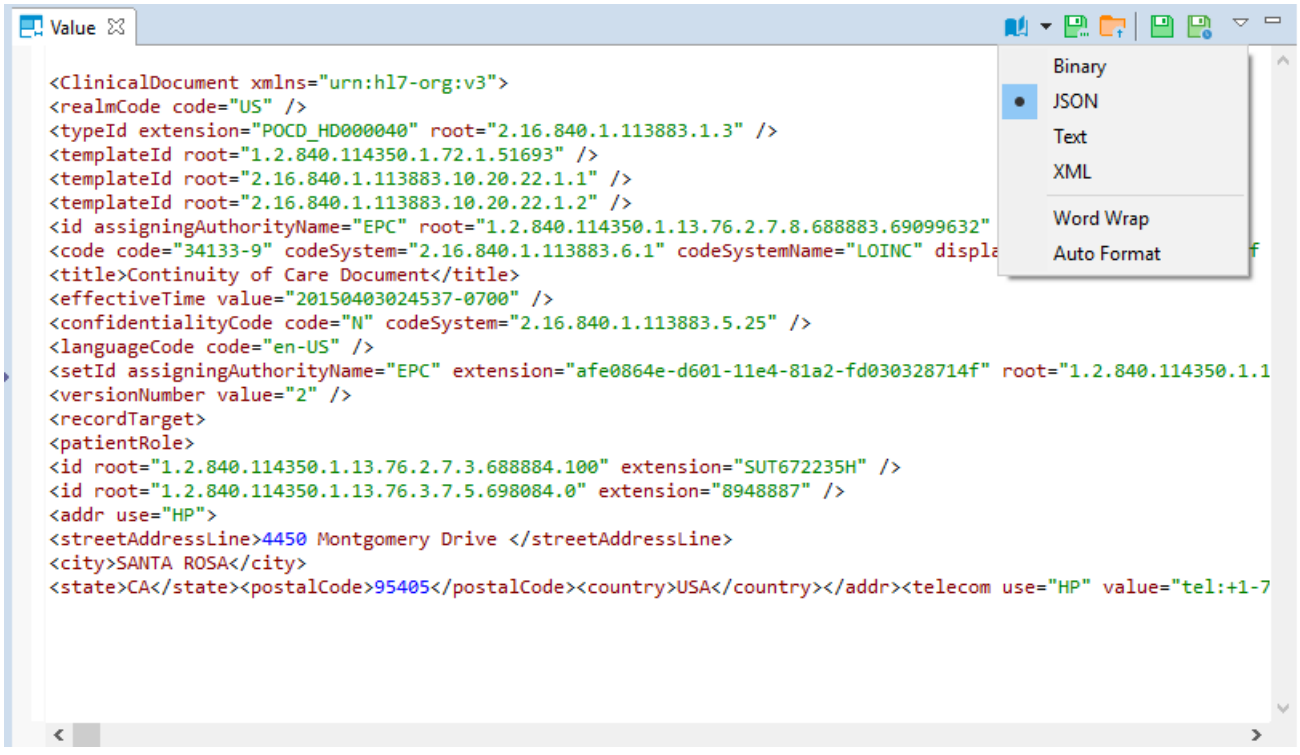
You can also edit XML/JSON content, save it locally and upload it from a local file with the help of **Value** panel toolbar.



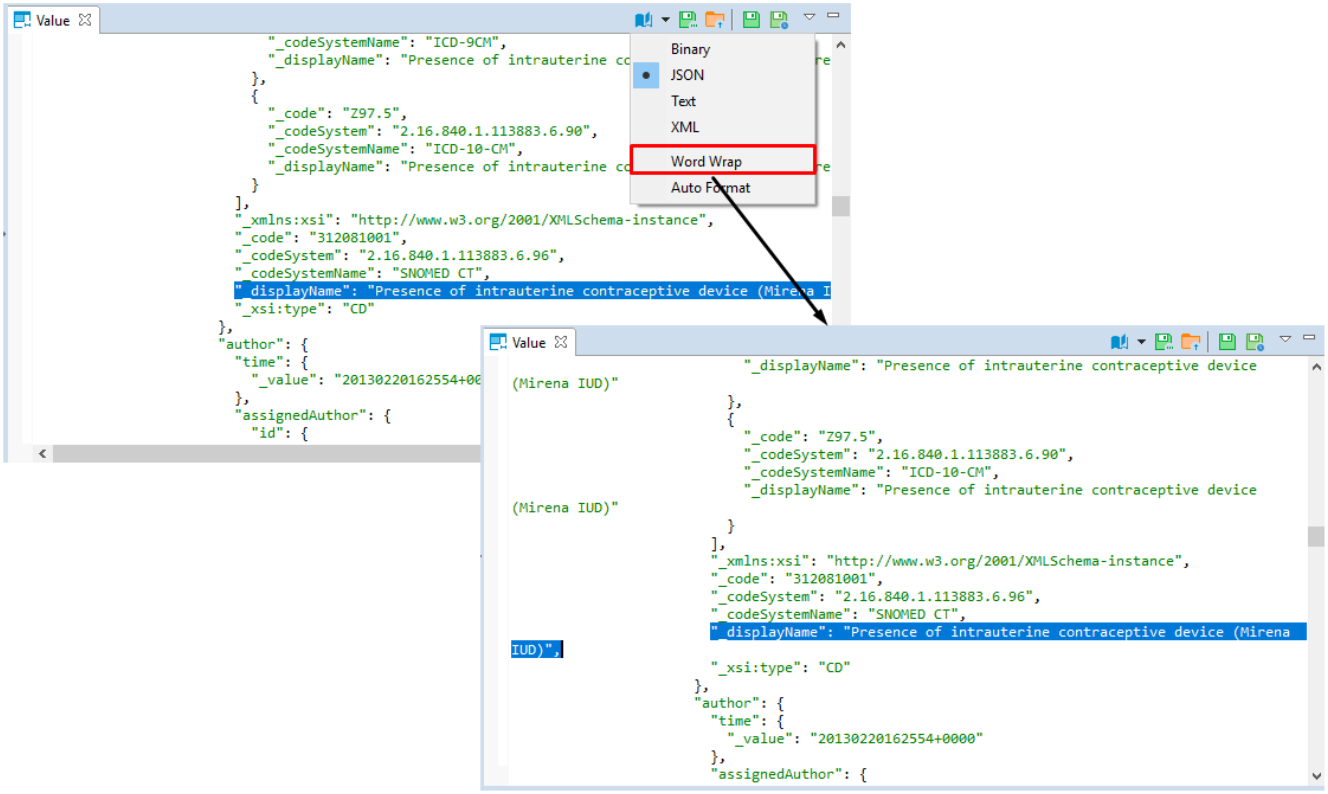
To upload data from a local file, press the **Load from file...** button .

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

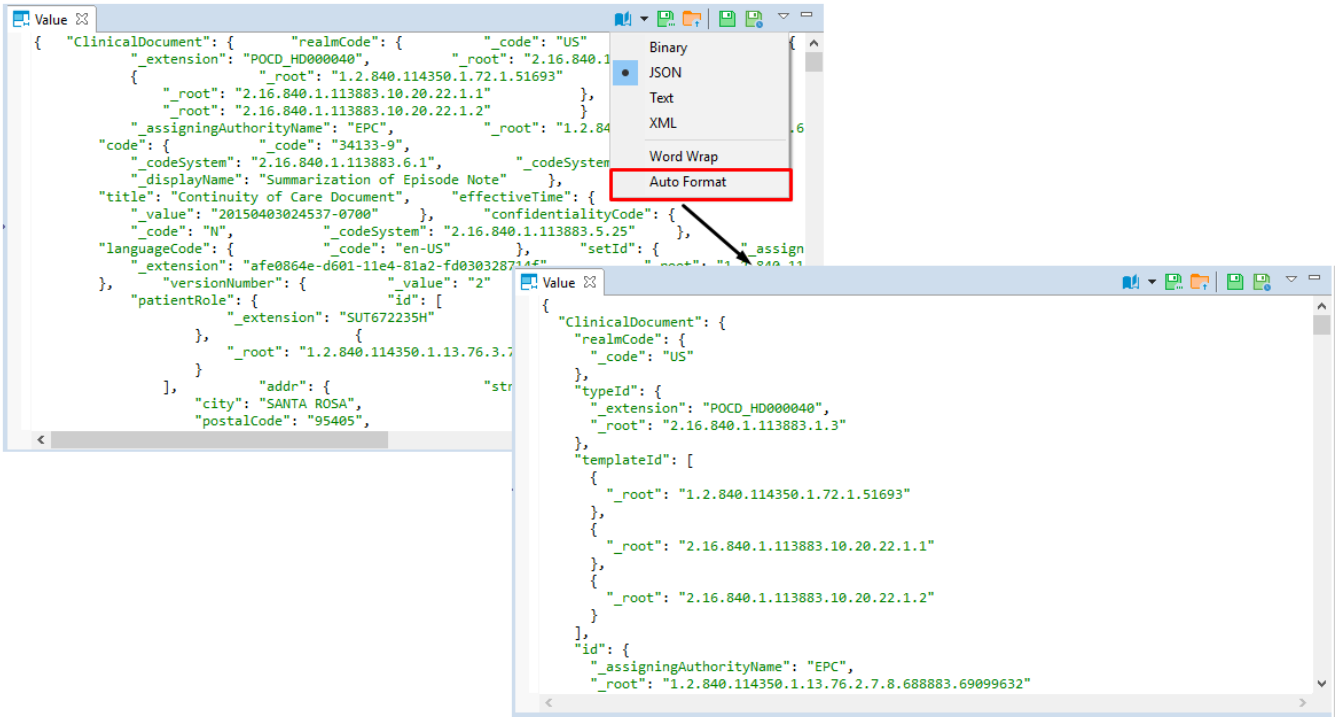
To switch between the formats, press the **Content viewer settings** button and select the format.



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



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

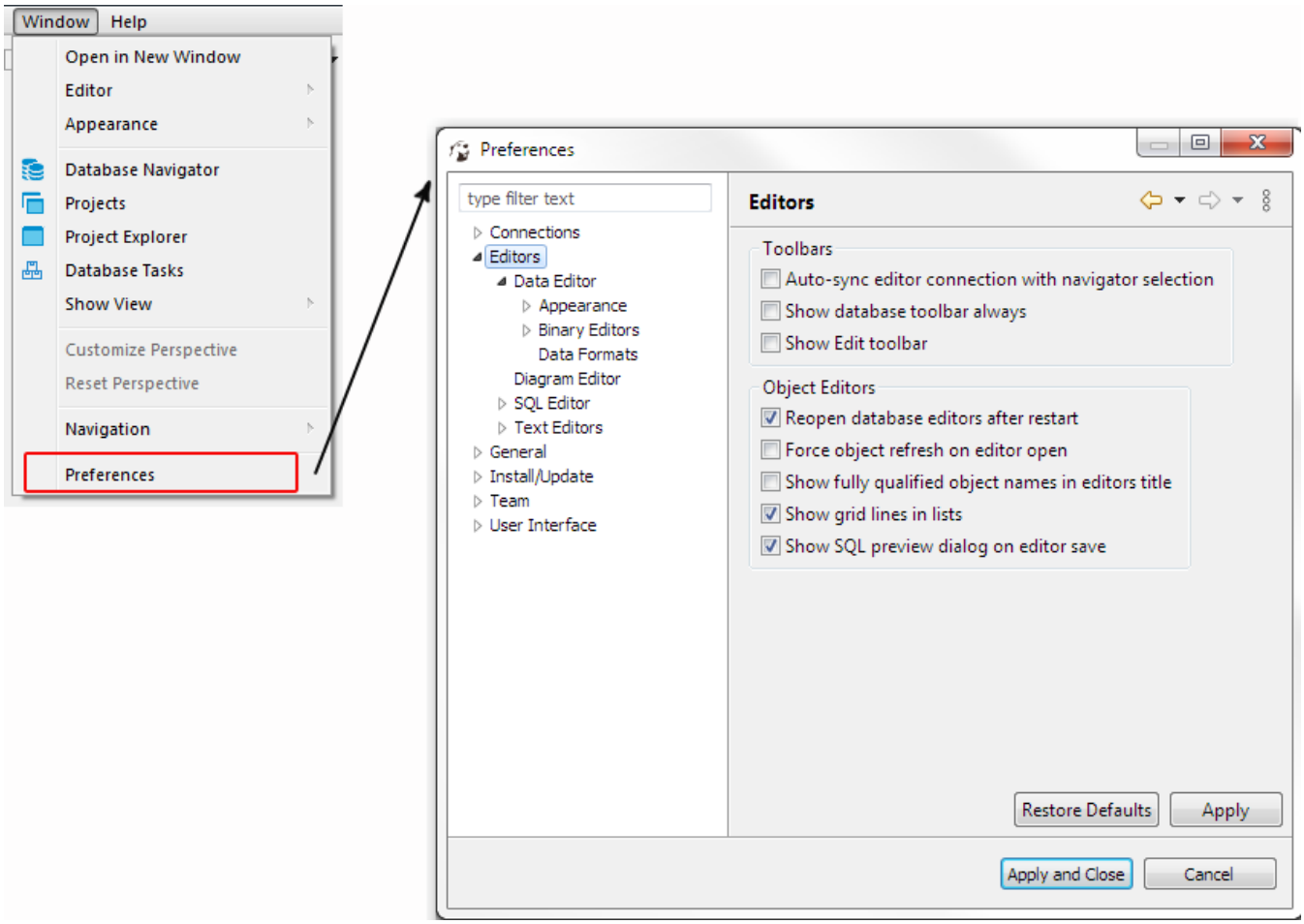


To learn more about **Value** panel, see Panels.

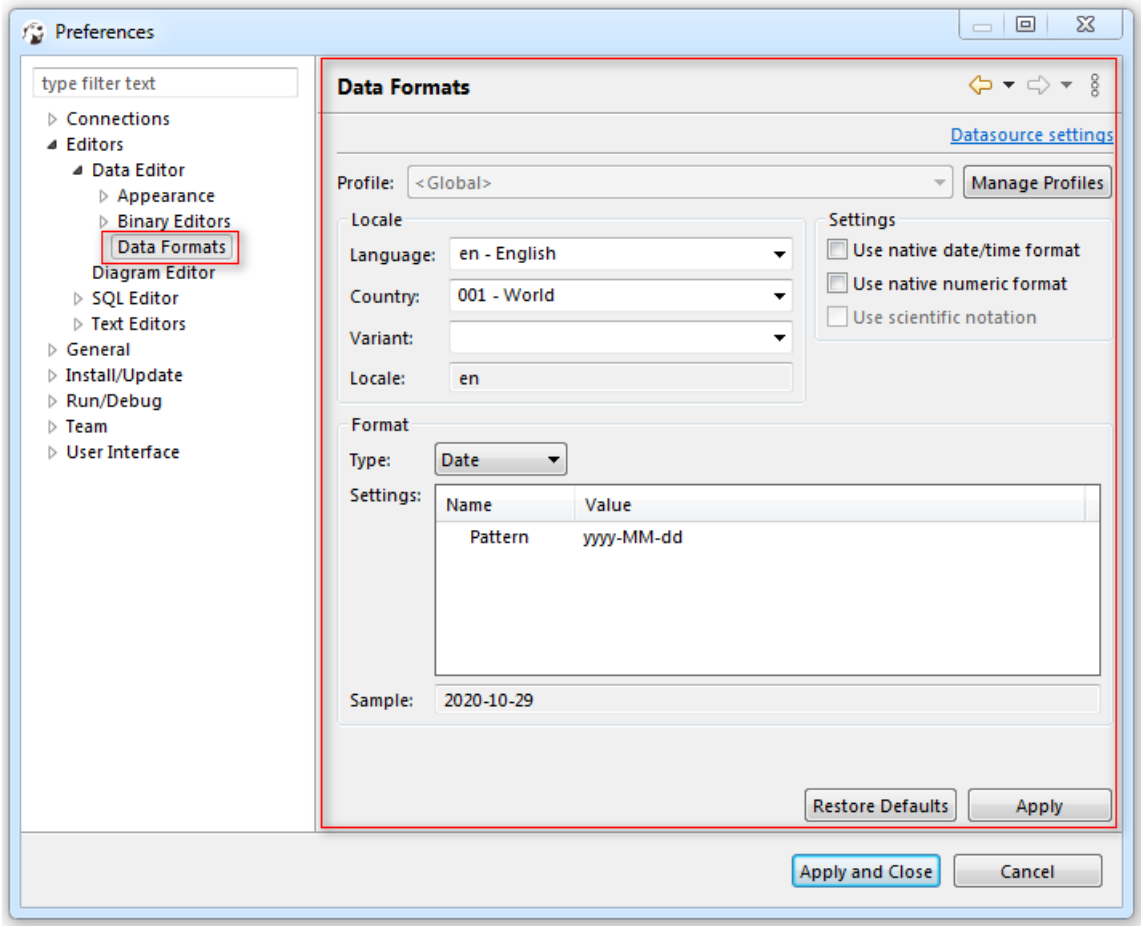
Managing Data Formats

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

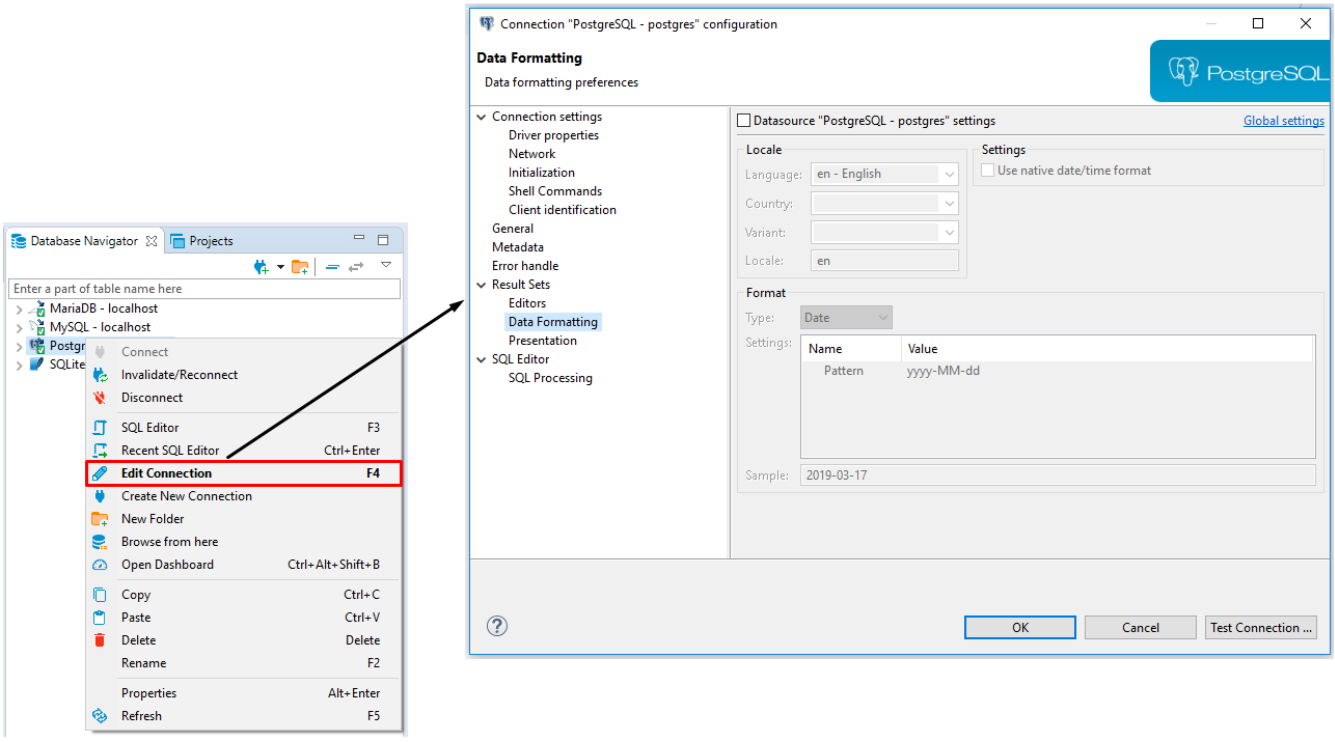
To change the data format settings use the option Window -> Preferences in main menu.



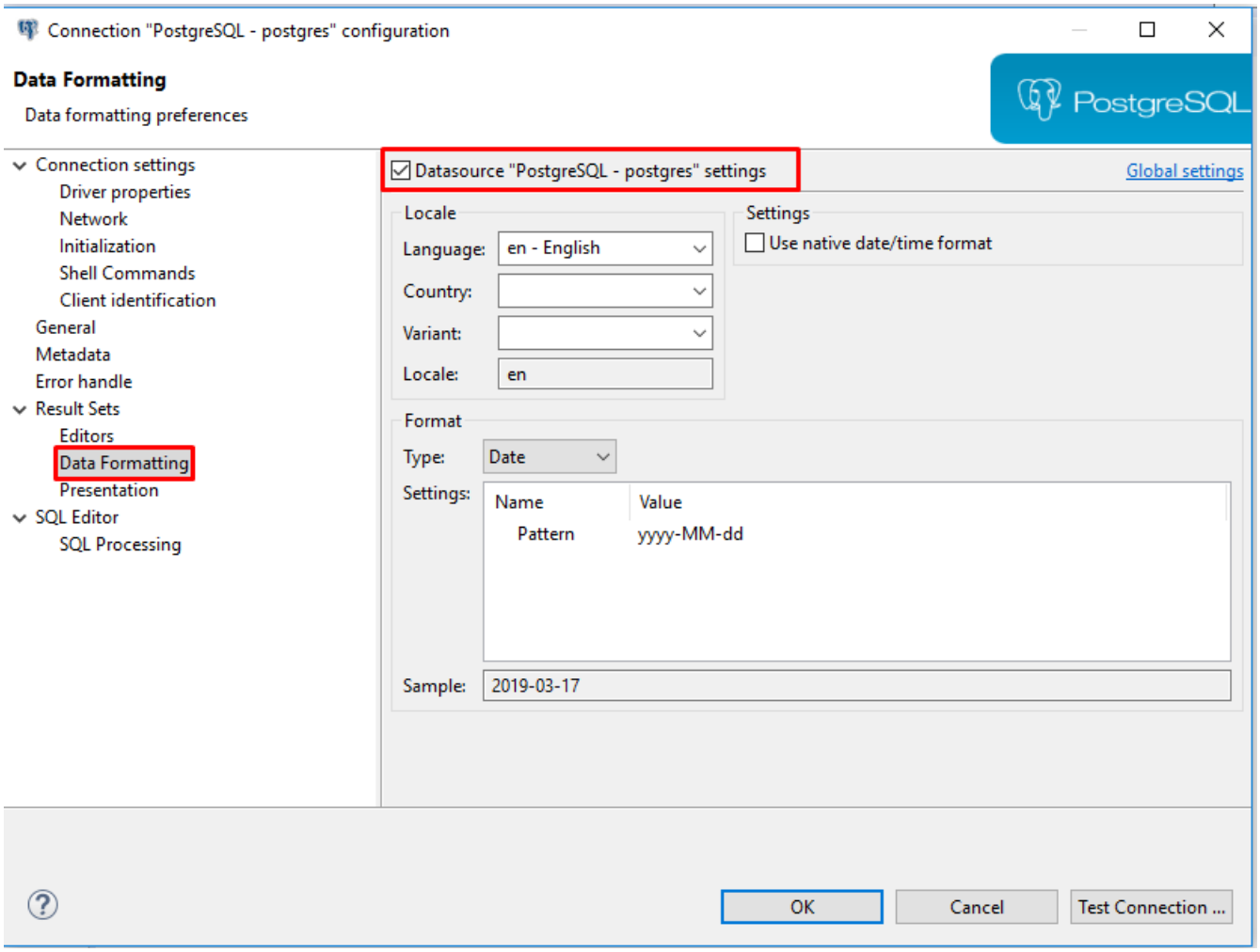
In the Preferences dialog box go to Editors -> Data Editor -> Data Formats.



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



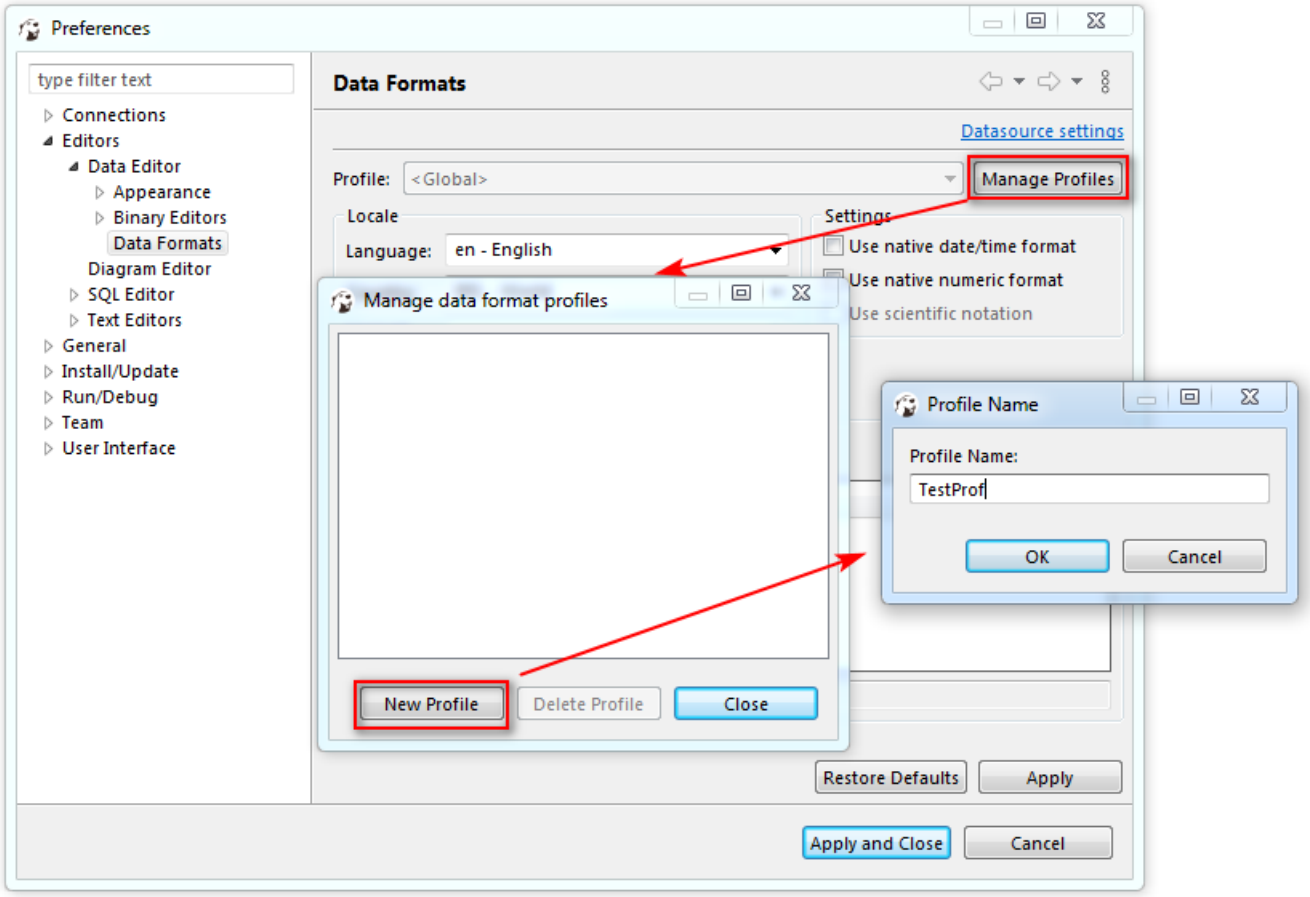
In the right area of the opened **Data formatting preferences** dialog window go to Data editor -> Data Formats and select the **Datasource settings** check box in the left area to customize the data format settings.



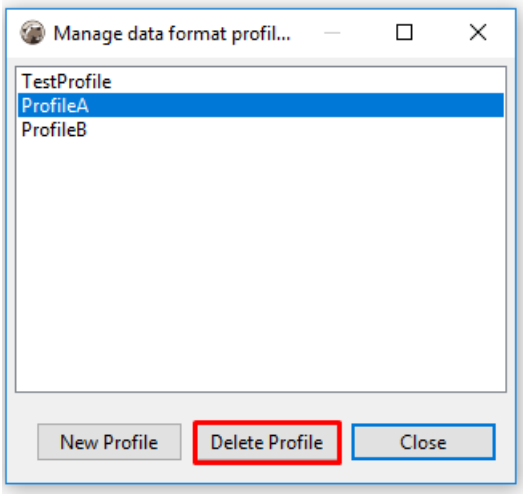
Data Format Profiles

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

To create a data format profile press the **Manage Profiles** button. In the opened dialog window press button **New Profile**, define the name and press **Create**.



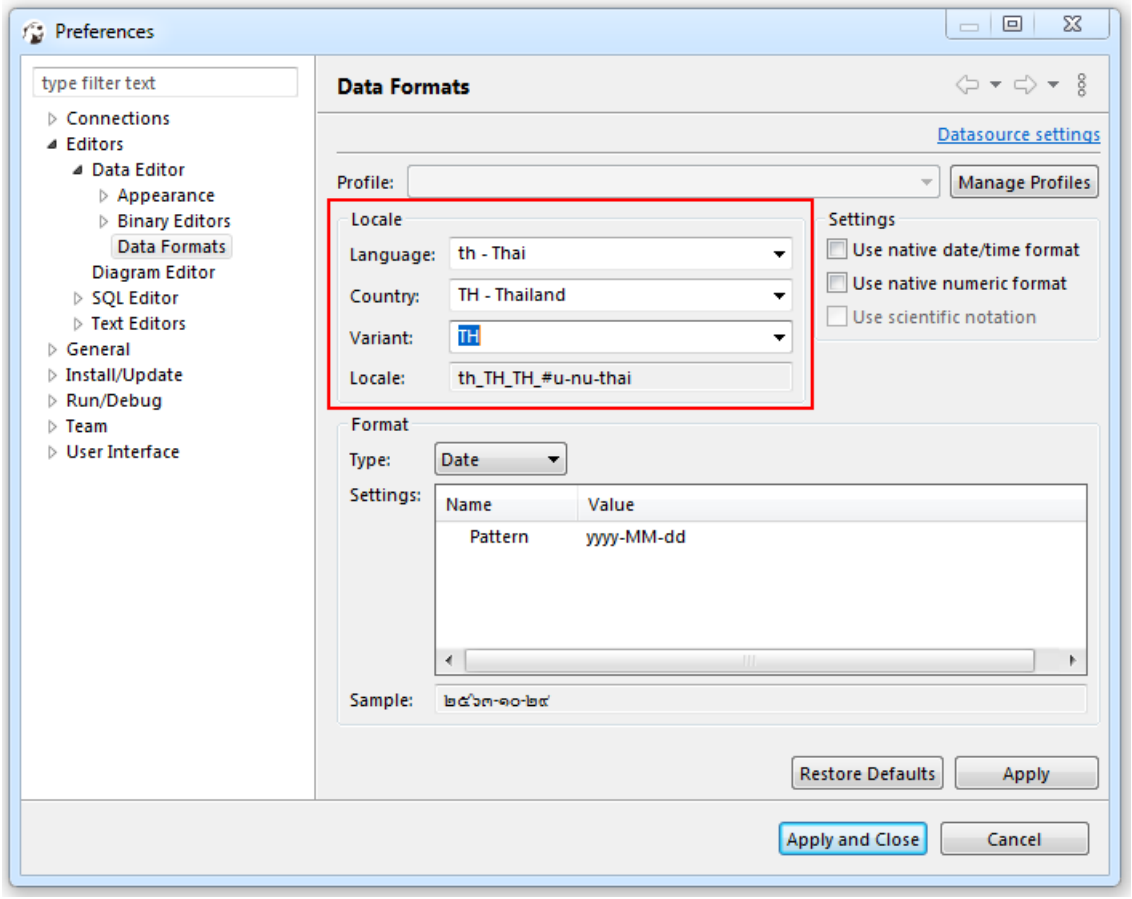
To delete a data format profile press the **Manage Profiles** button, then in the opened dialog window select the profile you want to delete and press the button **Delete Profile**.



Changing Data Formats

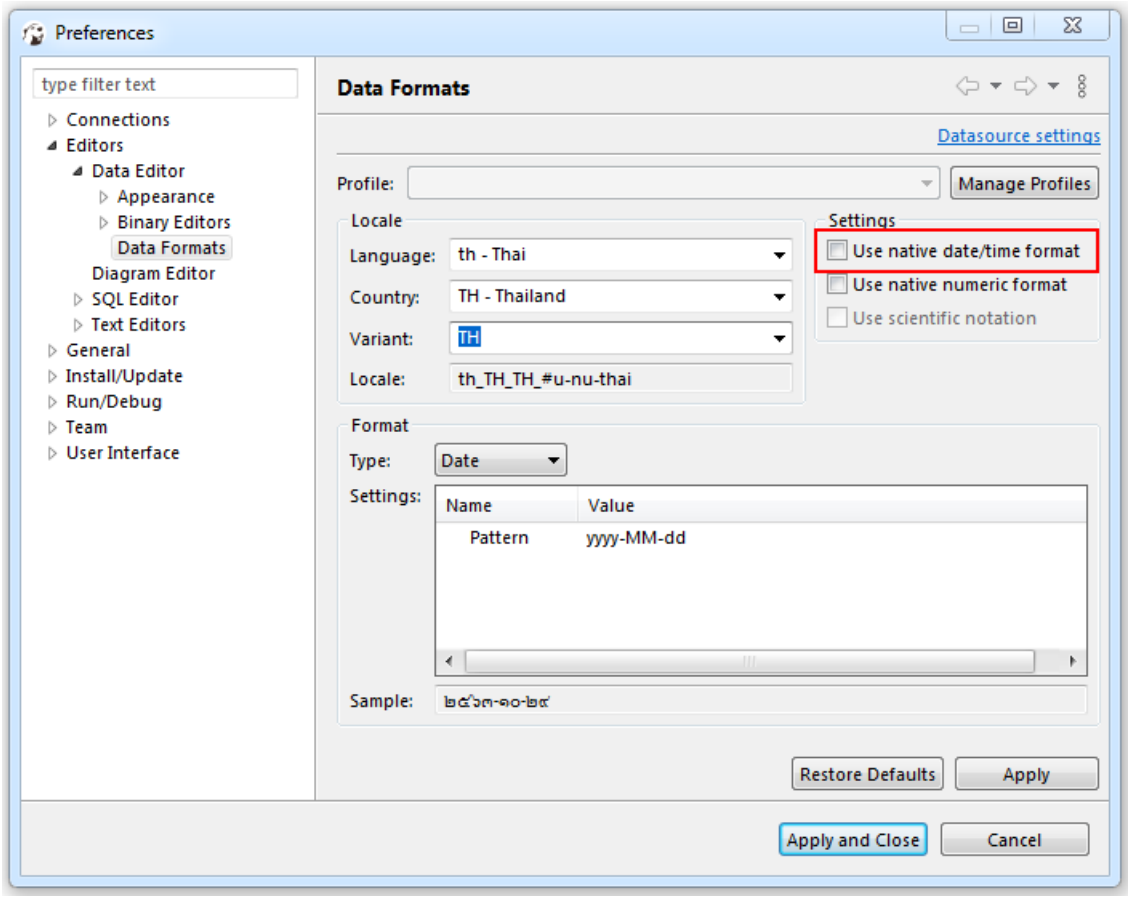
The following groups of data format settings can be adjusted:

Locale



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

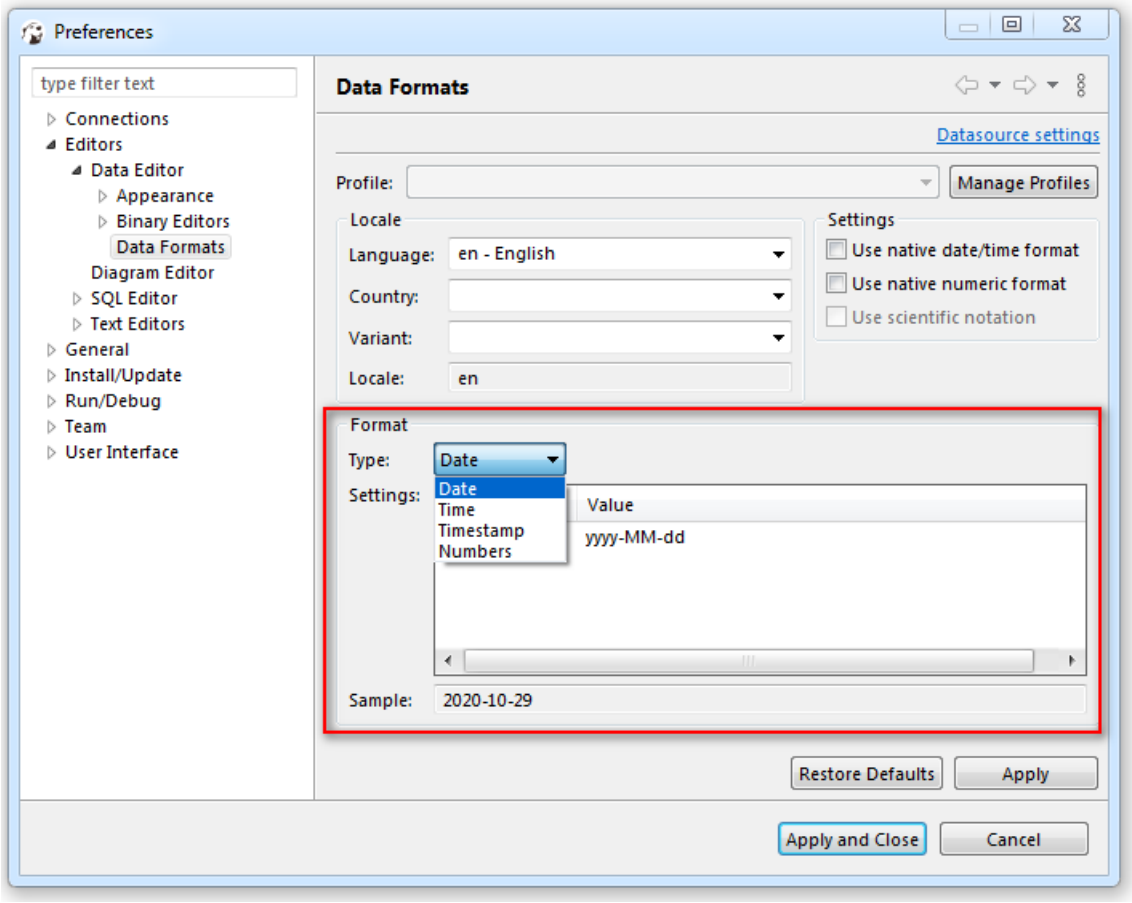
Native Date/Time Mode



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

You can change the format of the following data types:

Data Type Format



The format of the following data types can be customized:

- Date
- Time
- Timestamp
- Numbers

Date

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

Pattern	Description
yyyy or y	Year of era (4 digits)
yy	Year of era (2 last digits)
YYYY	Week year
M	Month in year without leading zeros
MM	Month in year
MMM	Short month name in year
MMMM	Month name in year
D	Day in year
d	Day in month without leading zeros
dd	Day in month
E	Day name in week
G	Era designator
z	General time zone

Pattern	Description
Z	RFC 822 time zone

Time

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

Pattern	Description
H	Hour in day (0-23) without leading zeros
h	Hour in day in am/pm (1-12) without leading zeros
HH	Hour in day (0-23)
hh	Hour in day in am/pm (1-12)
a	Am/pm marker
m	Minute in hour without leading zeros
mm	Minute in hour
s	Second in minute without leading zeros
ss	Second in minute
S	Millisecond
ffffff	Microseconds

Timestamp

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

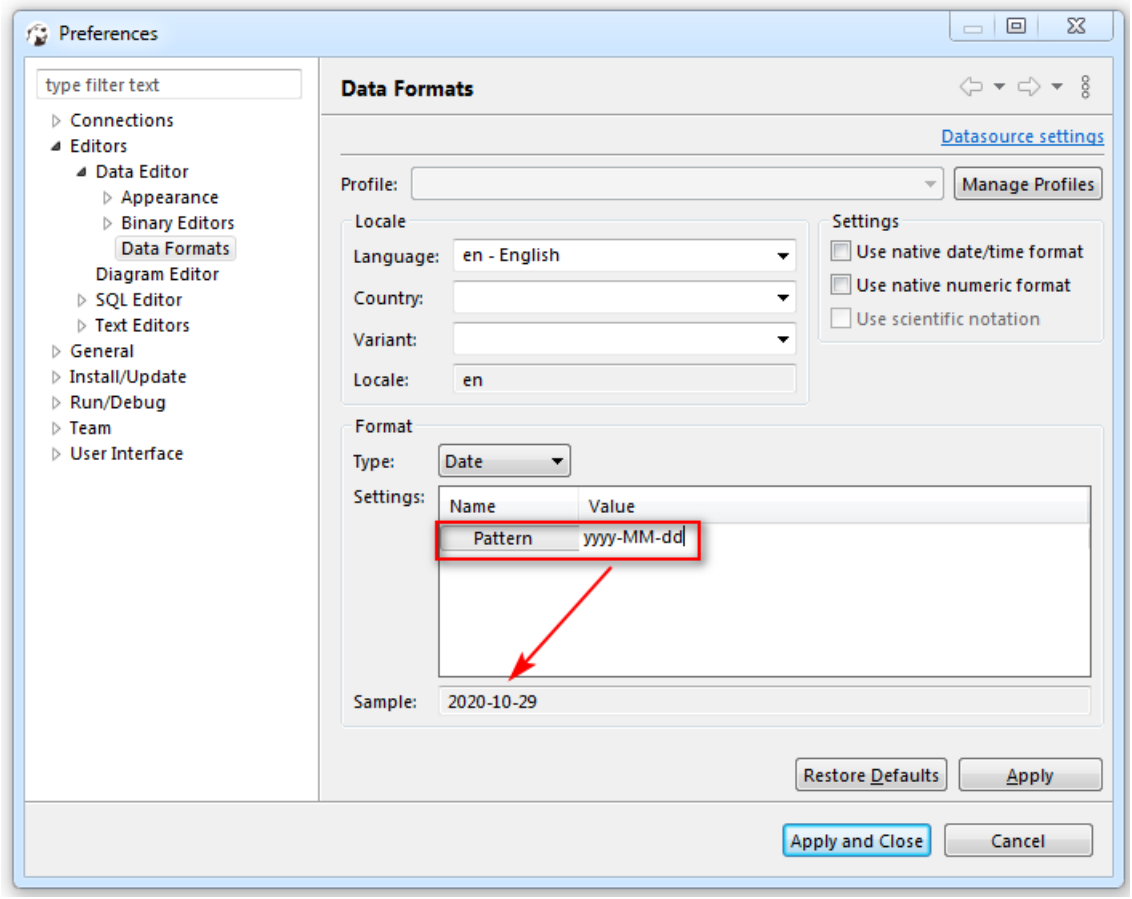
Link on Java date pattern documentation

Numbers

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

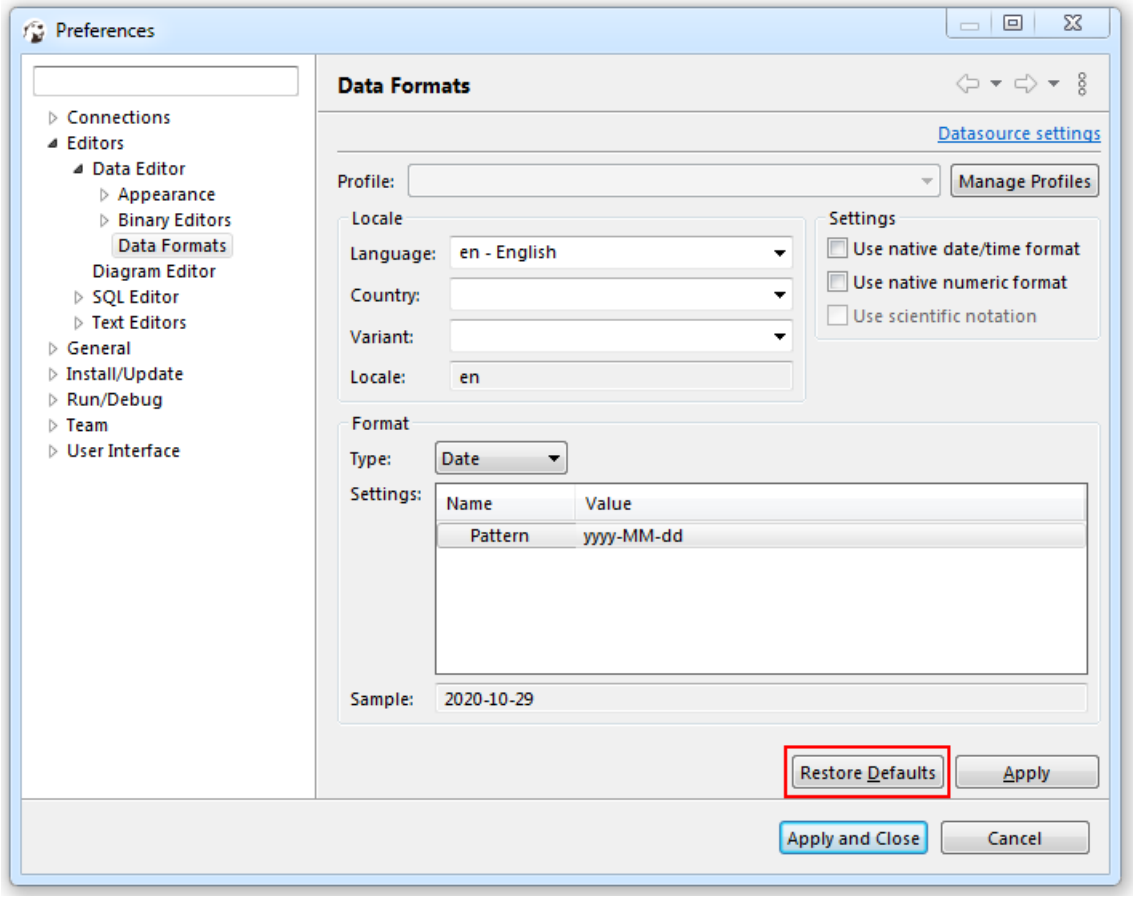
- **Use Grouping** - Long numbers can be hard to read if they have too many digits. For example, the factorial of 30 is 33 digits long! Select this check-box to enable Grouping mode, in which digits are displayed in clumps of 3 or 4 (depending on the current radix) separated by commas.
- **Maximum integer digits** - Defines the maximum number of digits to the left of the decimal point.
- **Minimum integer digits** - Defines the minimum number of digits to the left of the decimal point.
- **Maximum fraction digits** - Defines the maximum number of digits to the right of the decimal point.
- **Minimum fraction digits** - Defines the minimum number of digits to the right of the decimal point.
- **Use data type scale for fraction digits** - Some numeric columns or parameters may have a predefined scale, that is the maximum number of digits to the right of the decimal point. Select this check-box if you want the predefined precision to be used.
- **Rounding mode** - Specifies a rounding behavior for numerical operations capable of discarding precision. Each rounding mode indicates how the least significant returned digit of a rounded result is to be calculated. To learn more, please refer to Oracle documentation.

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



Restoring Default Data Formats

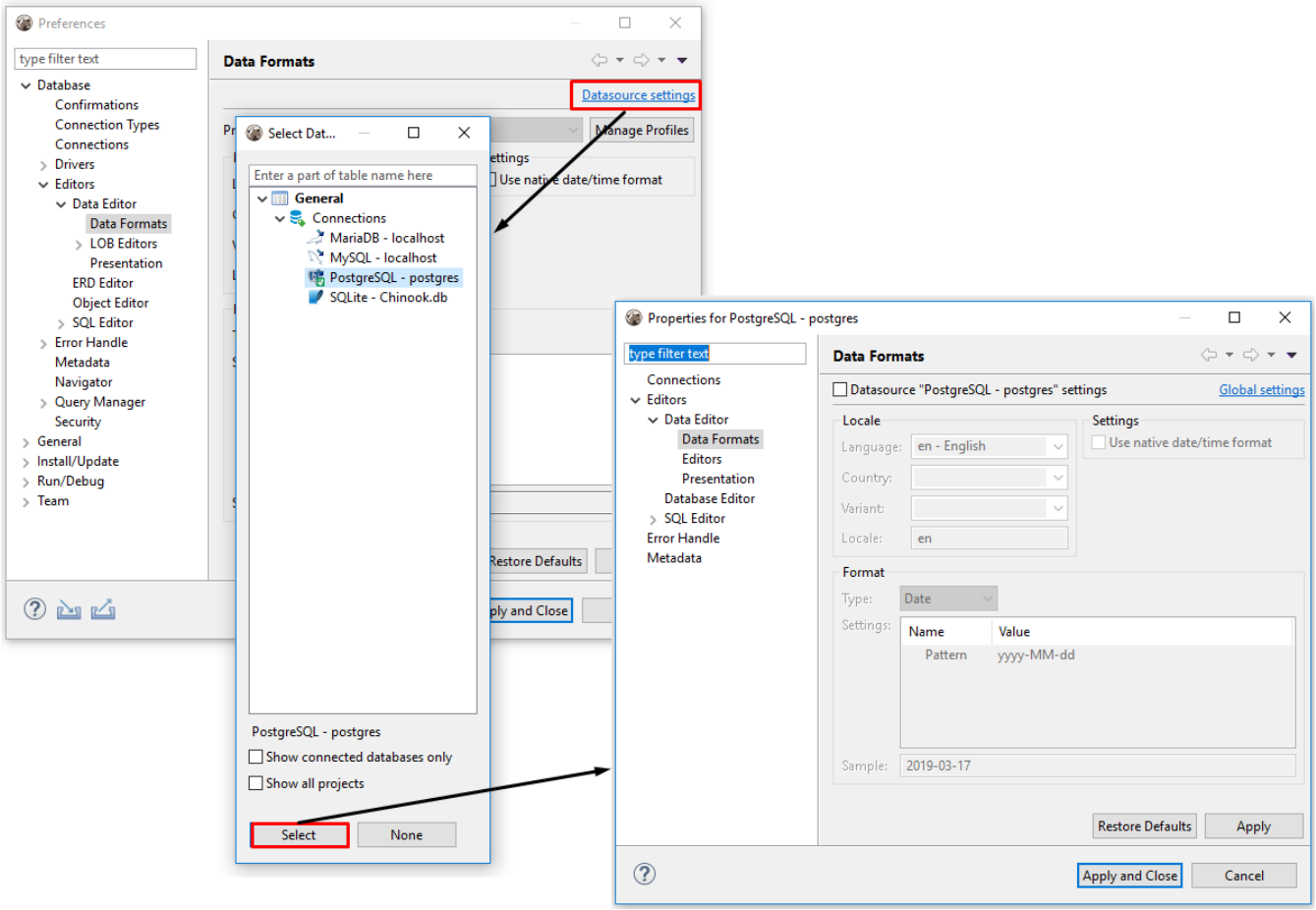
To restore the default data format settings, press the **Restore Defaults** button.



Datasource Settings

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

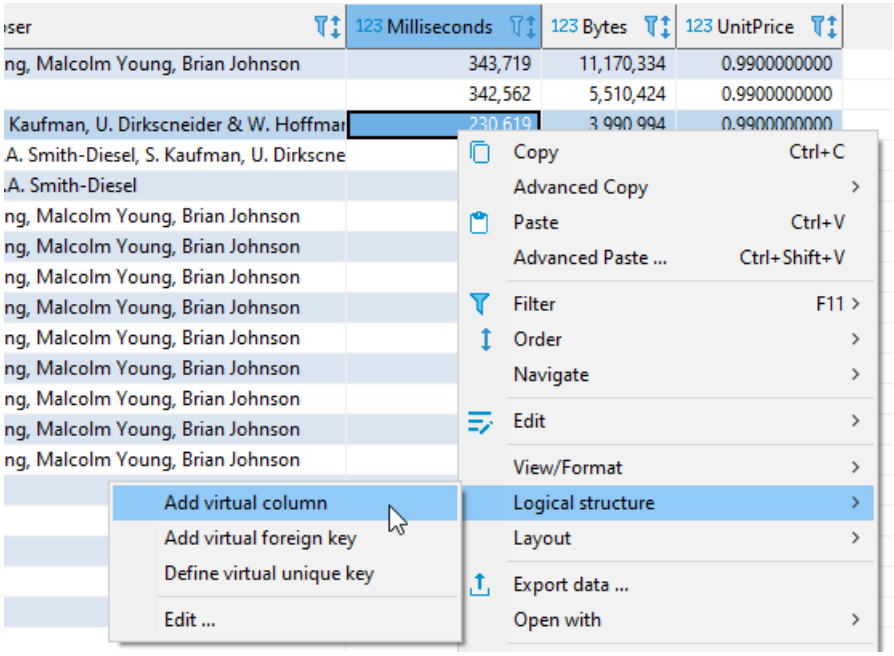
dialog box.



To save changes made press **Apply**.

Virtual column expressions

Expression language



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

Column values

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

Standard functions

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

math

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

geo

Function	Parameters	Description
wktPoint	(longitude, latitude)	Produces WKT (geometry) point out of two coordinates. Default SRID is 4326.
wktPoint	(longitude, latitude, srid)	Produces WKT (geometry) point out of two coordinates and SRID

content

If you have JSON or XML columns in your table, you can add a virtual column with an expression for these columns.

Use content.json()[parameter1][parameter2] pattern to create expression for JSON column.

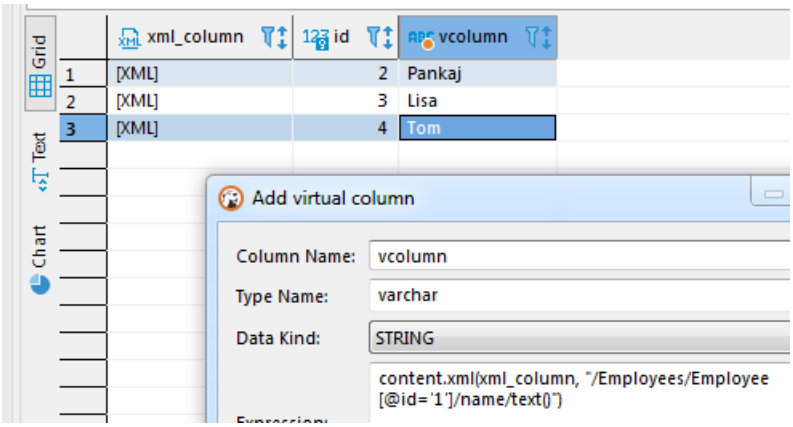
Expression example: `content.json(column1)['glossary']['GlossDiv']['title']`

You can read more about JSON parameters in the link on the top.

Use content.xml(columnName, "expression") or content.xml(columnName, "returnType", "expression") patterns to create expression for XML column. The quotation is important for parsing processes. XML expression can return types: string, number, boolean, node, nodeset - all these types can be used (!in quotes) for return data type clarification. content.xml(columnName, "expression") returns string by default

Expression example: `content.xml(column1, "nodeset", "/Employees/Employee[gender='Female']/name/text()")`

You can read more about XPath here: https://en.wikipedia.org/wiki/XPath



SQL Editor

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

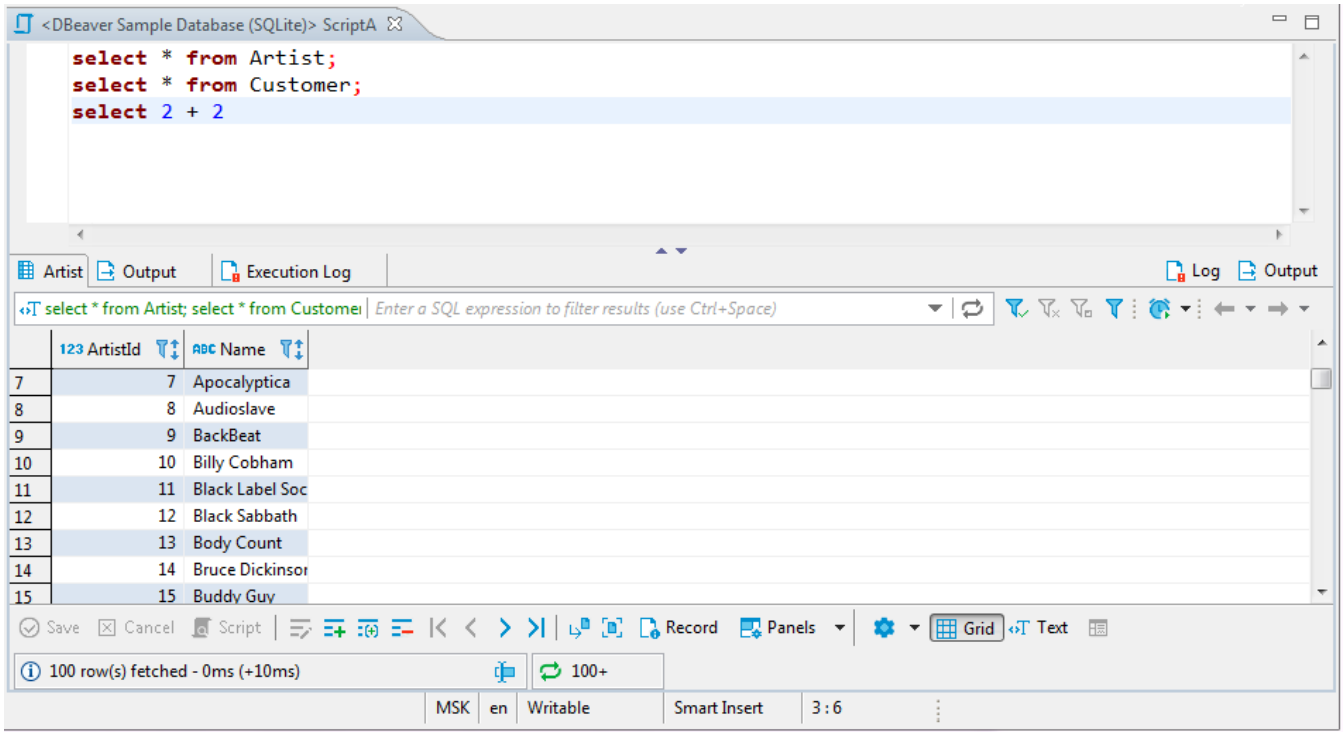
- Click this connection in the Database Navigator view and press `F3` or click **SQL Editor -> SQL Editor** on the main menu. Alternatively, click **SQL Editor** on the context menu of this connection. DBeaver opens the Choose SQL script editor with saved SQL scripts linked to this connection. Click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

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

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

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

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



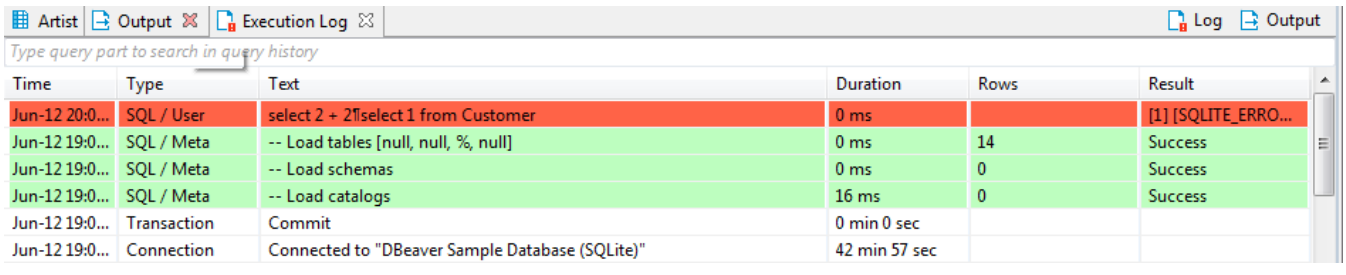
You can open the SQL editor preferences by pressing `Alt+Enter`.

Results Panel

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

The results panel provides **Output** and **Log** views of results.

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



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

Layout Adjustment

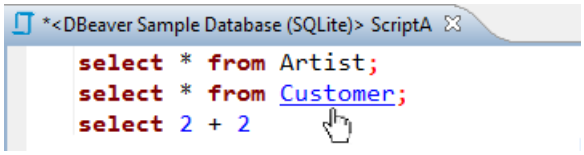
You can modify the layout of the SQL Editor by showing/hiding the results panel and changing the horizontal/vertical position of the panes.

- To toggle (hide/show) the results panel, press `CTRL+6` or right-click anywhere in the script pane and, on the context menu, click **Layout -> Toggle results panel**.
- To maximize the results panel, press `CTRL+Shift+6`, or double-click the results tab name, or right-click anywhere in the script panel and, on the context menu, click **Layout -> Maximize results panel**.
- To switch between the script panel and the results pane, press `Alt+6` or right-click anywhere in the script panel and, on the context menu, click **Layout -> Switch active panel**.

To position both panels horizontally, right-click anywhere in the script panel and, on the context menu, click **Layout -> Horizontal**.
To position both panels vertically, right-click anywhere in the script panel and, on the context menu, click **Layout -> Vertical**.

Hyperlinks

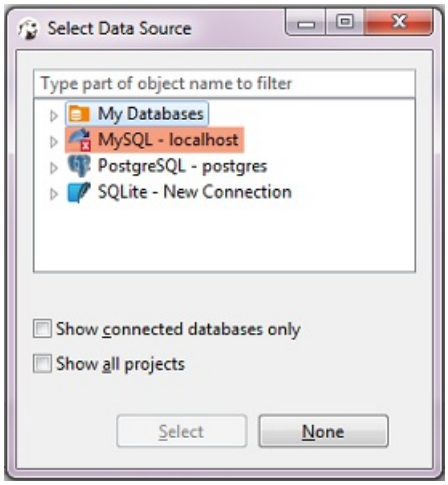
You can press and hold `Ctrl` and at the same time move the mouse over the SQL text. If DBeaver recognizes some identifier as a table/view name, it presents it as a hyperlink. You can click the hyperlink to open this object`s editor:



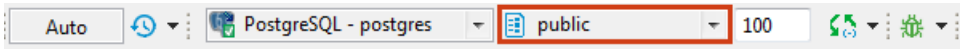
Active Database/Schema Selection

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

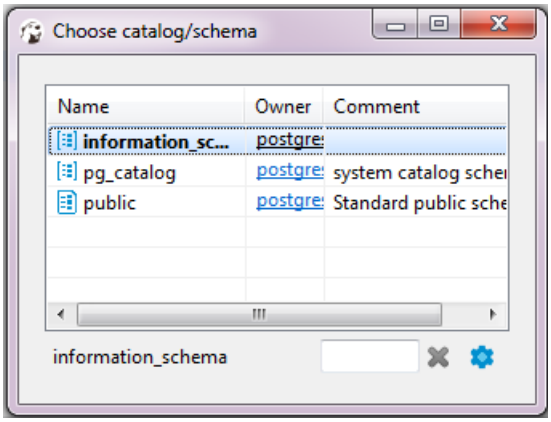
The Select Data Source dialog box opens. In the tree of connections, click the required connection and then click **Select**. To disassociate the SQL Editor with any connection, click **None**:



To change the active schema, press `Ctrl+O` or click the **Active Catalog/Schema** box in DBeaver`s main toolbar:



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

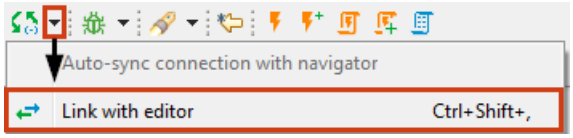


If there are many schemas and they do not fit in the dialog box use the search field to find the schema you need: 

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 on DBeaver`s main toolbar:

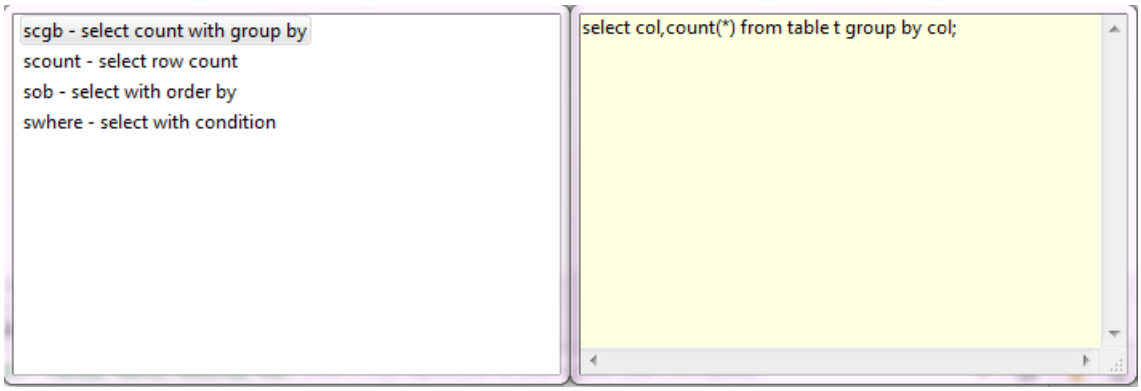
The reverse action is also possible: you can set the focus of the Database Navigator to the active connection of the SQL Editor - press `Ctrl+Shift+,` or click the arrow next to the **Set connection from navigator** button in DBeaver`s main toolbar and then click **Link with editor**:



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 with a list of available templates appears:



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

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

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

Standard Eclipse templates:

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

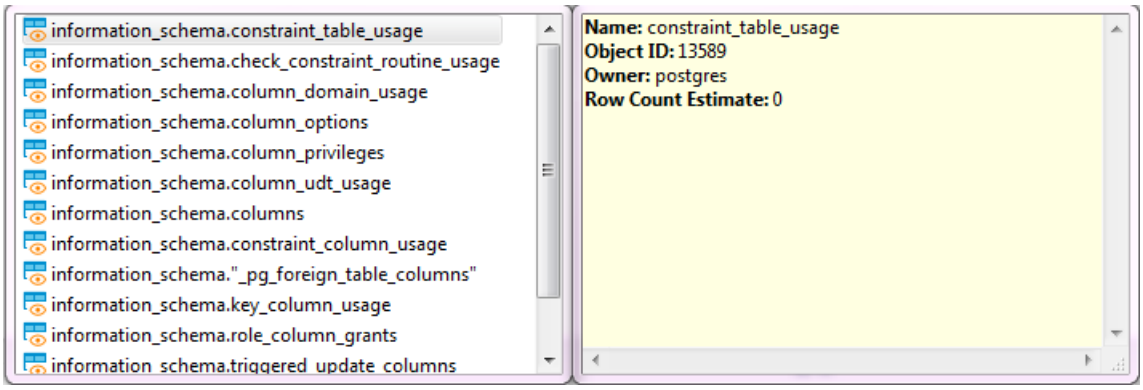
DBeaver-specific templates:

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

SQL Assist and Auto-Complete

The SQL Assist feature provides auto-completion of database object names and SQL commands, and other keywords in queries.

To perform some object name auto-complete, press `Ctrl+Space` or right-click the required place in the query and click **SQL Assist** on the context menu. DBeaver searches for objects in a database by their names and/or descriptions.



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

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

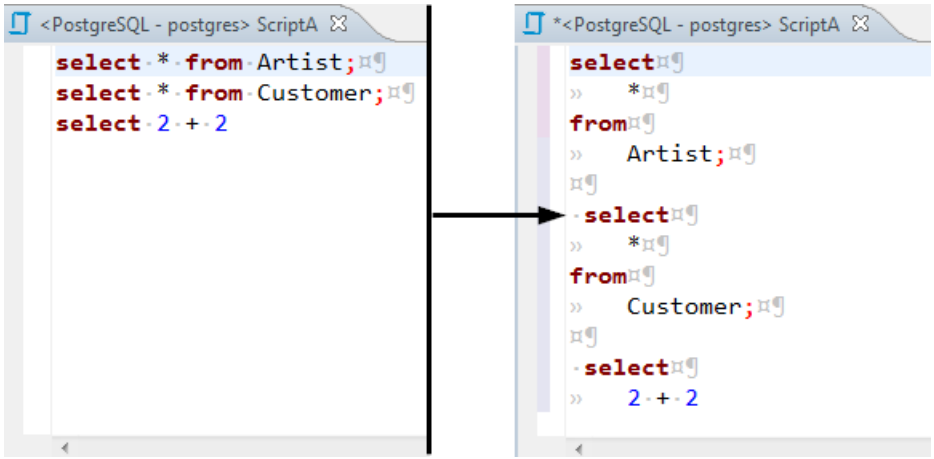
Hippie Engine

Hippie Engine provides autocompletion based on information of the current script file. It scans files looking for words and adds proposals based on similarity with the given string.

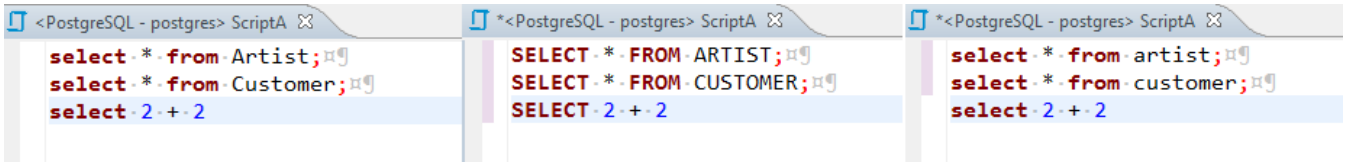
It can be turned off/on: `Window -> Preferences -> Editors -> SQL Editor -> Activate Hippie Engine for autocompletion`

SQL Formatting

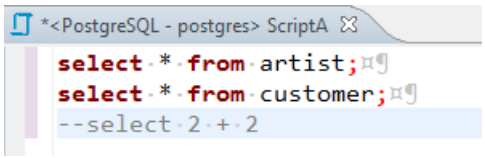
To format an SQL text, select it and press `Ctrl+Shift+F` or right-click the selected text and click **Format -> Format SQL** on the context menu.



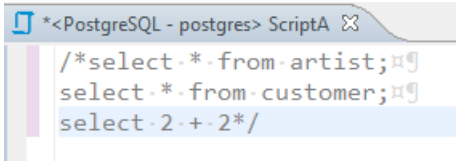
To format a script to upper or lower case, highlight the SQL text, then right-click it and click **Format -> To Upper Case** / **To Lower Case**, respectively, on the context menu.



To comment out an SQL line, press `Ctrl+/` or right-click the line and click **Format -> Toggle Line Comment** on the context menu. To uncomment a commented line, manually remove the commenting syntax, or press the same button combination, or right-click the line and click the same item on the context menu.

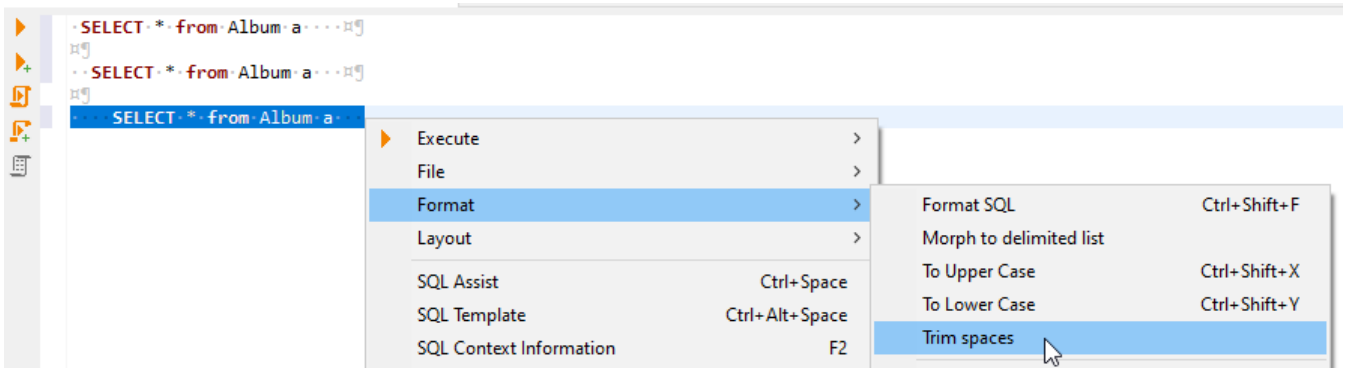


To comment out a block of text, select the text, then press `Ctrl+Shift+/` or right-click it and click **Format -> Toggle Block Comment** on the context menu. To uncomment a commented block of text, you can either manually remove the commenting syntax or select the same block of text, right-click it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

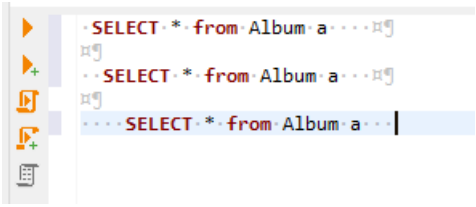


To trim spaces (leading and trailing) SQL text right-click the selected text or end of string and click **Format -> Trim spaces** on the context menu.

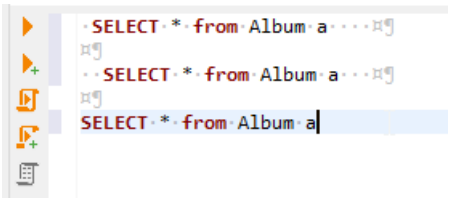
You can choose a part of the text



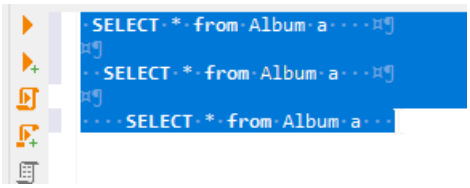
or put the cursor at the end of the row you want to trim



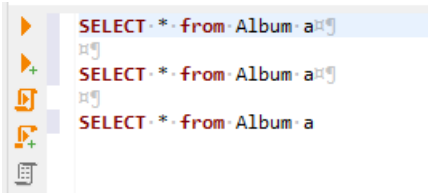
You will get the following result:



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



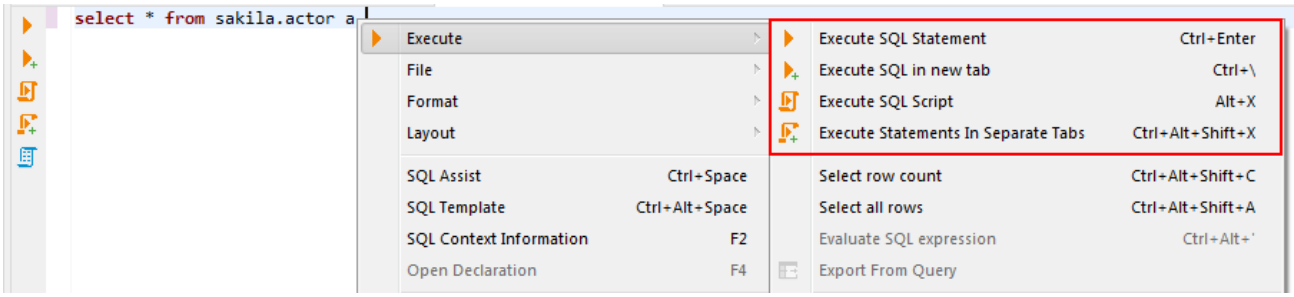
Then each of the selected rows will trim, and the final result will be like this:



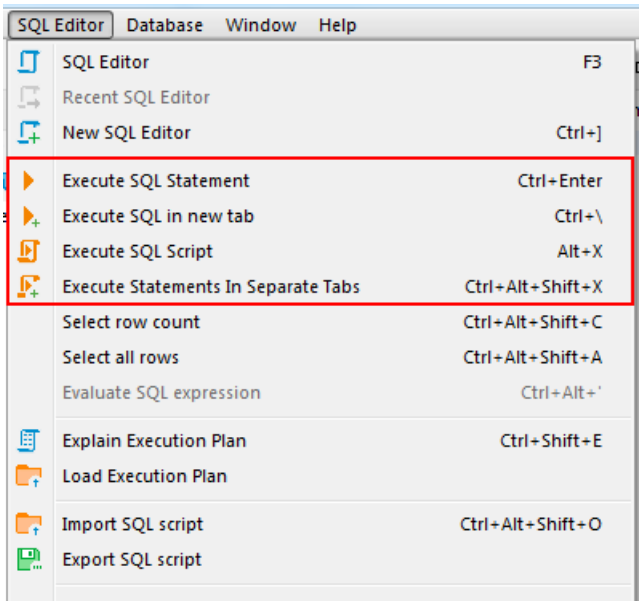
SQL Execution

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

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



- DBeaver main menu:



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

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

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

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

Result tabs

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

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

Naming

A tab is often named after the primary table of your query. For example, after executing the following query you will see a single tab called `Album` (assuming that your database has a table called `Album`):

```
SELECT * FROM Album;
```

If a query has joins or, in other words, has multiple source tables, a `(+)` is shown right to the table name. The following query will result in a tab called `Album(+)`:

```
SELECT * FROM Album al, Artist ar WHERE al.AlbumId = ar.ArtistId;
```

Additionally, you can change the name of a given tab via its context menu or by using a special comment:

```
-- title: DBeaver is cool
SELECT * FROM Album;
```

In other cases, tabs are named in a form of `Results <A> (<B>)`, where:

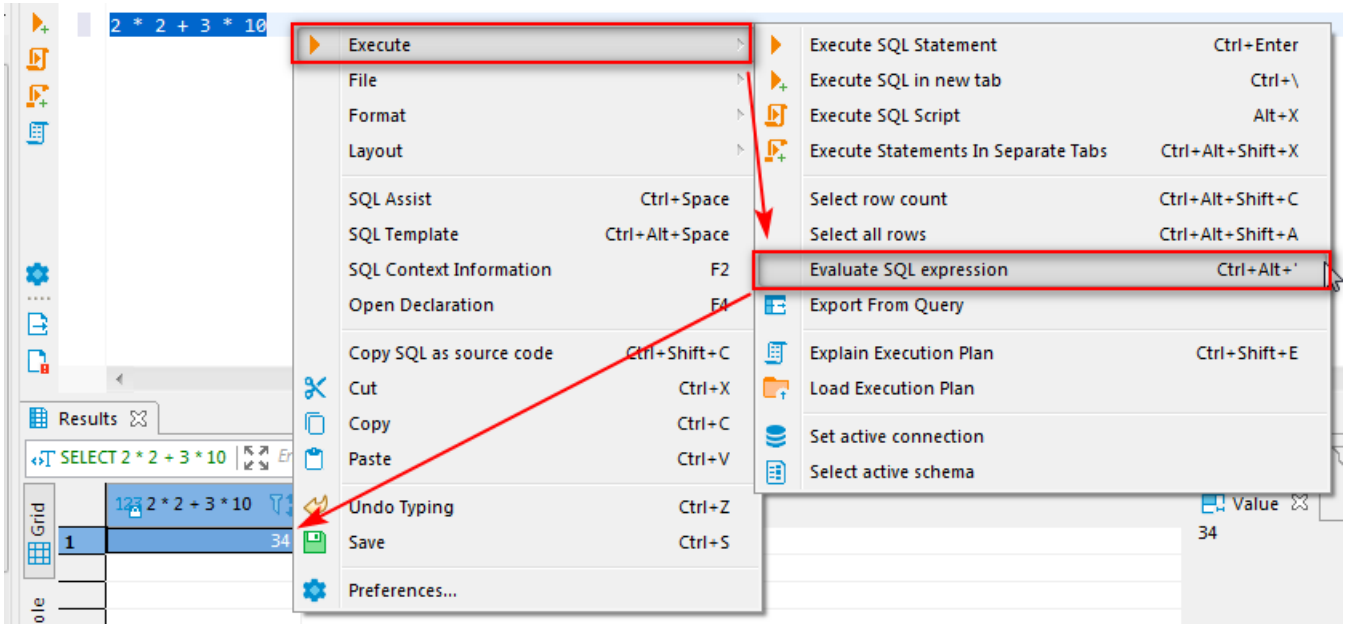
- **A** is *an index of query*
- **B** is *an index of result set of this query*

Pinning

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

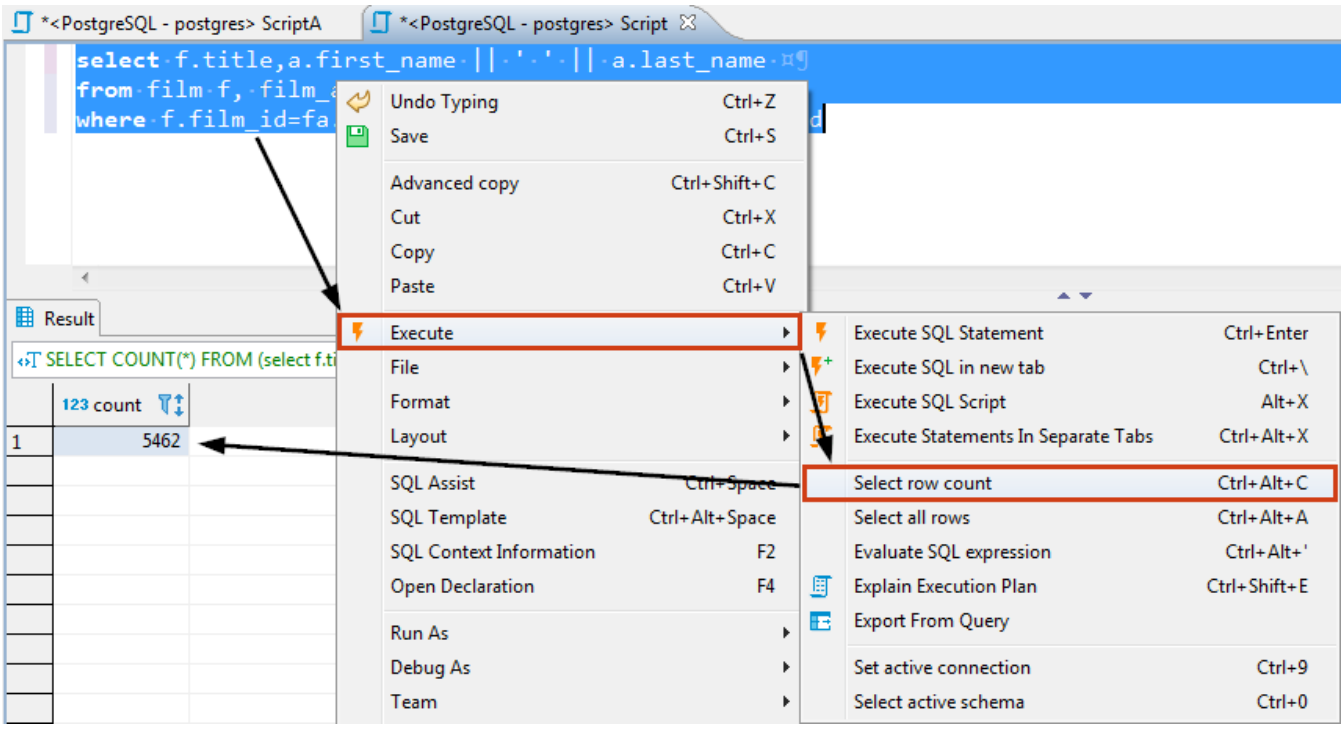
SQL Expression Evaluation

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



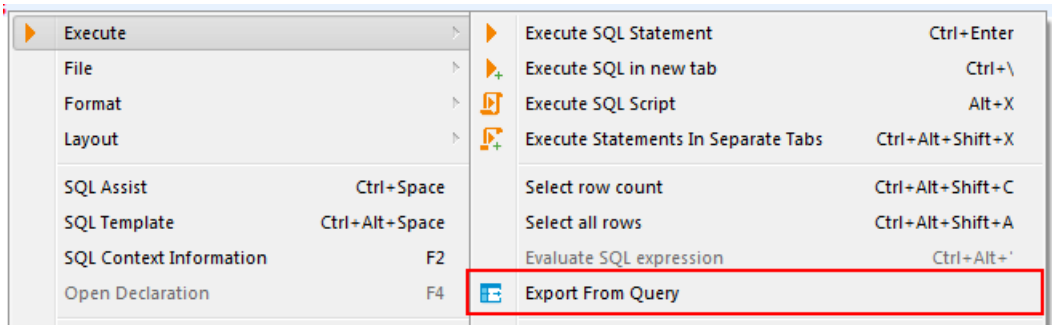
Row Count

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

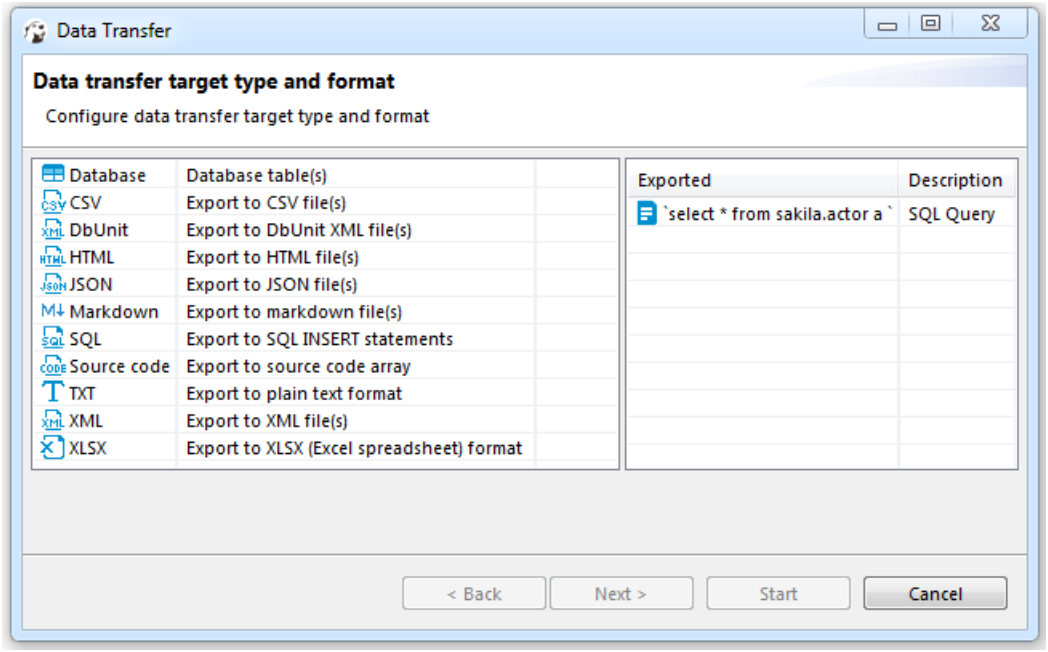


Query Export

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

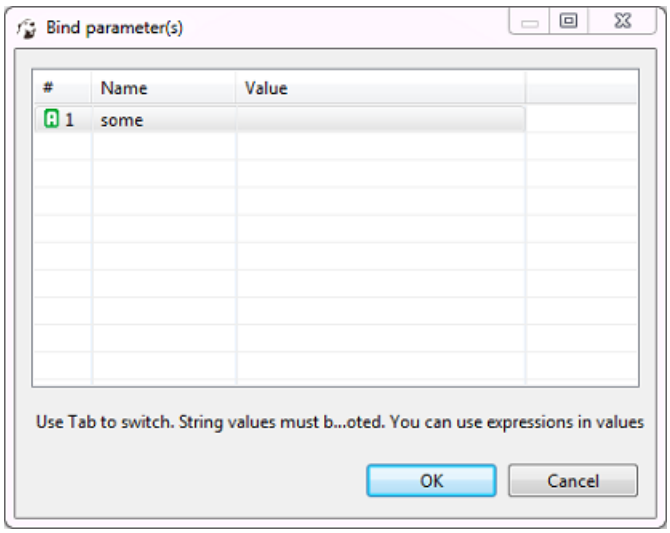


The Data transfer wizard opens. Go through its steps to complete the export of the query.

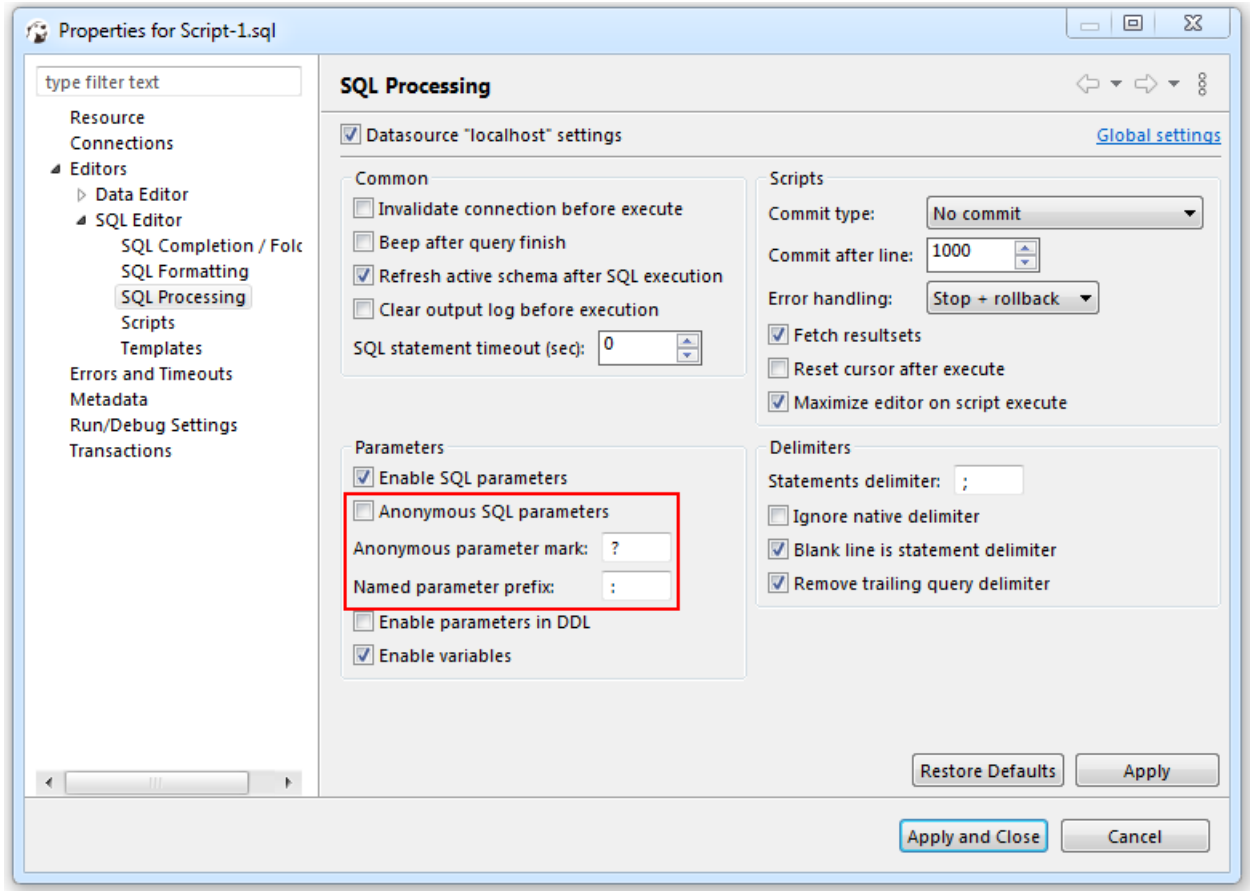


Dynamic Parameter Bindings

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



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



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

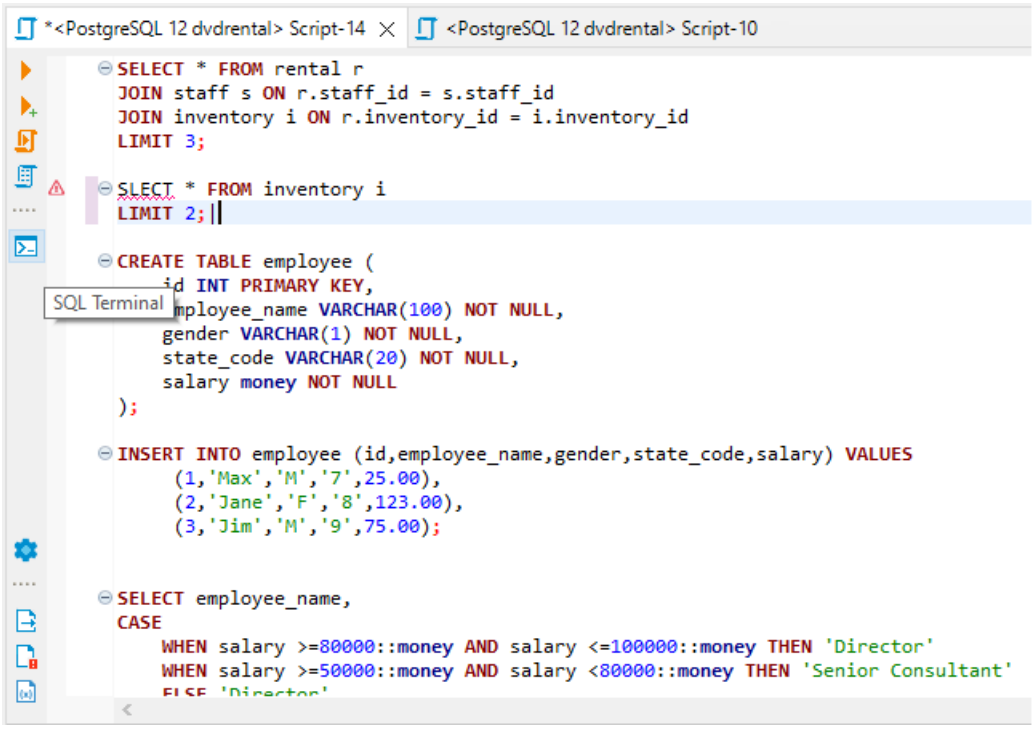
Miscellaneous

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

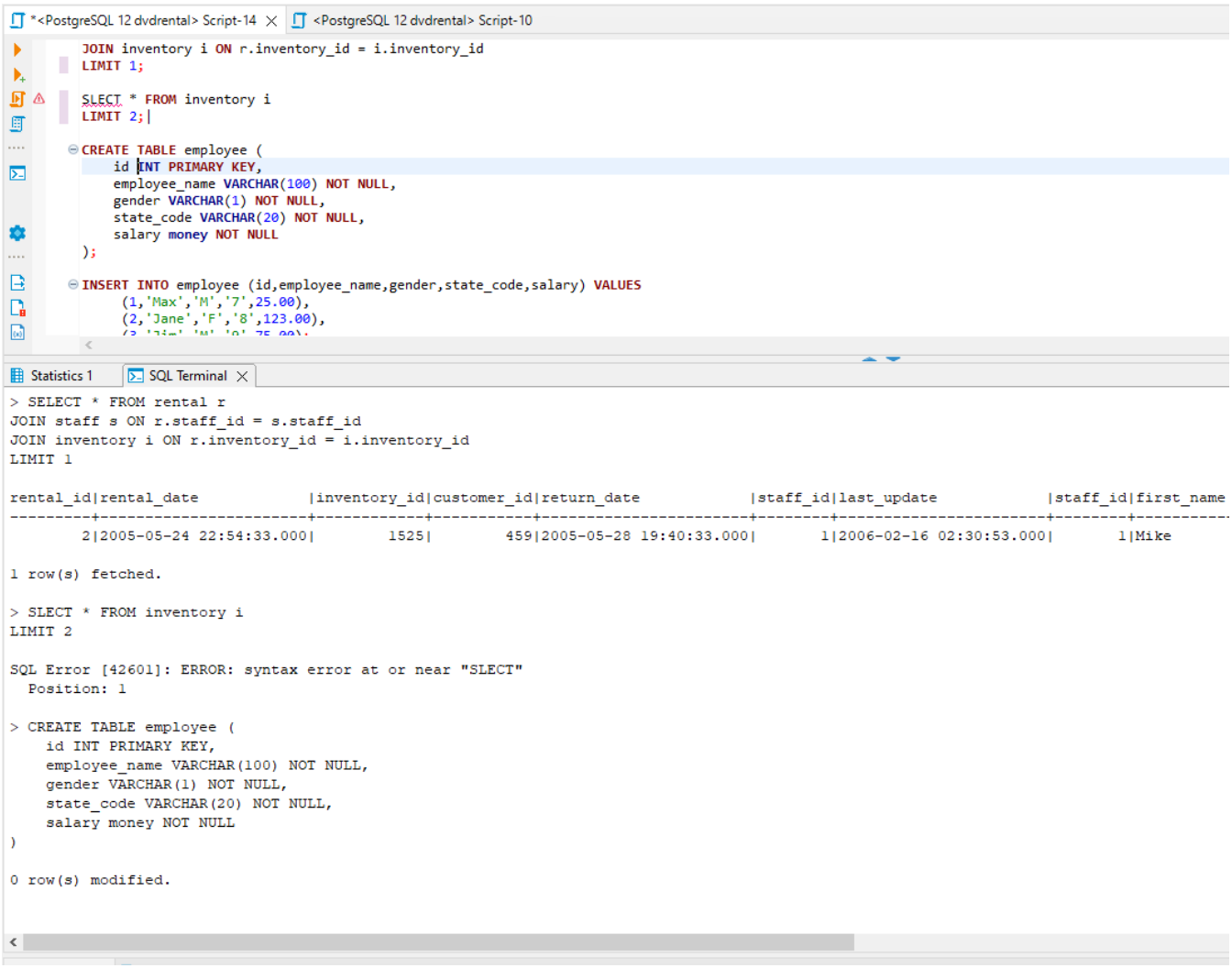
SQL Terminal

SQL Terminal is an SQL Editor result tab where you can find the results for all executed queries in a text format.

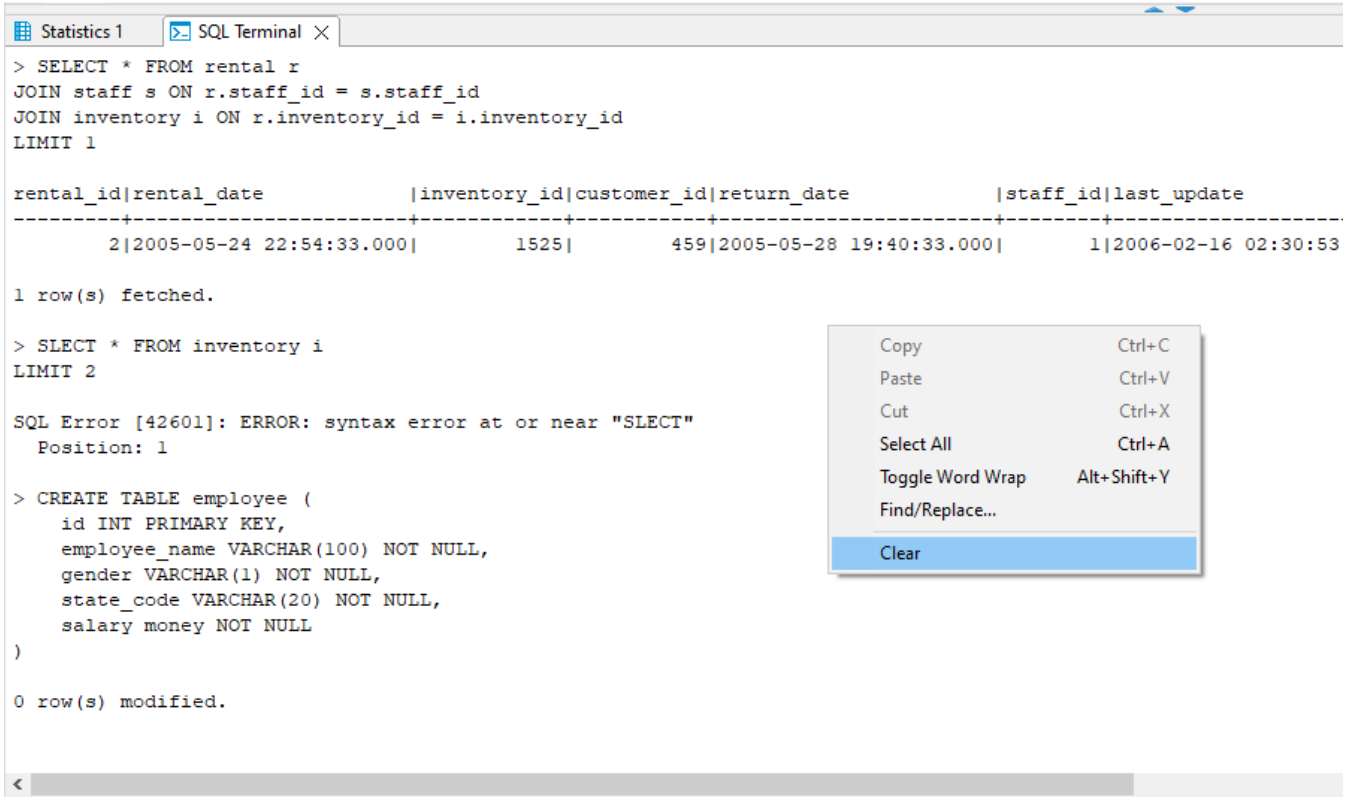
To open SQL Terminal you should press the corresponding button on the left SQL Editor toolbar.



You can execute a single statement or a sql script and see the result set with data, errors, and statistical information along with the corresponding query text.

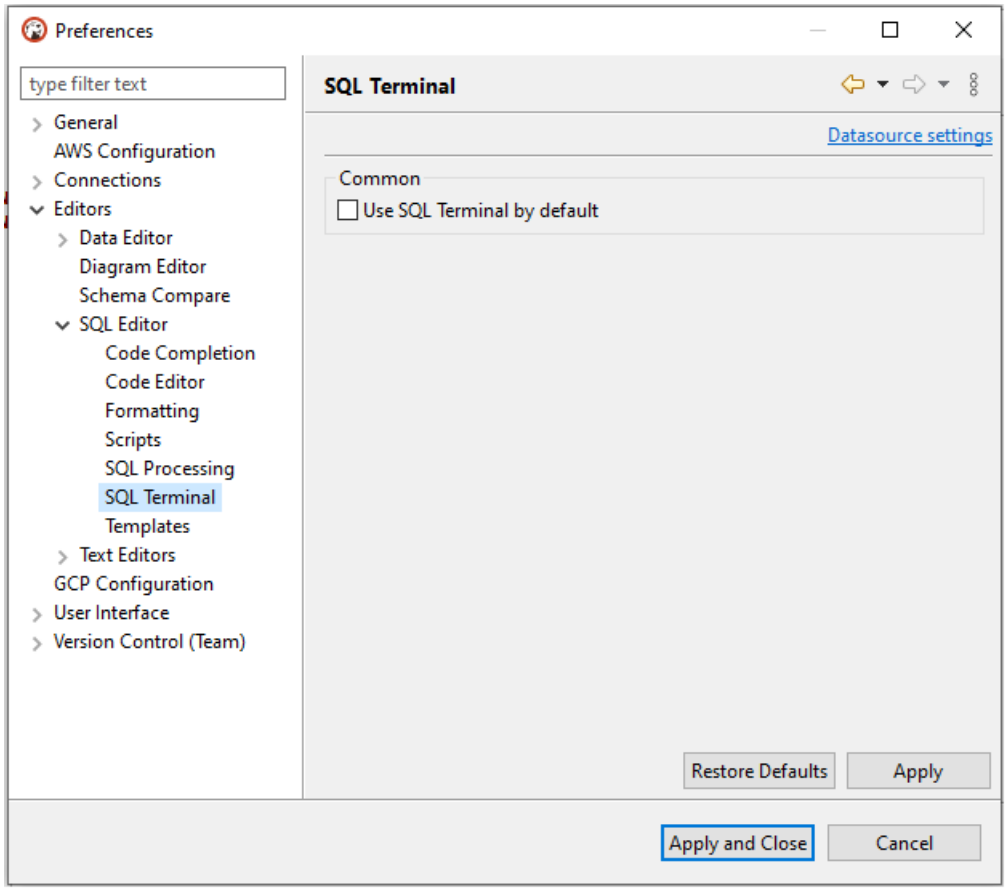


The context menu is available by right-clicking on the SQL Terminal area.



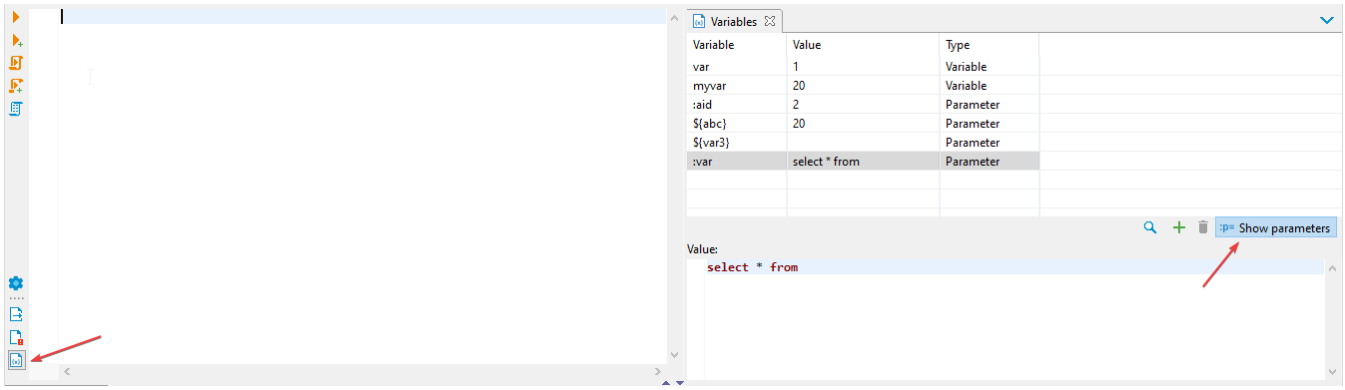
For each sql script file, it will be saved whether the SQL Terminal is enabled, and the state of the SQL Terminal button will be restored when the file is opened.

You can set up whether to enable or disable the SQL Terminal for new scripts by default in `Preferences->Editors->SQL Editor->SQL Terminal`.



Variables panel

You can see all of the currently assigned local variables for SQL Editor. You need to click a "Show SQL variables" button in the SQL editor.
A new tab alongside Output and Execution log panels will be opened with a list of assigned variables. On this panel, you can also show assigned parameters by clicking the corresponding button.



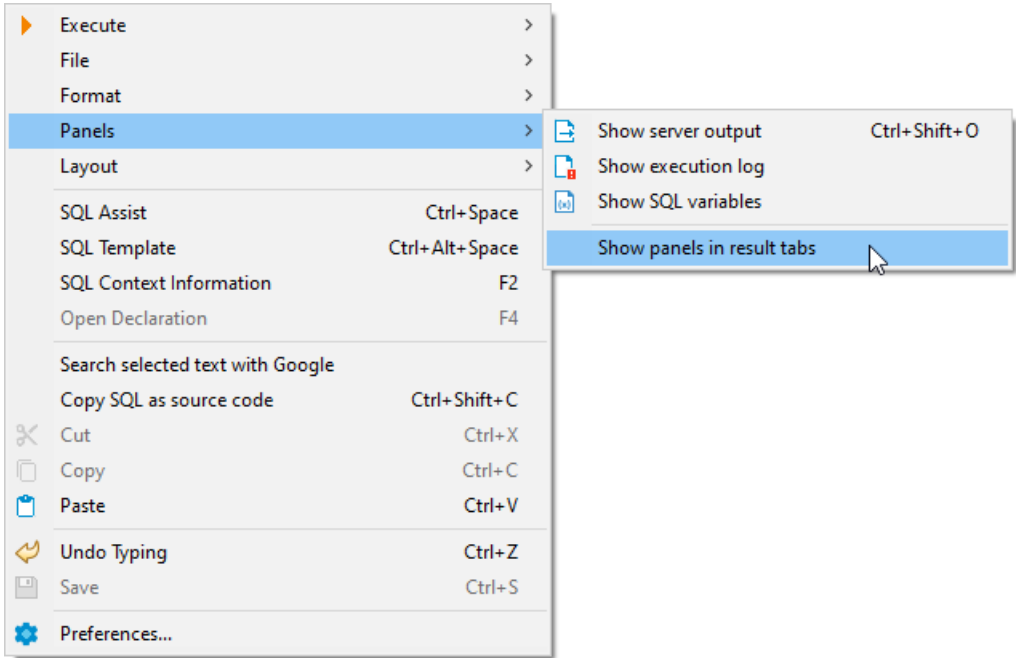
Manipulating variables

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

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

Moving a panel

Variables panel is always connected to output and execution log panels, but this group of tabs can be configured to be shown either at the right side of the editor or at the bottom alongside the results panel. By default, the panels are shown on the right side. To change their location you need to either check or uncheck `Show panels in result tabs` in the context menu.

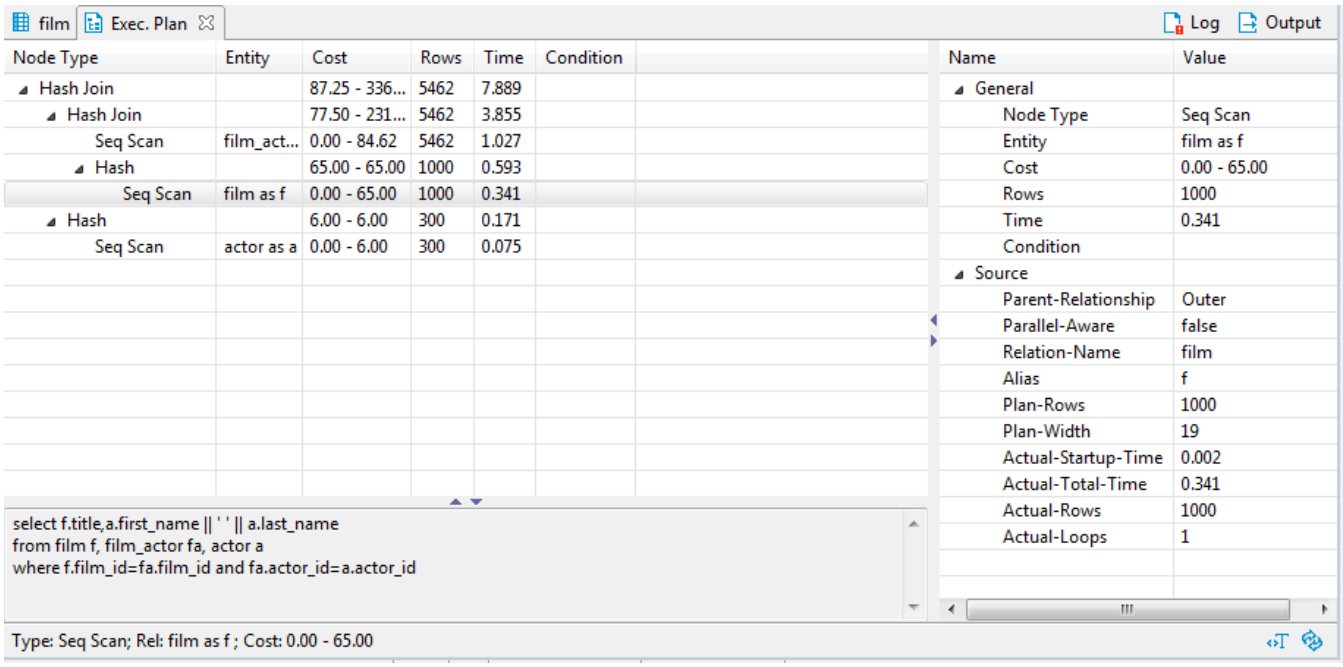


Query Execution Plan

Execution Plan

Simple plan view

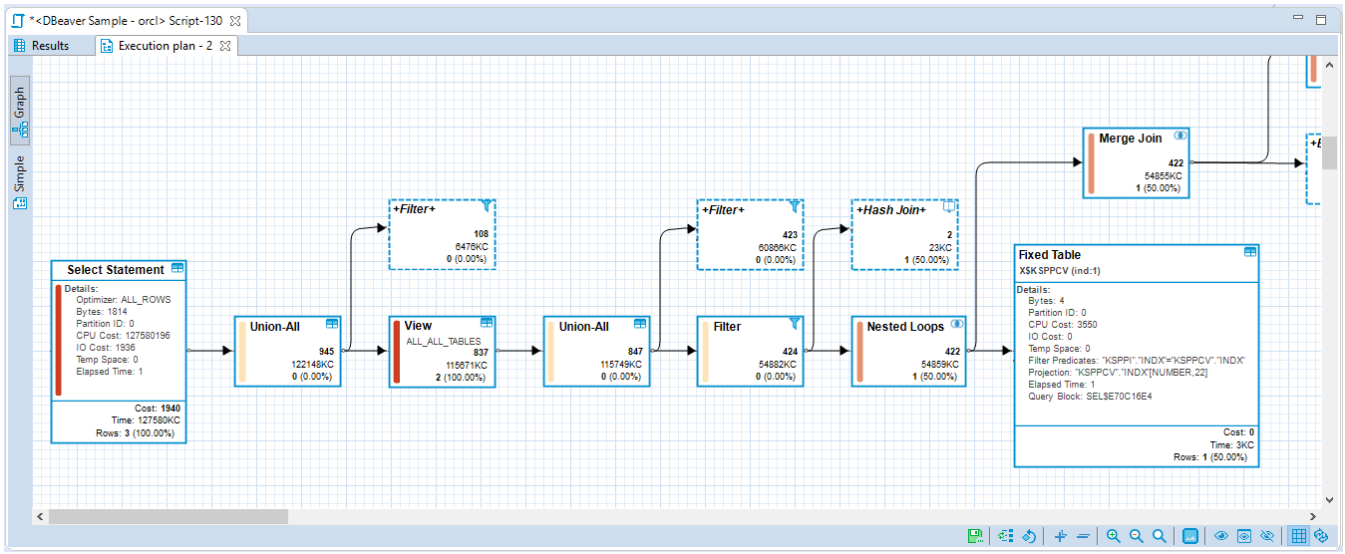
If a database driver supports the visualization of the execution plan, you can see the execution plan of the current query (under cursor) by pressing `Ctrl+Shift+E` or clicking **Explain execution plan** on the context menu or in the main toolbar: The execution plan command generates a tree of query 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 ( ).

Advanced plan view

In DBeaver Lite, Enterprise, and Ultimate editions you can use an advanced (graph) visualization of the execution plan.
This visualization shows the most expensive (cost-based) plan nodes. You can hide all irrelevant nodes (see node details), use horizontal or vertical plan layouts, export it to an image or save it as JSON to send to a colleague.



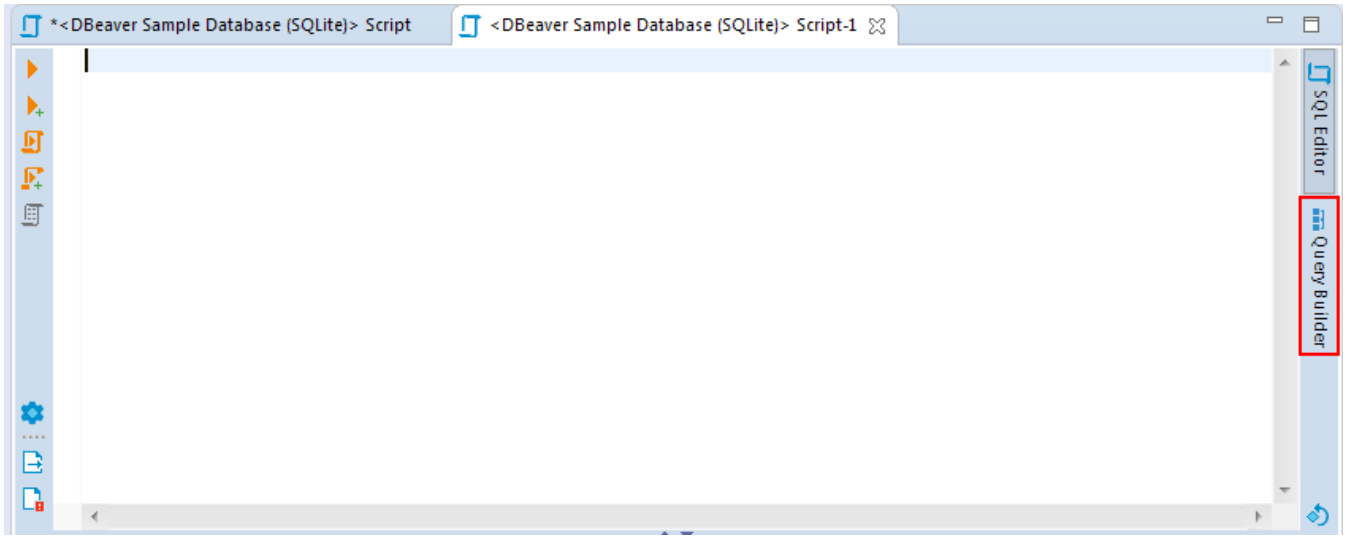
Visual Query Builder

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

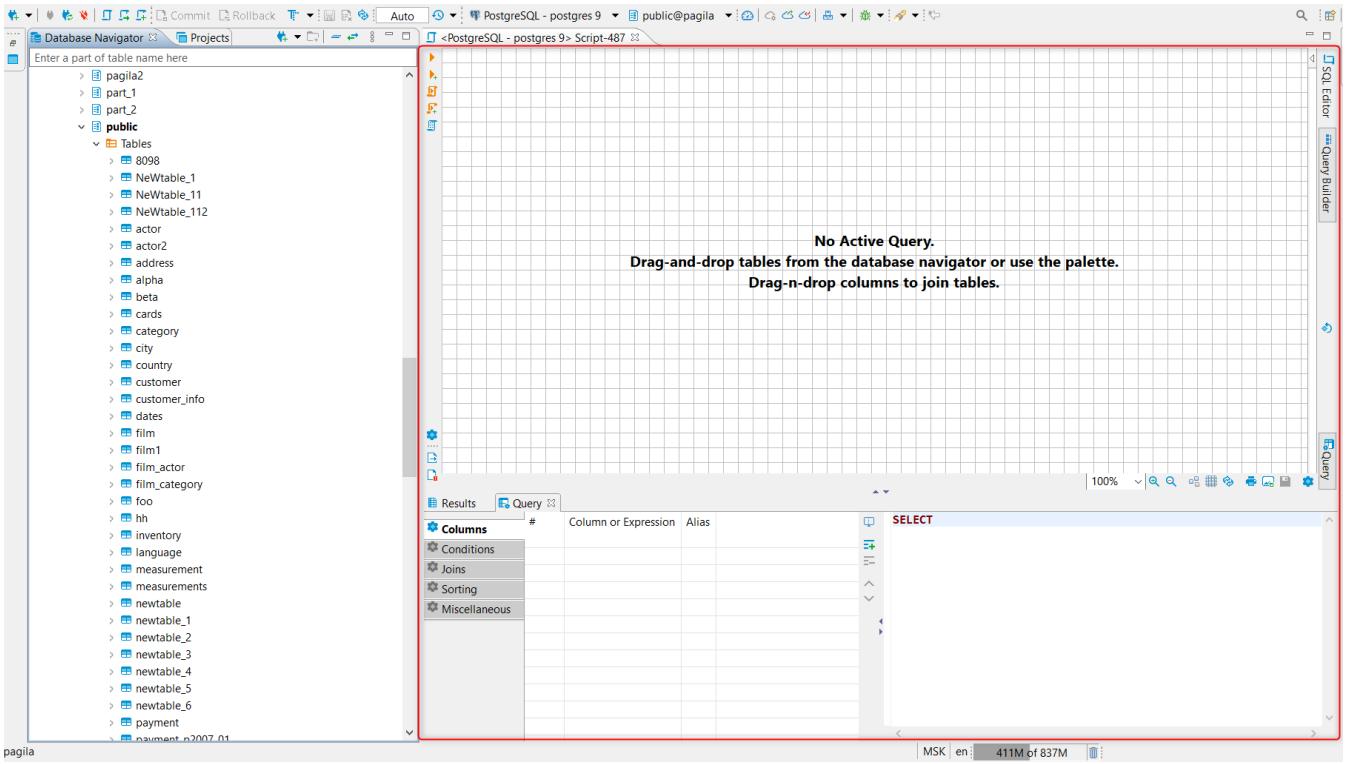
Query Builder is a user-friendly visualization tool that will help you make sense of your complex database designs. It can be useful when you need to understand the various relationships between different tables. Also, it can be helpful for those who are not very familiar with SQL scripting or if you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

Opening Visual Query Builder

To open **Visual Query Builder** click the **Open Query Builder** button in the SQL Editor tool bar.

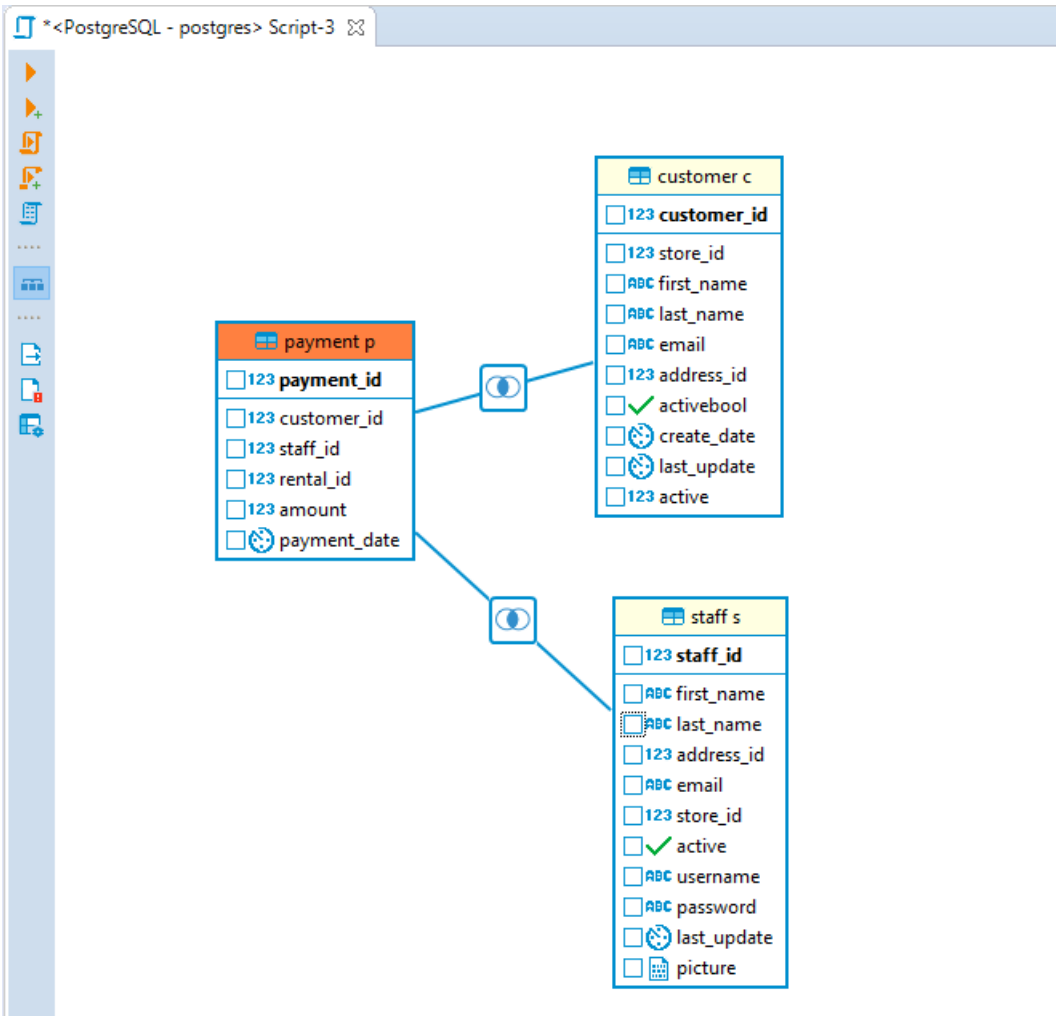


The **Visual Query Builder** will appear on the right.

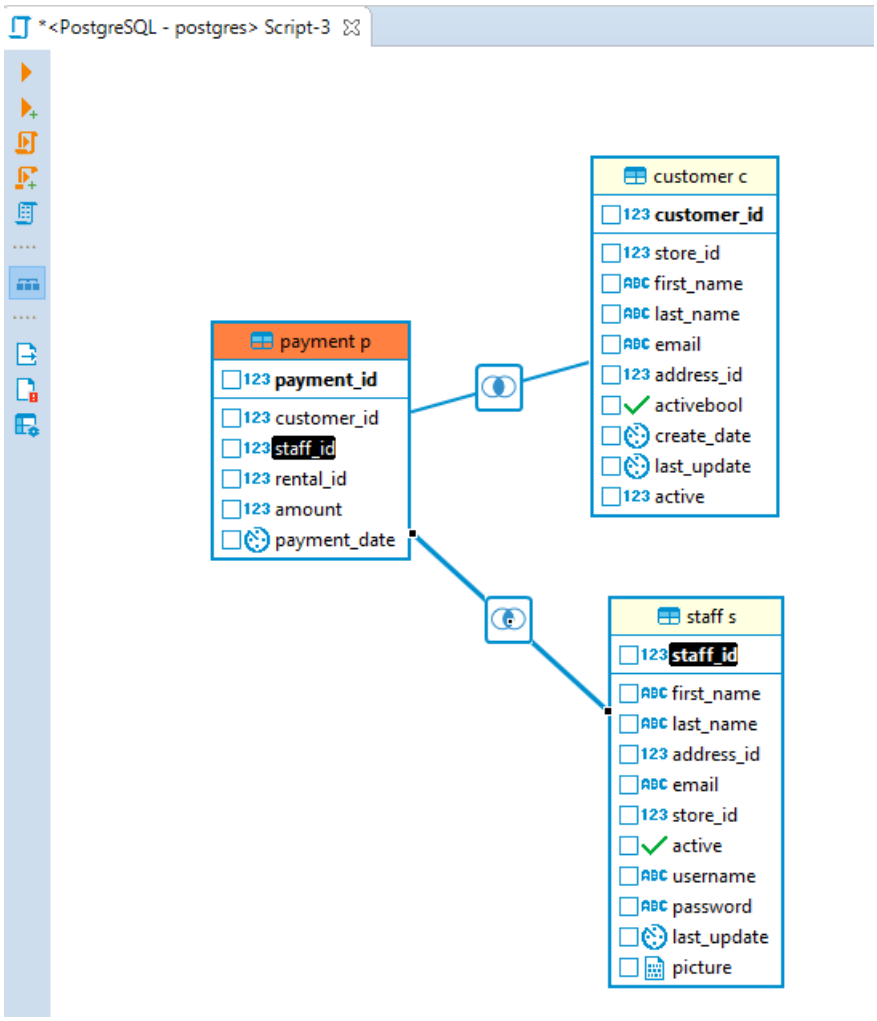


Creating Visual Query

Start creating a query by selecting a query data source: drag-and-drop tables you want to work with from the **Database Navigator** pane into the **Visual Query Builder** area. All the connections existing between the tables will be shown automatically.



To create a new join between the tables, press the left mouse button when the cursor is over the column of one table, holding the right mouse button drag the cursor to the column of another table and release the right mouse button. The connection between the selected columns of the tables will be created visually and in the SQL script a new join will be added.



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

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

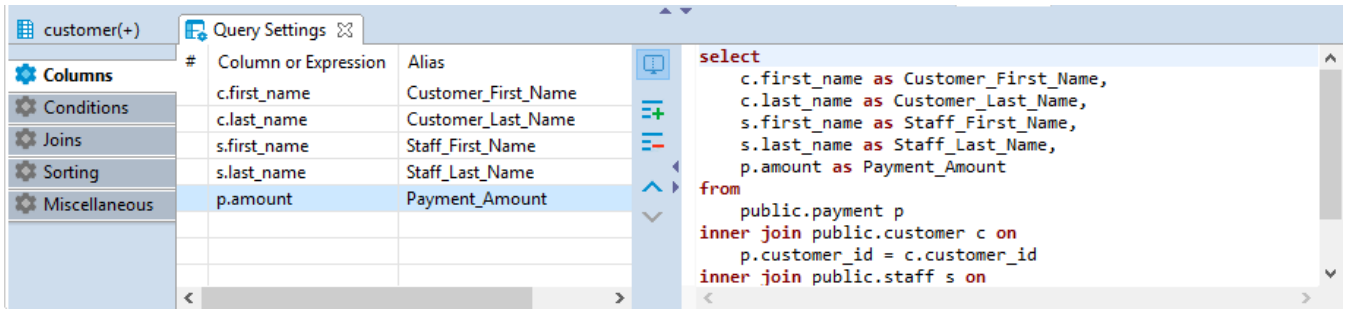
Adjusting Query Settings

Visual Query Builder also allows you to set the query conditions and adjust the representation of query results by means of **Query Settings Editor**.

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

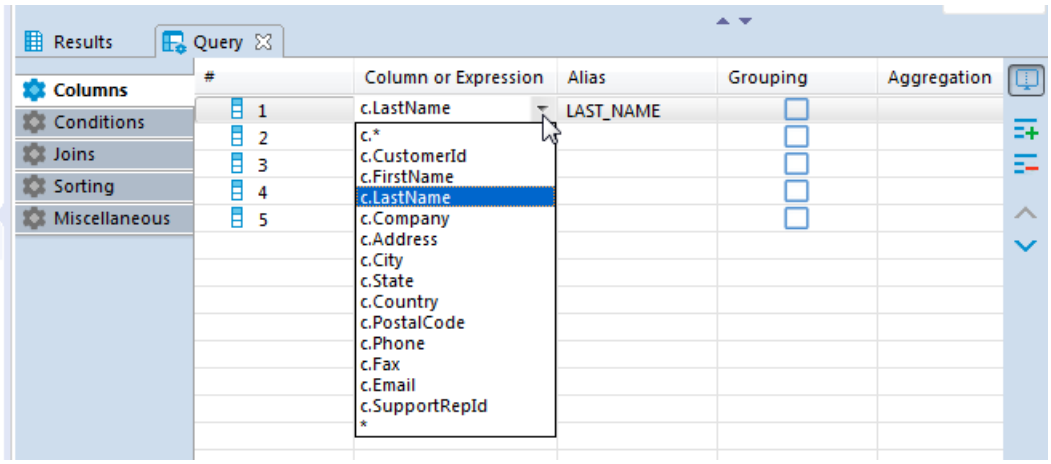
Query Settings window contains five tabs described below.

Columns



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

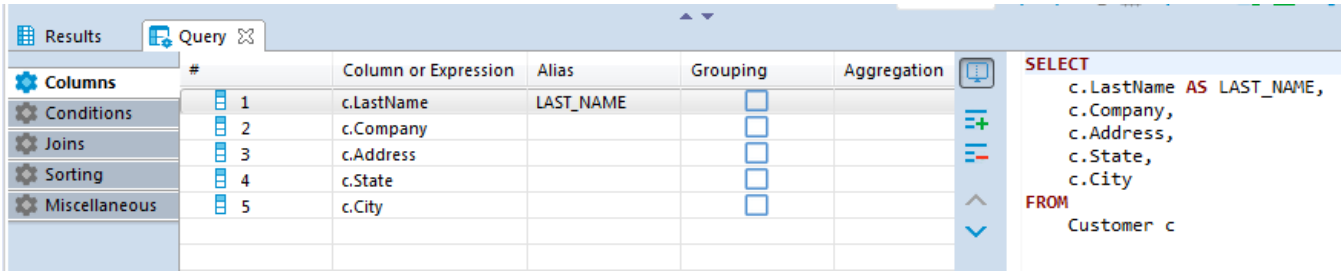
To add a column, press **Add** button and a new instance will be added to the table. Click on the first cell in **Column or Expression** column and select a column from the list of available columns displayed in the dropdown list appeared.



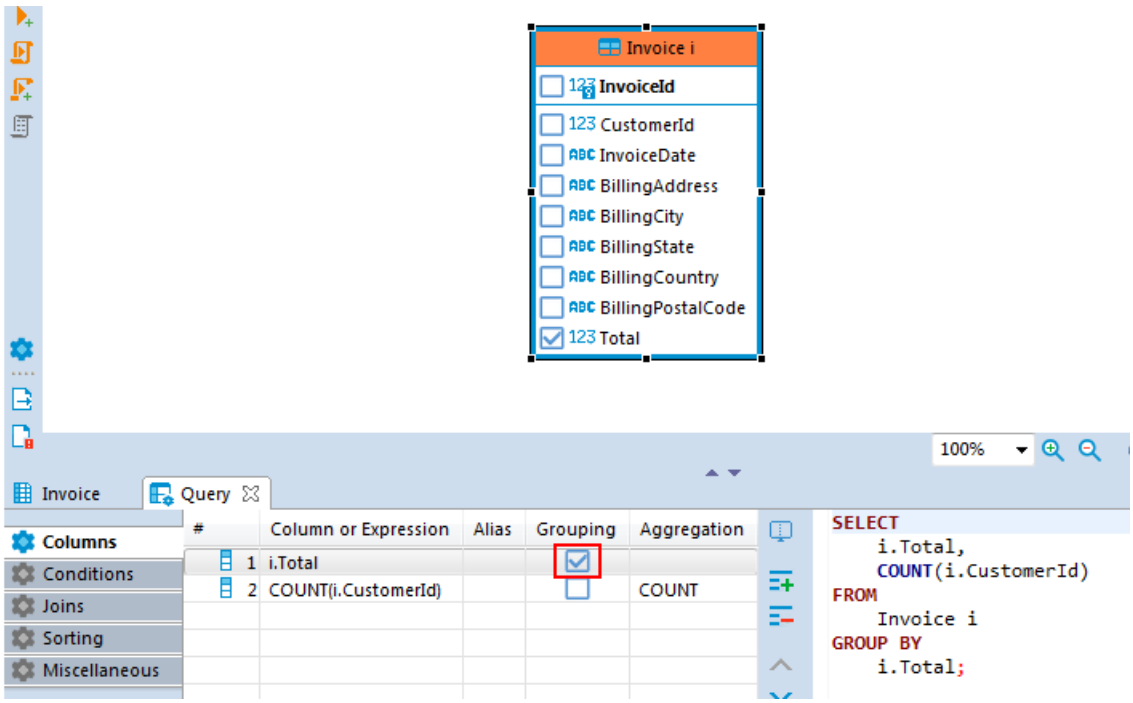
To remove a column, click on the row containing its name and press the **Remove** button on the right.

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

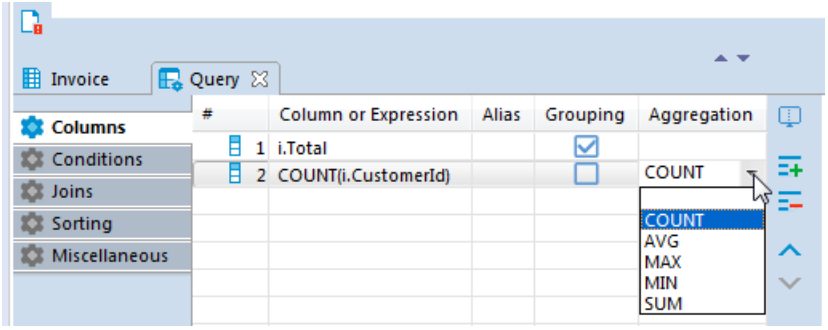
You can also define a user-friendly name of the column to be displayed in the result table. To set a user-friendly name, click on a cell in **Alias** column and insert the name. The change will be immediately displayed in the SQL script area.



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



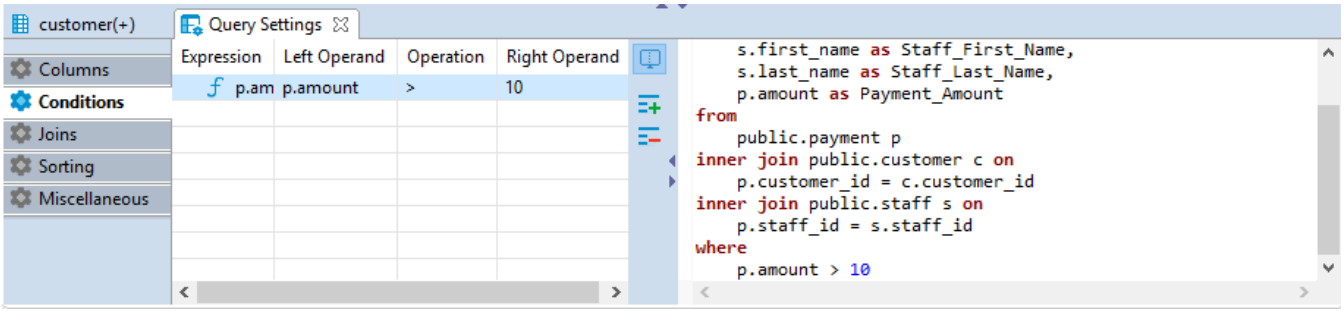
You can select other aggregation functions from the drop-down list. Or enter your own version in the cell.



When removing columns from the list, they will be removed from the grouping expression as well. When adding new columns to the list, it is added to the grouping expression.

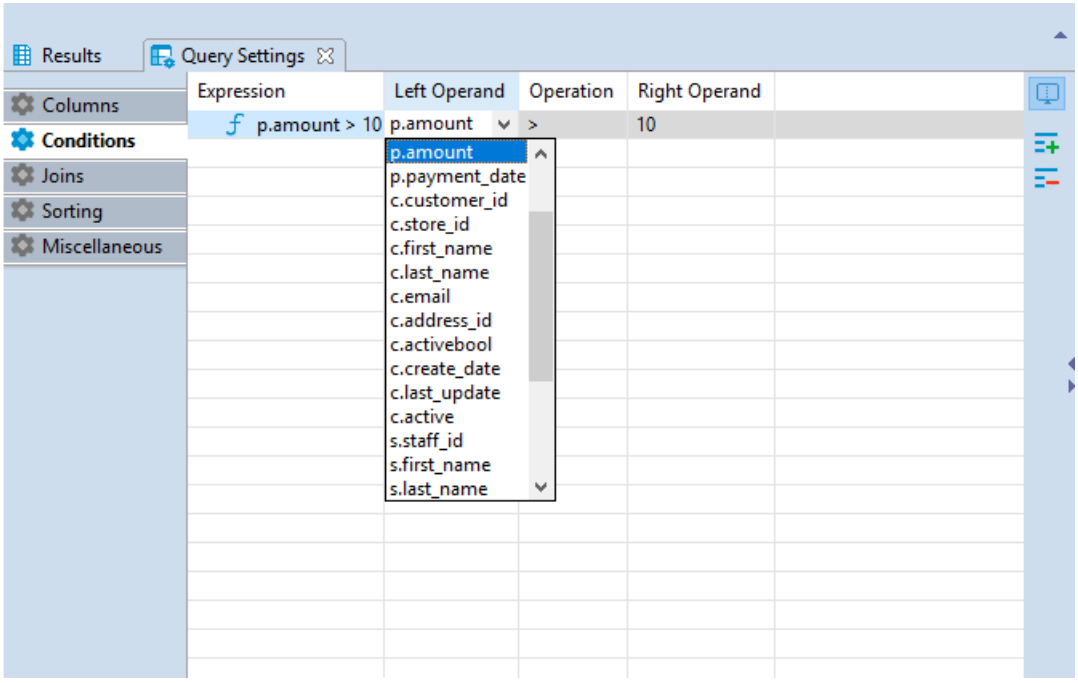
Conditions

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

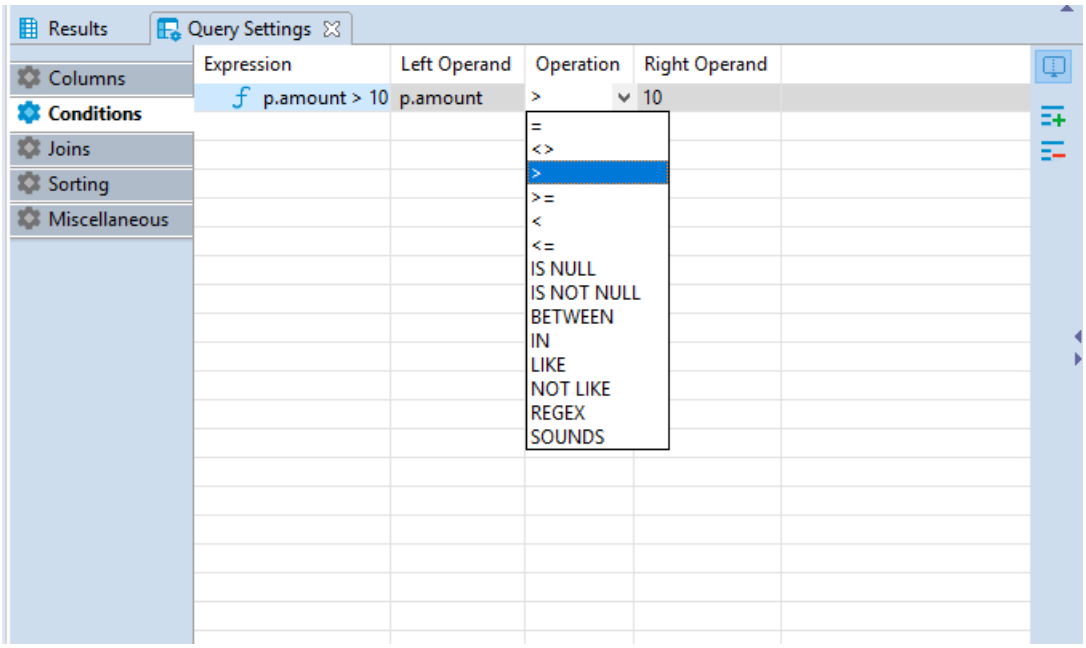


To add a new conditional expression, use the **Add** button on the right - a new instance will be added and the default conditional expression WHERE will be added to the SQL script area automatically. This default conditional expression can then be adjusted to the one you need:

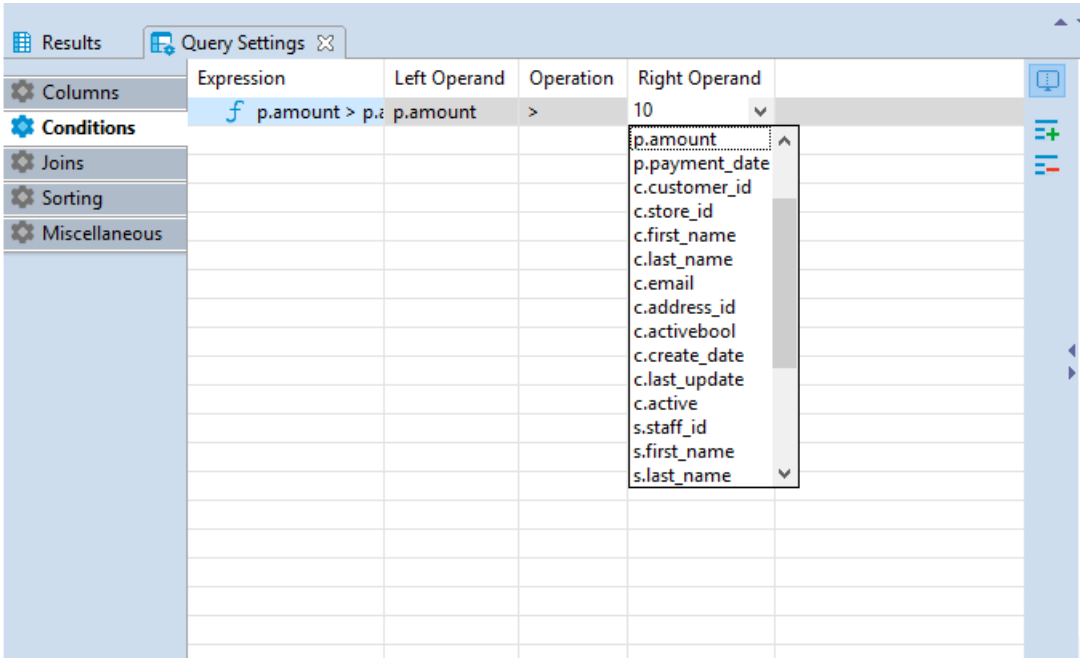
- **Left Operand** setting defines the left operand of the conditional expression. To set the left operand, click the cell in the **Left Operand** column and a drop down list of all available columns will be displayed. Select a column you want to use as the left operand in your conditional expression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



- **Operation** setting defines the comparison rule between the left and the right operands of the conditional expression. To set a comparison rule, click the cell in the **Operation** column and select the rule you need from the drop down list which will appear.



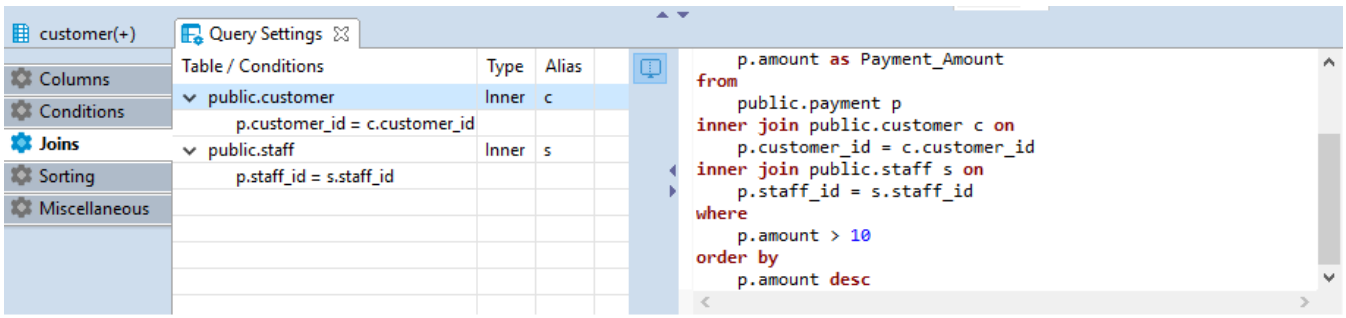
- **Right Operand** setting defines the right operand of the conditional expression. To set the right operand, click the cell in the **Right Operand** column and a drop down list of all available columns will be displayed. Select a column you want to use as the left operand in your conditional expression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



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

Joins

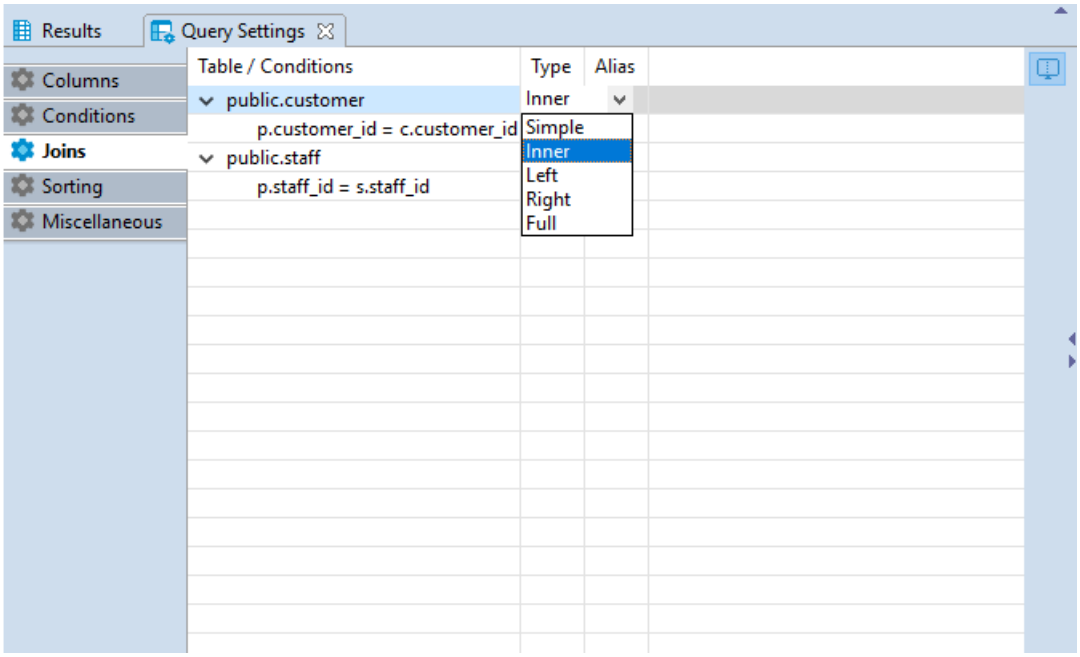
All the joins existing between the tables in **Visual Query Builder** main window are displayed in the **Joins** tab of **Query Settings Editor**.



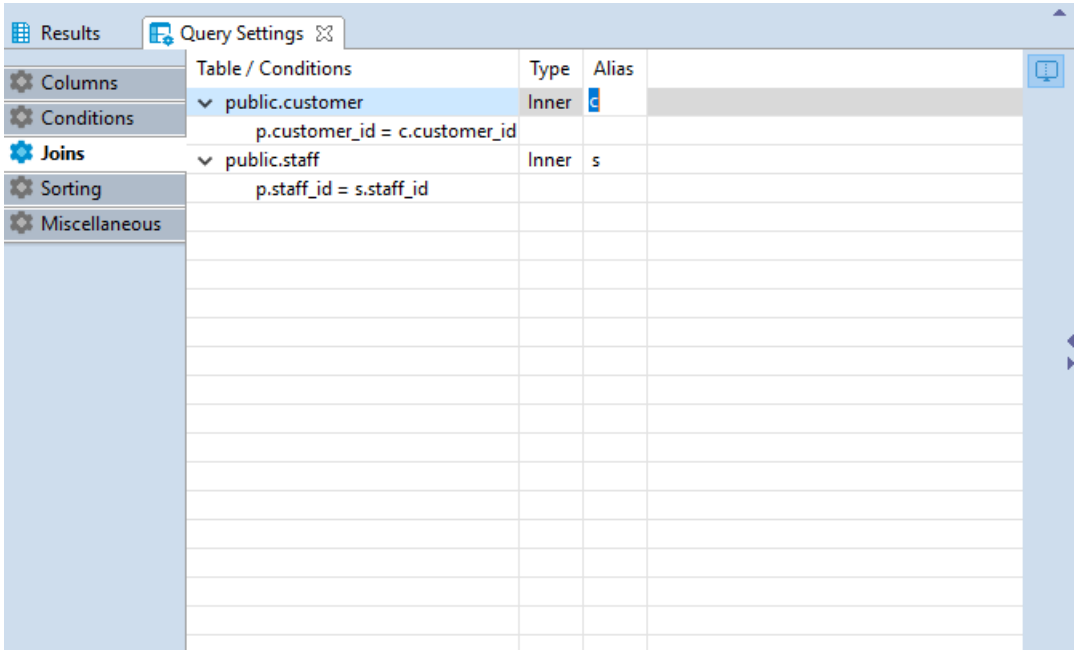
Joins cannot be added or removed by means of **Query Settings Editor**, however, the following join settings can be adjusted here:

- **Type** - defines the type of the join. Click the cell in the **Type** column - a drop down with available join types will be displayed.

Select the required option from the list by clicking on it.

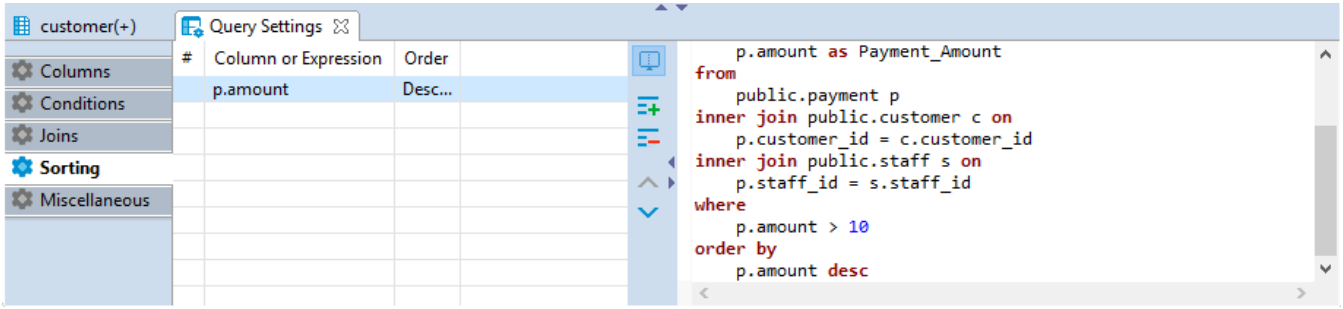


- **Alias** - defines a user friendly name of the join. To define this setting click on the cell in the **Alias** column and input the name.



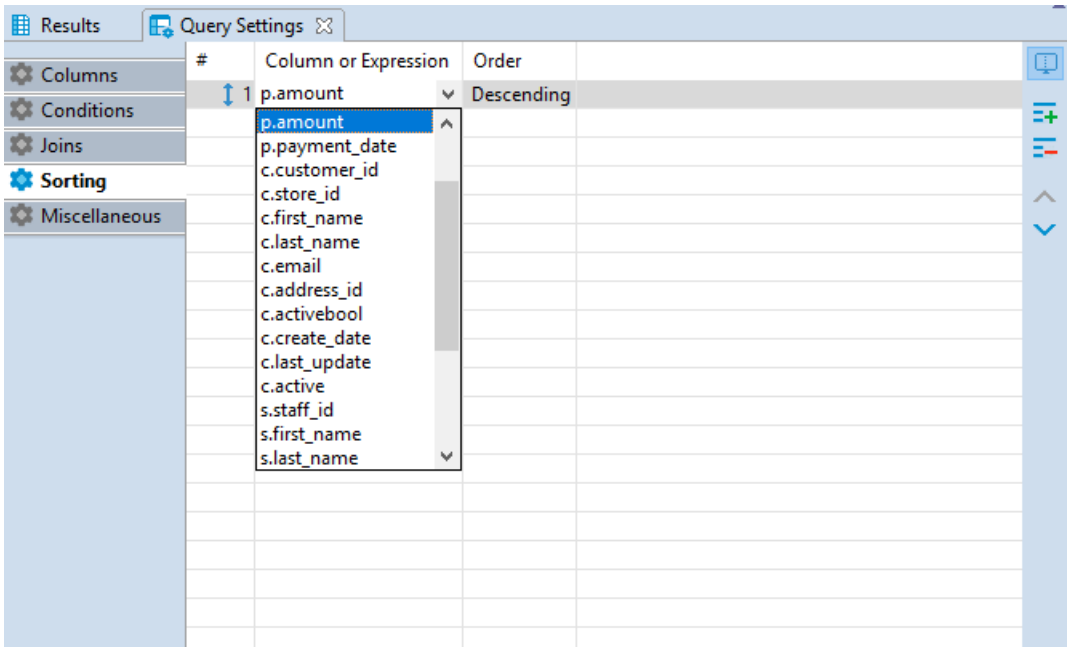
Sorting

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

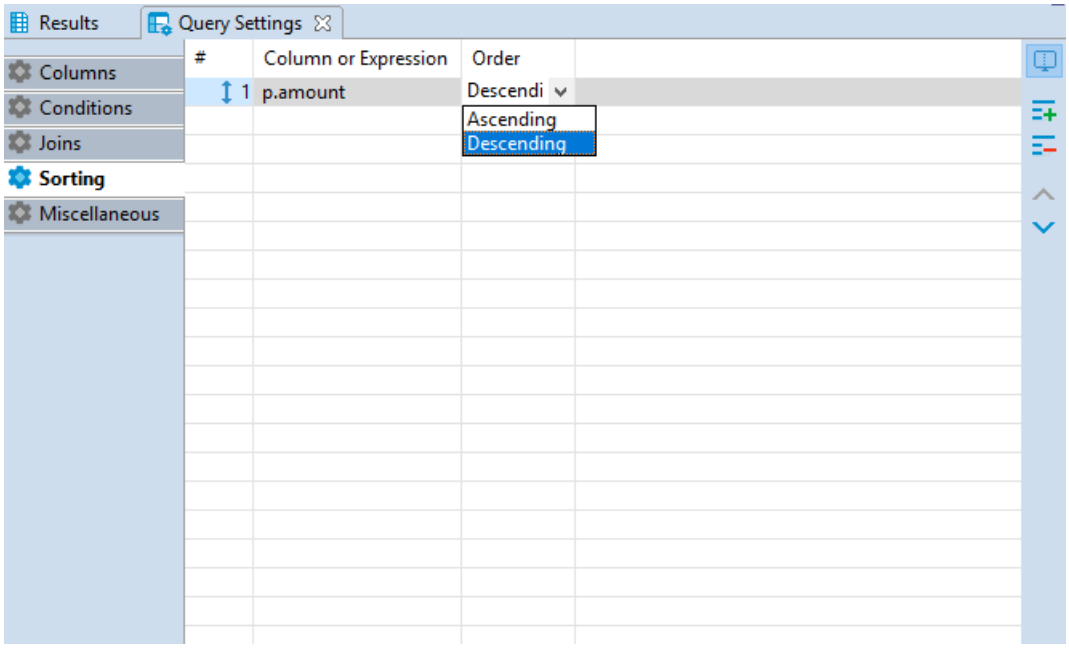


To add a new sorting condition press the **Add** button on the right and the default conditional expression ORDER BY will be added to the SQL script area automatically. This default conditional expression can then be adjusted to the one you need:

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



- In the **Order** column you can define whether the rows of the selected column should be sorted in ascending or descending order in the result table. To set the order, click the cell in **Order** column and select the required option from.The order by command will be added to the script.

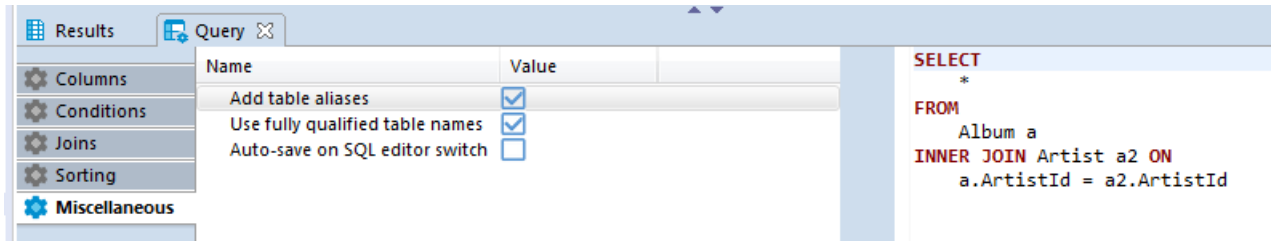


To remove a condition use the **Remove** button on the right.

Miscellaneous

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

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



Executing Visual Query

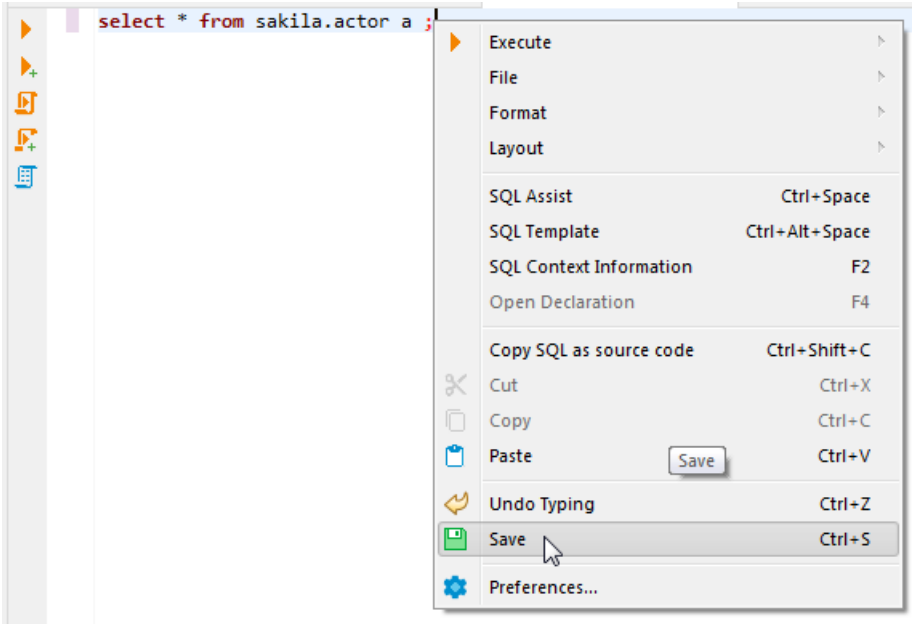
To execute a query, use the **Execute SQL statement** button  to get the results in the same tab or **Execute SQL statement in new tab** button  to get the results in a new tab. Both buttons are located in the **Visual Query Builder** vertical toolbar.

Script Management

Saving Scripts

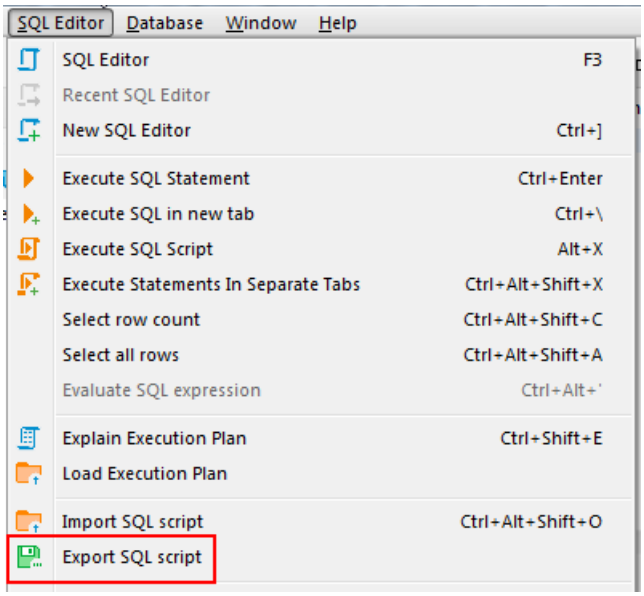
You can save scripts to a predefined space in the currently active project or somewhere in the file system.

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

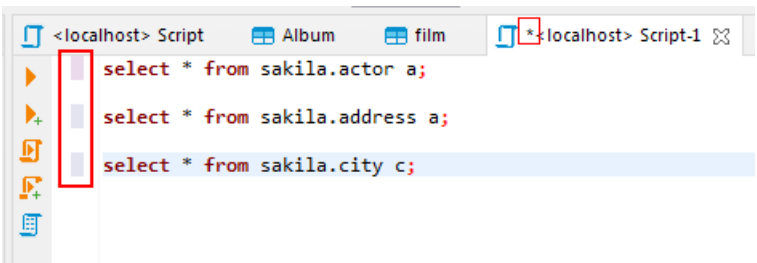


You can find the script saved this way in the Project Explorer view in the **Scripts** folder.

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



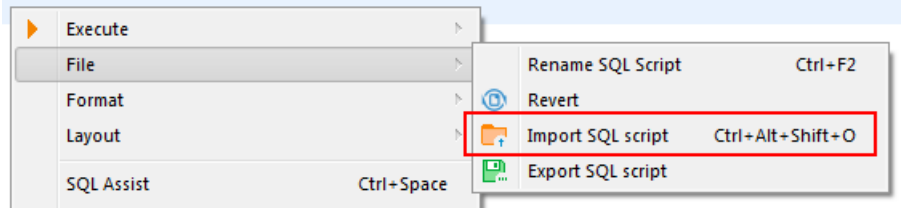
Unsaved data is highlighted in color on the left side of the editor, in addition to having an asterisk in the name of the script.



Loading Scripts

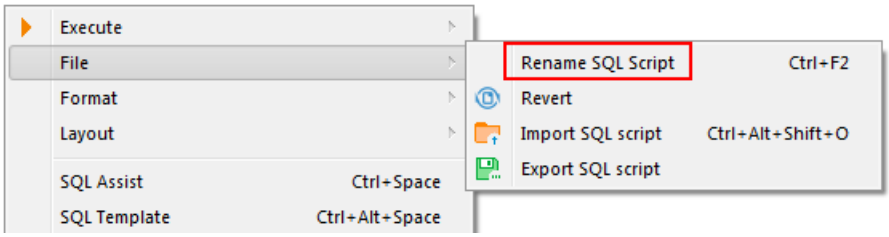
To load a script stored in the file system to the SQL Editor, press `CTRL+SHIFT+O`, or click **SQL Editor -> Import SQL script** on the

main menu, or right-click the script panel and click **File -> Import SQL script** on the context menu:

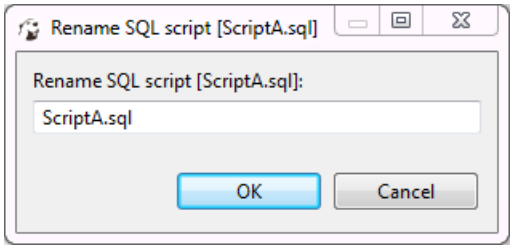


Renaming Scripts

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



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

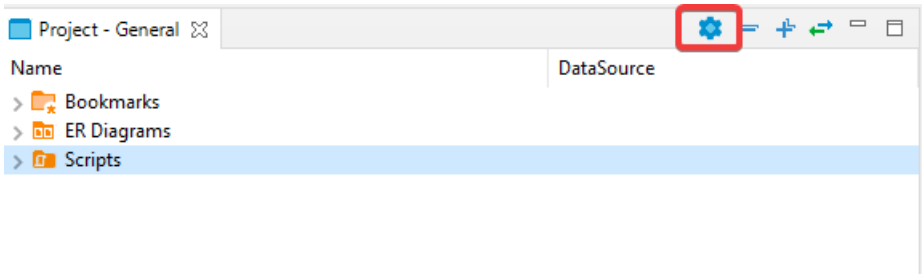


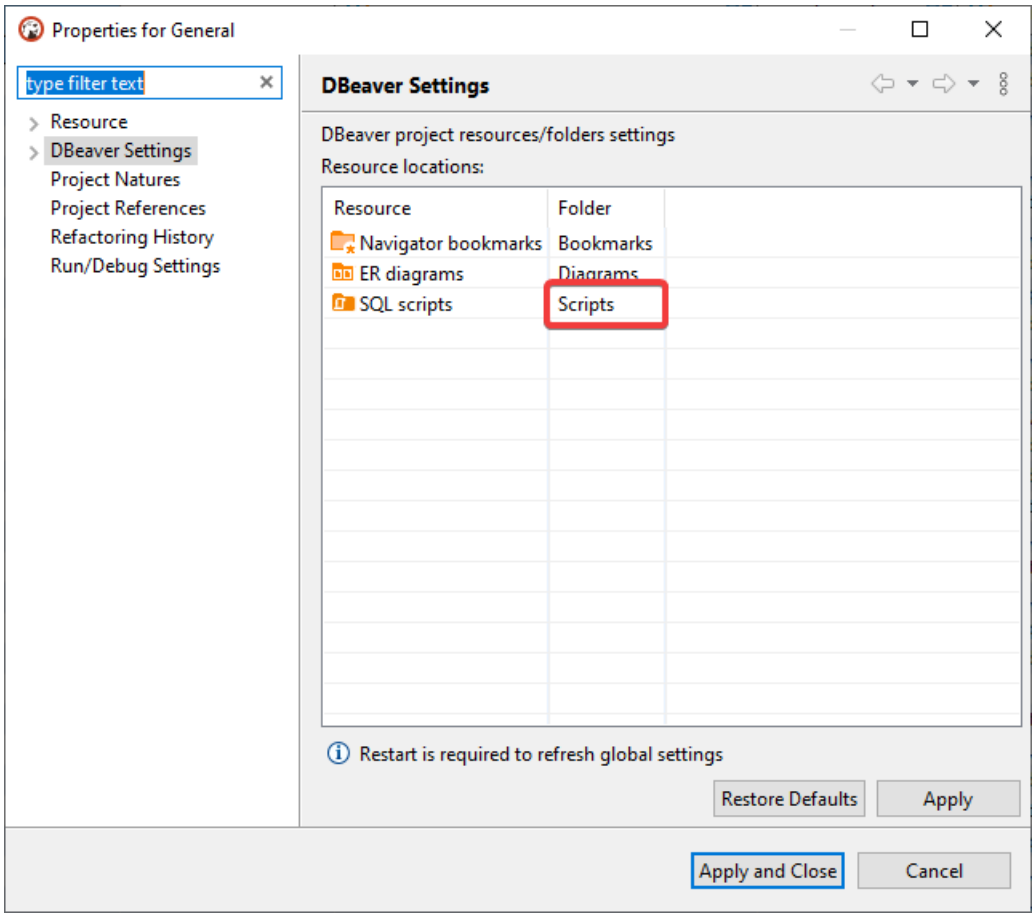
Reverting Changes

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

Changing default scripts directory

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





Adding external directory

You can also link an external directory to your project to either save your scripts into it, or to access scripts that were created outside of DBeaver.
To link an External directory right-click anywhere in the Project Explorer and pick **Create -> Link Folder**. There you can link any directory on your drive to a project. This will allow you to open any externally created scripts through Project explorer and to set this folder as default to save new scripts into.

SQL Console

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

Client Side Commands

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

DBeaver supports the following commands:

Command	Database	Description
`@set var = value`	All	Sets a script variable. You can use expressions as a value. Variables can be used as SQL queries input parameters. For more information see Dynamic parameter bindings
`@unset var`	All	Unsets a script variable.
`@echo message`	All	Prints message to output log. You can use a macro in a message (for example `${var}`).
`@include fileName`	All	- Executes a specified file name, - Can be used in scripts, - Opens a new SQL console with the specified file and processes SQL queries as in a regular SQL editor.
`@export { ... }`	All	Opens the data transfer wizard with predefined settings. For more information see the main article.
`source fileName`	MySQL	The same as `@include` but in MySQL CLI syntax
`define var = value`	Exasol	The same as `@set` but in Exasol EXAPlus syntax.

Export Command

The `@export` command allows you to open the data transfer wizard with prefilled settings.

It may be helpful in case you're editing several SQL queries and want to quickly perform the export of the produced results without creating any additional data transfer tasks.

Disclaimer: This article describes supported settings by the `@export` command, their purpose, and allowed values. Generally, this article contains every setting accessible in the data transfer wizard. Settings are written in the order they appear in the wizard, so you can always look at the wizard to quickly locate any of these settings.

The body of the command consists of JSON text, which looks like this:

```
{
    "type": <ID of the processor>,
    "producer": {
        <producer settings>
    },
    "consumer": {
        <consumer settings>
    },
    "processor": {
        <processor-specific settings>
    },
}
```

Due to certain limitations, it must be written on a single line, without line delimiters:

```
@export { "type": "csv", "producer": { ... }, "consumer": { ... }, "processor": { ... } }
```

Here's the description of each attribute:

Attribute	Description
`type`	Type of the processor.
`producer`	Settings that affect how the data is extracted. See the full table of supported settings in the main section.
`consumer`	Settings that affect how the data is transformed before processing. See the full table of supported settings in the main section.
`processor`	Settings that affect how the data is processed. This includes formatting, transformations, etc. These settings are specific to the processor specified by the `type` attribute. See the full table of supported processors in the main section.

Producer Settings

Id	Name	Description	Type	Default Value	Allowed Values
`extractType`	Extract type	Data extraction mode. Denotes whether a single query or multiple segmented queries should be used to extract data.	String	`SINGLE_QUERY`	`SINGLE_QUERY`, `SEGMENTS`
`segmentSize`	Segment size	Specifies how many rows are read per segment during data extraction. See `extractType`	Integer	`100000`	*Any*
`fetchSize`	Fetch size	Number of rows to fetch per one server round trip. May greatly affect extraction performance.	Integer	`10000`	*Any*
`openNewConnections`	Open new connection(s)	Open new physical connection for data reading. Makes great sense if you are going to continue to work with your database during the export process.	Boolean	`true`	*Any*

Id	Name	Description	Type	Default Value	Allowed Values
`queryRowCount`	Select row count	Query row count before performing export. This will let you track export progress but may cause performance faults in some cases.	Boolean	`true`	*Any*

Consumer Settings

Id	Name	Description	Type	Default Value	Allowed Values
`formatterProfile`	Formatting Profile	Specifies the profile used for formatting data.	String	*<empty>*	*Any*
`valueFormat`	Value Formatting	Specifies how the data is interpreted.	String	`UI`	`UI`, `EDIT`, `NATIVE`
`lobExtractType`	Binaries Policy	Specifies how binaries are processed.	String	`INLINE`	`SKIP`, `FILES`, `INLINE`
`lobEncoding`	Binaries Encoding	Specifies how binaries are encoded.	String	`BINARY`	`BASE64`, `HEX`, `BINARY`, `NATIVE`
`outputClipboard`	Copy to Clipboard	Specifies that the data should be copied to the clipboard rather written to files on a disk.	Boolean	`false`	*Any*
`outputFolder`	Output Directory	Output directory pattern. Specifies there the output files should be located.	String	*N/A*	*Any*
`outputFilePattern`	Output Filename	Output filename pattern.	String	`${table}_${timestamp}`	*Any*
`outputEncoding`	Output Encoding	Specifies the file encoding.	String	`UTF-8`	*Any*
`outputEncodingBOM`	Insert BOM	Specifies whether the byte order mark should be written to the output file. Common for encoding such as `UTF-16LE`, `UTF-16BE`, `UTF-32LE`, and `UTF-32LE`.	Boolean	`false`	*Any*
`outputTimestampPattern`	Timestamp Pattern	Pattern used for the `${timestamp}` variable in `outputFolder` and `outputFilePattern`.	String	`yyyyMMddHHmm`	*Any*
`appendToFile`	Append to the end of the file	If file already exists, appends data at end of it. Only works against compatible processors.	Boolean	`false`	*Any*
`useSingleFile`	Write to the single file	Write all streams to the single file. Only works against compatible processors.	Boolean	`false`	*Any*
`compressResults`	Compress	Specifies whether the output file should be compressed using ZIP.	Boolean	`false`	*Any*

Id	Name	Description	Type	Default Value	Allowed Values
`splitOutFiles`	Split output file	Specifies whether the output file should be split using the `maxOutFileSize` threshold. If size exceeds this threshold, a separate file is created and so on.	Boolean	`false`	*Any*
`maxOutFileSize`	Maximum file size	Maximum size of a single file. See `splitOutFiles`	Integer	`10000000`	*Any*

Processor Settings

CSV (`csv`)

Id	Name	Description	Type	Default Value	Allowed Values
`extension`	File extension		String	`csv`	*Any*
`delimiter`	Delimiter	Column delimiter. You can use special characters \ + t,n,r	String	`,`	*Any*
`rowDelimiter`	Row delimiter	Row delimiter. Default is system-specific line feed delimiter. You can use special characters \ + t,n,r	String	`default`	`default`, `\n`, `\r`, `\r\n`, `\n\r`
`header`	Header	CSV header settings	String	`top`	`none`, `top`, `bottom`
`headerFormat`	Header format	Header format	String	`label`	`label`, `description`, `both`
`escape`	Characters escape	Bad characters escaping model (surrounded with quotes or escaped with '\' character)	String	`quotes`	`quotes`, `escape`
`quoteChar`	Quote character	Character which will be used to quote strings (space means no quote)	String	`"`	*Any*
`quoteAlways`	Quote always	Quote all cell values. Cannot be used with "quoteNever"	String	`disabled`	`disabled`, `all`, `strings`, `all but numbers`, `all but nulls`
`quoteNever`	Quote never	Do not quote cell values. Cannot be used with "quoteAlways"	Boolean	`false`	*Any*
`nullString`	NULL string	String which will be used instead of NULL values	String	*<empty>*	*Any*
`formatNumbers`	Format numbers	Format numeric values using locale settings	Boolean	`false`	*Any*

DbUnit (`dbunit`)

Id	Name	Description	Type	Default Value	Allowed Values
`upperCaseTableName`	Force upper case table name		Boolean	`true`	*Any*
`upperCaseColumnNames`	Force upper case column names		Boolean	`true`	*Any*
`extension`	File extension		String	`xml`	*Any*
`includeNullValues`	Include NULL values in export		Boolean	`true`	*Any*
`nullValueString`	Replace NULL values with		String	`[NULL]`	*Any*

HTML (`html`)

Id	Name	Description	Type	Default Value	Allowed Values
`extension`	File extension		String	`html`	*Any*
`tableHeader`	Output table header	Output query or table name as first row in generated table	Boolean	`true`	*Any*
`columnHeaders`	Output column headers	Output column names as extra row in generated table	Boolean	`true`	*Any*
`extractImages`	Images	Extract images to graphic files	Boolean	`true`	*Any*

JSON (`json`)

Id	Name	Description	Type	Default Value	Allowed Values
`printTableName`	Print table name		Boolean	`true`	*Any*
`formatDateISO`	Format dates in ISO 8601		Boolean	`true`	*Any*
`extension`	File extension		String	`json`	*Any*

Markdown (`markdown.table`)

Id	Name	Description	Type	Default Value	Allowed Values
`extension`	File extension		String	`md`	*Any*
`nullString`	NULL string	String which will be used instead of NULL values	String	*<empty>*	*Any*
`formatNumbers`	Format numbers	Format numeric values using locale settings	Boolean	`false`	*Any*
`showHeaderSeparator`	Show header separator	Print header separator (---). Required for GitHub markdown.	Boolean	`true`	*Any*
`confluenceFormat`	Confluence format	Use Confluence format (special format of header and no separator line)	Boolean	`false`	*Any*

SQL (`sql`)

Id	Name	Description	Type	Default Value	Allowed Values
`includeAutoGenerated`	Include generated columns	Include auto-generated columns (e.g. auto-increment) in SQL INSERT	Boolean	`false`	*Any*
`extension`	File extension		String	`sql`	*Any*
`nativeFormat`	Native date/time format	Use native date/time format in INSERT statements	Boolean	`true`	*Any*
`omitSchema`	Omit schema name	Omit schema/catalog name in INSERT statements	Boolean	`false`	*Any*
`rowsInStatement`	Data rows per statement	Number of data rows per single insert statement	Integer	`10`	*Any*
`lineBeforeRows`	Insert line before rows	Insert line feed before values (for multi-row inserts)	Boolean	`true`	*Any*
`keywordCase`	Keyword case	You can choose lower or upper keyword case	String	`upper`	`upper`, `lower`

Id	Name	Description	Type	Default Value	Allowed Values
`identifierCase`	Identifier case	You can choose lower or upper keyword case for table and column names	String	`as is`	`as is`, `upper`, `lower`
`upsertKeyword`	Upsert keyword	You can choose different upsert keywords	String	`INSERT`	`INSERT`, `INSERT ALL`, `UPDATE OR`, `UPSERT INTO`, `REPLACE INTO`, `ON DUPLICATE KEY UPDATE`, `ON CONFLICT`
`insertOnConflict`	On conflict expression	Expression for the end of the statement. Enter the required value in this field. This is database specific setting	String	*<empty>*	*Any*

Source code (`source.code`)

Id	Name	Description	Type	Default Value	Allowed Values
`language`	Language	Programming languages	String	`PHP < 5.4`	`PHP < 5.4`, `PHP 5.4+`
`formatDateISOPHP`	Format dates in ISO 8601		Boolean	`true`	*Any*
`extension`	File extension		String	`php`	*Any*
`quoteChar`	Quote character	Character which will be used to quote strings	String	`"`	`"`, `'`
`rowDelimiter`	Row delimiter	Row delimiter. Default is system-specific line feed delimiter. You can use special characters \ + t,n,r	String	`default`	`default`, `\n`, `\r`, `\r\n`, `\n\r`

TXT (`txt`)

Id	Name	Description	Type	Default Value	Allowed Values
`extension`	File extension		String	`txt`	*Any*
`batchSize`	Batch size		String	`200`	*Any*
`minColumnLength`	Min column length		String	`3`	*Any*
`maxColumnLength`	Max column length		String	`0`	*Any*
`showNulls`	Show NULLs		Boolean	`false`	*Any*
`delimHeader`	Show header delimiter		Boolean	`true`	*Any*
`delimLeading`	Show leading delimiter		Boolean	`true`	*Any*
`delimTrailing`	Show trailing delimiter		Boolean	`true`	*Any*
`delimBetween`	Show in-between delimiter		Boolean	`true`	*Any*

XML (`xml`)

Id	Name	Description	Type	Default Value	Allowed Values
`extension`	File extension		String	`xml`	*Any*

XLSX (`xlsx`)

Id	Name	Description	Type	Default Value	Allowed Values

Id	Name	Description	Type	Default Value	Allowed Values
`extension`	File extension		String	`xlsx`	*Any*
`rownumber`	Row number(s)	Set row index as first column	Boolean	`false`	*Any*
`border`	Border style	Cell borders style	String	`THIN`	`NONE`, `THIN`, `THICK`
`nullString`	NULL string	String which will be used instead of NULL values	String	*<empty>*	*Any*
`header`	Column names as header	Use column name as first row	Boolean	`true`	*Any*
`headerfont`	Header row font	First row font properties	String	`BOLD`	`NONE`, `BOLD`, `ITALIC`, `STRIKEOUT`, `UNDERLINE`
`trueString`	Boolean string TRUE	String which will be used instead of TRUE boolean values	String	`true`	*Any*
`falseString`	Boolean string FALSE	String which will be used instead of FALSE boolean values	String	`false`	*Any*
`exportSql`	Export SQL	Export SQL to a second sheet	Boolean	`false`	*Any*
`splitSqlText`	Split SQL Text	Split exported SQL on rows by CR	Boolean	`false`	*Any*
`splitByRowCount`	Max row on sheet	Split by row count	Integer	`1048575`	*Any*
`splitByColNum`	Column group	Column number for grouping rows on sheet by column value	Integer	`0`	*Any*
`dateFormat`	Excel date format	Excel date and time format (e.g. m/d/yy h:mm) it can be changed in Excel application	String	`m/d/yy`	`m/d/yy`, `d-mmm-yy`, `d-mmm`, `mmm-yy`, `h:mm AM/PM`, `h:mm:ss AM/PM`, `h:mm`, `h:mm:ss`, `m/d/yy h:mm`

PostgreSQL Debugger

Prerequisites for Debugging

To enable interactive debugging of PL/pgSQL procedures on a Postgres server, you need to use the _plugin_debugger_. The _plugin_debugger_ is a typical interactive debugger delivered as an extension. It requires a shared library preload in Postgres to operate the shared_preload_libraries parameter in the settings. The debugger is developed and maintained by EDB. Its source code is available for examination and improvement.

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

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

Installation

PostgreSQL 12 on Ubuntu-based distros

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

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

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

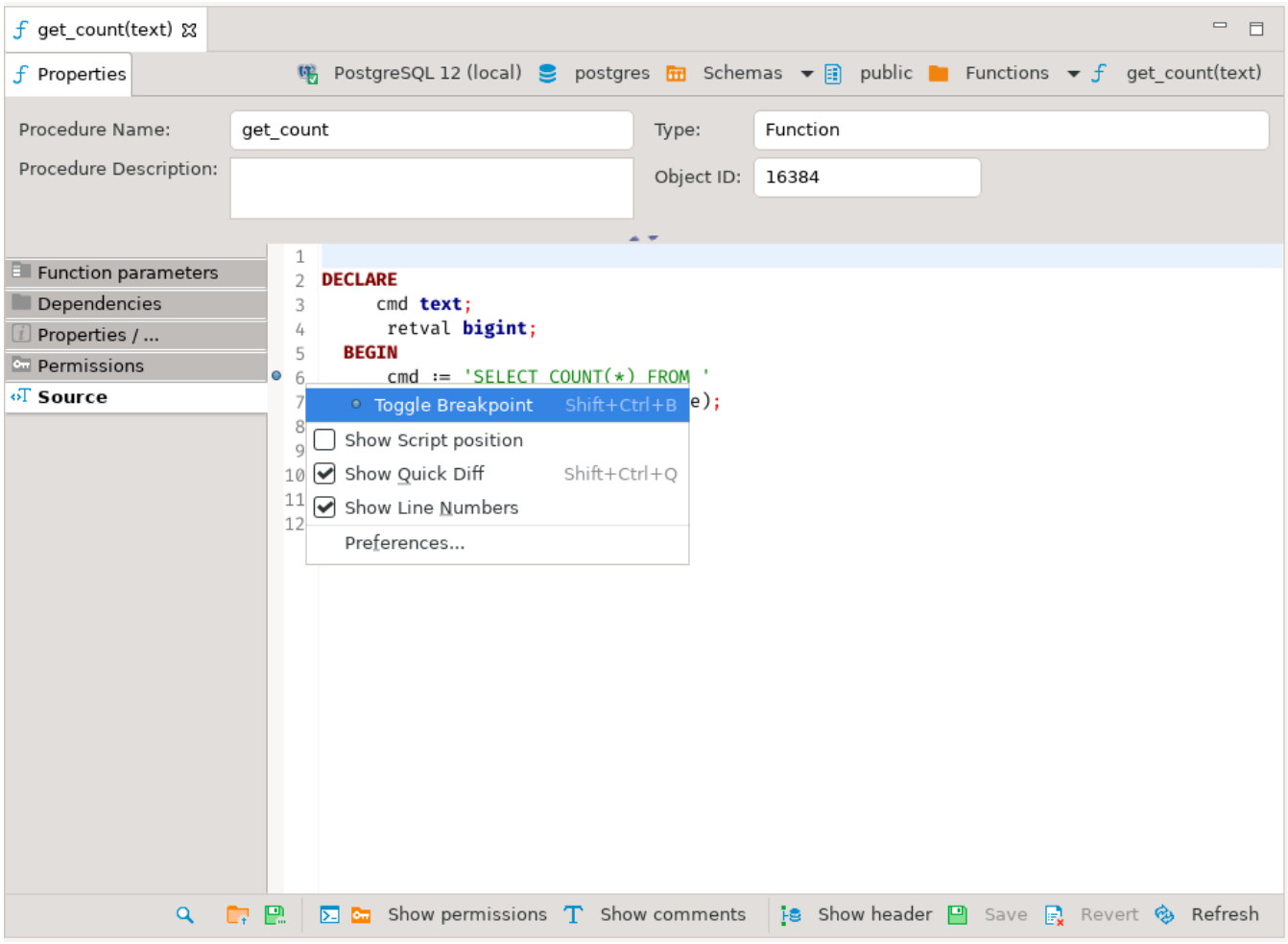
```
CREATE EXTENSION pldbgapi;
```

Installation from source code

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

Running debugger in the DBeaver interface

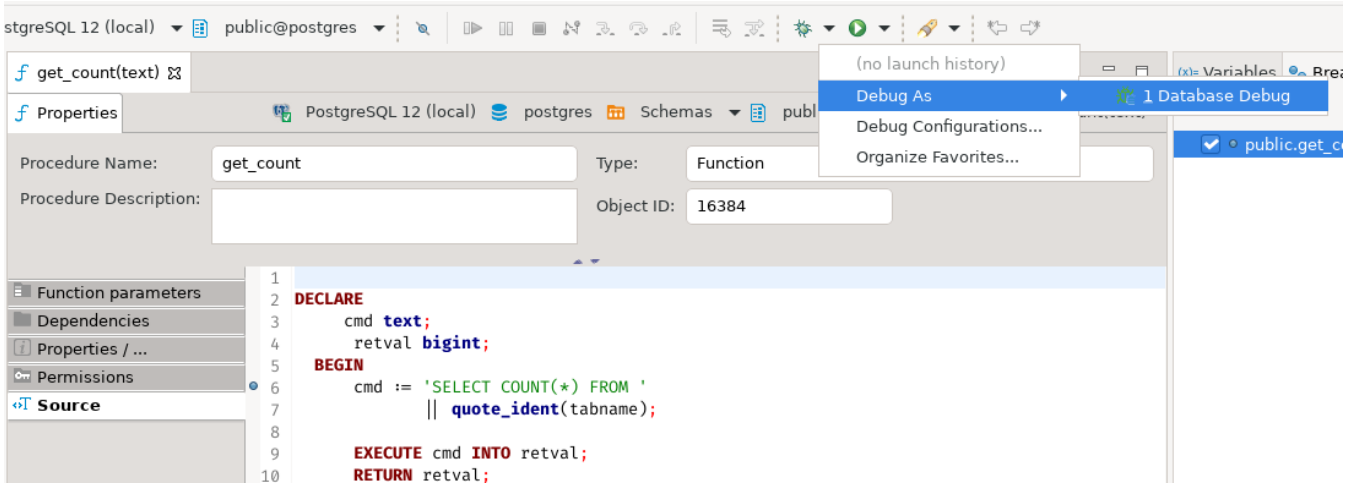
Open the source code of the function you want to debug. To toggle breakpoints, place the caret on the line you want the function to be stopped at and use a shortcut `Shift + Control + B`. Alternatively, you can toggle the breakpoint with your mouse by clicking on a ruler, as demonstrated in the screenshot below:



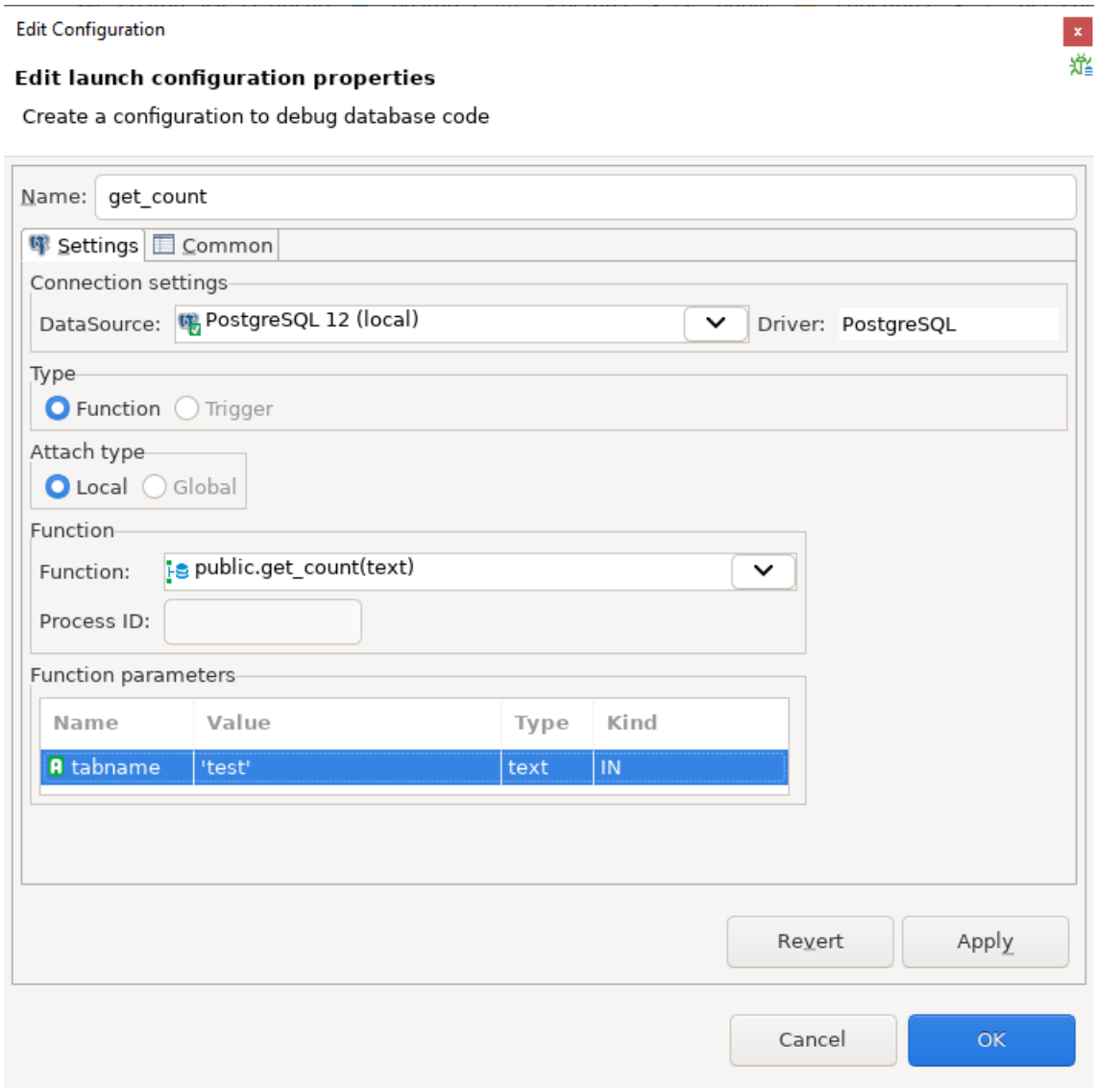
NB. You must only toggle the breakpoints when *Show header* option is **not** on.

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

Debug As -> Database Debug:

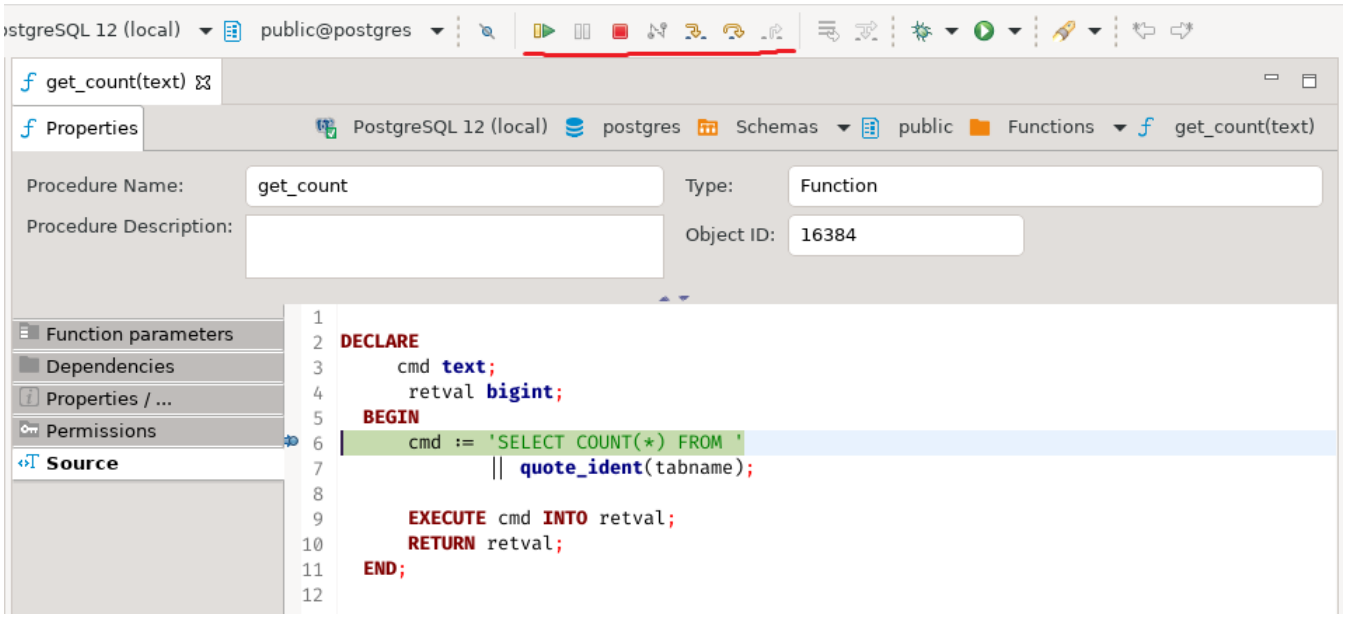


The *Edit Configuration* dialog opens. Set up input values in the table *Function parameters*.



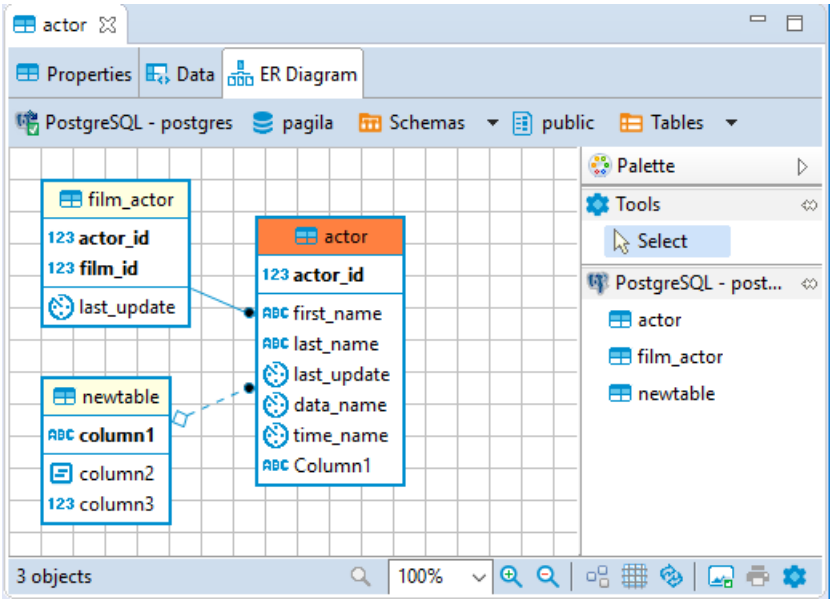
Click on _OK_ button, and you are ready to go!

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



ER Diagrams

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



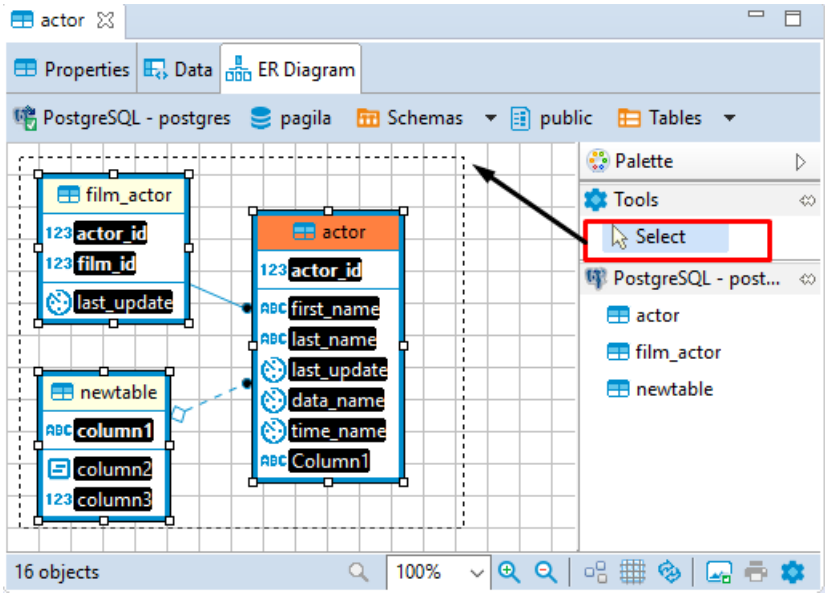
Entity Relation Diagrams (ERD) are graphic presentations of database entities and the relations between them. DBeaver allows you to view the diagrams of existing tables and whole database schemas, see Database Structure Diagrams. DB also allows the creation of custom diagrams, see Custom Diagrams.
By default DBeaver uses IDEF1X notation.

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

Selection of Elements in Diagrams

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

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



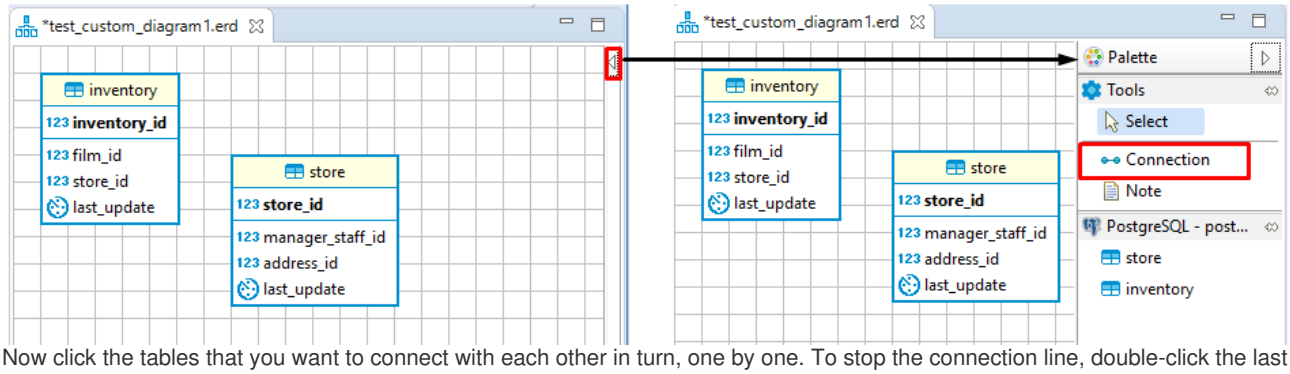
Structure Adjustment

NOTE: All changes to existing database schemas cannot be saved and are intended for exploration purposes only.
You can do the following structural changes in the diagrams.

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

location.

- Auto-arrange tables into a compact view after manual rearrangements: click the **Arrange Diagram** ( ) in the toolbar or on the context menu (right-click anywhere on the diagram tab).
- (Available for Custom Diagrams only) - connect tables with a connector: click the **Show Palette** button ( ) in the upper-left corner of the diagram tab and then, in the Palette panel, click **Connection**:



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

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

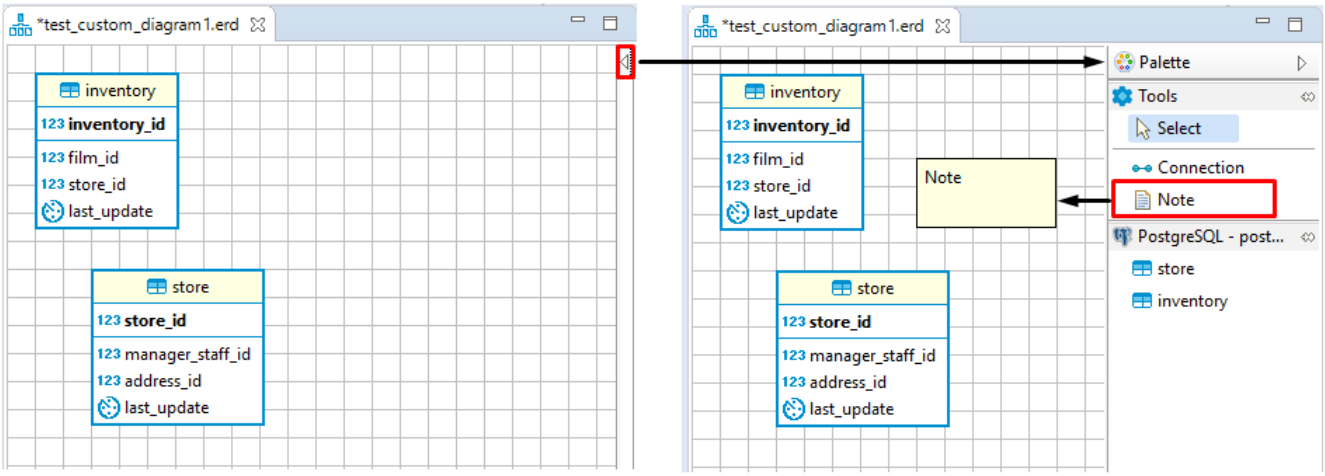
- Enable/disable the diagram grid: Click **Toggle Grid** ( ) in the toolbar.
- Modify attributes visibility: Right-click the diagram and, on the context menu, click **Show Attributes** and then select one of the options:
  - **All** - all attributes
  - **Any keys** - primary and foreign keys
  - **Primary key** - only primary keys
  - **None** - no attributes
- Modify attributes presentation: Right-click the diagram and, on the context menu, click **View Styles** and then select one of the options:
  - **Show Icons**
  - **Show Data Types**
  - **Show Nullability**
  - **Show Comments**
  - **Show Fully qualified names**
- Change the color of the entities/notes: Right-click the header of the entity or comment and then click **Set color** on the context menu. Then you can select the color and click **OK**.
- For elements located in front of/behind others, bring an element to the front or send it to the back: Right-click the element and then click **Bring to front** / **Send to back** on the context menu.
- Zoom the diagram in/out: Click the **Zoom In**/**Zoom Out** buttons or choose the scaling value in the dropdown list in the toolbar: 100%

Refresh

To see changes made by others to the database schema, you might need to refresh the diagram: click **Refresh Diagram** ( ) in the toolbar.

Notes

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



Search in Diagram Entities

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

Diagram Export

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

Diagram Printing

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

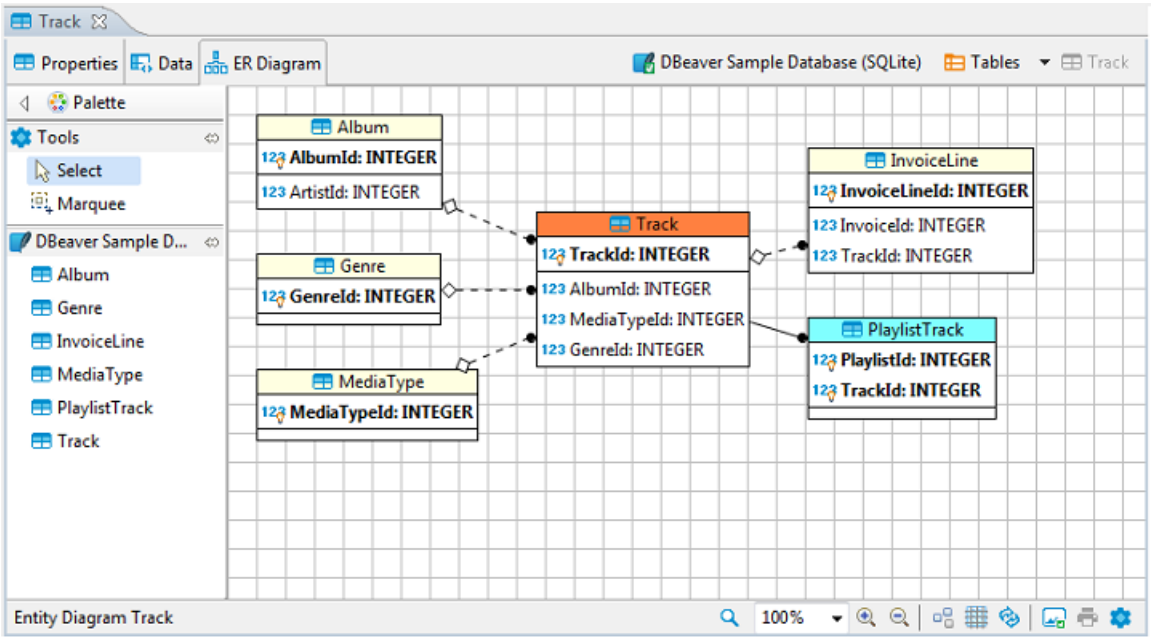
Settings

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

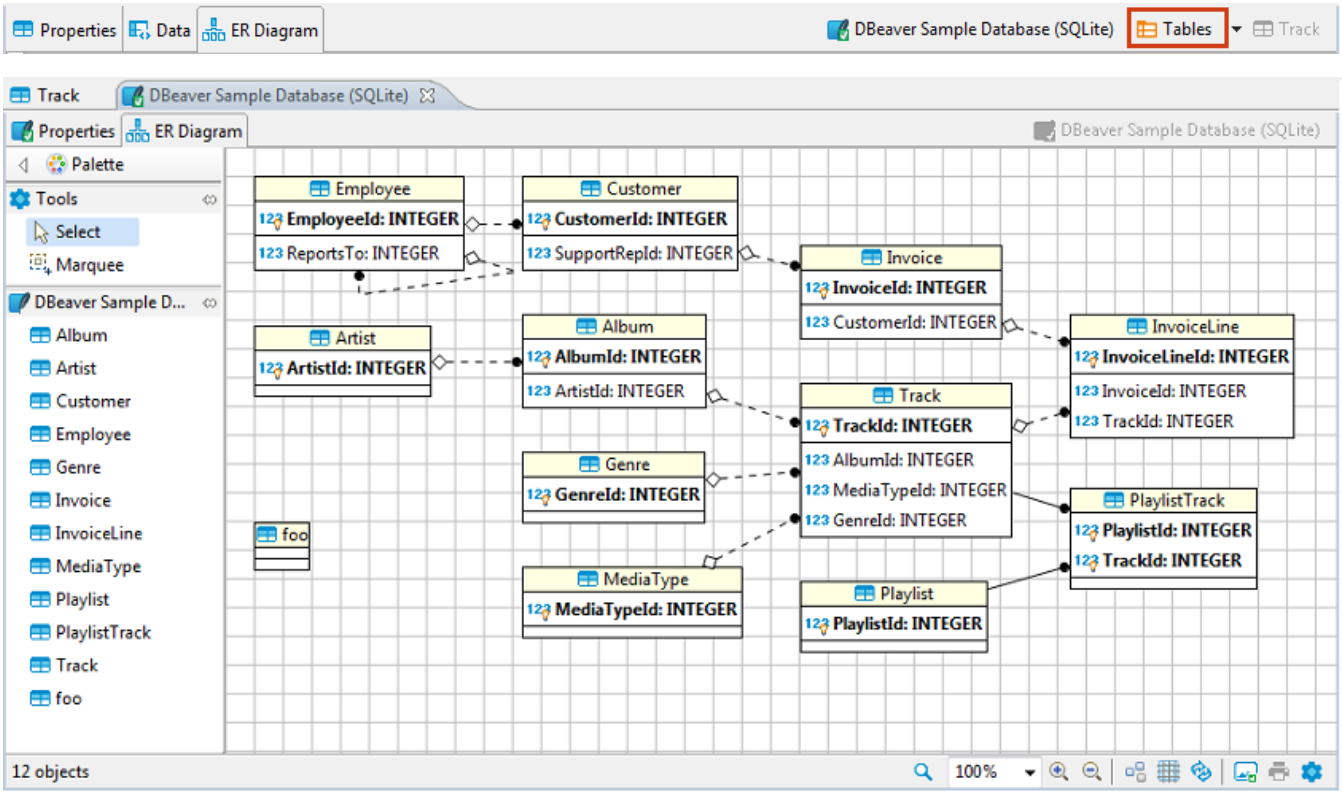
Database Structure Diagrams

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

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



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



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

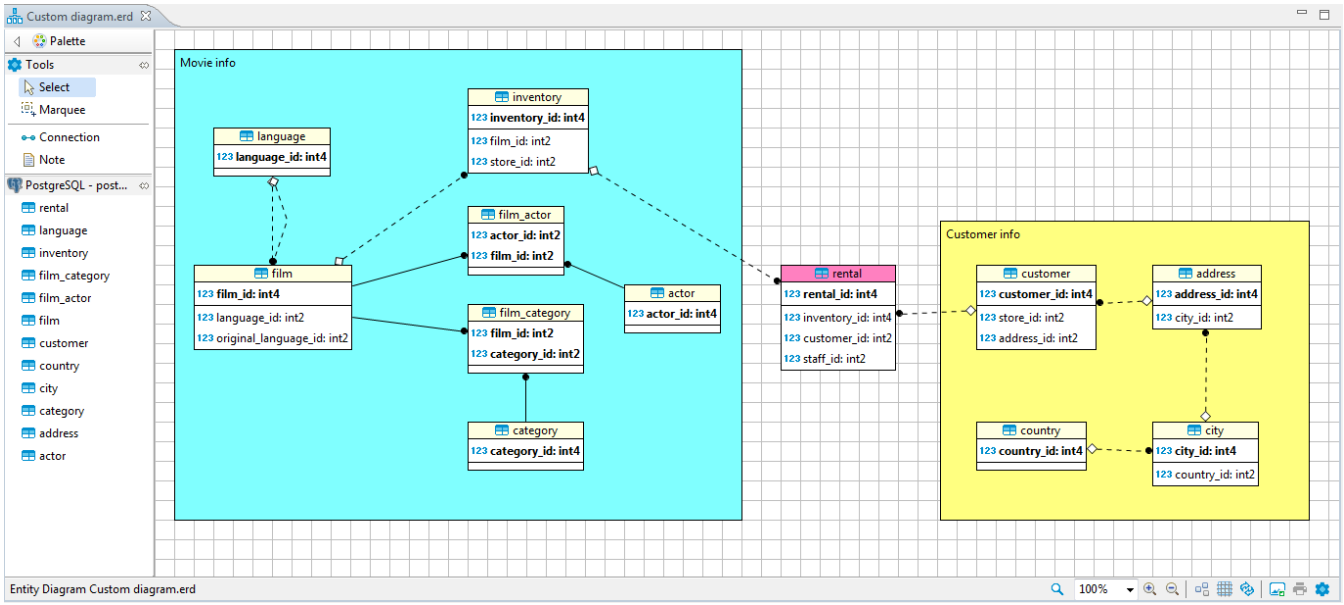
Relationship Notation

Lines representing the relationship between tables can look different depending on the nature of the relationship. Please note that any line can have only one beginning and one end. Even for one-to-many and other relationships that imply otherwise. In those situations you will just see more than one line.

Notation	Description
	The solid line means that the foreign key column is also a primary key in a referencing table
	The dashed line means that the foreign key column is not a primary key in a referencing table
	If the relationship between two tables is other than one-to-one, you will see multiple lines that all start at the same one point and all merge at the end point.
	The black dot represents the beginning of the line and is attached to a table that has a foreign key referencing another table
	The white rectangle represents the end of the line and is attached to a referenced table. It only appears at the end of the dashed line

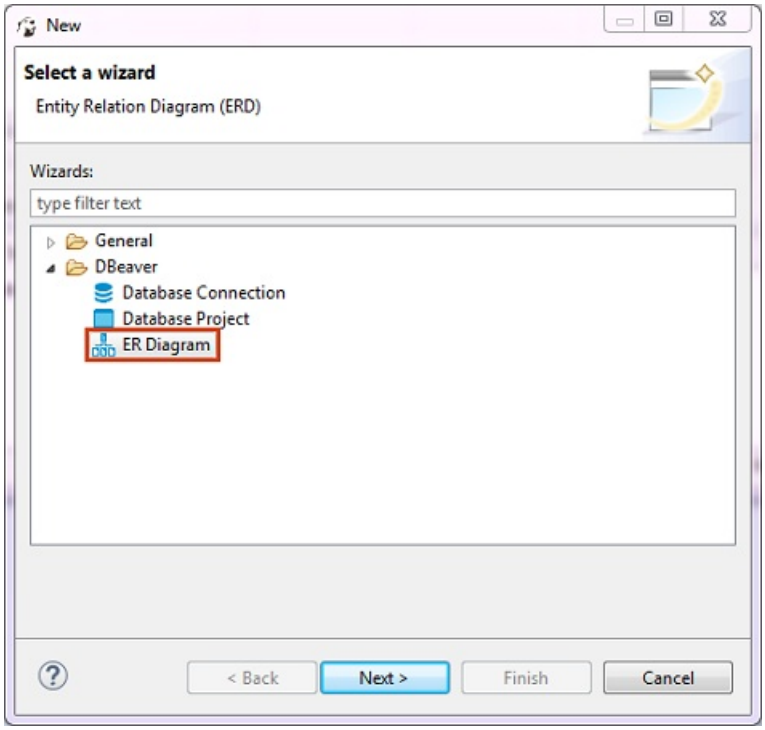
Custom Diagrams

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

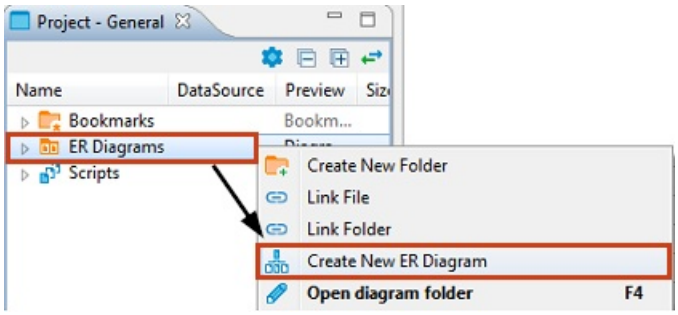


You can create a custom diagram in one of the ways:

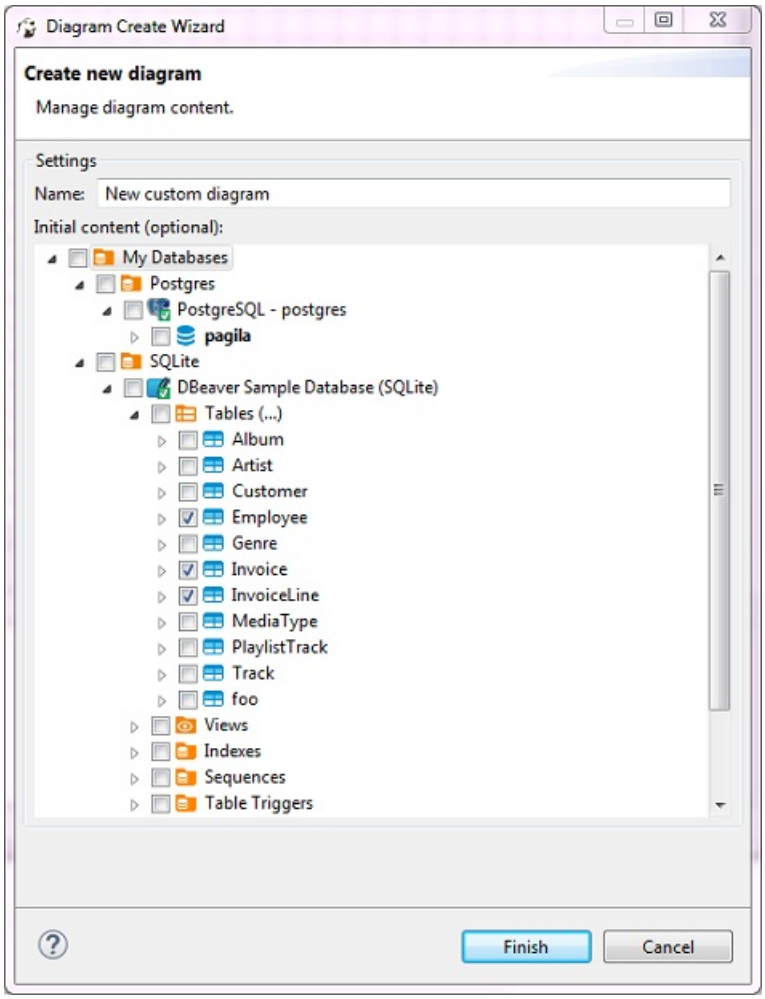
1. On the DBeaver main menu, click **File -> New**. Then in the new diagram wizard, click **DBeaver -> ER Diagram**, and then **Next**:



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



In both cases, in the Diagram Create Wizard, specify the diagram name and (optional) choose the initial diagram contents (set of tables):



The new diagram appears in a separate editor. Now you can drag-n-drop any number of tables into it. You can add tables from different connections as well as from different database types (for example, combine Oracle and MySQL tables in one and the same diagram).

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

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

Undo/redo functions are fully supported in diagram editing.

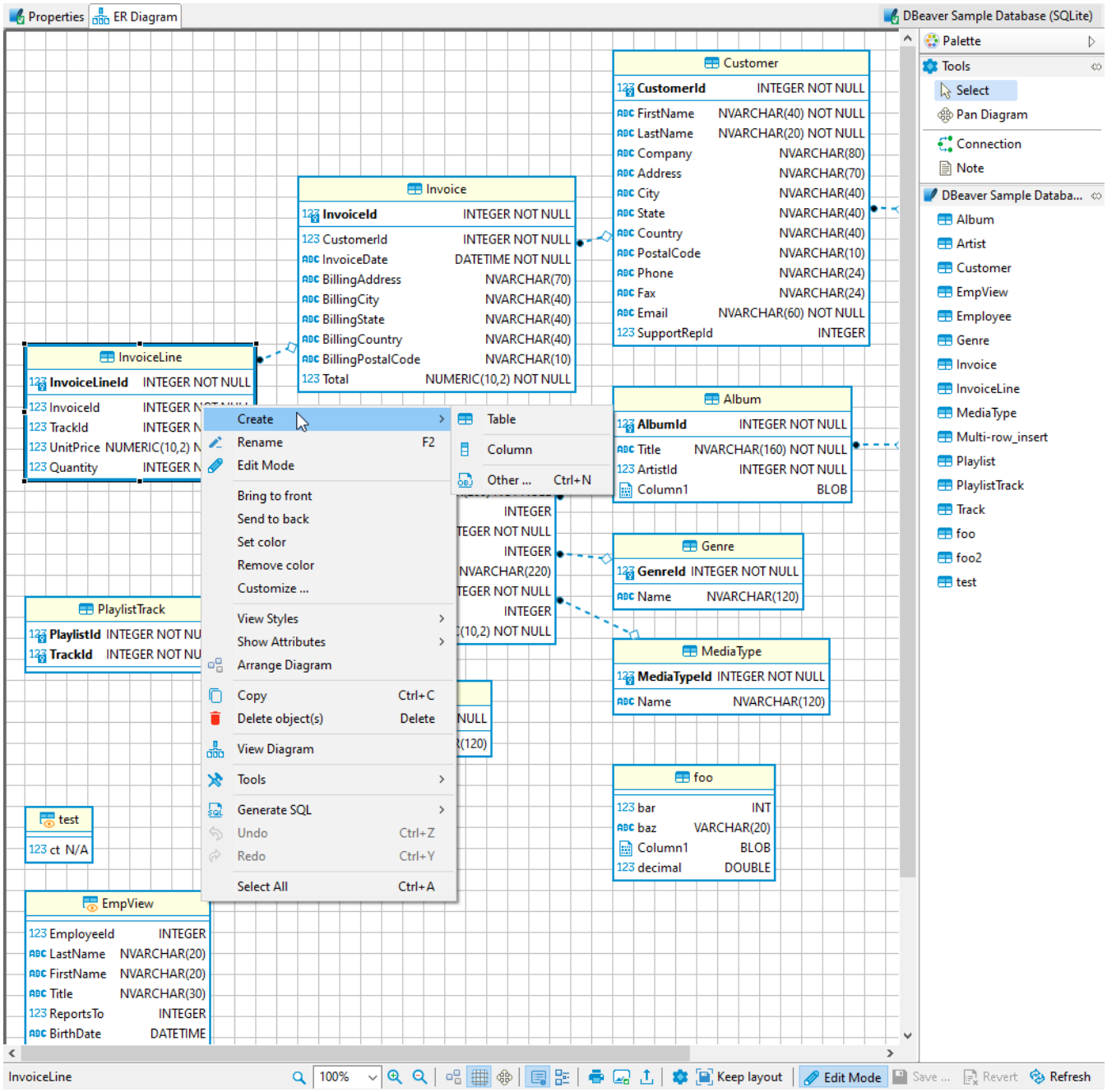
Edit mode

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

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

Working in Edit mode

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



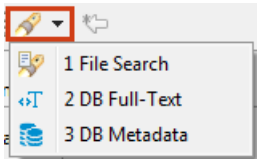
Saving changes to a database is performed by clicking a save button. This will show you a preview of a generated script that you then can execute or cancel. The revert button will cancel all the changes that you did to a diagram after the last save.

Search

DBeaver provides:

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

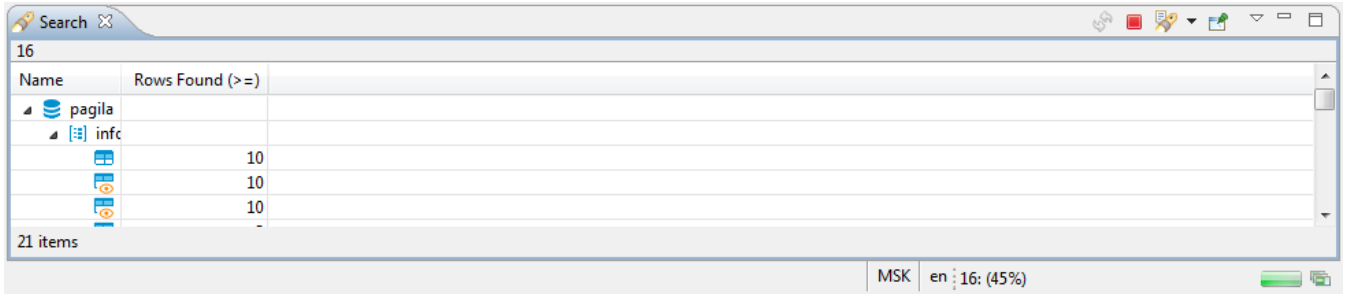
To use search, click the Search button on the main toolbar:



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

Search View

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



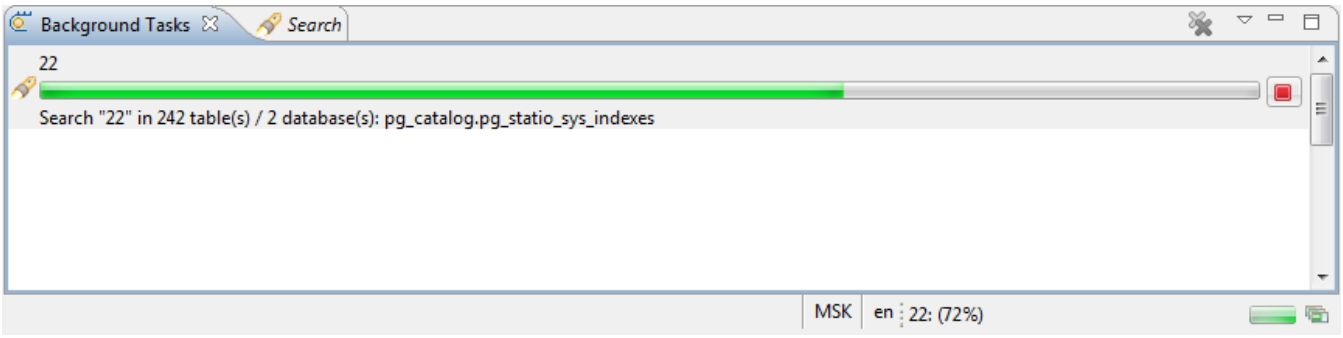
The view contains a toolbar that provides common tools for all types of search as well as specific tools for the File Search type. The following are common tools:

Button	Name	Description
	Run the Current Search Again	Refreshes the search results
/	**Cancel Current Search**	- Active state (red) indicates that the search is still in progress and appears if the search takes some time to complete. Clicking the button in this state stops the current search. - Inactive state (grey) indicates that the search is complete. The button in this state is non-actionable.
	Show Previous Searches	- Clicking the button itself opens the Previous Searches window. - Clicking the arrow next to the button opens a dropdown menu. See the *Search History* section further in this article.
	Pin the Search View	Ties the current search results to the Search view. If you click this button, the current results stay in the view while the results of the next search appear in a new Search view. Otherwise, every new search replaces the previous results with new results.

For information about specific tools of File search, see the File Search article.

If the search is short, the results appear almost instantly. But if it takes some time, the Search view indicates the progress in the following ways:

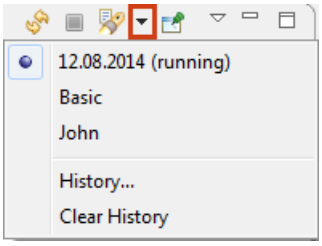
- The **Cancel Current Search** button in the toolbar has the Active state ( )
- The progress bar appears in the bottom-right corner of the view indicating the process:
- The button to show the search progress in a separate view ( ) appears in the bottom-right corner of the view next to the search progress bar. Clicking the button opens the Background Tasks view:



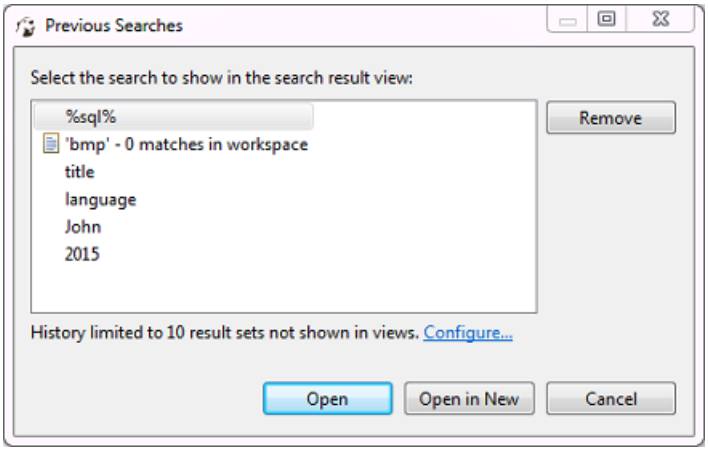
Search History

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

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



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



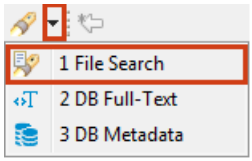
To remove one or more of the previous search queries:

1. Click the **Show Previous Searches** button in the toolbar or click the arrow next to it and then **History** on the dropdown menu. The Previous Searches window opens.
2. Click the query to remove or select several of them by clicking and simultaneously holding the `Ctrl` key.
3. Click **Remove**.

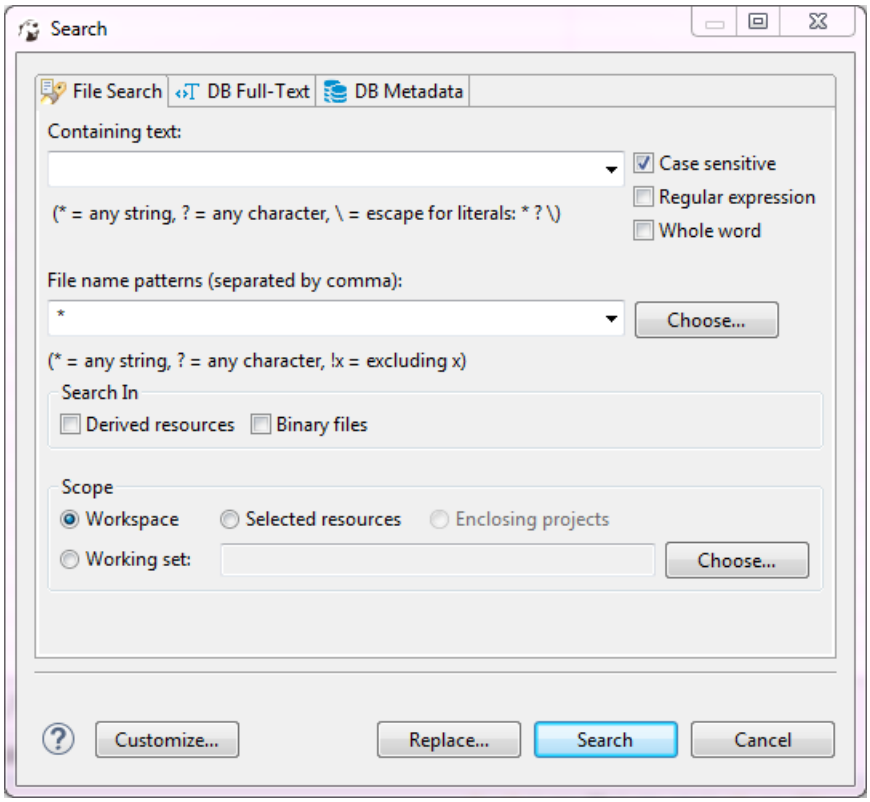
To clear the history by removing all previous queries, click the arrow next to the **Show Previous Searches** button on the toolbar and then click **Clear History** on the dropdown menu.

File Search

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

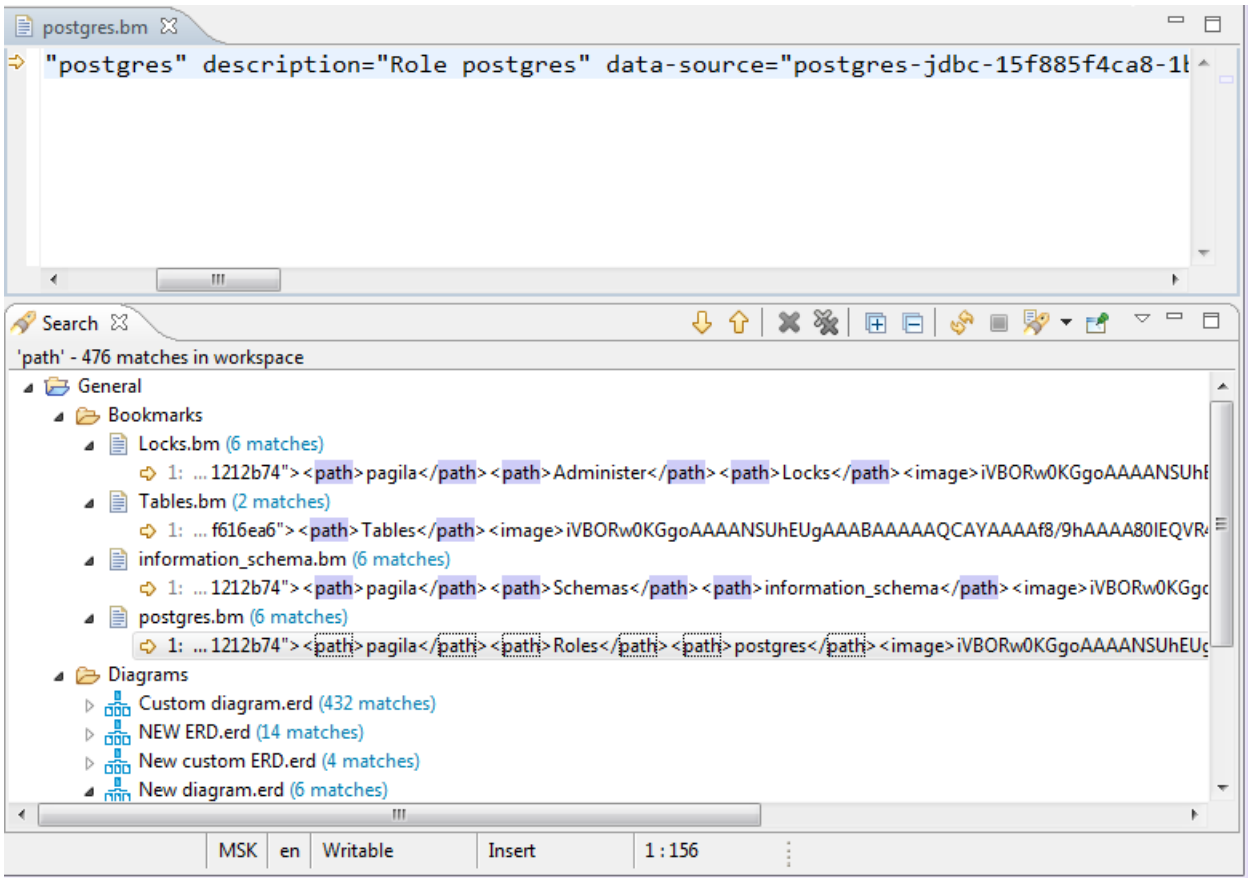


The Search window opens displaying the File Search tab:



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

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



The toolbar of the Search view for File search provides more tools in addition to those available for all search types:

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

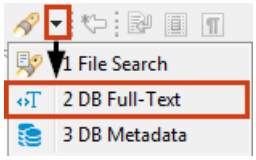
The view also provides a view menu (click the **View Menu** button ( ) in the upper-right corner of the view) that contains the following items:

Menu item	Description
Show as List	Presents the results in the form of list
Show as Tree	Presents the results in the form of tree
Filters	Opens Search Filters dialog box
Preferences	Opens the Preferences window on the Search page

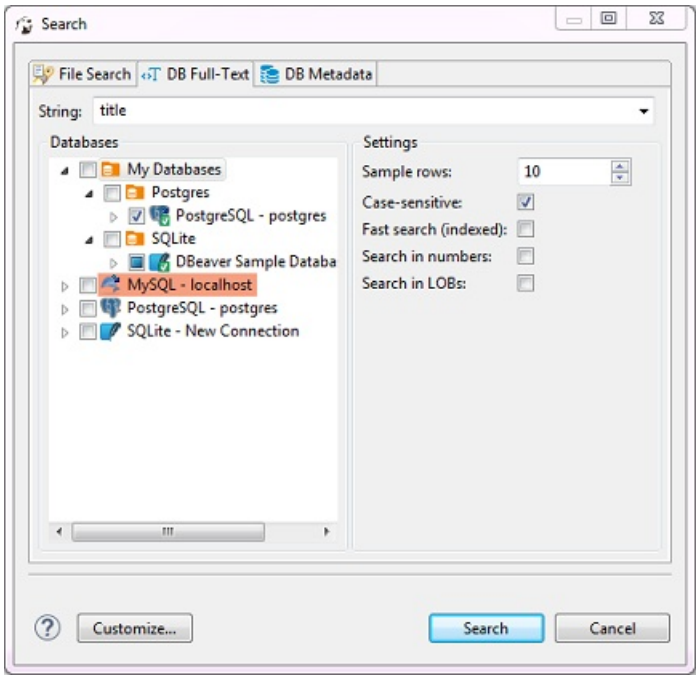
Double-clicking a results row opens it in a separate viewer.

DB Full-Text Search

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



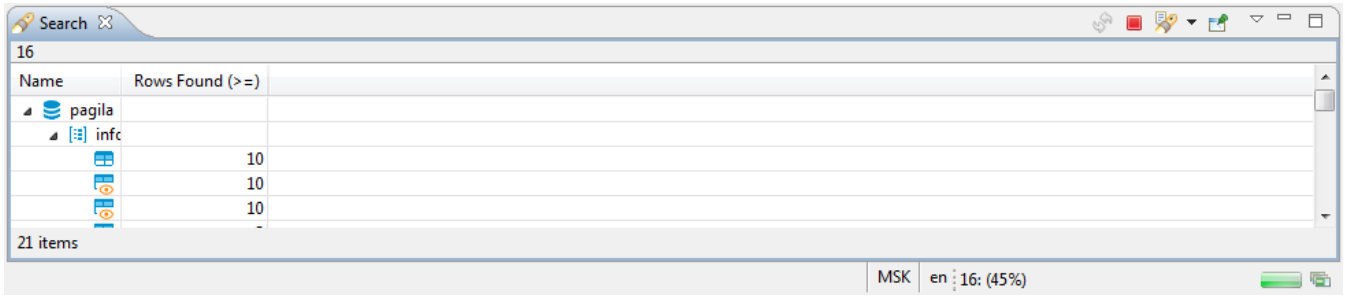
Alternatively, you can click the **Search** button on the main menu and then click the **DB Full-Text** tab in the Search window:



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

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

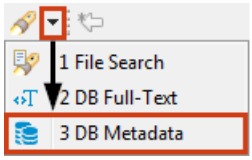
After you click **Search**, the results will open in Search view:



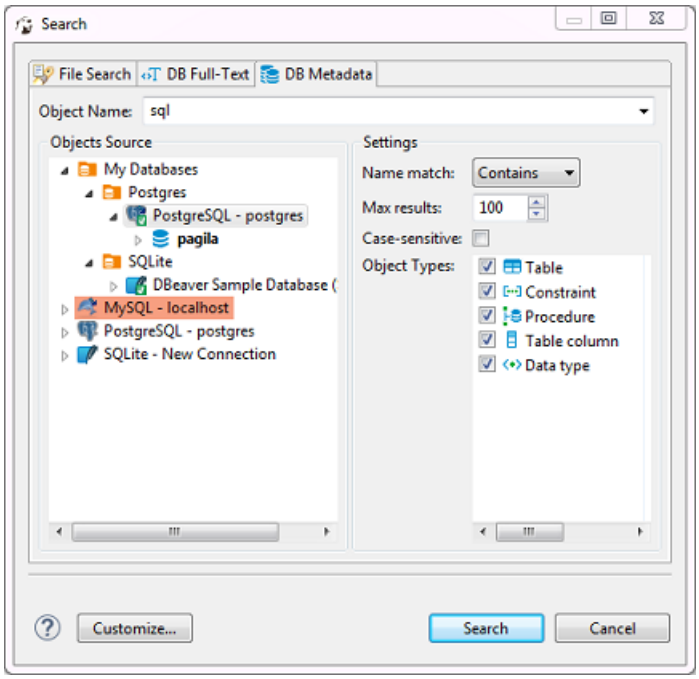
Double-clicking a row in the Search view opens the respective object in a dedicated Database Object editor.

DB Metadata Search

To search for database metadata, click the arrow next to the **Search** button in the main toolbar and then click **DB Metadata Search** on the dropdown menu:

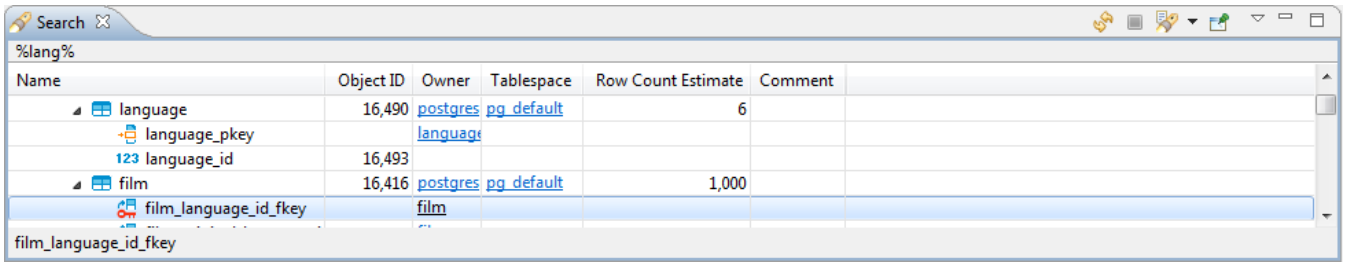


Alternatively, you can click the **Search** button on the main menu and then click the **DB Metadata** tab in the Search window:



Now you need to choose the database connection against which to run the search. You only need to select the database connection(s) in the **Objects Source** field.
In the **Object Types** field, you can select the database objects among which DBeaver will run the metadata search – select or clear the checkboxes.
You can specify that the object name starts with, contains or is similar to the search combination (**Name match** field). You can also set the maximum number of results to display (**Max results** field) and apply **Case-sensitive** search.

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



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

Schema compare

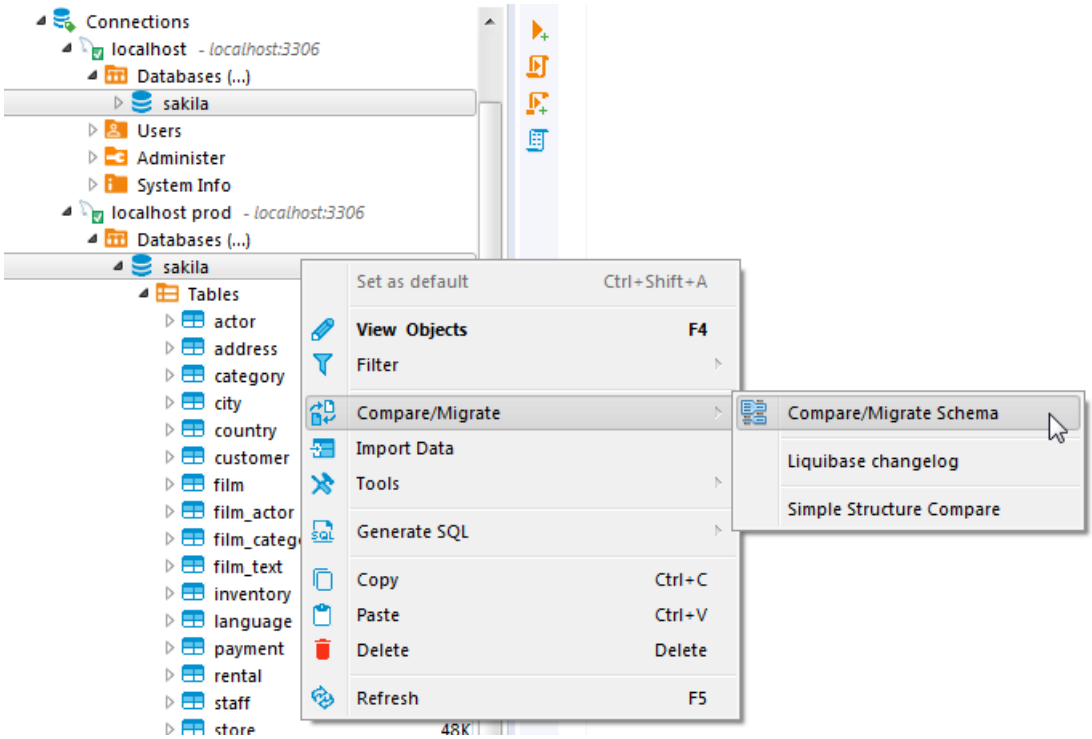
NB: This feature is available in Enterprise and Ultimate editions only.

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

- DDL script (series of create/alter/drop statements)
- Diff diagram (sort of ER diagram)
- Liquibase changelog
- Liquibase change report (JSON, YAML, or plaintext)

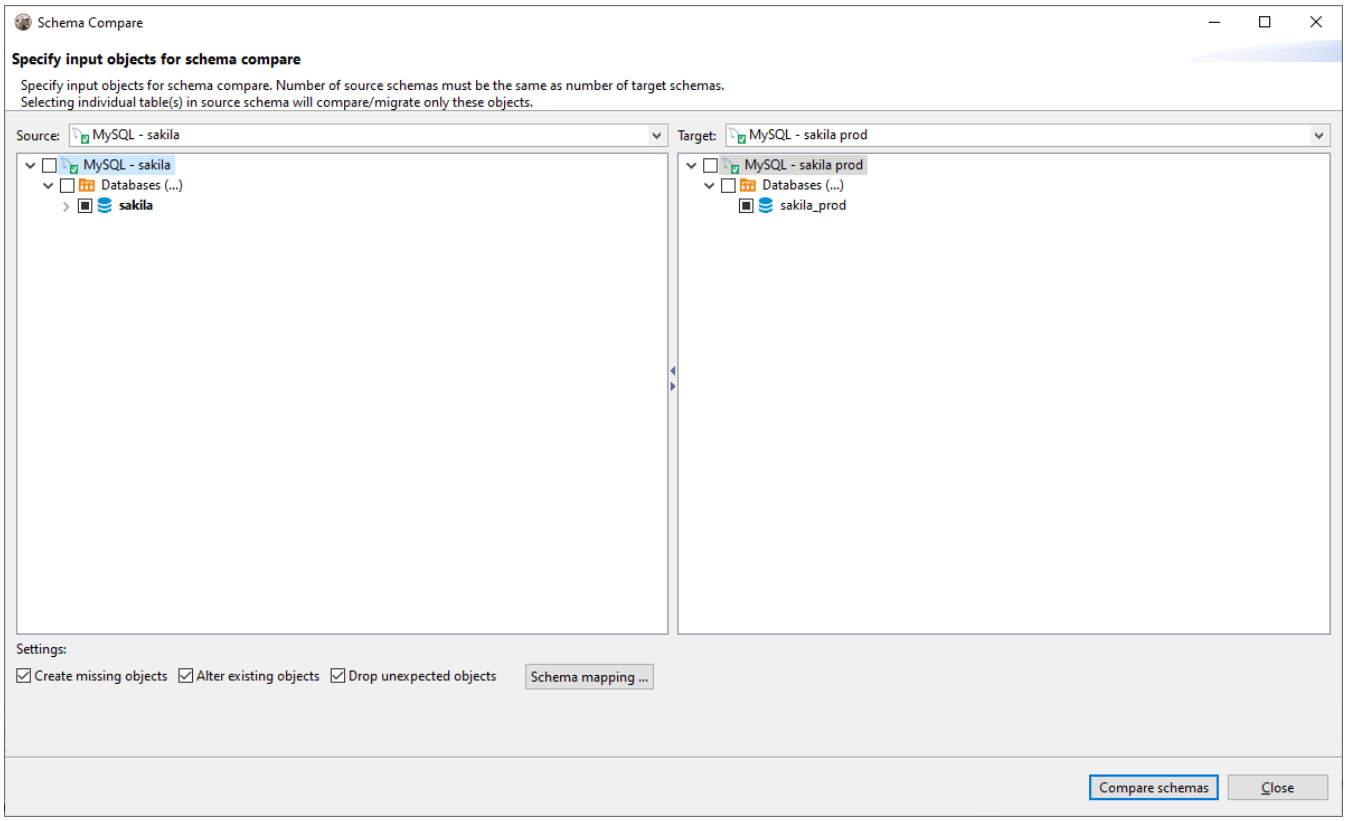
Selecting objects to compare

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

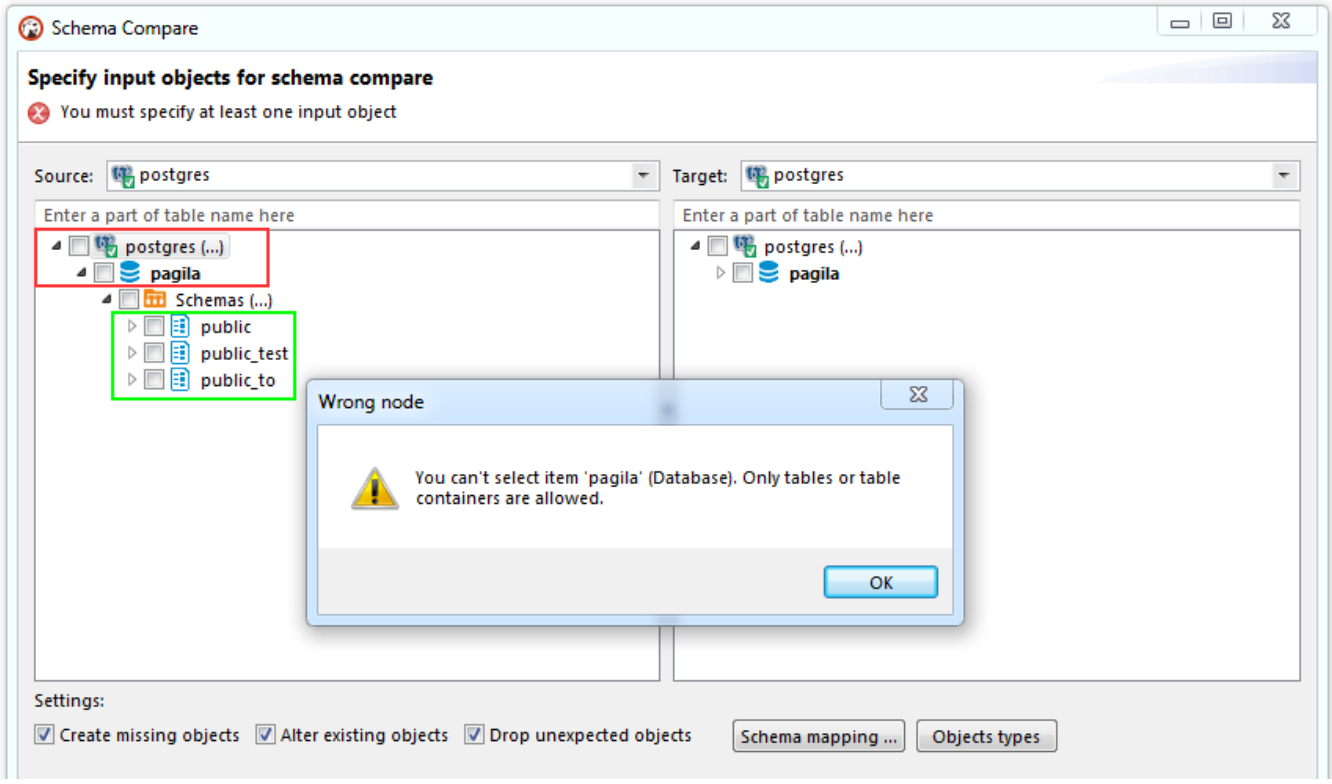


Compare settings

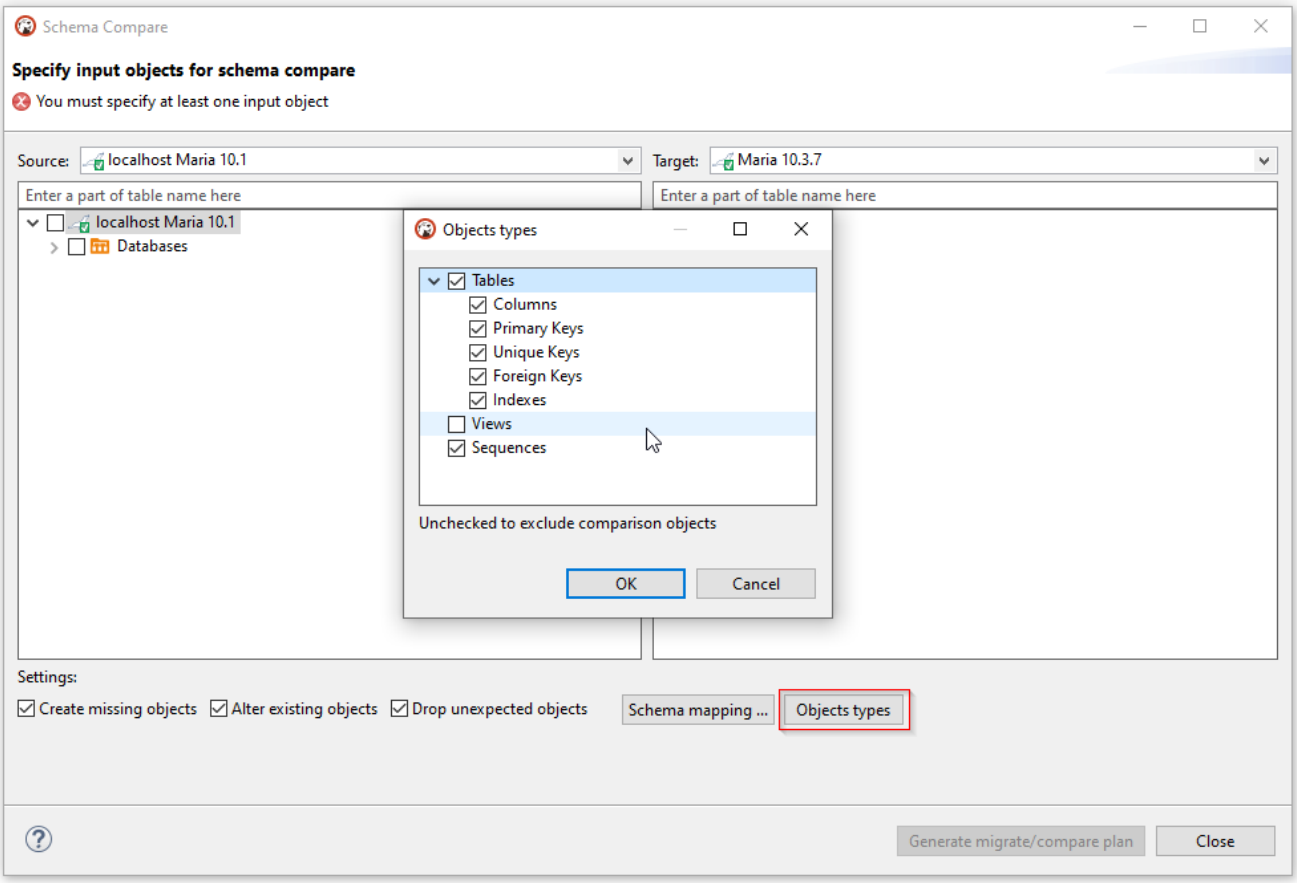
Re-validate that you have chosen the correct objects to compare. You can also specify the types of changes to be processed: create, drop, or alter. By default, all kinds of changes are enabled.



For comparisons, table containers should be used. Schemes - if the database supports the schemas. Databases - if the database supports catalogs and does not support the schemes. Datasources - if there is no support schemas or catalog support (you can find an example below in "Compare schemaless bases").



You can exclude the specific compared types of objects.
For example, you can do this if you do not want to see the sequences, views, or external keys in the final comparison result.

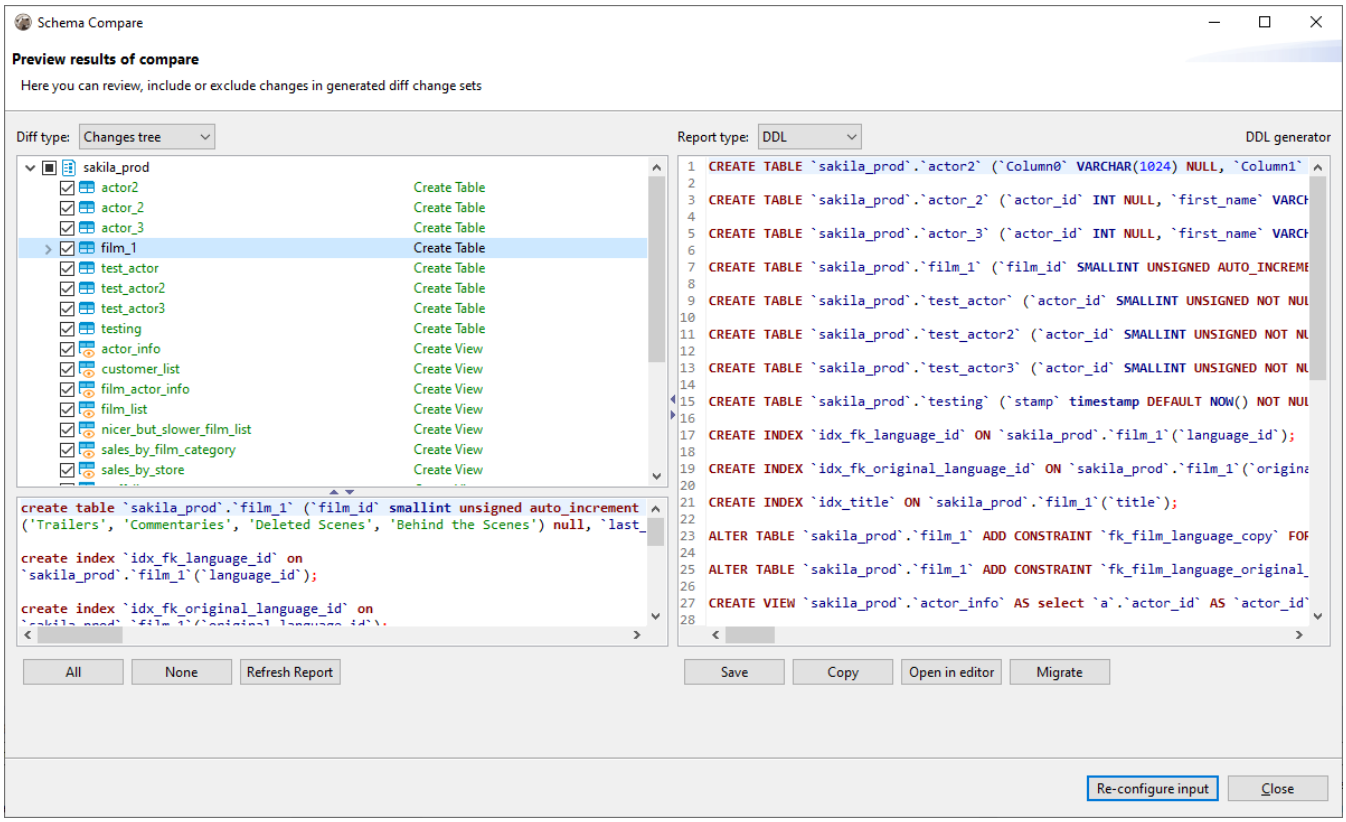


Compare results

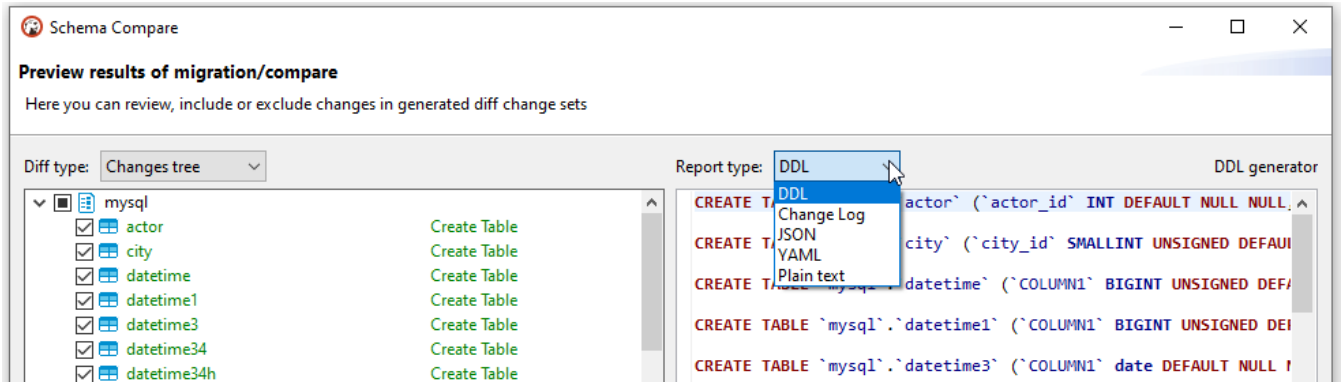
Click on `Compare Schemas` to generate a diff report.

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

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

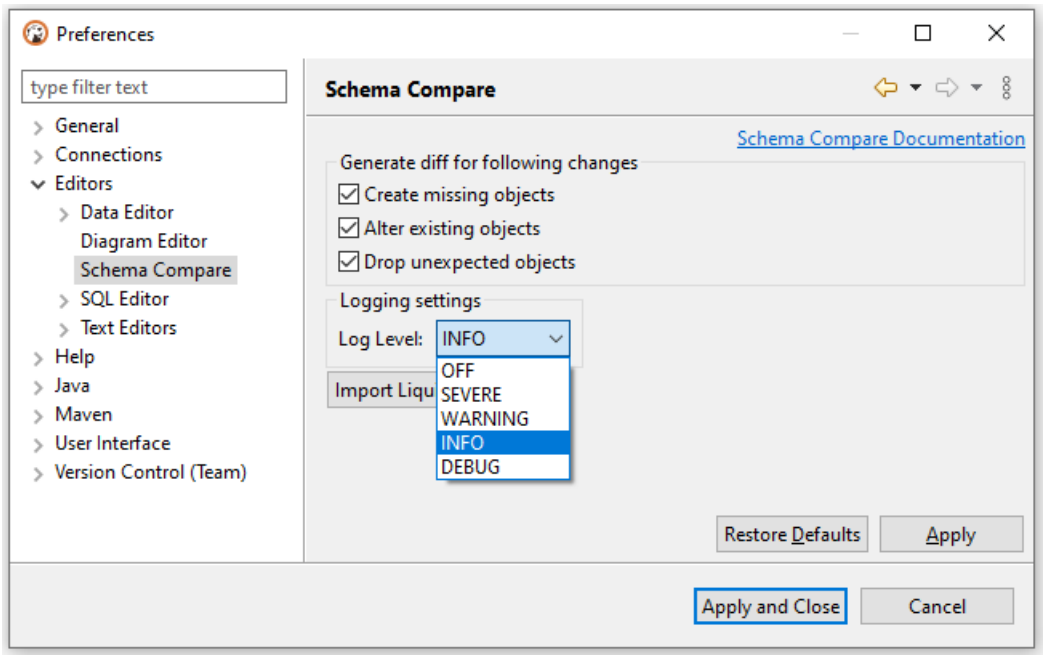


You can also switch to another diff report representation (diagram, JSON, YAML, plaintext).

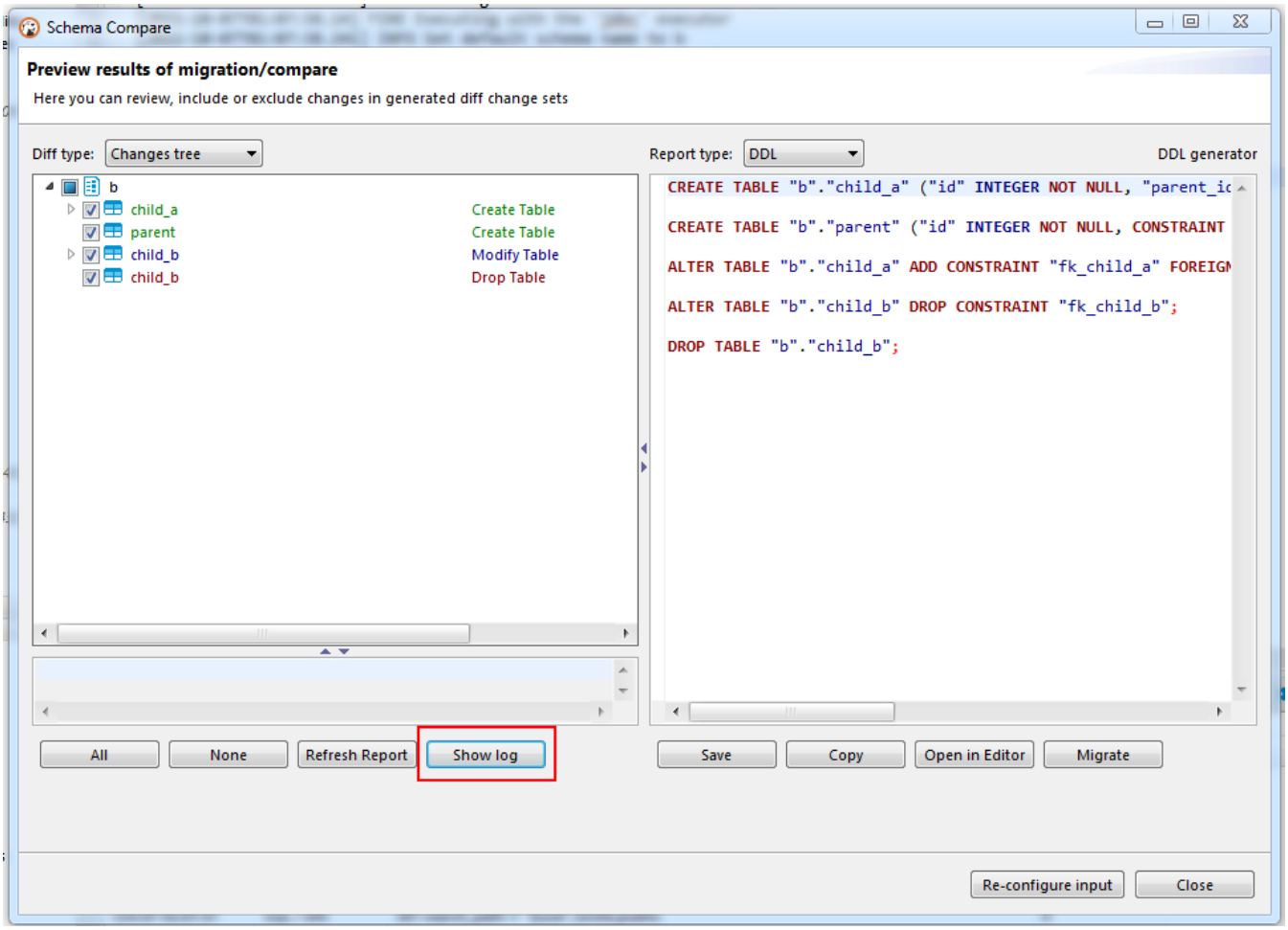


Compare logs

To get acquainted with the comparison logs, you first specify the logging level on the Preferences-> Editors-> Schema Compare preference page. Specify one of the logging levels and click on `Apply`. By default, the logging level is the OFF level. To get complete information, you can choose the DEBUG level.

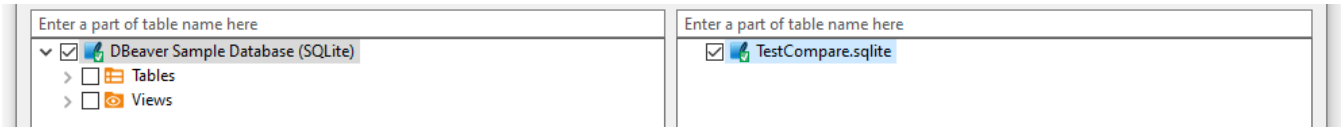


After comparing operations, you can click on the `Show log` button. A log will be open in the Editor, and the content of this log will depend on the logging level you choose in the settings. Log level changes from preferences will not be applied to the comparison wizard if it is already open in another window. Close and open the schema compare wizard in this case.



Compare schemaless bases

Some bases (like SQLite and Firebird) do not have catalogs and schemes that can be selected for comparison. In this case (and only for these databases), it is possible to compare the entire datasource entirely.



Databases supporting schema comparison

- MySQL/MariaDB
- Oracle
- PostgreSQL
- SQLServer
- Snowflake
- SQLite
- Firebird
- Redshift
- DB2
- Informix
- Derby
- Greenplum
- Netezza
- Cockroach

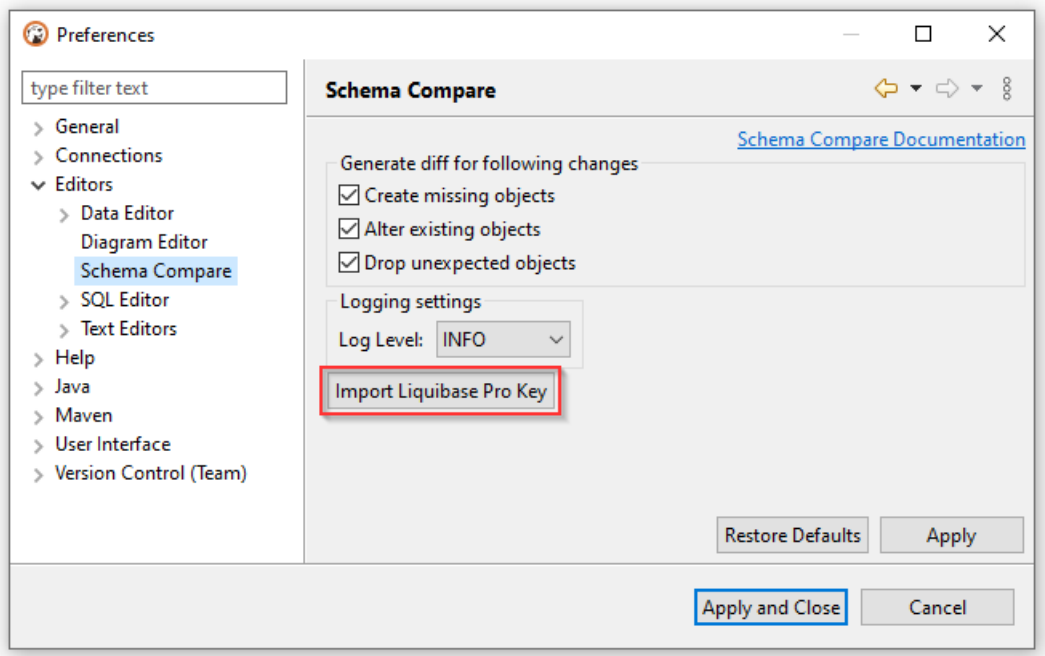
- Vertica
- SAP HANA

Using schema compare with Liquibase PRO key.

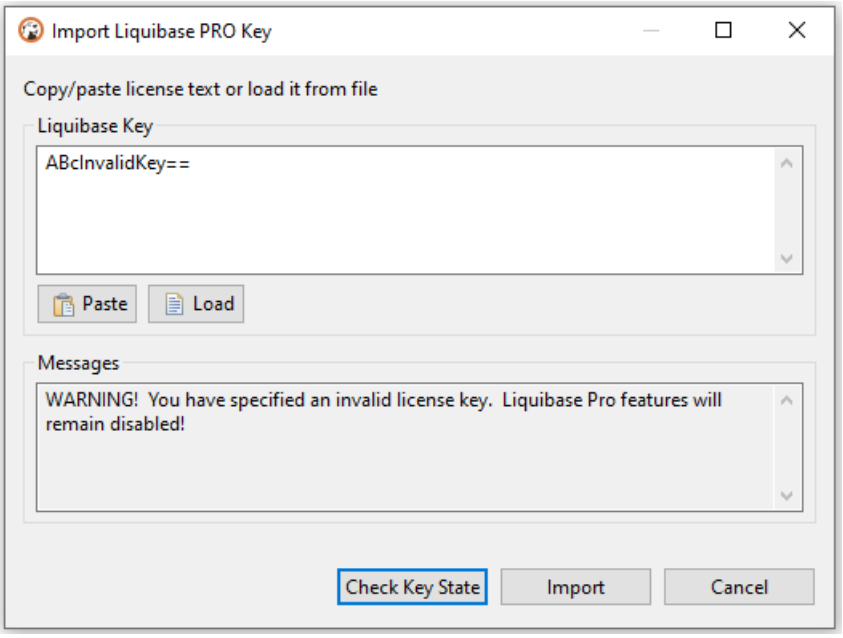
If you have a Liquibase PRO key, then you can use it with DBeaver. Steps you need:

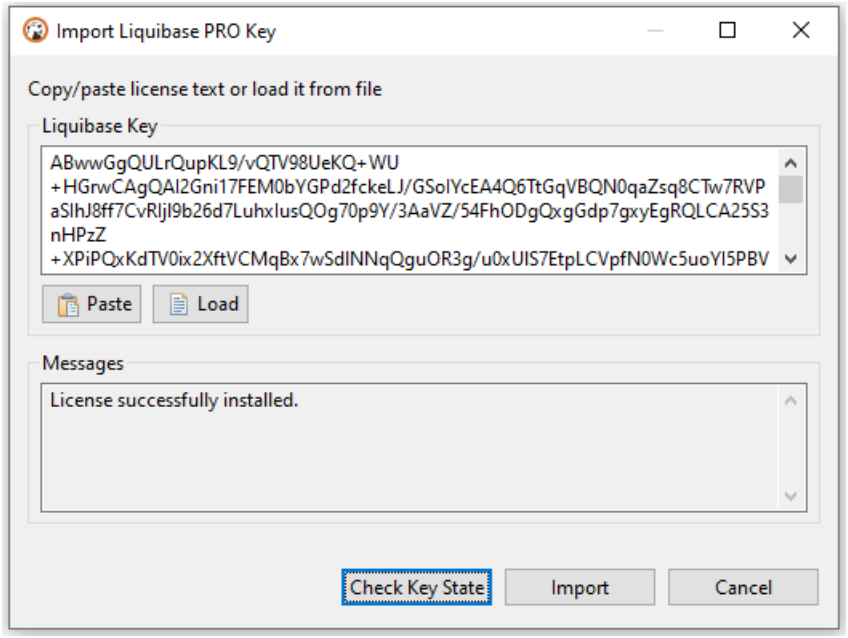
- Find and open your dbeaver.ini file. It is located in the DBeaver root directory.
- Find -vmargs command
- Add a new line after this command: -Dliquibase.license.key=yourKey (example: -Dliquibase.license.key=ABwwGgQU...)
- Open DBeaver and "Schema compare" window. The key will be checked at this step.

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



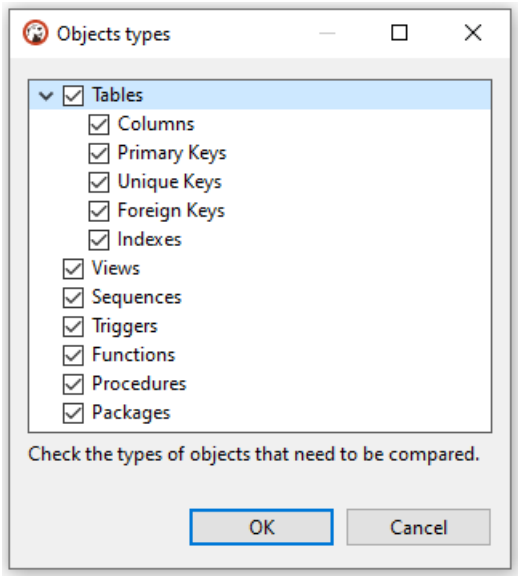
You can add your key in the Liquibase Key text field manually, throw the `Paste` button, or use the `Load` button to download a file. You can check the license state with the `Check Key State` button. After pressing the button, you can see the result of the checking in the `Messages` field.





We advise you to restart the program after adding a key for correct program work. Settings changes will not be applied to the comparison wizard if it is already open in another window. The key will be saved in the DBeaver settings. If you specified the key in the .ini file and installed another key through the Import Key dialog, then the key from the .ini file will be in priority.

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



Object types being compared by LiquibasePRO

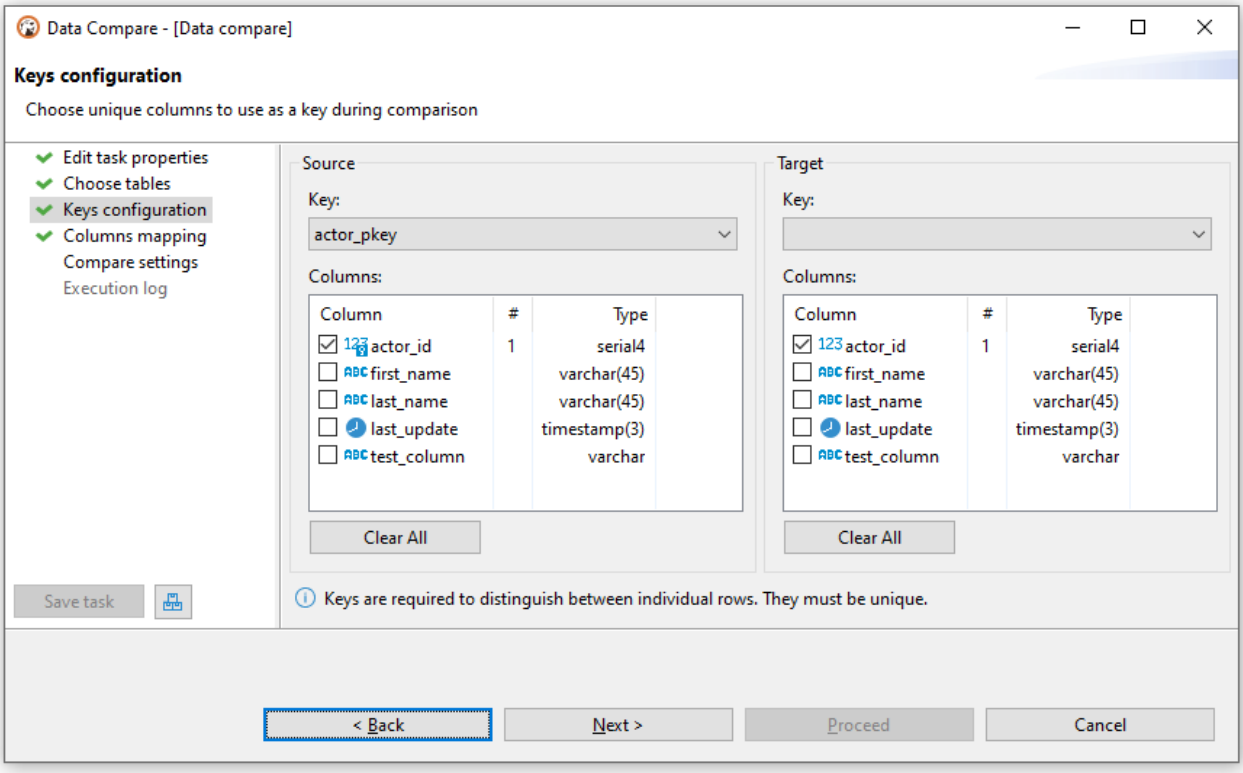
- Check Constraints
- Procedures
- Functions
- Triggers
- Synonyms (Oracle)
- Package with the body (Oracle)

Data compare

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

Preparing the tool

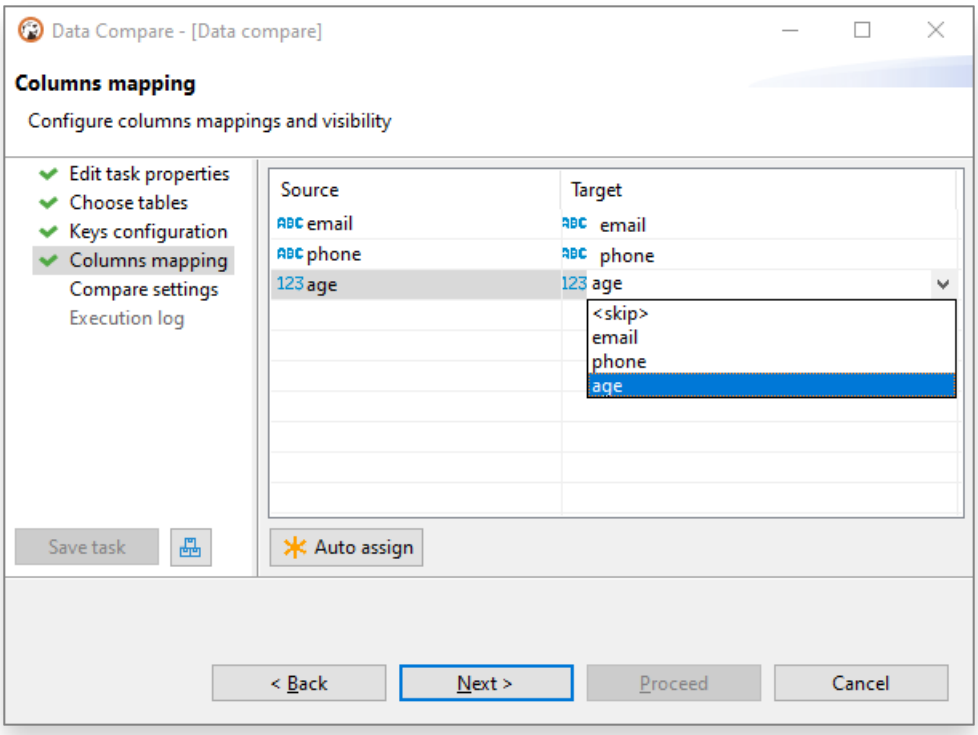
1. You will need to choose one or two desired tables in the Database Navigator - it may be the tables from the same databases or from different databases or even from different RDBMS (e.g. PostgreSQL and MySQL).
2. Then choose `Database` ⇒ `Compare/Migrate` ⇒ `Data Compare` from the menu and the Data Compare Wizard will appear. Here you can preview selected tables or choose other ones. Then you can navigate to the second page.
3. On the second page you should choose the columns that will be used as a **unique key** during the comparison (the amount of columns must be equal). If the *keys* chosen are wrong, it may lead to invalid results. By default, if the table has a unique key in it, it will be chosen automatically during the initial setup:



   Without unique keys the rows cannot be compared properly, since there is no way to distinguish between individual rows.
4. After reaching the last page, you can tweak limits (e.g. you only care about the `N` first rows) and exclude categories of the resulting rows (e.g. you are not interested in modified rows). Also, you can press the `Save task` button to save the configuration in Task to use it later, or Schedule it.

Changing columns mapping

If tables you want to compare have different ordering, naming or quantity of columns, you can tweak their mappings and/or entirely skip columns you're not interested in:



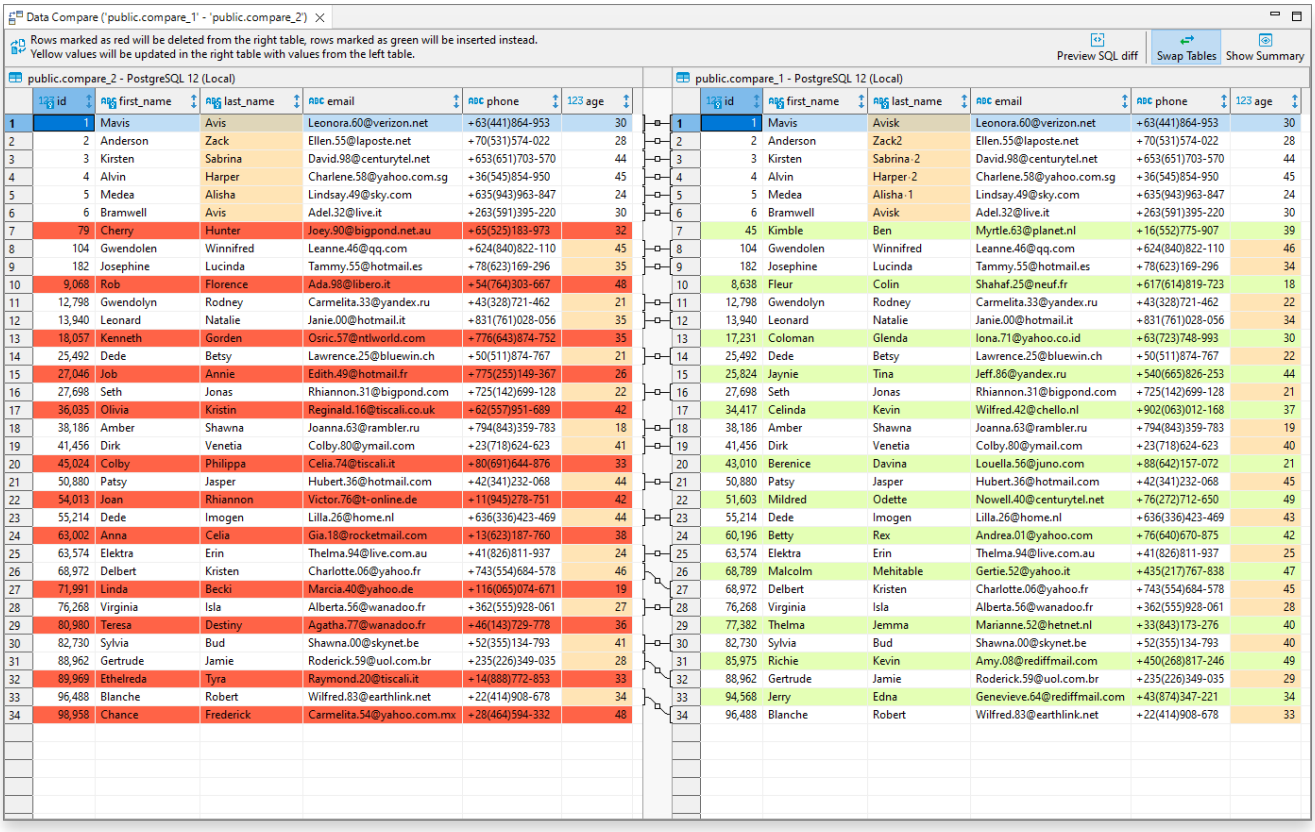
Please note that this page only features columns that aren't used as primary key configured on a previous page.

Pressing the **Auto assign** button will attempt to link columns with same names.

Viewing the results

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

When the results are ready, DBeaver will play a *beep* sound, and the editor will open:



Here you can examine the differences, swap the panels using the `Swap Containers` button in the toolbar, or preview the summary including the statistics by using the `Show Summary` button in the toolbar too. Everything you can do in the regular data editor is possible to perform here - you can copy data or transfer it to another database, except modify the values.

** - Please note that the actual preview is only available when Data Compare is launched through the wizard - otherwise only the*

statistics will be written into the Task Log.

Exporting the results

You can export results as an SQL script containing statements that will turn the second table into the first one by `DELETE`-ing unrelated rows, `INSERT`-in new ones, and `UPDATE`-ing existing ones, so the data will look the same.

You can export results either by:

1. Setting the **Export compare results to file** option found on the last page
2. Pressing the **Preview SQL diff** button found in the compare viewer

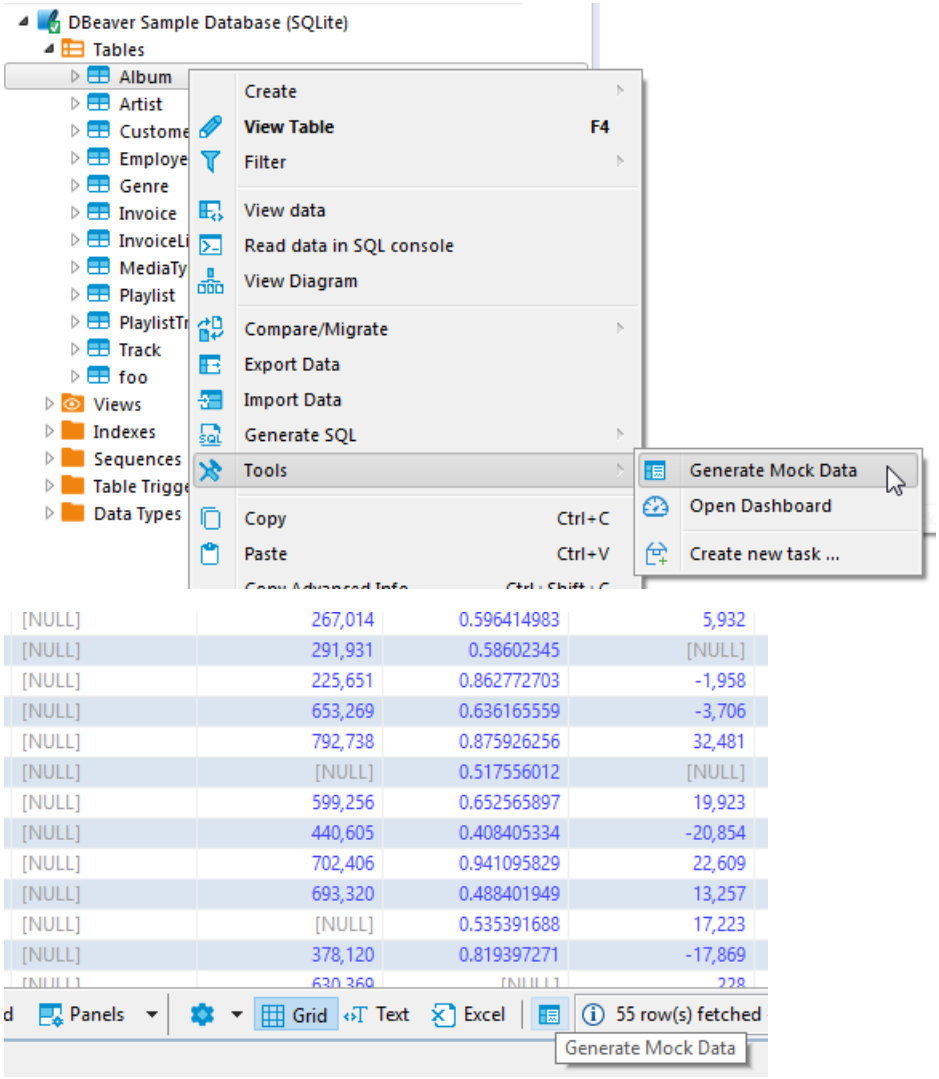
Under the hood

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

MockData generation

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

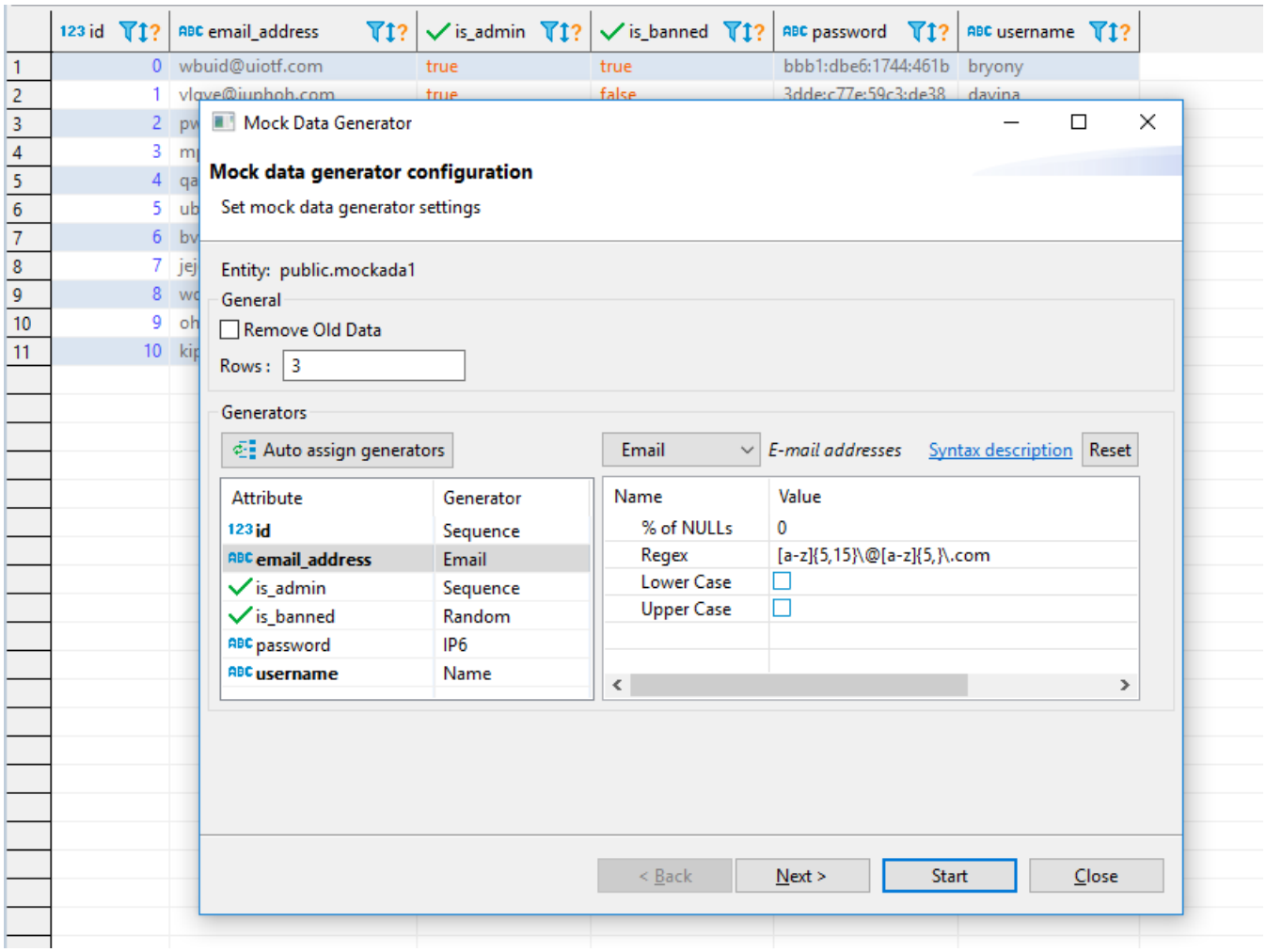
Sometimes in software development we need to generate mock, but valid, data for testing. Populating a database manually is a time-consuming and exhausting process. It can be very complicated when you need to generate not just 5–10 users, but thousands of entities of different types. DBeaver Mock Data generator helps you generate test data much easier.



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

Th following are features of the DBeaver Mock Data generator:

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

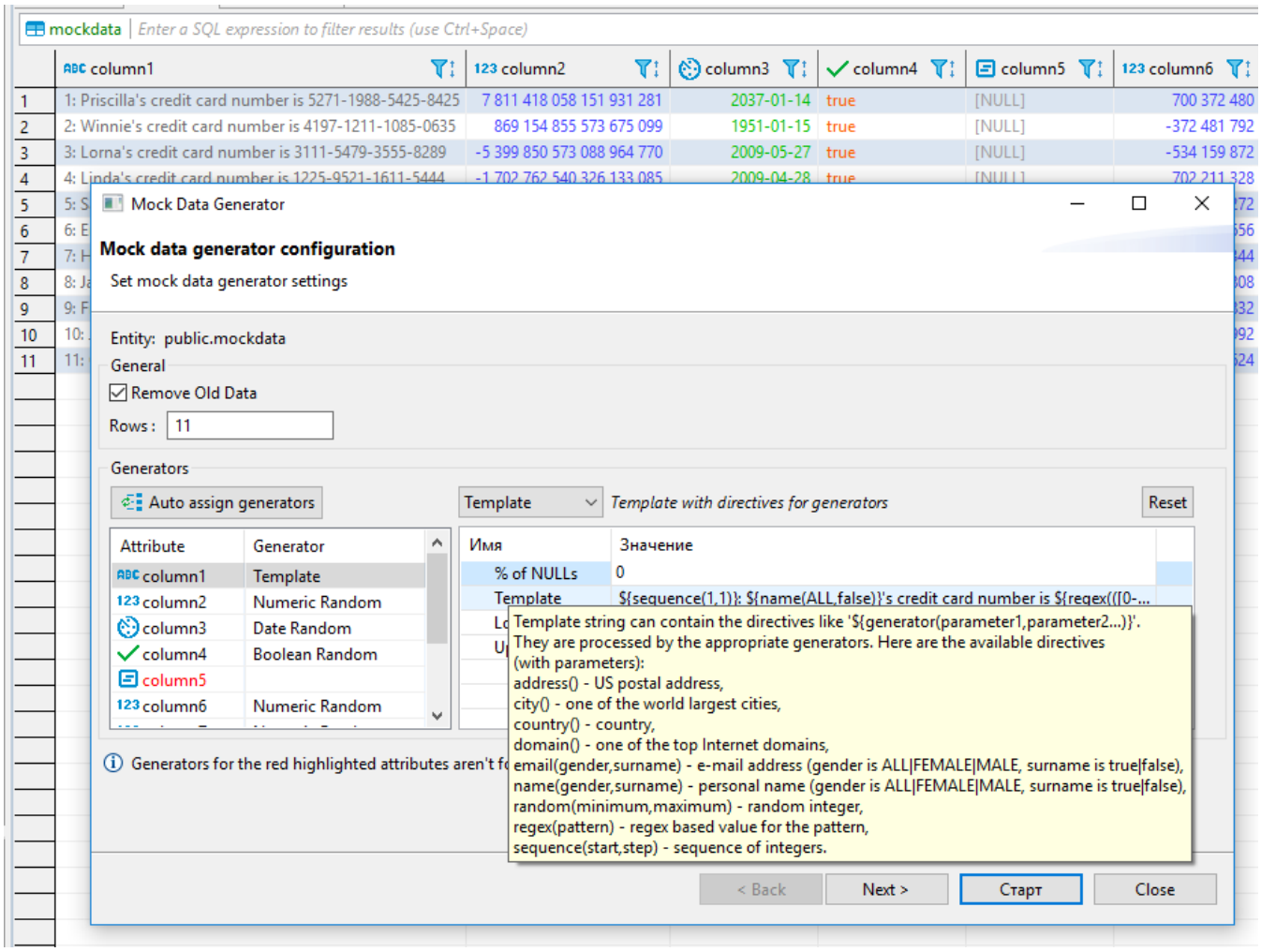


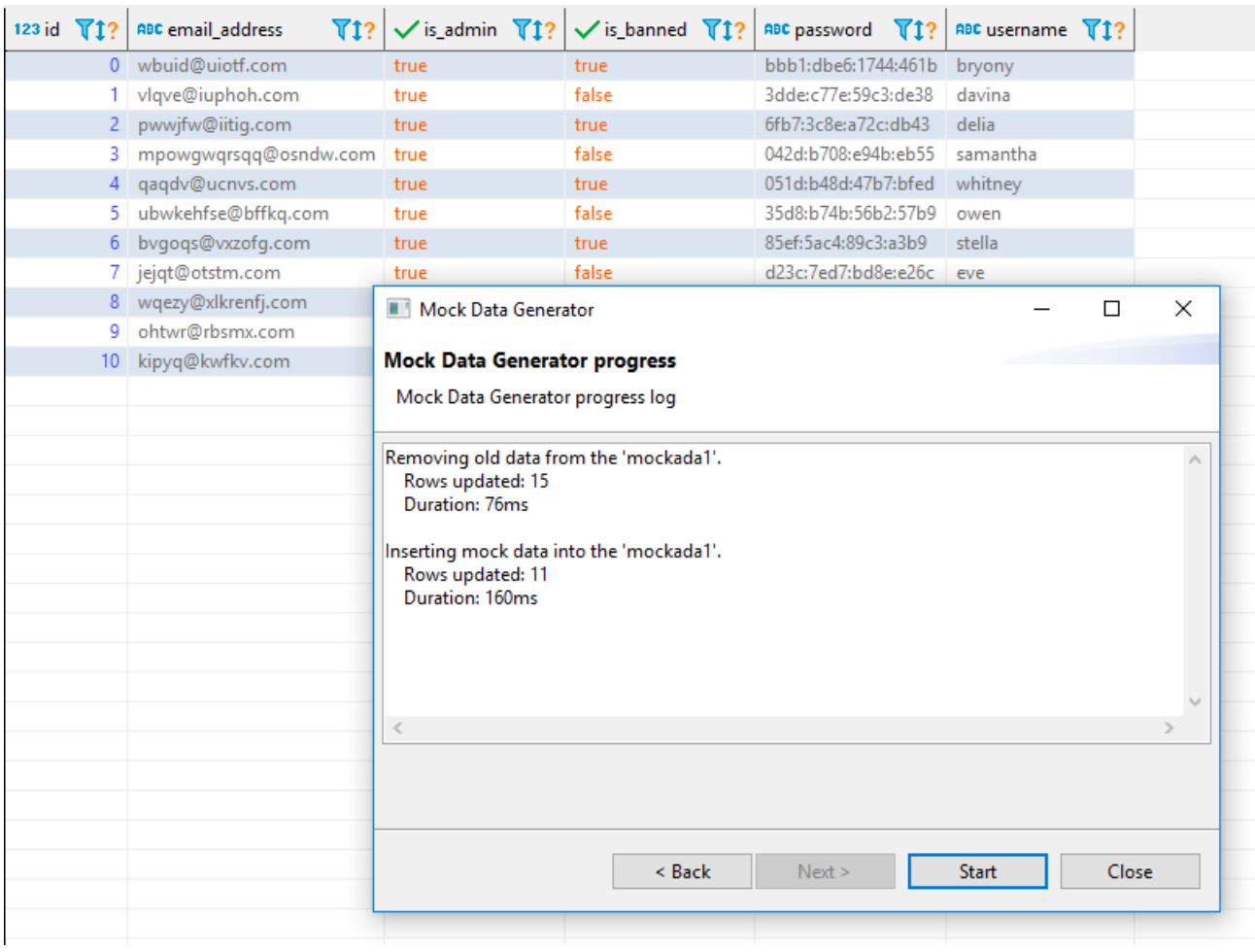
The following are mock data generators for data types with their configurable parameters:

- Boolean
  - Random
  - Sequence (initial, order)
- Date
  - Random (start, end)
  - Sequence (start, step, reverse)
- Numeric
  - Random
  - Sequence (start, step, reverse)
  - Advanced (min, max, precision, scale) *
    - Price preset *
    - Coordinate preset *
- String
  - Text (template, min length, max length)
  - UUID
  - Address *
  - City *
  - Country *
  - Domain *
  - Email (gender, with surname, numeric suffix) *

- Name (gender, with surname) *
- Price (country, min, max) *
- Regex based random (regex template) *
  - Credit Card preset *
  - Email preset *
  - Gender preset *
  - HEX Color preset *
  - IP4 address preset *
  - IP6 address preset *
  - Phone Number preset *
  - Postal Code preset *
  - Price preset *
- Template with parametrized directives for other generators *:
  - address() - US postal address
  - city() - one of the world's largest cities
  - country() - country
  - domain() - one of the top Internet domains
  - email(gender,surname) - e-mail address (gender is ALL|FEMALE|MALE, surname is true|false)
  - name(gender,surname) - personal name (gender is ALL|FEMALE|MALE, surname is true|false)
  - random(minimum,maximum) - random integer
  - regex(pattern) - regex based value for the pattern
  - sequence(start,step) - sequence of integers

- NULL values
- FK - data from the referenced table according to the constraint





* These features are available in Enterprise and Ultimate editions only.

Dashboards, DB monitoring

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

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

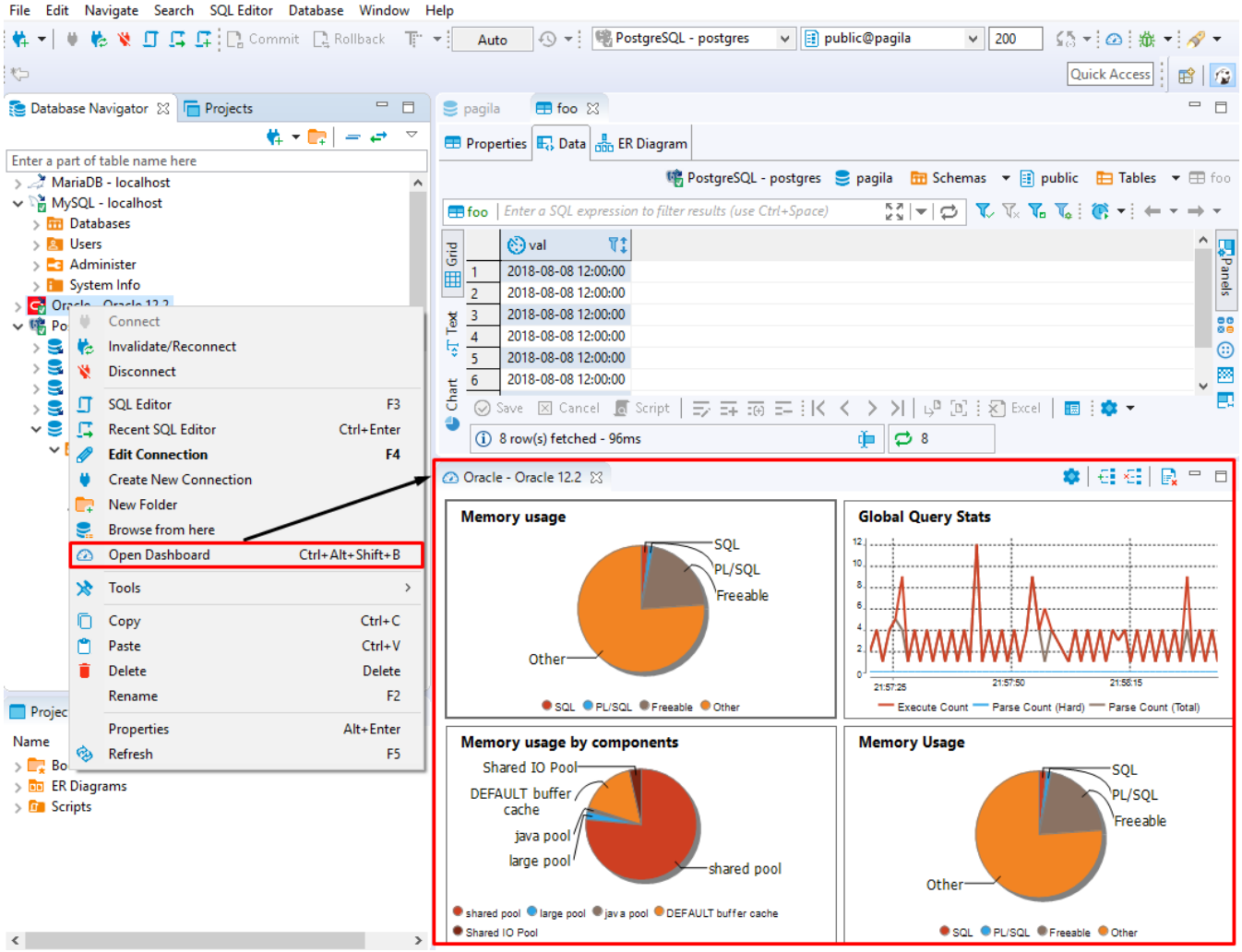
Managing Dashboards Panel

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

Opening Dashboard Panel

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

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



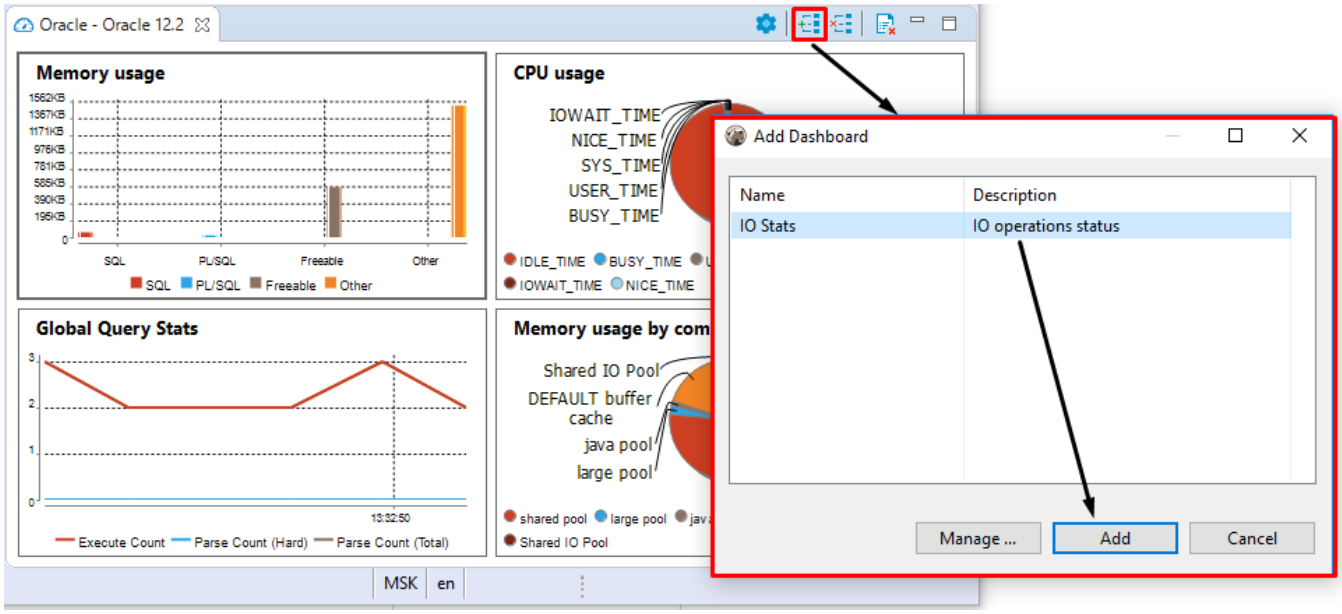
The following controls are available on the dashboards panel toolbar:

Icon	Name	Description
	Settings	Opens dashboard's configuration.
	Add dashboard	Adds dashboard to the dashboard panel.

Icon	Name	Description
	Remove dashboard	Removes dashboard from the dashboard panel.
	Reset dashboards	Restarts dashboard's calculation.

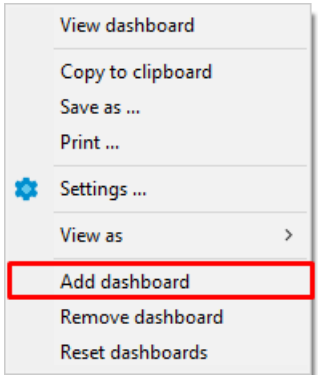
Adding Dashboards

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



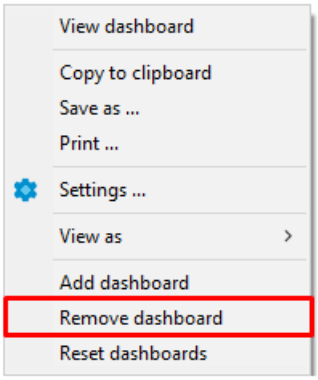
Note: Different databases have different sets of predefined dashboards. DBeaver is delivered with sets of predefined dashboards for such databases as Postgress SQL, MySQL, Oracle, and Exasol. It is also possible to create new custom dashboards, for more details see Managing Dashboards.

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.

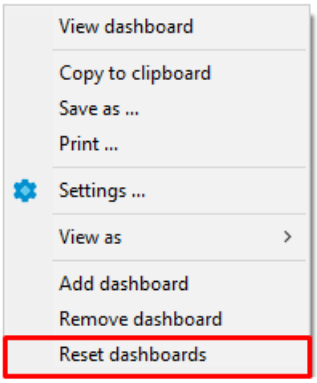


Resetting Dashboards

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

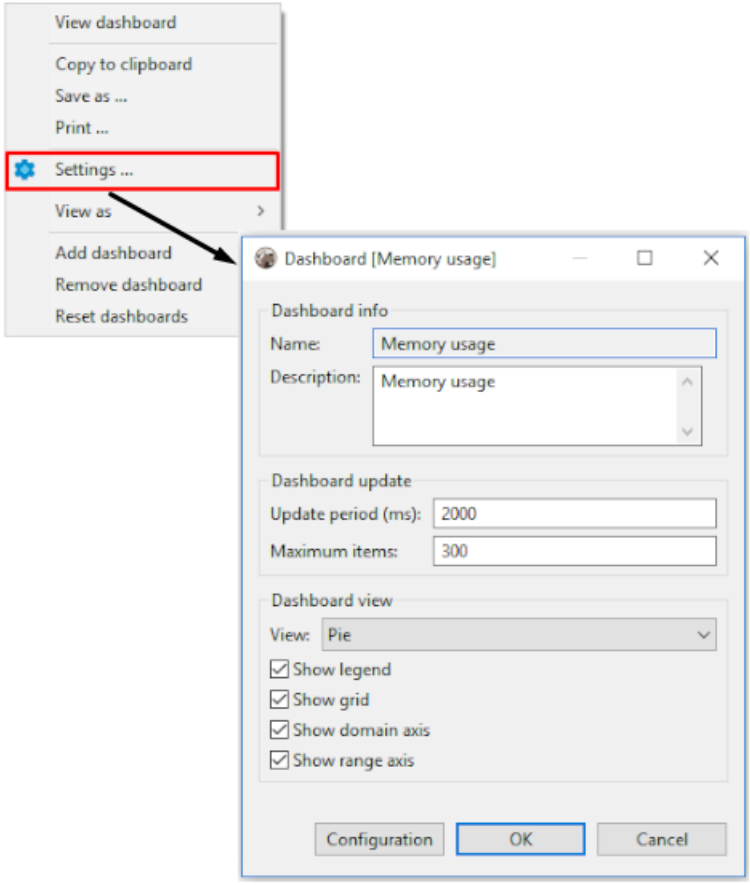
You can reset all the dashboards displayed on the dashboards panel by a single click on **Reset dashboards** button on the dashboard panel's toolbar.

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



Changing Dashboard Representation

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

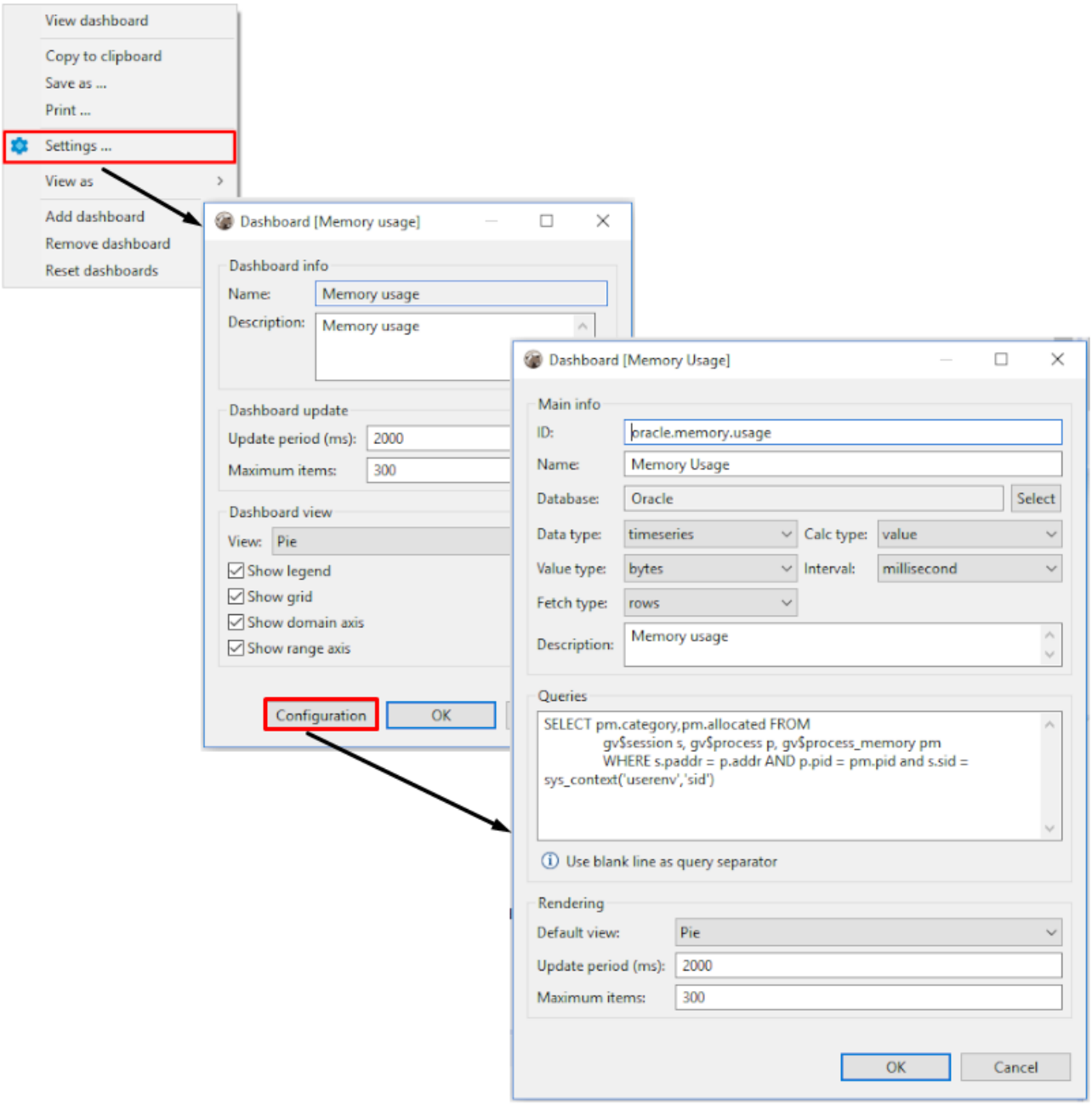


The following dashboard representation parameters can be adjusted:

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

Adjusting Dashboard Configuration

To adjust dashboard's configuration settings right-click on a dashboard, select the **Settings** menu option, then, in the opened dialog box press the **Configuration** menu option.



The following dashboard parameters can be configured:

Parameter	Description
ID	Defines dashboard's ID. Make sure that ID has numeric values in it.
Name	Defines dasboard's name.
Database	Defines the database driver. To learn more about database drivers, see Database Drivers.
Data type	Defines the data type. The following options are availabe: timerseries (the default option) and statistics. Select timeseries type if you want to track the actual value returned by the server. Select the statistics type if your dashboard will show historical data.
Calc type	Defines how the data should be calculated. The following options are available: value (the default option) and delta. Select value if you are interested in the current value. Select delta if you want to track the difference between the current value and the previous one. This may be very useful when you work with statistics data, for example.
Value type	Defines the value to be shown on the range domain. The following options are available: decimal (the default option), integer, percent, bytes. Choose the value type in accordance with your data, for example, memory usage is convinient to be tracked in KBytes.
Interval	Defines time interval to be shown on the domain axis. The following time intervals are available: millicecond(the default option), second, minute, hour, day, week, month, year.
Fetch type	Defines whether the query should fetch data from rows or columns.

Parameter	Description
Description	Defines the description of a dashboard. Use this field to make it easy to understand what kind of information the dashboard represents.
Queries	Defines an SQL query whose fetched data will be used to build the chart displayed on the dashboard.
Default view	Defines the default visual representation of a dashboard on the dashboard panel. The following options are available: Bar, Pie, Time series(the default option).
Update period(ms)	Defines how often the dashboard's rendering should be updated.
Maximum items	Defines the maximum number of items to be fetched for the dashboard.

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

Setting Connection Prefereces

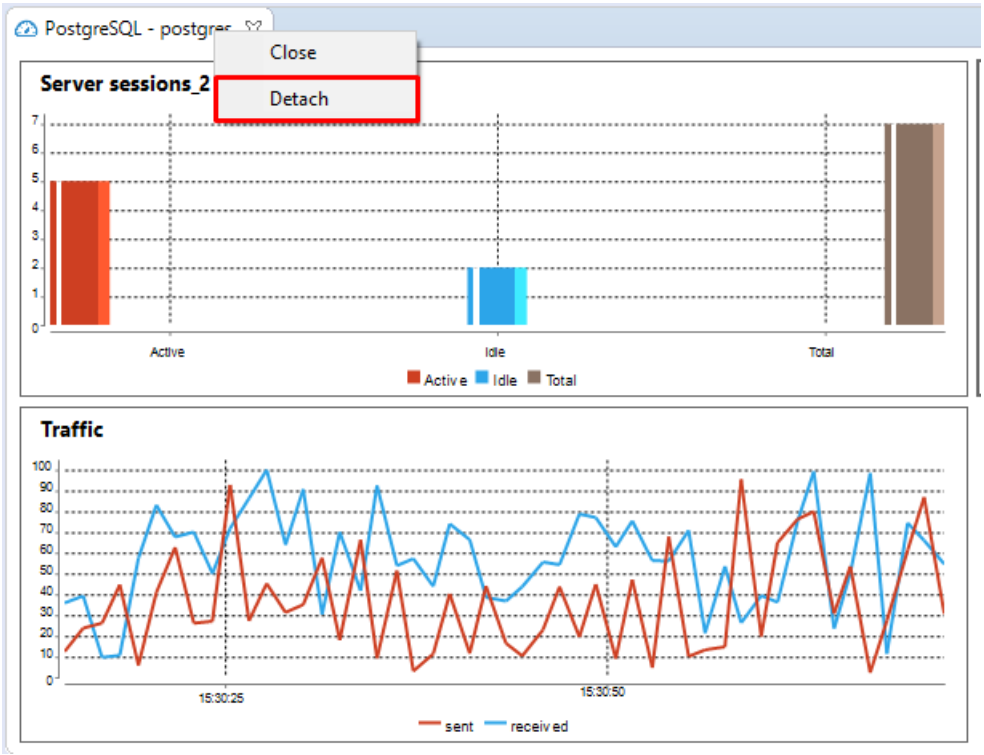
By default, if there is no active connection to the database and you open its dashboards panel, all the dashboards on the panel will be empty.

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

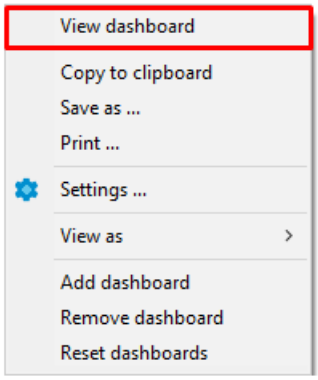
Detaching Dashboards

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

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

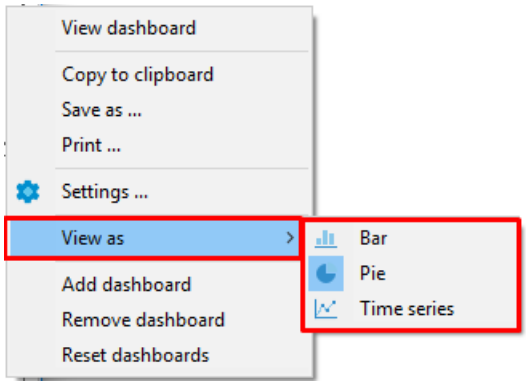


To detach a single dashboard, double left click over it. You can also right click the dashboard and then, select the **View Dashboard** menu option, the dashboard will be detached from the panel and you will be able to move it to any place on your screen.



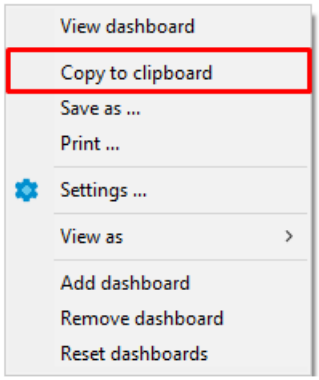
Changing Dashboard View

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



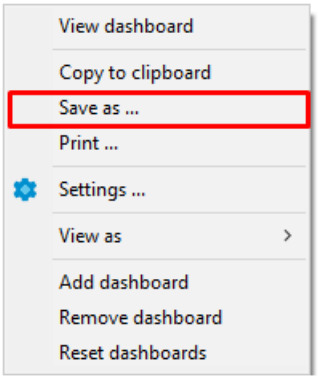
Copying Dashboards to Clipboard

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



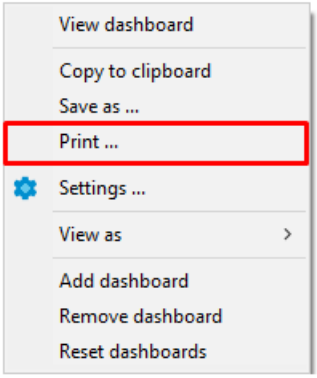
Saving Dashboards

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



Printing Dashboards

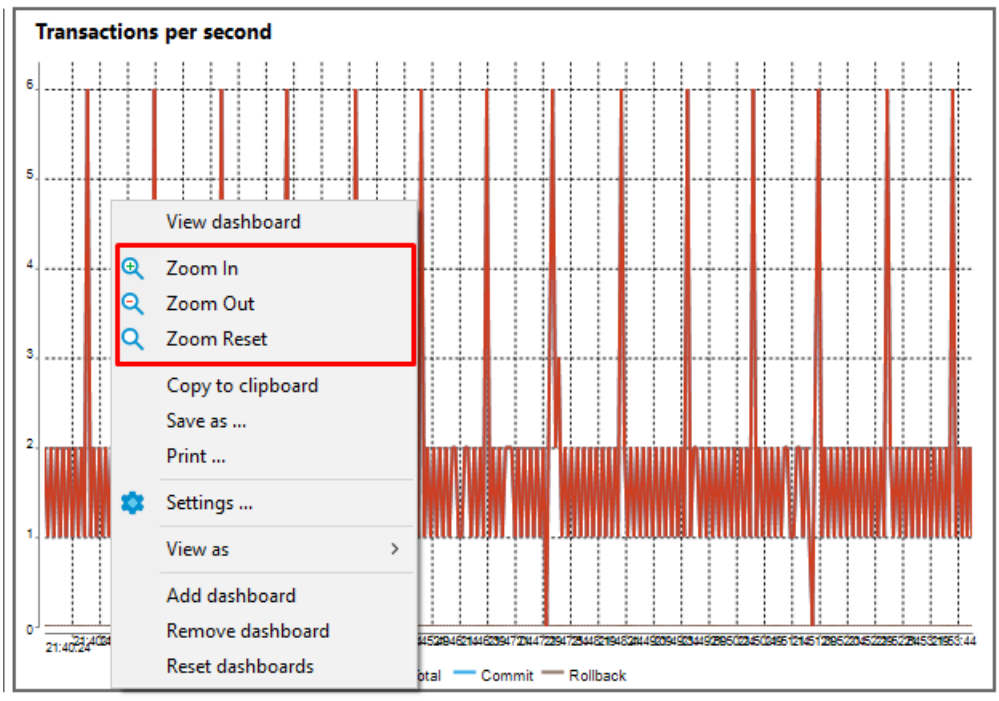
If you want to print out a screenshot of a dashboard, right-click the dashboard to be printed and select the **Print...** option.



Zooming

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

- Zoom In
- Zoom Out
- Zoom Reset



Managing Dashboards

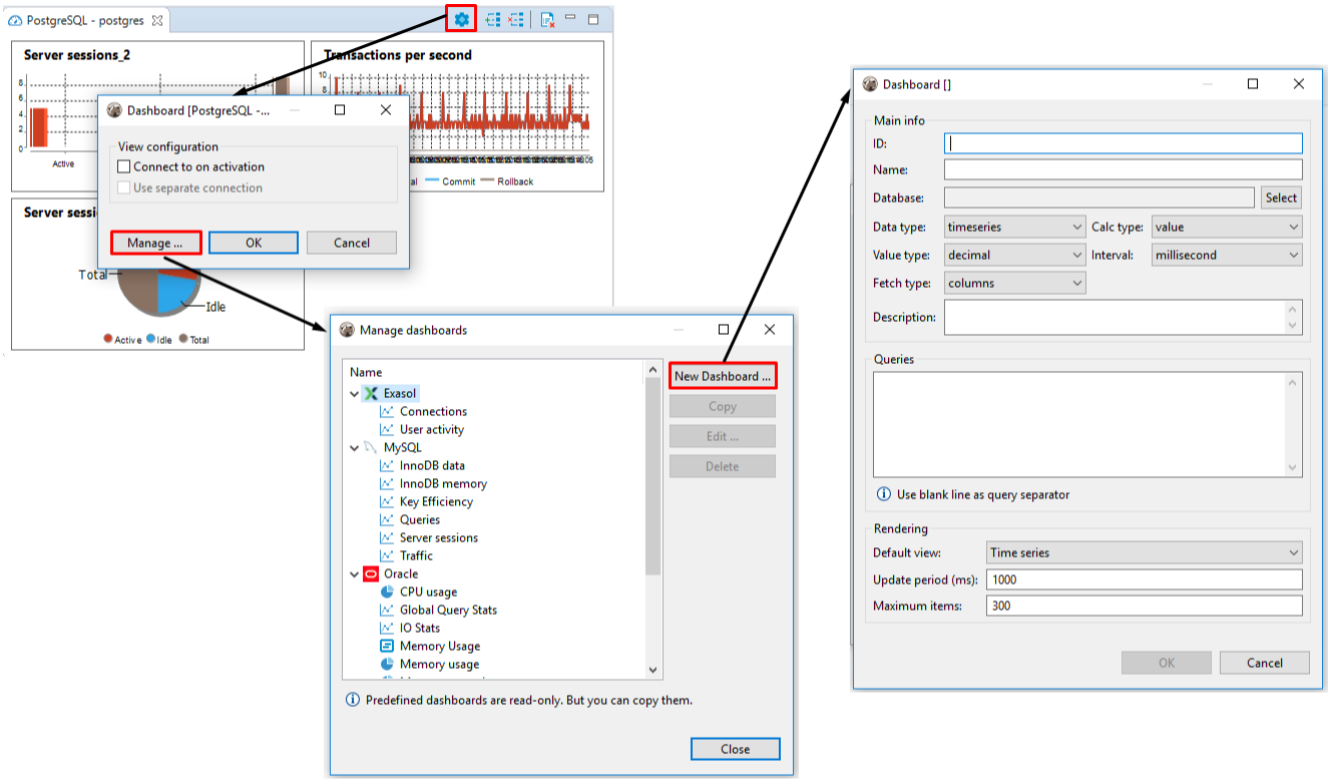
You can extend the list of predefined default dashboards by creating your own custom dashboards. This section describes dashboards' list management.

Creating Dashboards

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

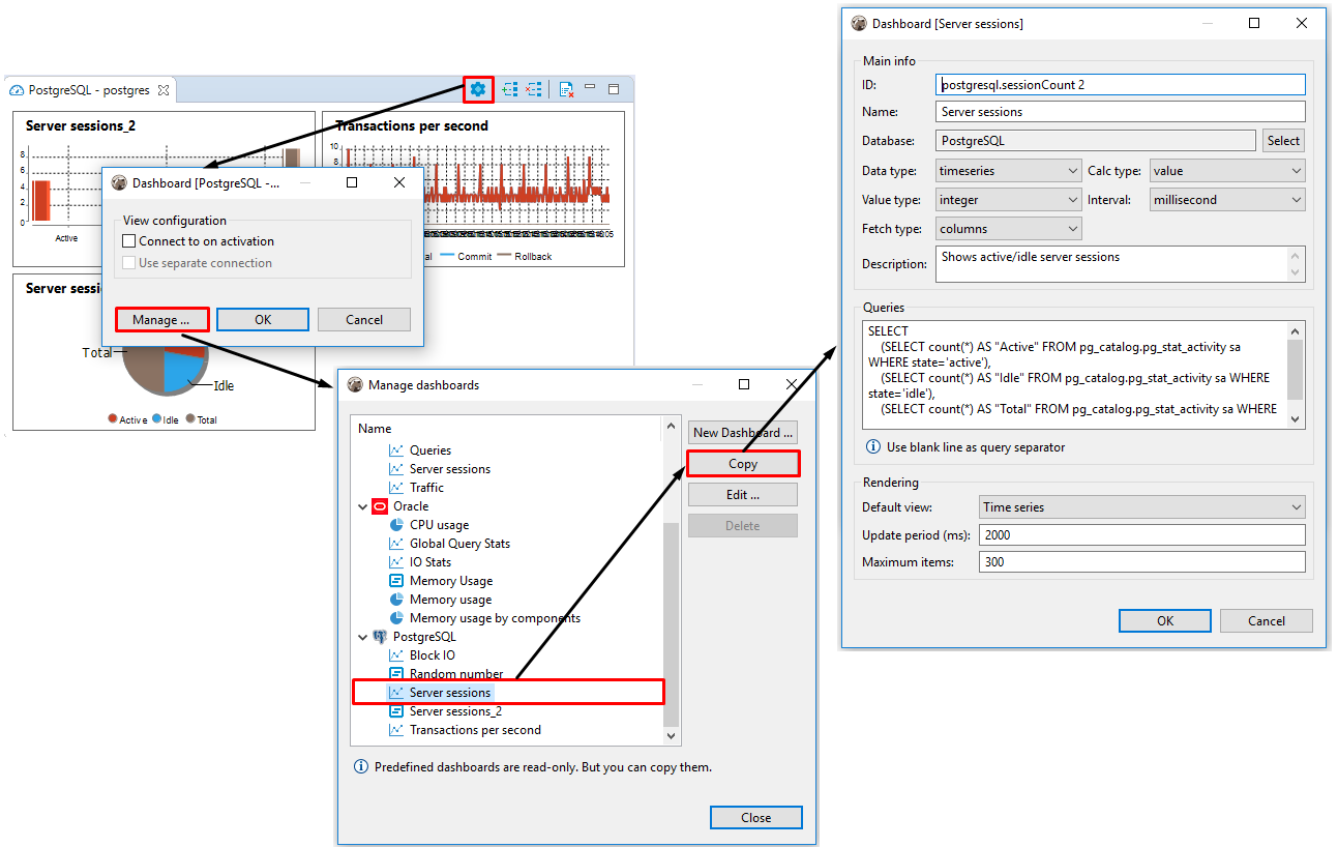
To create a dashboard from scratch:

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



To create a dashboard from a template:

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



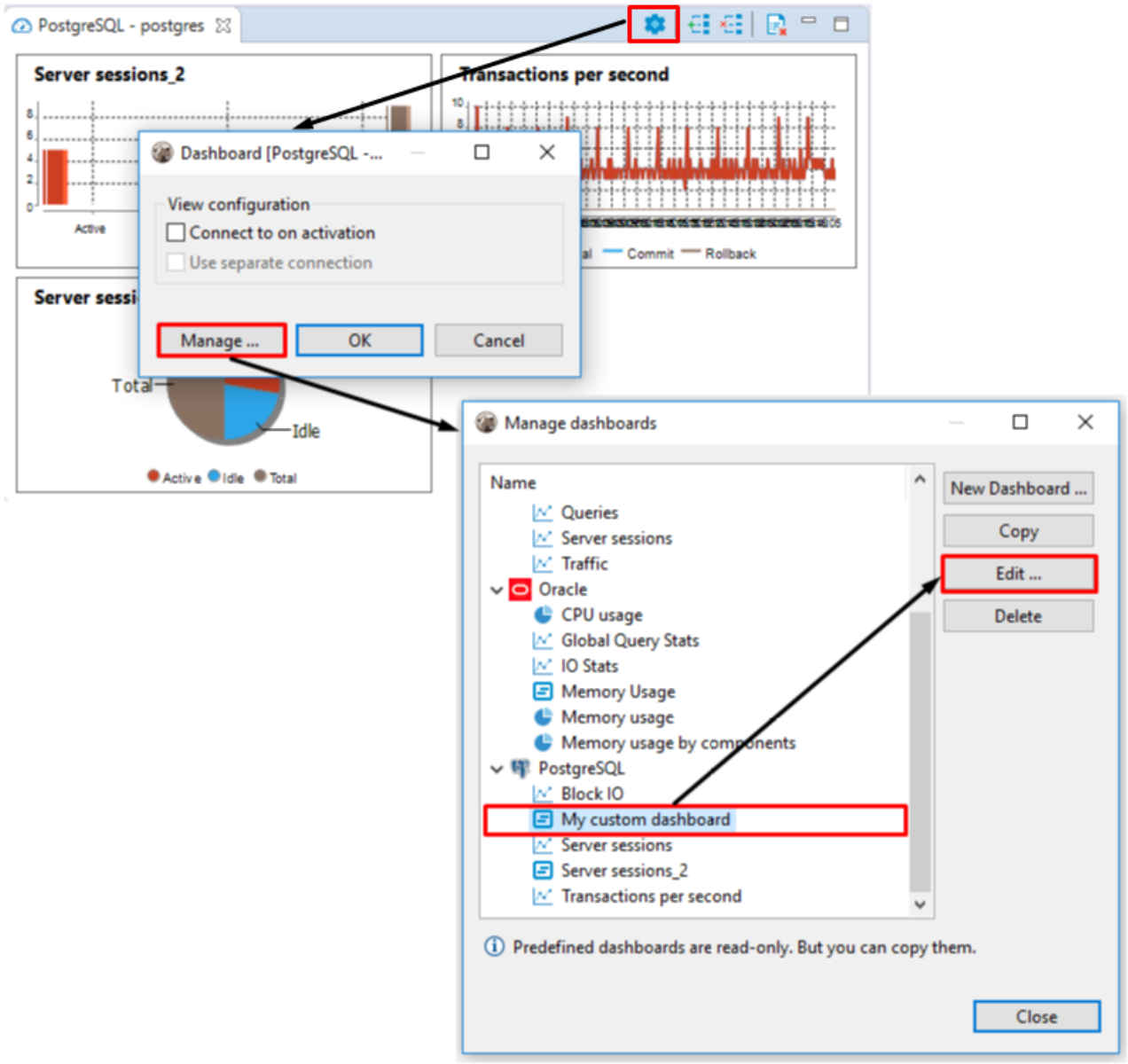
Editing Dashboards

If you need to change the dashboard's name , ID or any other configurational setting, you can edit a dashboard.

Note: Only custom dashboards can be edited, predefined dashboards are read-only, but you can use them as templates and create a custom dashboard whose parameters will be editable. To learn how to create dashboards from templates, see Creating Dashboards.

To edit dashboard's configuration:

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



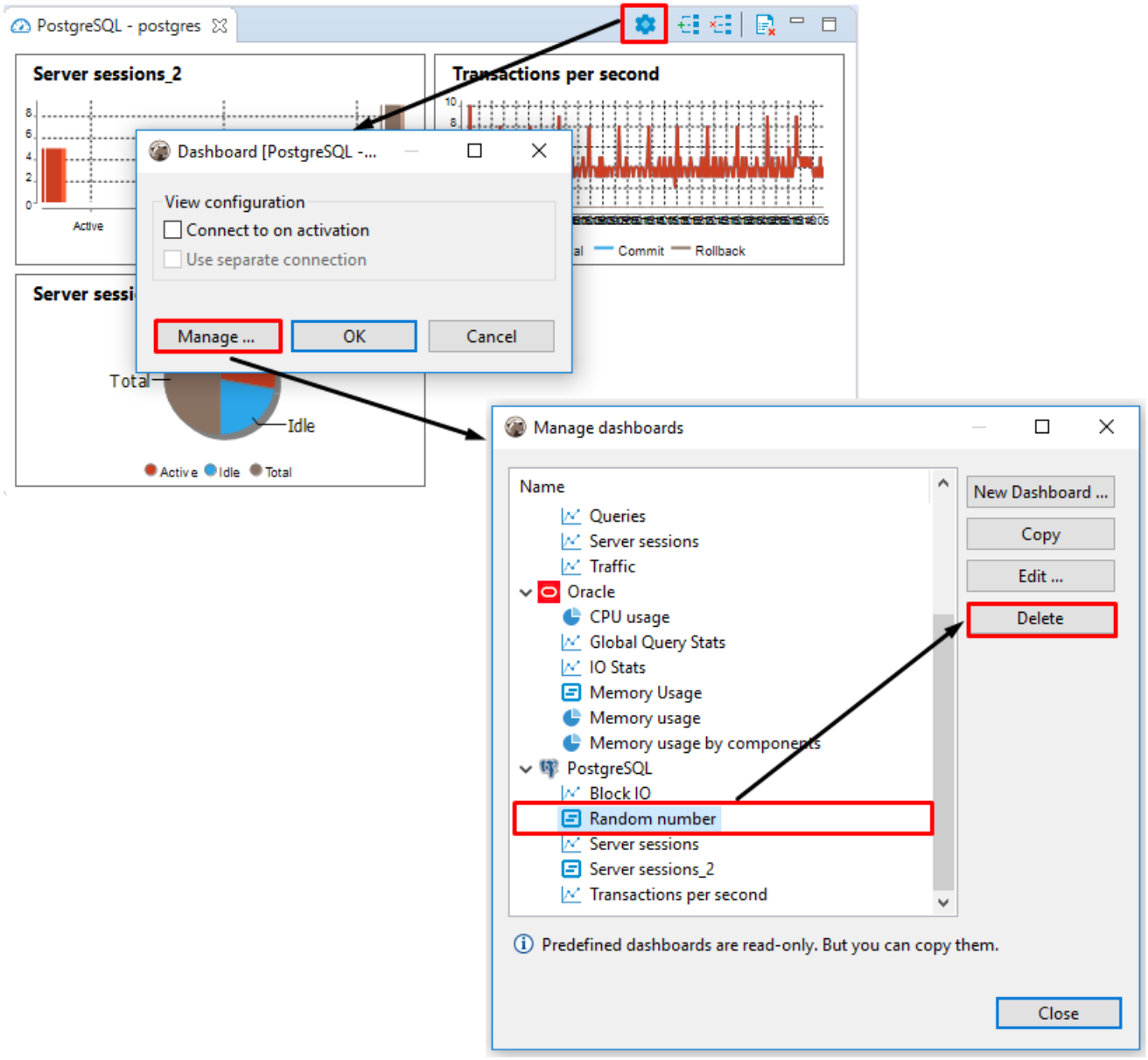
Deleting Dashboards

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

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

To delete a dashboard:

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

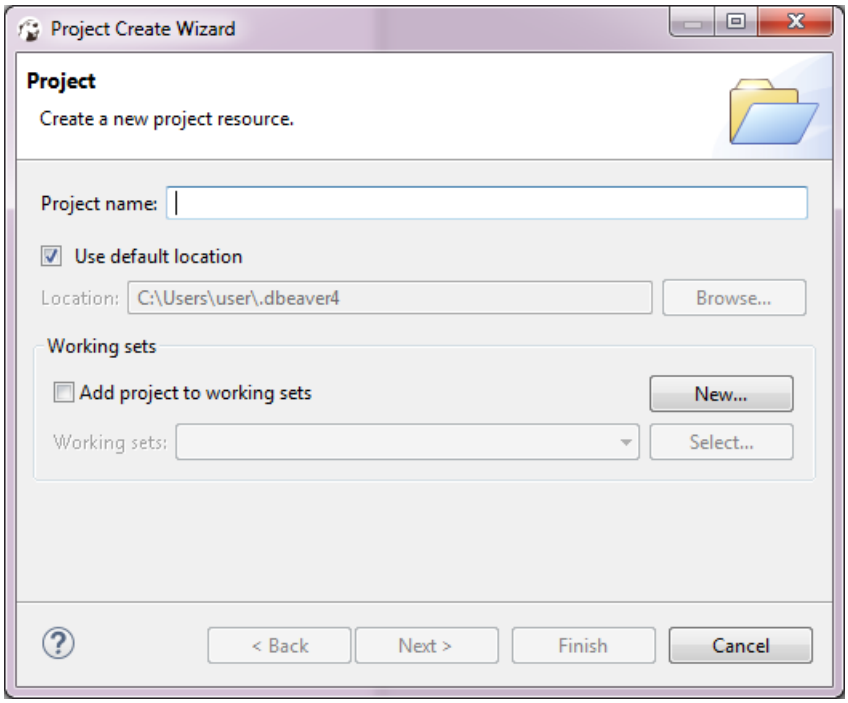

PostgreSQL - postgres
Server sessions_2
Active
Transactions per second
Commit
Rollback
Dashboard [PostgreSQL -...
View configuration
Connect to on activation
Use separate connection
Manage ...
OK
Cancel
Total
Idle
Active
Idle
Total
Manage dashboards
Name
Queries
Server sessions
Traffic
Oracle
CPU usage
Global Query Stats
IO Stats
Memory Usage
Memory usage
Memory usage by components
PostgreSQL
Block IO
Random number
Server sessions
Server sessions_2
Transactions per second
New Dashboard ...
Copy
Edit ...
Delete
Predefined dashboards are read-only. But you can copy them.
Close

Projects

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

Creating Project

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

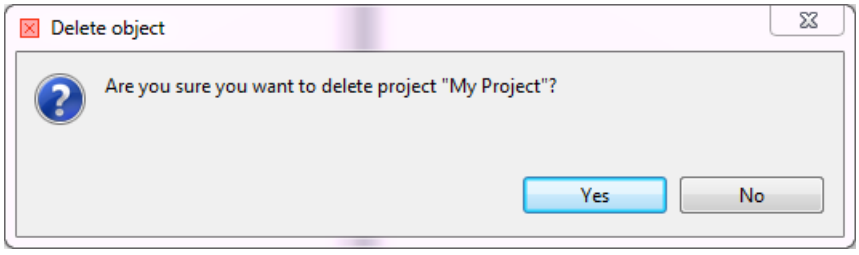


1. In the Project screen, in the **Project name** field, specify the name of the project.
2. To keep the default location to store the project, leave the **Use default location** checkbox selected. If you want to change the location, clear the checkbox and enter the name of the new directory into the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

Deleting Project

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

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



2. **Delete project** dialog box is to confirm the deletion of the project's contents. These are the project configuration files and scripts stored in the file system. Click **Yes** if you want the contents to be deleted as well. To keep the contents, click **No**.

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

Project security

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

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

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

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

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

Master password for local configuration

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

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

Use Windows Integration password provider

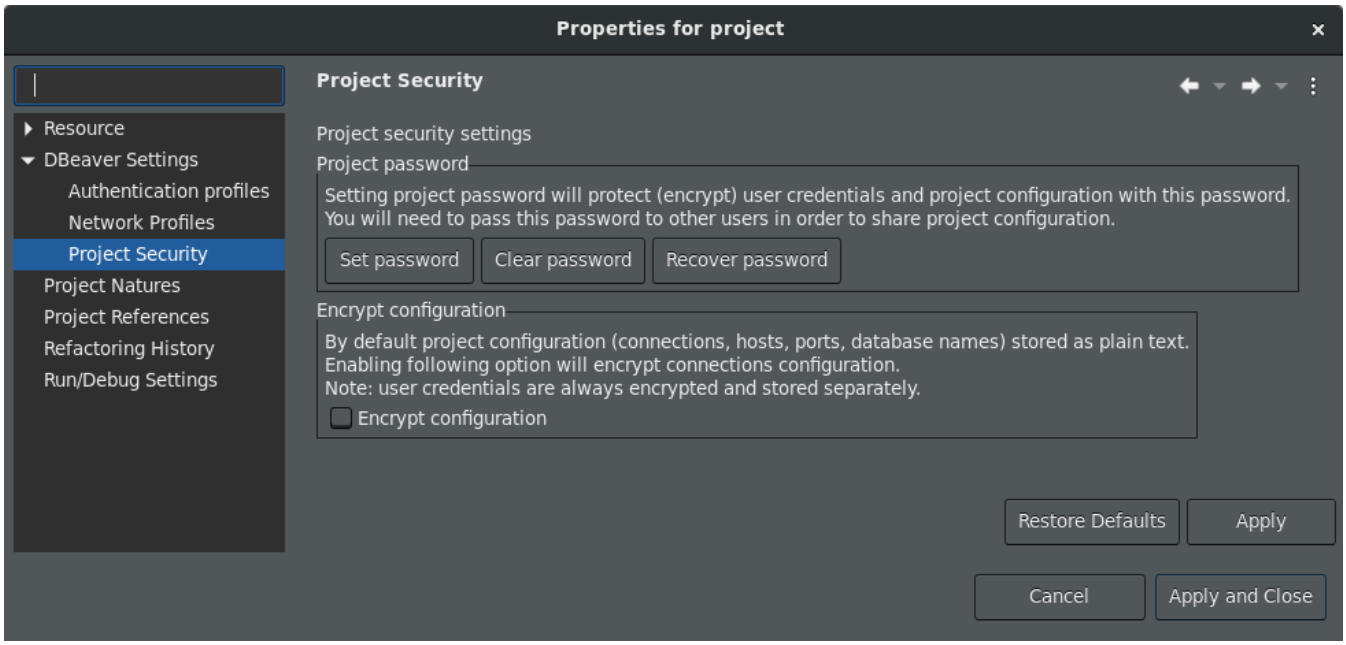
You can disable the default password provider and enable a "Windows Integration" provider. This provider does not need a master password but it uses a randomly generated password stored in a local user secure storage (in Windows). This is easier (as you don't need to remember the master password) but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

Project password

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

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

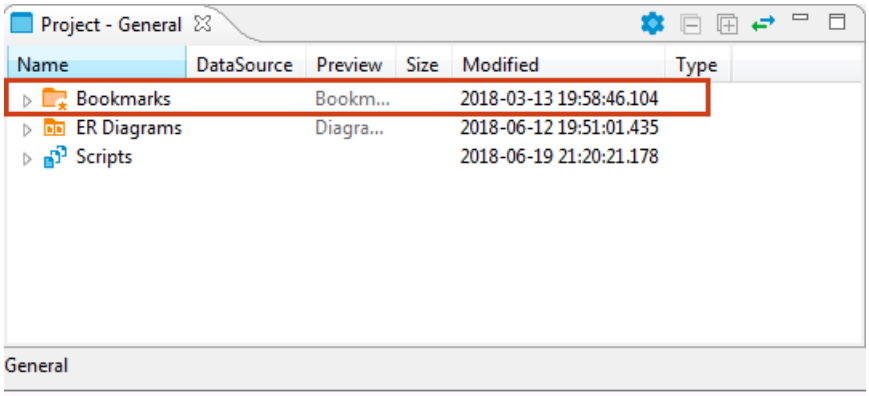
- Clicking on main menu File->Project security
- Clicking on "Configure" icon in the project explorer view toolbar and switching to the Project Security tab
- Pressing `ALT+Enter` on a project element in `Projects` view and switch to Project Security tab



On the project security page click on the "Set Password" button to enable the project password. Click on Clear to disable it (you will need to enter a current project password to clear it).

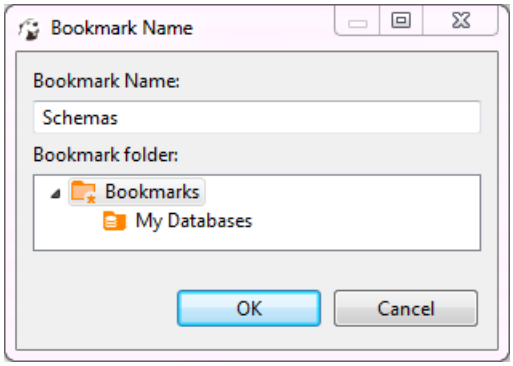
Bookmarks

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



To create a bookmark:

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



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

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

Shortcuts

Here is a complete list of default hotkeys in DBeaver UI for Windows, Linux, and macOS users. It will help you work in DBeaver faster and more efficiently. Remember that you can always change the keyboard shortcut in the DBeaver settings. Go through: *Window - Preferences - User Interface - Keys*. Select command and add a keyboard shortcut to the *Binding row.*

Connection

Shortcut for Windows/Linux	Shortcut for macOS	Action
F4	F4	Open object editor
Ctrl+Shift+D	⌃⇧D	Open database meta-object
Ctrl+Alt+Enter	⌃⌥↩	Open a new SQL console. No script file will be created.
Ctrl+]	⌃]	Create a new SQL script(***)
F3	F3	Open existing SQL script (***)
Ctrl+Enter	⌃↩	Open most recent SQL script(***) for an active connection
Ctrl+Alt+Shift+K	⌃4	Commit changes in current session
Alt+`	⌃V	Shows context menu with database tools
Ctrl+Alt+Shift+U	⌃⌥⇧U	Generates UUID/GUID and inserts into an active text editor or data editor
Ctrl+Alt+Shift+B	⌃⌥⇧B	Open database dashboard
Ctrl+Alt+Shift+R	⌃8	Rollback changes in current session
Ctrl+0	⌃0	Select active schema for current database
Ctrl+9	⌃9	Select connection for current editor

Result Set

Shortcut for Windows/Linux	Shortcut for macOS	Action
Alt+Insert	⌥Insert	Add new row
Ctrl+D	⌃D	Copy values from row above to current row
Ctrl+Alt+D	⌃⌥D	Copy values from row below to current row
Alt+Delete	⌥⌦	Delete current row
Ctrl+Alt+Insert	⌃⌥Insert	Duplicate current row
Enter	↩	Edit cell value with inline editor
Esc	Esc	Reset cell to original value
Ctrl+S	⌃S	Apply data changes
Ctrl+Alt+Shift+T	⌃⌥⇧T	Set focus to filter editor/data editor
Ctrl+Shift+7	⌃⇧7	Switch focus to results viewer/active panel
Ctrl+Alt+Shift+Enter	⌃⌥⇧↩	Apply cell changes
Shift+Enter	⇧↩	Edit cell value in separate dialog/editor
Ctrl+Shift+=	⌃⇧=	Fetch all rows
Ctrl+Alt+;	⌃⌥;	Fetch next page of results
Ctrl+F11	⌃F11	Filter by unique attribute values
F11	F11	Filter context menu
Ctrl+Alt+Shift+Left	⌃⌥⇧←	Move to first row
Ctrl+Alt+Shift+Right	⌃⌥⇧→	Move to last row

Shortcut for Windows/Linux	Shortcut for macOS	Action
Alt+Space	⌥Space	Follow foreign key link
Ctrl+Alt+Right	⌃⌥→	Move to next row
Ctrl+Alt+Left	⌃⌥←	Move to previous row
Ctrl+1	⌃1	Foreign keys and tables referencing current table
Ctrl+R	⌃R	Reject data changes
Ctrl+Backspace	⌃⌫	Set cell to default value
Ctrl+`	⌃`	Switch results presentation
Tab	Tab	Toggle results Grid/Record view
F7	F7	Toggle extra result panels
Ctrl+2	⌃2	Toggle results sort order (ascending/descending/default)

Data Editor

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Space	⌃Space	Enable autocomplete
Ctrl+F	⌘F	Find and replace text
Ctrl+Shift+Space	⌃⇧Space	Show Context Information in DIalogs and Windows
Ctrl+Alt+Space	⌃⌥Space	Show Context Information (SQL Editor Context)
Ctrl+X	⌘X	Cut the selection to the clipboard
Ctrl+C	⌘C	Copy the selection to the clipboard
Ctrl+V	⌘V	Paste from the clipboard
Delete	⌦	Delete the selection
Ctrl+O	⌘O	Export Diagram
Ctrl+K	⌘K	Find next item
Ctrl+Shift+K	⇧⌘K	Find previous item
Ctrl+J	⌘J	Incremental find
Ctrl+Shift+J	⇧⌘J	Incremental find reverse
Ctrl+Shift+Q	⌃⇧Q	Toggles quick diff information display on the line number ruler
Ctrl+1	⌘1	Suggest possible fixes for a problem
Ctrl+Y	⇧⌘Z	Redo the last operation
Ctrl+Z	⌘Z	Undo the last operation
Shift+Delete	⇧⌦	Remove selected Pictogram Elements
Ctrl+A	⌘A	Select all
Alt+Shift+A	⌥⌘A	Toggle block/column selection in the current text editor
Ctrl+Shift+Insert	⇧⌘Insert	Toggle insert mode
Alt+Shift+Y	⌥⌘Y	Toggle word wrap in the current text editor
Ctrl+F5	⌘F5	Update selected Pictogram Elements
Alt+/	⌃.	Context insensitive completion
Ctrl+Shift+Space	⌃⇧Space	Context insensitive completion (SQL Editor Context)

Text Editor

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Shift+Y	⇧⌘Y	Changes the selection to lowercase
Ctrl+Shift+X	⇧⌘X	Changes the selection to uppercase
Ctrl+Numpad_Subtract	⌘Numpad_Subtract	Collapses the folded region at the current selection
Ctrl+Shift+Numpad_Divide	⇧⌘Numpad_Divide	Collapses all folded regions
Ctrl+Alt+Down	⌥⌘↓	Duplicates the selected lines and moves the selection to the copy
Ctrl+D	⌘D	Delete a line of text
Ctrl+Delete	⌥⌦	Delete the next word
Ctrl+Backspace	⌥⌫	Delete the previous word
Ctrl+Shift+Delete	⇧⌘⌦	Delete to the end of the line of text
Ctrl+Alt+Up	⌥⌘↑	Duplicates the selected lines and leaves the selection unchanged
Ctrl+Numpad_Add	⌘Numpad_Add	Expands the folded region at the current selection
Ctrl+Numpad_Multiply	⌘Numpad_Multiply	Expands all folded regions
Ctrl+Shift+Enter	⇧⌘↩	Adds a new line above the current line
Shift+Enter	⇧↩	Adds a new line below the current line
Ctrl+Alt+J		Join lines of text (Editing Text)
Ctrl+Shift+J	⌃⇧J	Join lines of text (SQL Editor Context)
End	⌘→	Go to the end of the line of text
Home	⌘←	Go to the start of the line of text
Alt+Down	⌃⇧↓	Moves the selected lines down (Editing Text)
Ctrl+Shift+Down	⌥↓	Moves the selected lines down (SQL Editor Context)
Alt+Up	⌃⇧↑	Moves the selected lines up (Editing Text)
Ctrl+Shift+Up	⌥↑	Moves the selected lines up (SQL Editor Context)
Ctrl+Right	⌥→	Go to the next word
Ctrl+Left	⌥←	Go to the previous word
Ctrl+Shift+Numpad_Multiply	⇧⌘Numpad_Multiply	Resets the folding structure
Ctrl+Down		Scroll down one line of text
Ctrl+Up		Scroll up one line of text
Shift+End	⇧⌘→	Select to the end of the line of text
Shift+Home	⇧⌘←	Select to the beginning of the line of text
Ctrl+Shift+Right	⌥⇧→	Select the next word
Ctrl+Shift+Left	⌥⇧←	Select the previous word
	⇧↘	Select to the end of the text
	⇧↖	Select to the beginning of the text
F2	F2	Displays information for the current caret location in a focused hover
Ctrl+End	↘	Go to the end of the text
Ctrl+Home	↖	Go to the beginning of the text
Ctrl+Numpad_Divide	⌘Numpad_Divide	Toggles folding in the current editor
Insert	Insert	Toggle overwrite mode
Ctrl++	⌘+	Zoom in text, increase default font size for text editors

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+=	⌘=	Zoom in text, increase default font size for text editors
Ctrl+-	⌘-	Zoom out text, decrease default font size for text editors

SQL Editor

Shortcut for Windows/Linux	Shortcut for macOS	Action
Alt+X	⌥X	Execute script(**)
Ctrl+Enter	⌃↩	Execute SQL statement(*)
Ctrl+	⌃	Execute SQL statement in a new tab
Ctrl+Alt+Shift+X	⌃⌥⇧X	Execute script's statements in separate results tabs
Ctrl+/	⌃/	Add or remove single line comment
Ctrl+Shift+/	⌃⇧/	Add or remove multi line comment
Alt+6	⌥6	Switch active SQL editor panel
Ctrl+6	⌃6	Show/hide results panel
Ctrl+Shift+	⌃⇧	Close results tab
Ctrl+Shift+F	⌃⇧F	Format text
Ctrl+Alt+'	⌃⌥'	Select value of the selected SQL expression
Ctrl+Shift+E	⌃⇧E	Explain execution plan
Ctrl+Shift+P	⌃⇧P	Position cursor on the matching bracket
Ctrl+Alt+Shift+O	⌃⌥⇧O	Load SQL script from file system
Ctrl+Shift+6	⌃⇧6	Maximize/normalize results panel
Alt+Down	⌥↓	Switch to the next query
Alt+Up	⌥↑	Switch to the previous query
Ctrl+F2	⌃F2	Rename current SQL script
F4	F4	Open editor of current (highlighted) database object
Ctrl+Alt+Shift+A	⌃⌥⇧A	Select and show all rows (no fetch size limit)
Ctrl+Alt+Shift+C	⌃⌥⇧C	Select row count for query under cursor
Ctrl+Shift+.	⌃⇧.	Set active connection from database navigator selection
Ctrl+Shift+O	⌃⇧O	Show server output console
Ctrl+Alt+Shift+W	⌃⌥⇧W	Toggles text editor soft word wrap

Visual Query Builder

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+B	⌃B	Open Visual Query Builder
Ctrl+Shift+B	⌃⇧B	Show/hide generated SQL query text

Search

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+H	⌃H	Open the Search dialog
Ctrl+Alt+G	⌥⌘G	Searches the files in the workspace for specific text
Ctrl+Alt+Shift+F	⌥⇧⌘L	Quick search in Windows

Data viewer

Shortcut for Windows/Linux	Shortcut for macOS	Action
Tab	Tab	Cycle through hand tool and previously used tool
Ctrl+=	⌘=	Zoom In
Ctrl+-	⌘-	Zoom Out
Alt+Shift+Q, Q	⌥⌘Q Q	Shows a particular view
Alt+Shift+Q, B	⌥⌘Q B	Shows a particular view (Breakpoints)
Alt+Shift+Q, C	⌥⌘Q C	Shows a particular view (Console)
Alt+Shift+Q, L	⌥⌘Q L	Shows a particular view (Error Log)
Alt+Shift+Q, Z	⌥⌘Q Z	Shows a particular view (History)
Alt+Shift+Q, O	⌥⌘Q O	Shows a particular view (Outline)
Alt+Shift+Q, X	⌥⌘Q X	Shows a particular view (Problems)
Alt+Shift+Q, S	⌥⌘Q S	Shows a particular view (Search)
Alt+Shift+Q, Y	⌥⌘Q Y	Shows a particular view (Synchronize)
Alt+Shift+Q, V	⌥⌘Q V	Shows a particular view (Variables)

Window

Shortcut for Windows/Linux	Shortcut for macOS	Action
F12	⌘F12	Activate Editor
	⌘W	Closes the active Dialog
Ctrl+3	⌘3	Find actions. Quickly access UI elements
Ctrl+M	⌃M	Maximize/restore a state of active view or editor
Ctrl+F6	⌘F6	Switch to the next editor
Ctrl+F8	⌘F8	Switch to the next perspective
Ctrl+F7	⌘F7	Switch to the next view
Ctrl+Shift+F6	⇧⌘F6	Switch to the previous editor
Ctrl+Shift+F8	⇧⌘F8	Switch to the previous perspective
Ctrl+Shift+F7	⇧⌘F7	Switch to the previous view
Ctrl+E	⌘E	Open the editor drop-down list
	⇧F10	Show the context menu
Alt+Shift+F3	⌥⇧F3	Shows contribution information for the currently selected element
Ctrl+Shift+L	⇧⌘L	Show the key assist dialog
Ctrl+F10	⌘F10	Show the context menu for the ruler
Alt+-	⇧⌘F10	Show the system menu
Ctrl+F10	⌘F10	Show the view menu
Ctrl+Shift+E	⇧⌘E	Switch to an editor
Alt+F11	⌃⌘F	Toggles the window between full screen and normal
Ctrl+_	⇧⌘-	Split or join the currently active editor (Horizontal)
Ctrl+_	⇧⌘[	Split or join the currently active editor (Vertical)

Database Navigator

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Shift+,	⌃⇧,	Link with editor
Ctrl+Alt+Shift+D	⌃⌥⇧D	Add Bookmark
Enter	↩	Opens task configuration
Ctrl+Shift+A	⌃⇧A	Set as default object

Navigation tree

Shortcut for Windows/Linux	Shortcut for macOS	Action
Alt+Left	⌘[	Move backward in the editor navigation history
Ctrl+Shift+Numpad_Divide	⇧⌘Numpad_Divide	Collapse the current tree
Ctrl+Shift+Numpad_Multiply	⇧⌘Numpad_Multiply	Expand the current tree
Alt+Right	⌘]	Move forward in the editor navigation history
Ctrl+G	⌃G	Go to a specified line of text in Windows
Ctrl+L	⌘L	Go to a specified line of text (Editing Text)
Ctrl+.	⌘.	Navigate to the next item
Ctrl+Alt+Right	⌃⌥→	Next edit location
Alt+F7	⌥F7	Switch to the next page
Alt+PageDown	⌥⇟	Switch to the next sub-tab
Ctrl+PageDown	⌃⇟	Switch to the next tab
Ctrl+Shift+R	⇧⌘R	Open an editor on a particular item
Ctrl+,	⇧⌘.	Navigate to the previous item
Ctrl+Alt+Left	⌃Q	Previous edit location
Alt+Shift+F7	⌥⇧F7	Switch to the previous page
Alt+PageUp	⌥⇞	Switch to the previous sub-tab
Ctrl+PageUp	⌃⇞	Switch to the previous tab
Alt+Shift+W	⌥⌘W	Open the Show In menu

File

Shortcut for Windows/Linux	Shortcut for macOS	Action
F2	F2	Rename the selected item
F5	F5	Refresh the selected items
Ctrl+S	⌘S	Save the changes in current file
Ctrl+Shift+S	⇧⌘S	Save changes in all open files
Ctrl+F4	⌘W	Close the active editor
Ctrl+Shift+F4	⇧⌘W	Close all editors
Ctrl+N	⌘N	Open the New item wizard
Alt+Shift+N	⌥⌘N	Open the New menu
Ctrl+O	⌃O	Open a file
Ctrl+P	⌘P	Print
Alt+Enter	⌘I	Display the properties of the selected item

Run/Debug

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Alt+M	⌥⌘M	Add memory block
Ctrl+W	⌘W	Close the selected rendering
F11	⌘F11	Launch in debug mode
Ctrl+Z	⌃D	Send end of file
Ctrl+G	⌘G	Go to Address
Ctrl+N	⌘N	Add a new rendering
Ctrl+Alt+N	⌥⌘N	Show renderings from next memory monitor.
Ctrl+Shift+.	⇧⌘.	Load next page of memory
Ctrl+Shift+,	⇧⌘,	Load previous page of memory
Ctrl+F11	⇧⌘F11	Launch in run mode
Ctrl+R	⌘R	Resume and break when execution reaches the current line
Ctrl+Alt+B	⌃⌥B	Sets whether or not any breakpoint should suspend execution
F5	F5	Step into
F6	F6	Step over
F7	F7	Step return
F8	F8	Resume
Ctrl+F2	⌘F2	Terminate
Ctrl+Shift+B	⇧⌘B	Creates or removes a breakpoint
Ctrl+T	⌘T	Toggle visibility of the Memory Monitors Pane
Shift+F5	⇧F5	Toggles enablement of debug step filters

Git

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+Left	⌘←	Collapse Working Tree
Ctrl+#	⌥⌘3	Commit
Ctrl+C	⌘C	Copy Path to Clipboard
Ctrl+O	⌘O	Show the quick outline for a unified diff
F2	F2	Rename Branch
Ctrl+Shift+K	⌃⇧K	Commit changes to Git
Ctrl+Shift+U	⌃⇧U	Update changes from Git

Project

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+B	⌘B	Build all projects

Oracle

Shortcut for Windows/Linux	Shortcut for macOS	Action
Ctrl+F9	⌃F9	Compile

Utility

Shortcut for Windows&Linux	Shortcut for macOS	Action
Ctrl+Shift+C	⌃⇧C	Advanced Copy
Ctrl+Shift+V	⌃⇧V	Paste with extra settings
CTRL+ALT+PAGE_UP CTRL+ALT+PAGE_DOWN		Switch between tabs in database object Properties view

Help

Shortcut for Windows&Linux	Shortcut for macOS	Action
F1	F1	Open the documentation

References

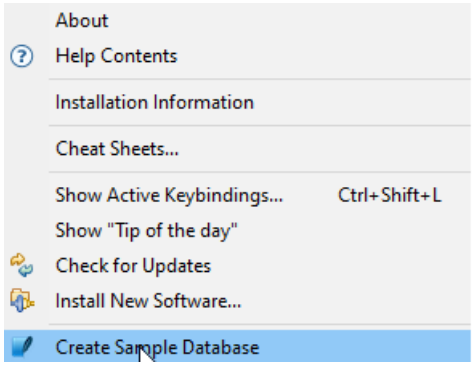
`*` - Current query is the query under cursor or the selected text. Query is separated from other script queries by delimiter (; by default) or by empty lines.
`**` - Current script is a set of all queries in the current SQL file. If there is a text selection then only queries in this selection are processed.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

Sample Database

Introduction

If you want to test **DBeaver** features, you can create a demo database. There is a specialized database that can be created. You can create a sample *SQLite* database at the first launch. The database can be added at any time by the following sequence. Chinook is used as a reference database

> Help -> Create Sample Base



Database Connections

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

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

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

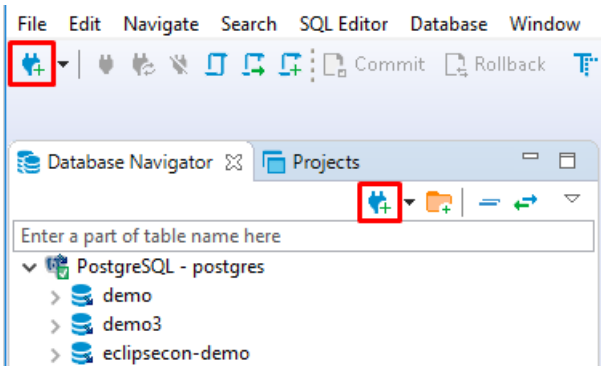
Database connections might have the following states:

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

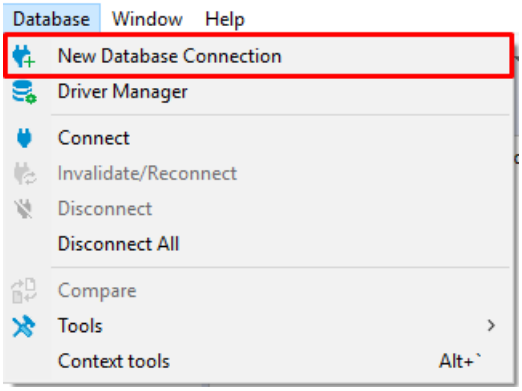
Create Connection

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

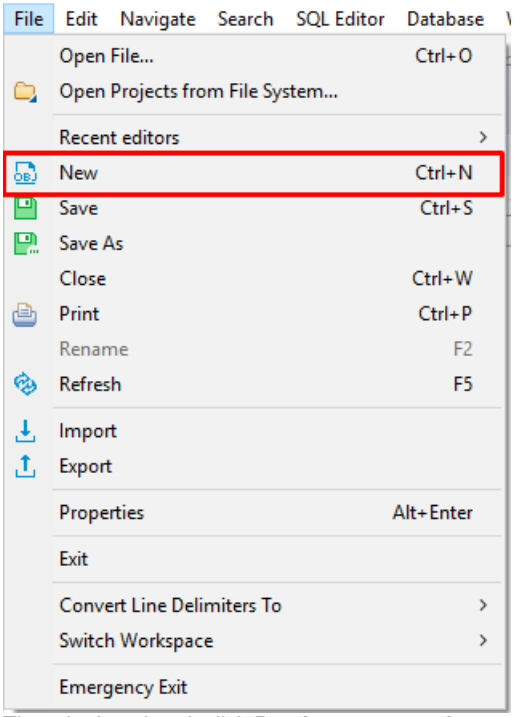
- Click the **New Connection Wizard** button in the application toolbar or in the Database Navigator view toolbar:



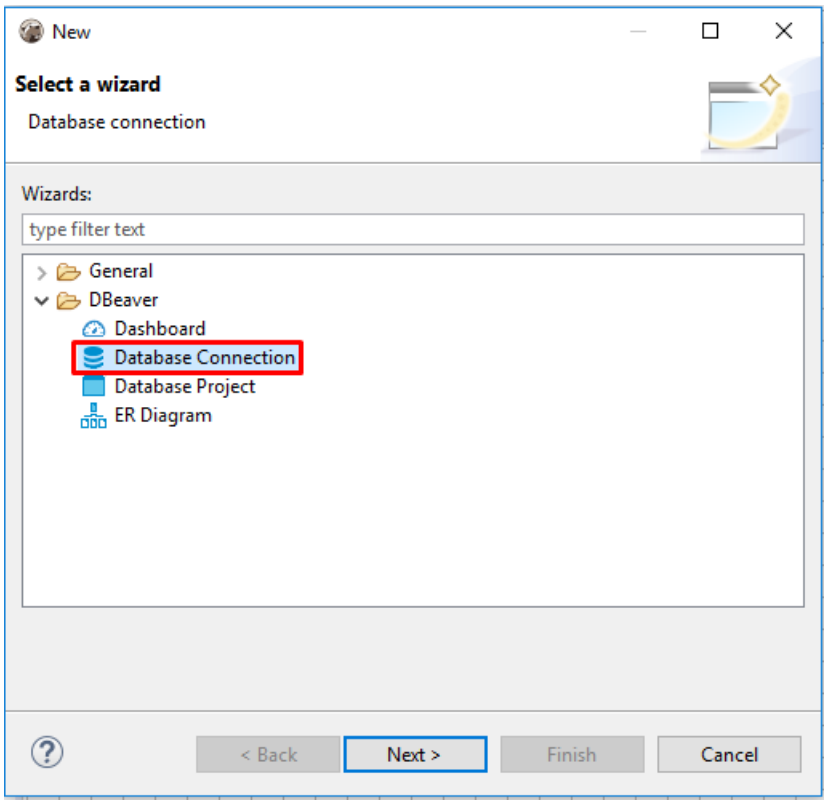
- Click **Database -> New Connection** in the menu bar:



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

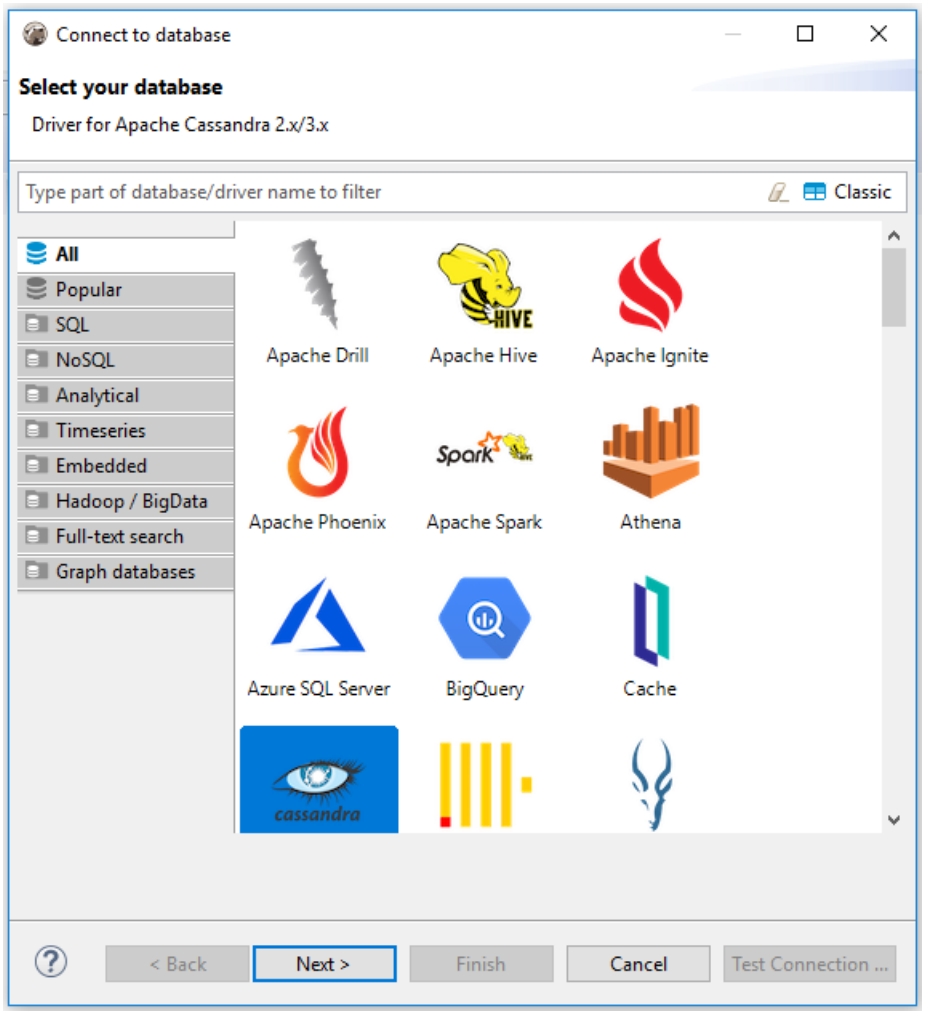


Then, in the wizard, click **Database connection** and then click **Next**:



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

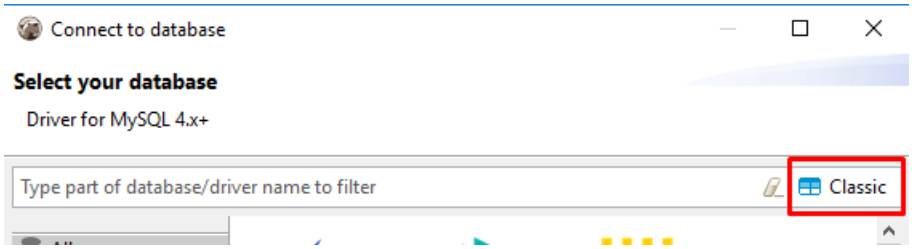
1. Choose a driver for the new connection: click the name of the suitable database type in the gallery. Then click **Next**.



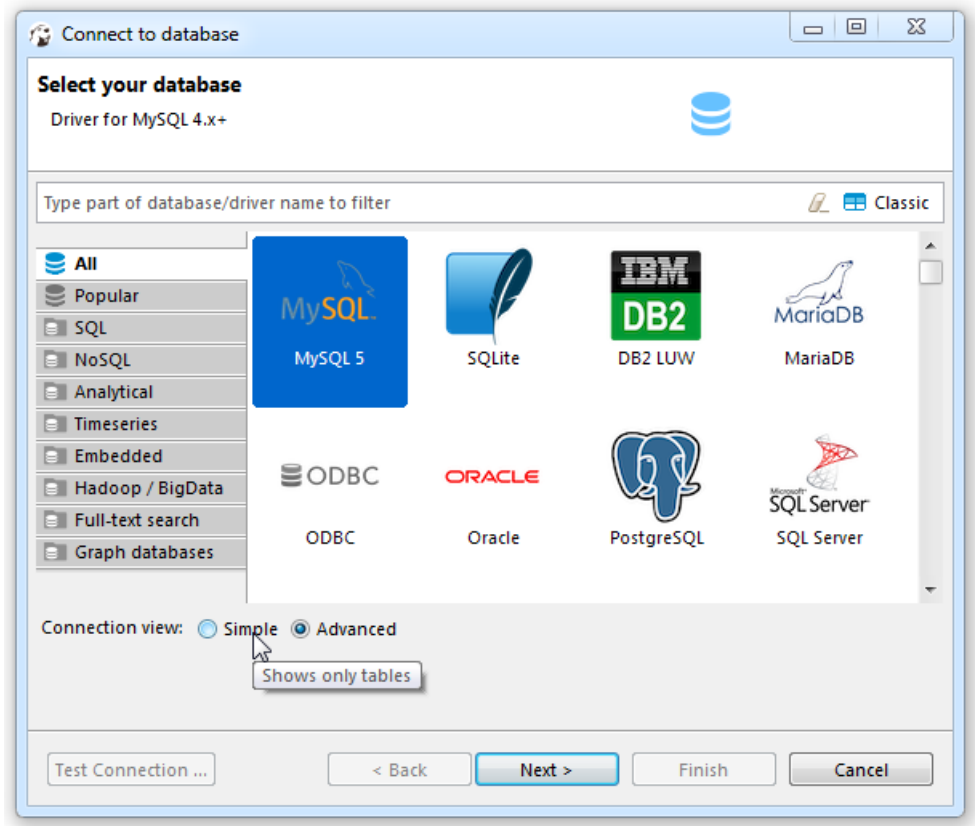
To quickly find the needed driver, you can type a hint in the text field above the list of drivers.
If you cannot find a driver for your database then probably there is no suitable driver and you need to create one. Please see Database Drivers article.

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

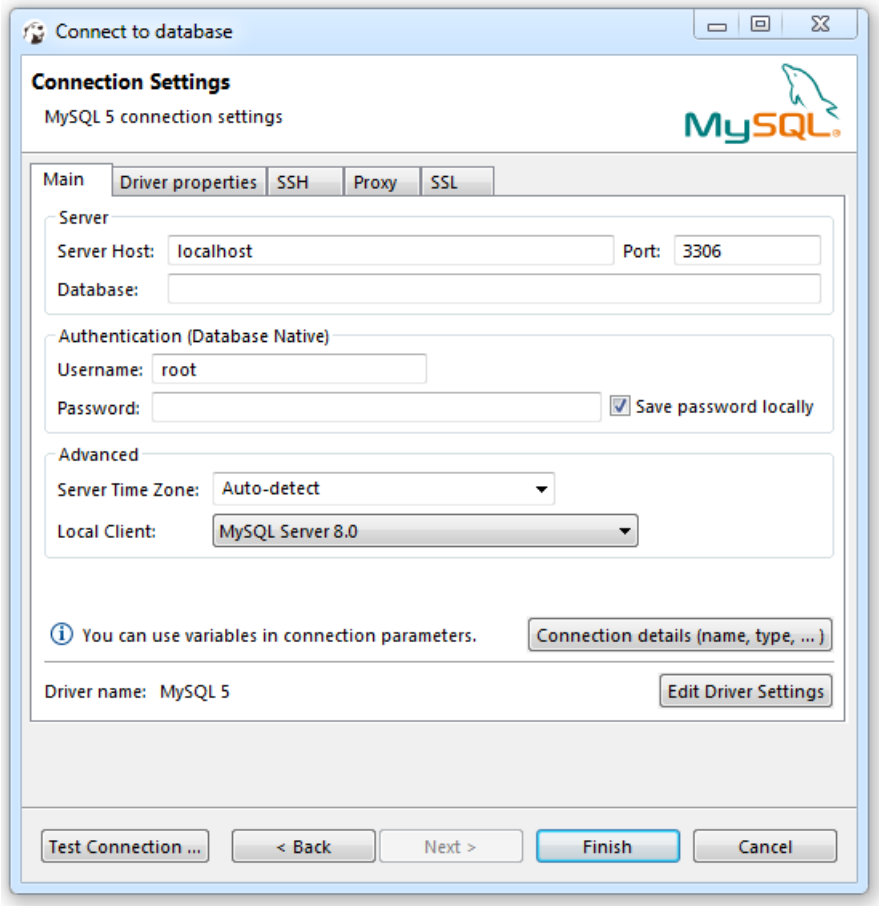
If you prefer the classic list view of the available drivers, use the **Classic** button.



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



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



For most drivers required settings include:

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

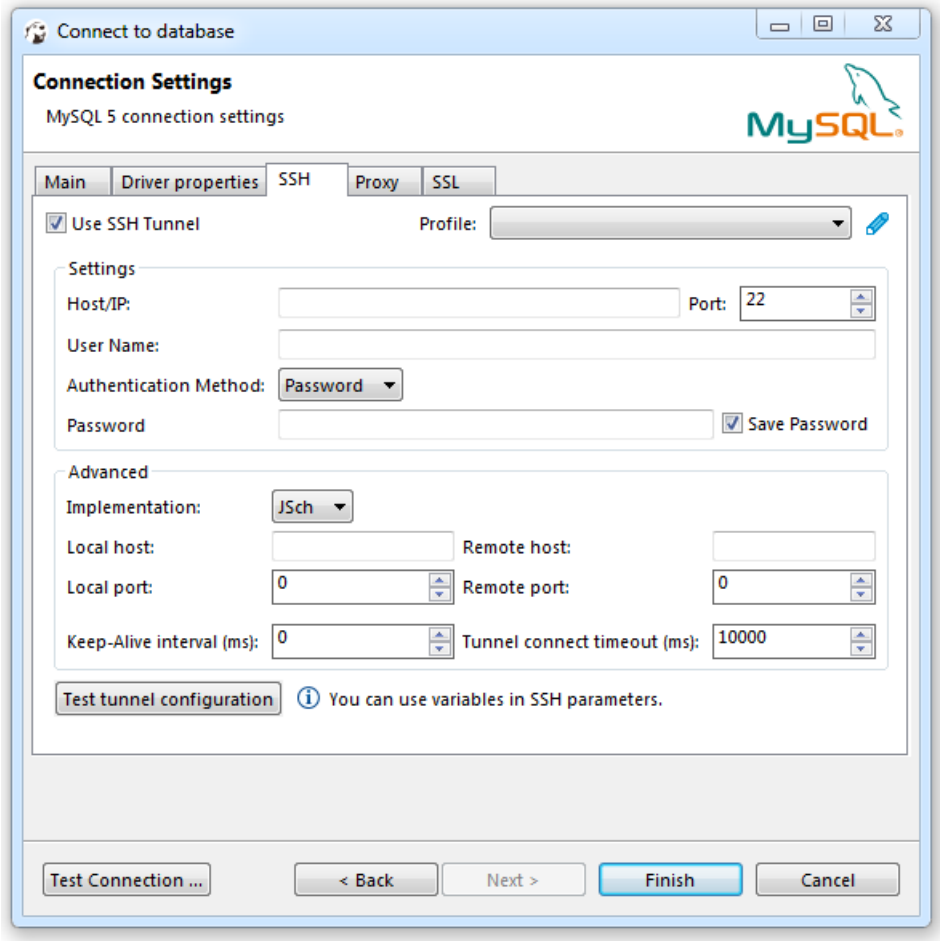
However, the number and type of connection properties are very dependent on the driver.
For example, embedded drivers (such as SQLite, Derby Embedded, HSQLDB, H2 Embedded), unlike remote ones, require only the path to the database.

3. If necessary, specify advanced settings, see **Advanced Settings** section below, and click **Next**.
4. To test if the connection works, click **Test Connection**.
5. Click **Finish**. The connection appears in the tree of connections in the Database Navigator and DBeaver actually connects to the database.

Advanced Settings

Network Settings (SSH, SOCKS, SSL)

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



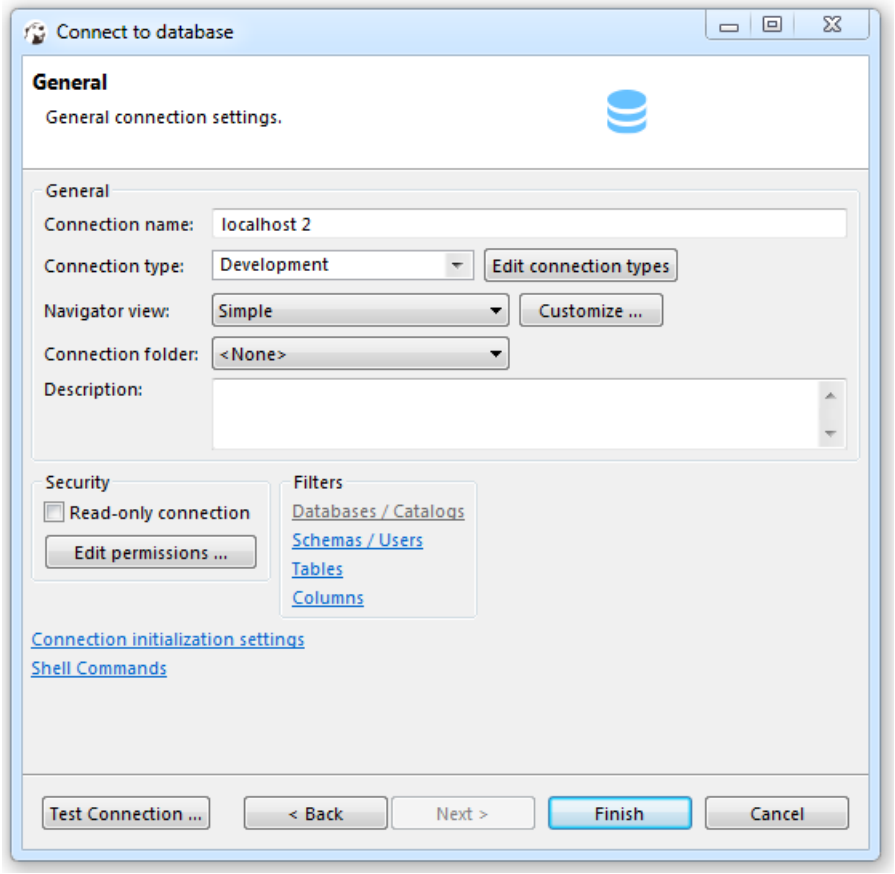
DBeaver supports following SSH authentication methods: user/password, public key authentication and agent authentication. Supported implementations for agent authentications are pageant and ssh-agent.

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

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

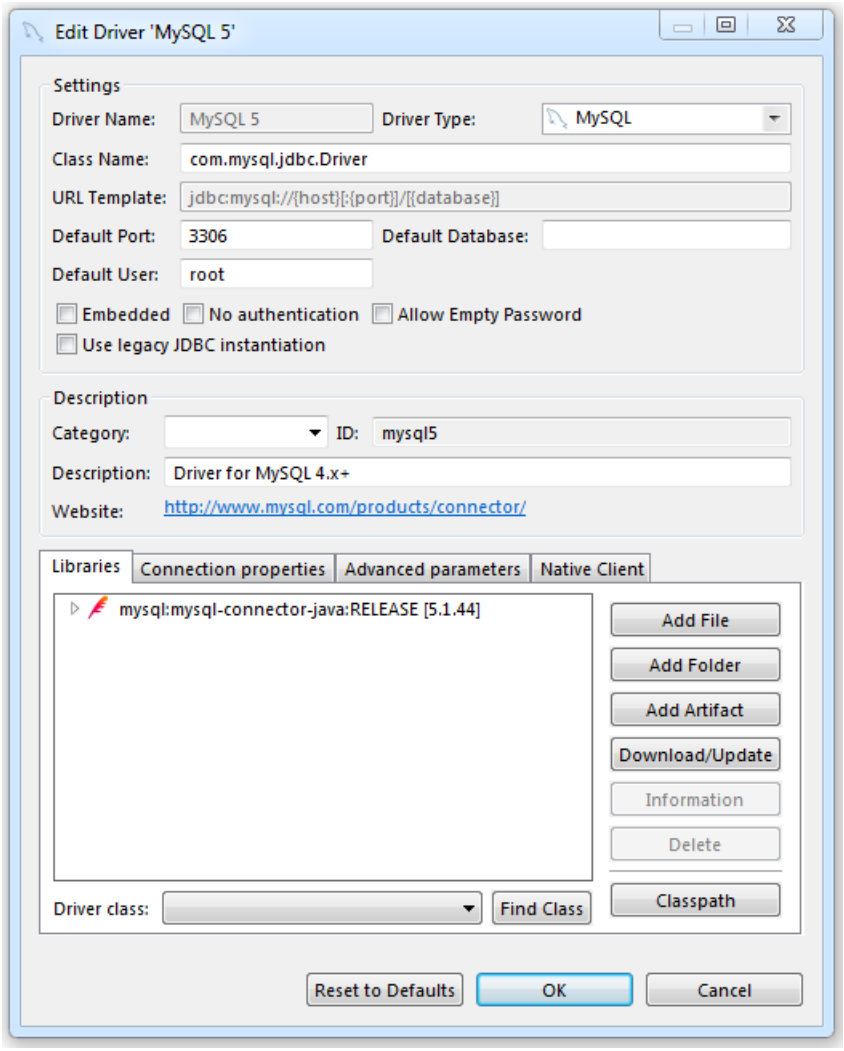
Connection Details (name, type, etc.)

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



Driver Properties

Each driver has its own set of additional properties. Refer to the driver documentation to get information about available properties and their values.



Variables in parameters

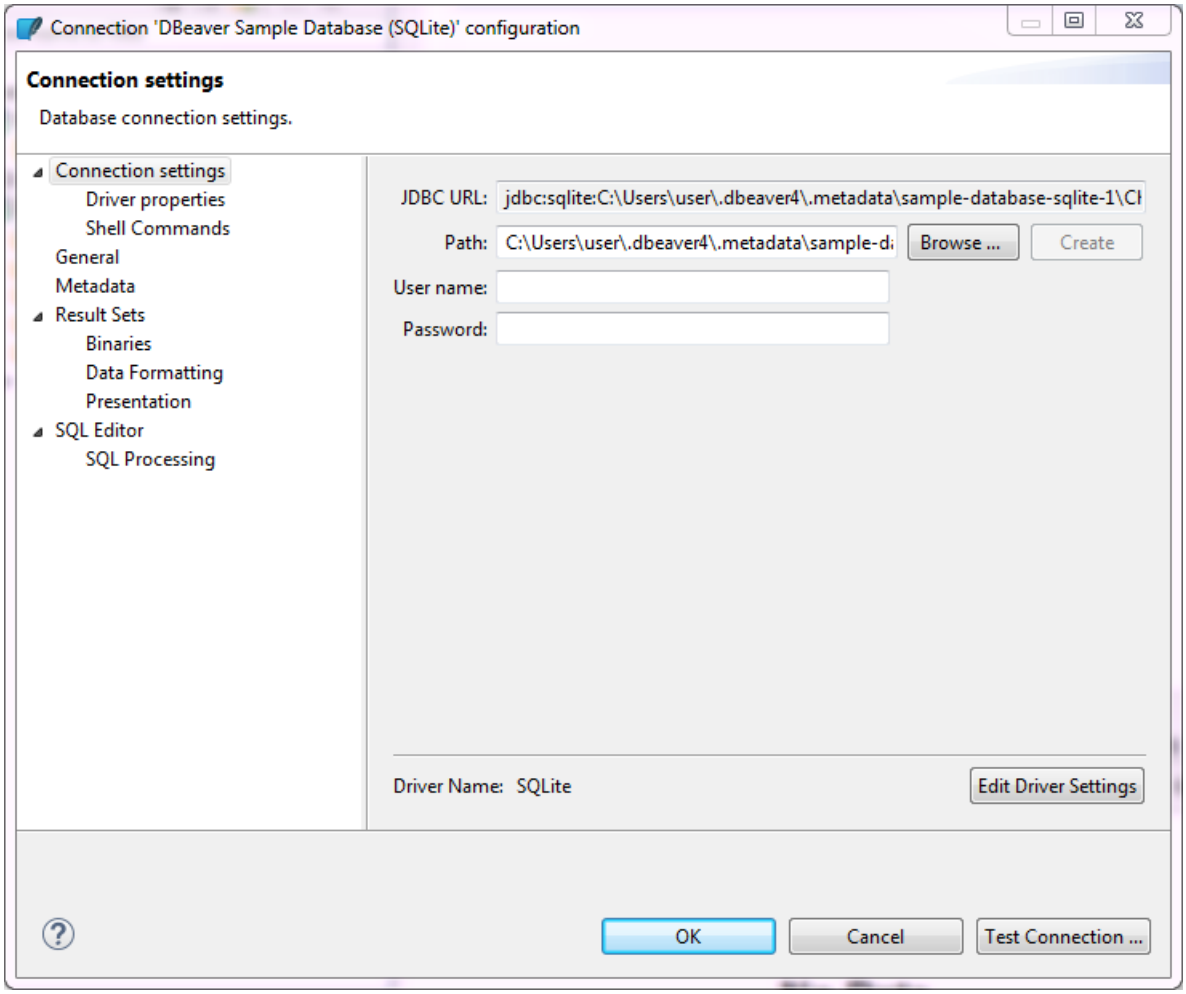
You can use variables in all connection parameters and in the driver properties. Variables are system environment variables or one of the following list:

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

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

Edit Connection

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



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

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

Driver settings

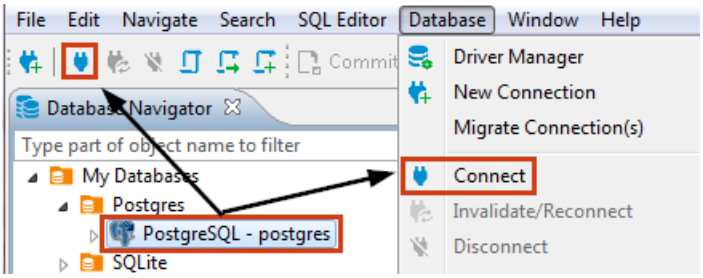
- In development

JDBC Time Zones

Connect to Database

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

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



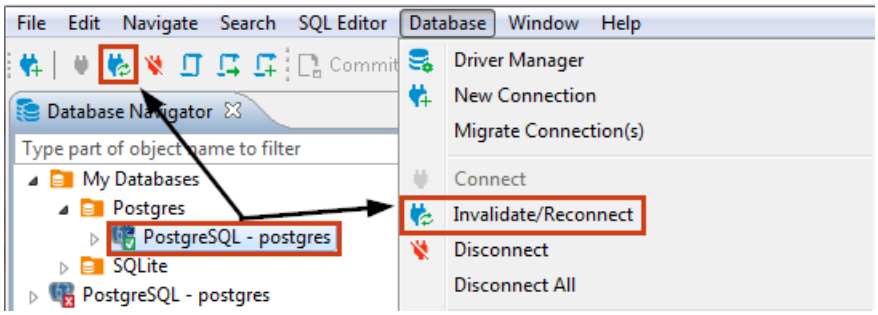
You can also right-click the connection and click **Connect** on the context menu.

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

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

Invalidate/Reconnect to Database

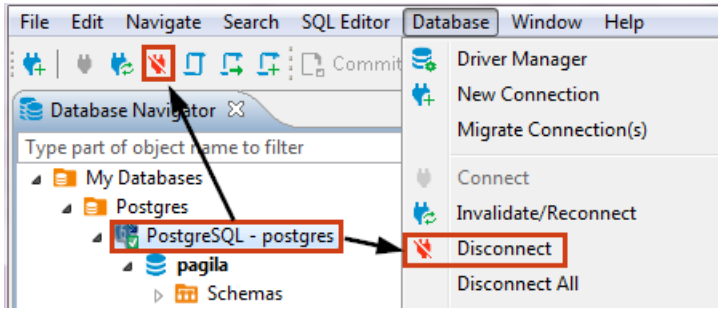
You might need to invalidate a database connection and then reconnect to it again in cases such as where the connection to the server is temporarily lost, etc. To invalidate a database connection and then reconnect to the database, click the database connection in the Database Navigator or Projects view,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



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

Disconnect from Database

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



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

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

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

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

Change current user password

DBeaver has a possibility to change credentials for the current database user.

Databases, which support this feature:

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

Changing expired user password

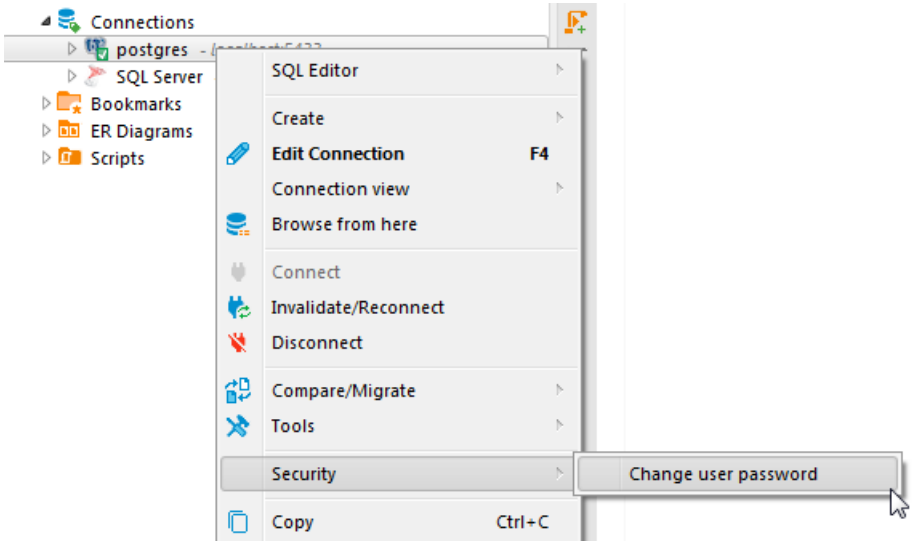
Different databases perform password management for users in a different ways. For some of them you can change the user password after the expiration date. It works for Oracle, PostgreSQL, and Netezza databases.

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

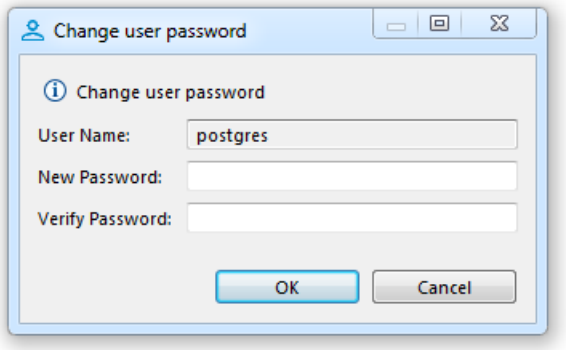
How to change user password

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

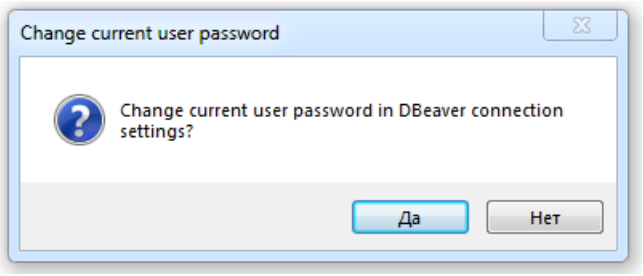
1. Connect to the database.
2. Open the context menu by right-clicking on the connection in the Navigation tree.
3. Select a "Security" point and click on a "Change user password" point in a sub-menu.



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



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



The password has been changed.

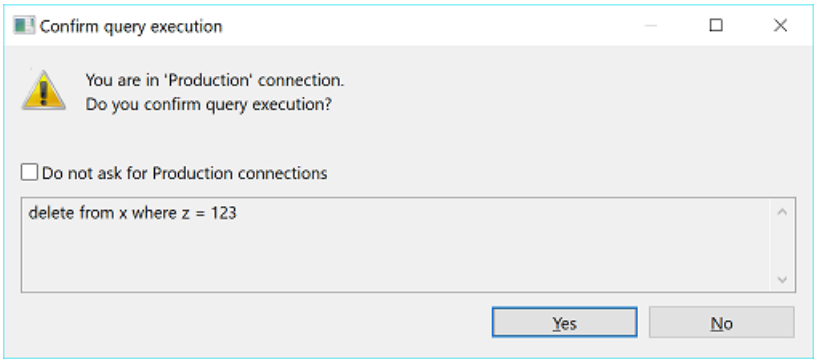
Connection Types

Connection types define how DBeaver behaves regarding:

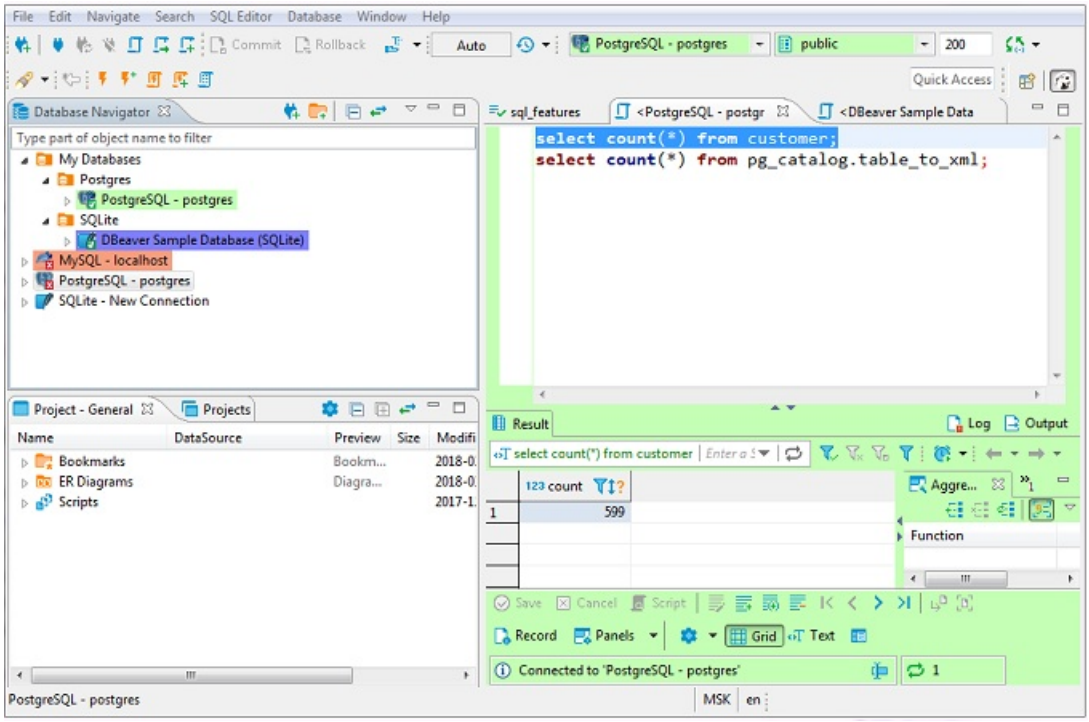
- Default transactions commit mode - with or without automatic commit of changes to the database.

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

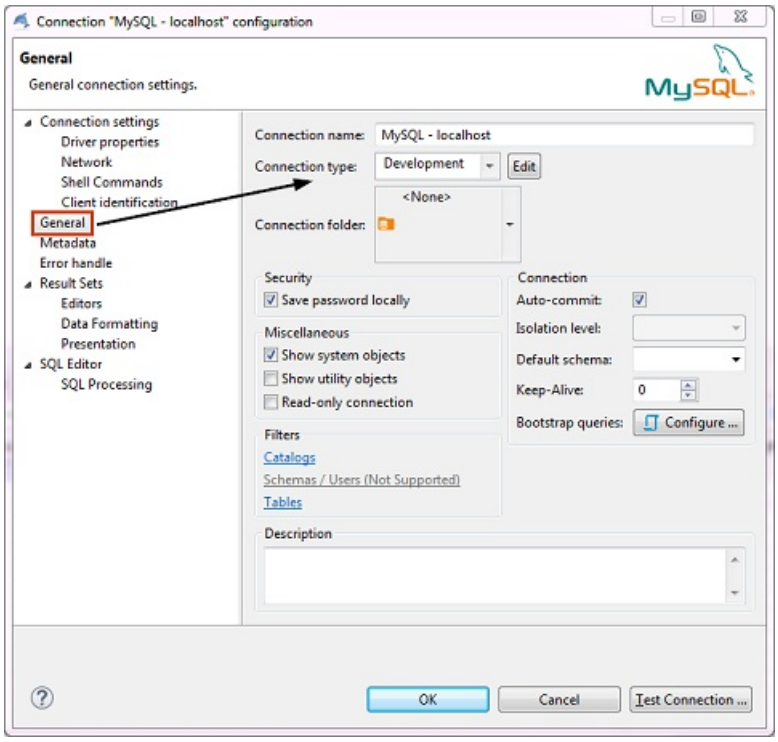
- SQL statements execution (with or without user confirmation): If set to the required user confirmation for SQL execution, DBeaver shows a confirmation message every time you attempt to execute a 'transaction' type of query (INSERT/DELETE/UPDATE, etc.):



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



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



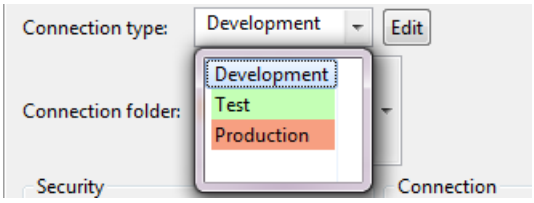
There are three default connection types – **Development**, **Test**, and **Production**. You can change the connection type for your database connection as well as you can create a new connection type, edit or delete an existing one.

Change Connection Type

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

To change the connection type:

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

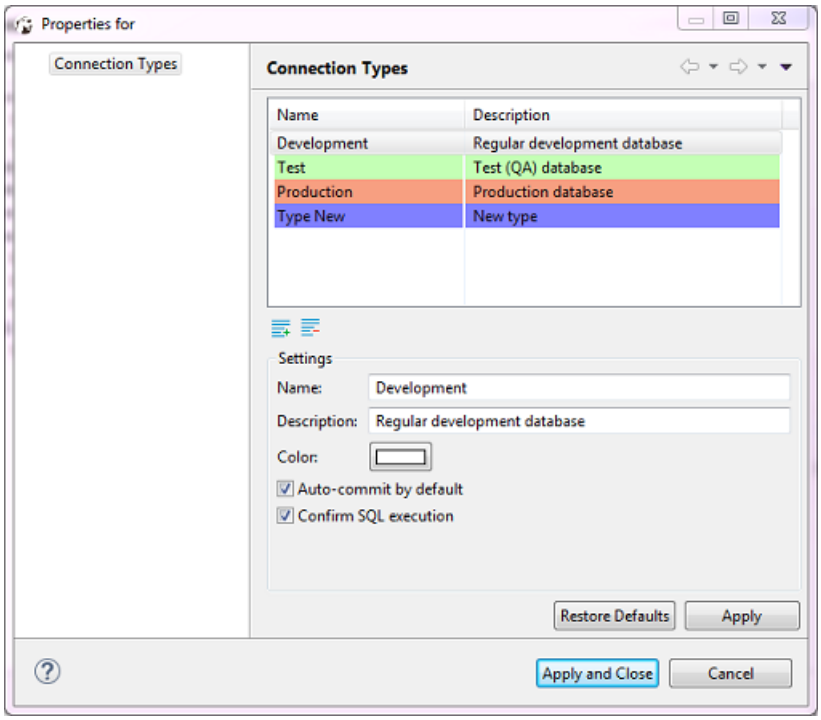


2. To test the connection, click **Test Connection**. To confirm the change, click **OK**.

Create Connection Type

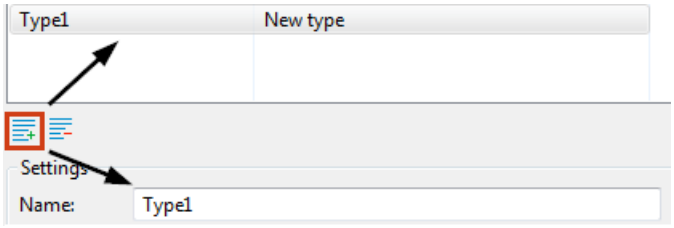
To create a connection type:

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



The window displays existing connection types and their settings.

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



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

Edit Connection Type

To edit a connection type:

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

Delete Connection Type

To delete a connection type:

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

2. In the Properties window, in the list of connection types, click the connection type to set the focus on it and then click the delete button under the list: 
3. Click **Yes** in the confirmation dialog box to confirm the deletion. Otherwise, click **No**.
4. Click **Apply** to apply the changes and keep the window open or click **Apply and Close** to apply the changes and close the window.

Auto and Manual Commit Modes

DBeaver supports two modes for committing changes to the database:

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

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

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

Auto-Commit Mode

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

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

Clicking the statistics field opens the Transaction Log.

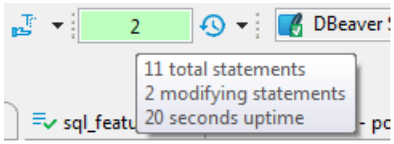
Manual Commit Mode

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

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

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

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



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

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

Clicking the statistics field opens the Transaction Log.

Smart Commit Mode

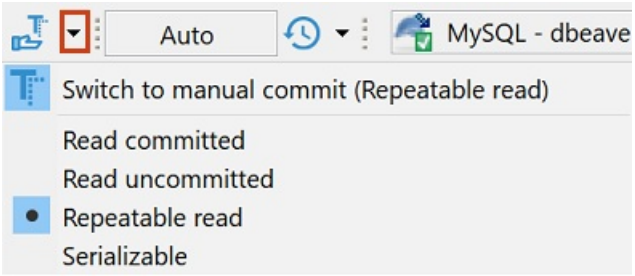
When smart commit is enabled and you are in auto-commit mode, which allows DBeaver to monitor your activity.
When you try to execute any data modifying query (UPDATE, INSERT, DELETE, UPSERT, MERGE, etc) DBeaver will switch to manual commit mode before executing your query. Also if you edit any table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

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

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

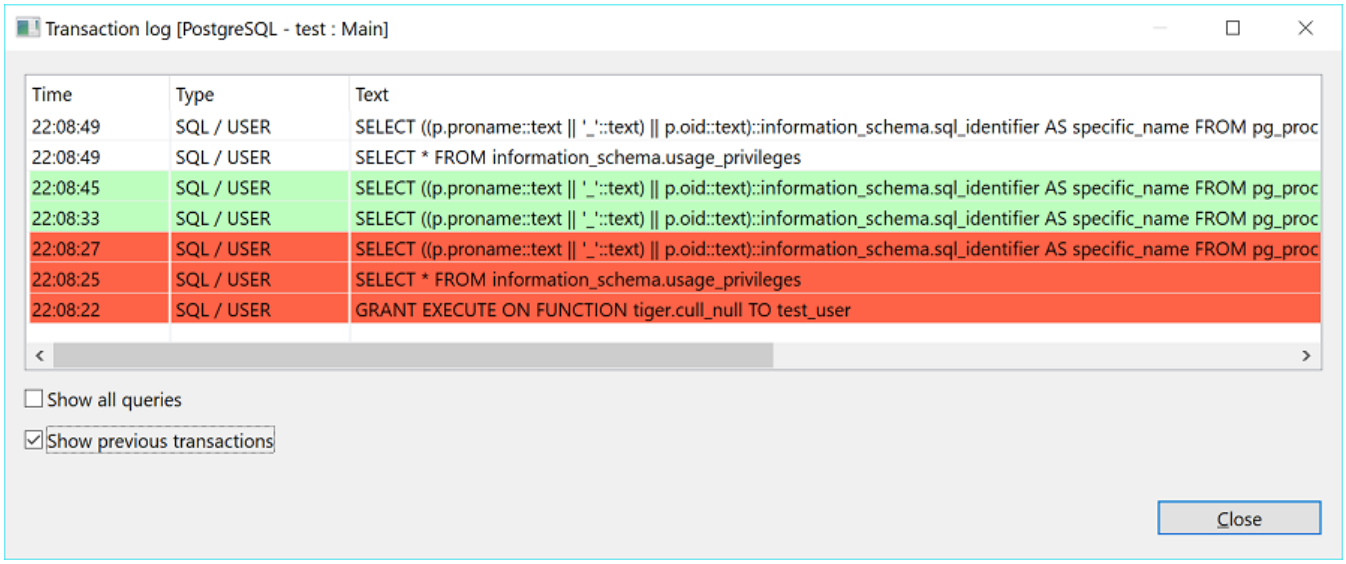
Transaction Isolation Level

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


Auto
MySQL - dbeave
Switch to manual commit (Repeatable read)
Read committed
Read uncommitted
Repeatable read
Serializable

Transaction Log

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



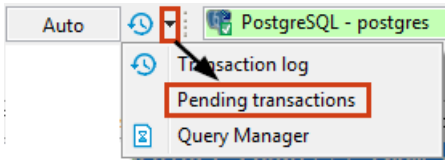
The Transaction Log window shows transactions that are:

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

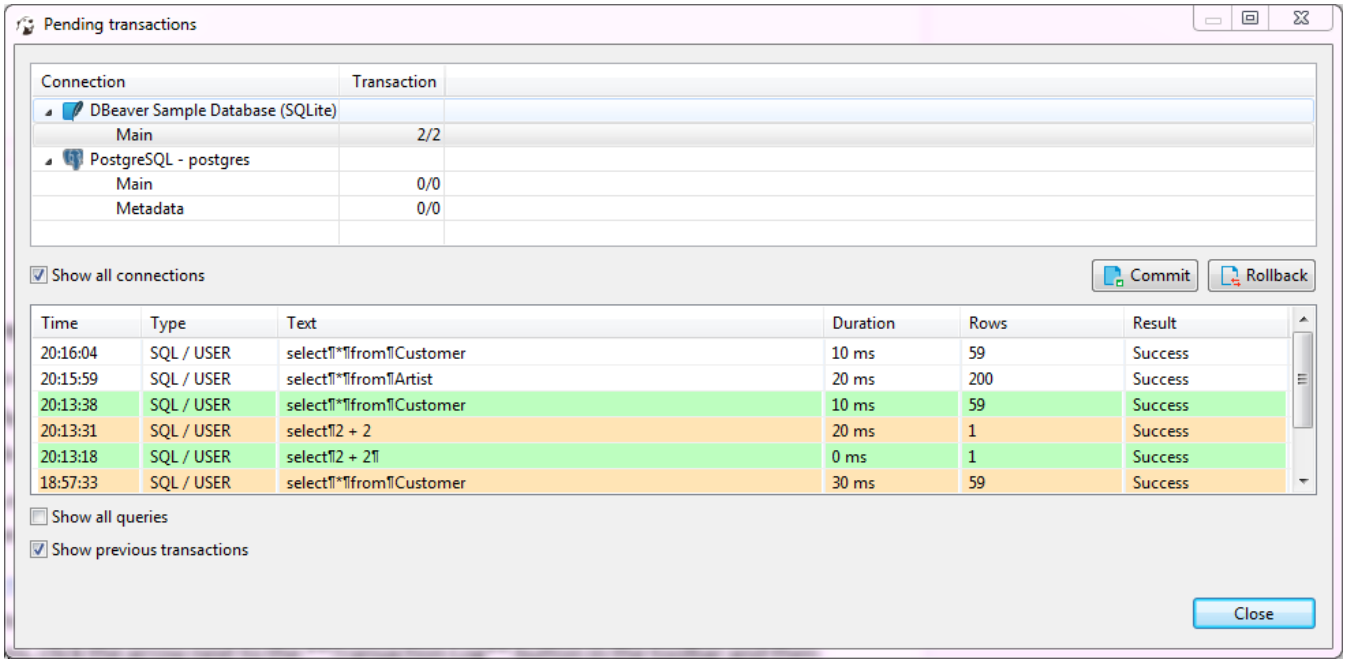
To see all previous transactions during the current session, select the **Show previous transactions** checkbox. To see all queries including non-transactional ones, select the **Show all queries** checkbox.

Pending transactions

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



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



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

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

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

Database drivers

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

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

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

Obtaining JDBC driver

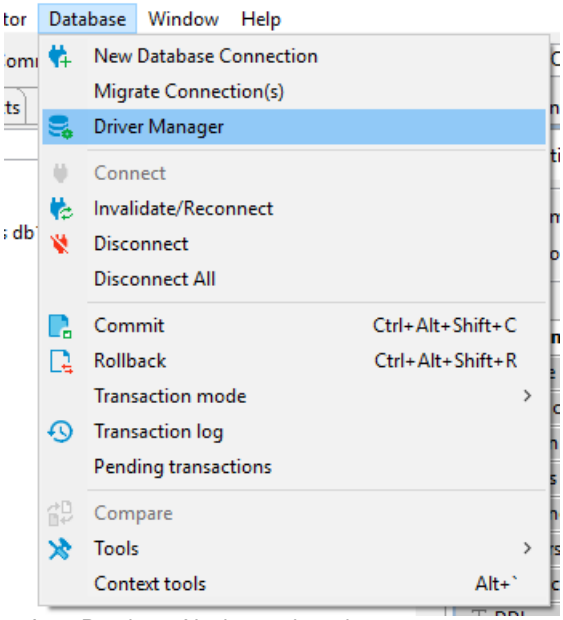
JDBC driver is a program (in Java) which can connect and operate with some local or remote database server. It usually provides all needed functionality to cover 100% of database functionality. The JDBC driver is usually provided by database vendors to allow customers to work with their databases.

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

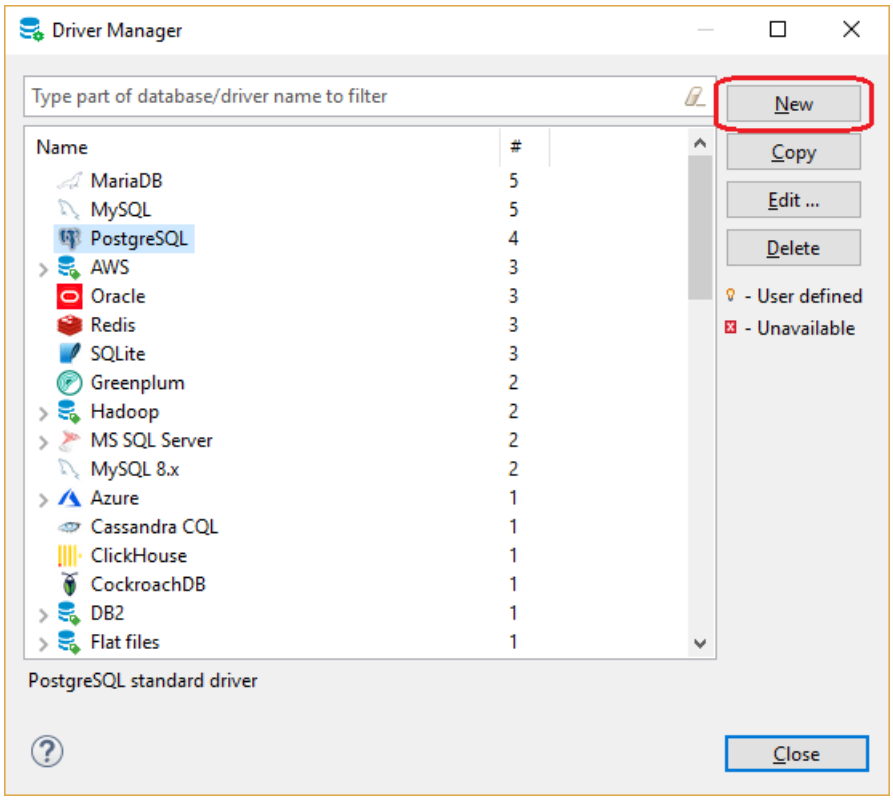
Adding driver configuration in DBeaver

Open driver manager dialog

You can open the driver manager from the main menu:



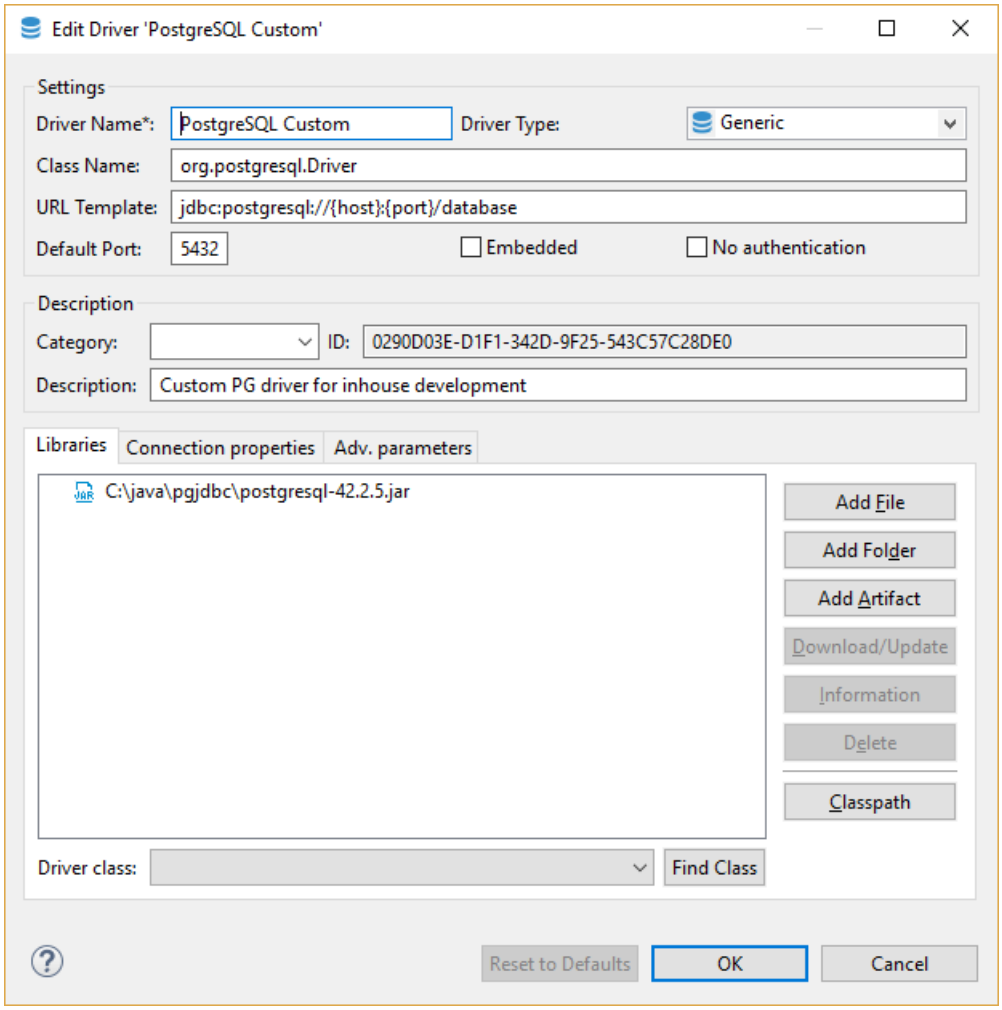
or from Database Navigator drop-down menu.



Add a new driver

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

Main parameters



Parameter	Description

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

Libraries

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

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

Maven artifacts

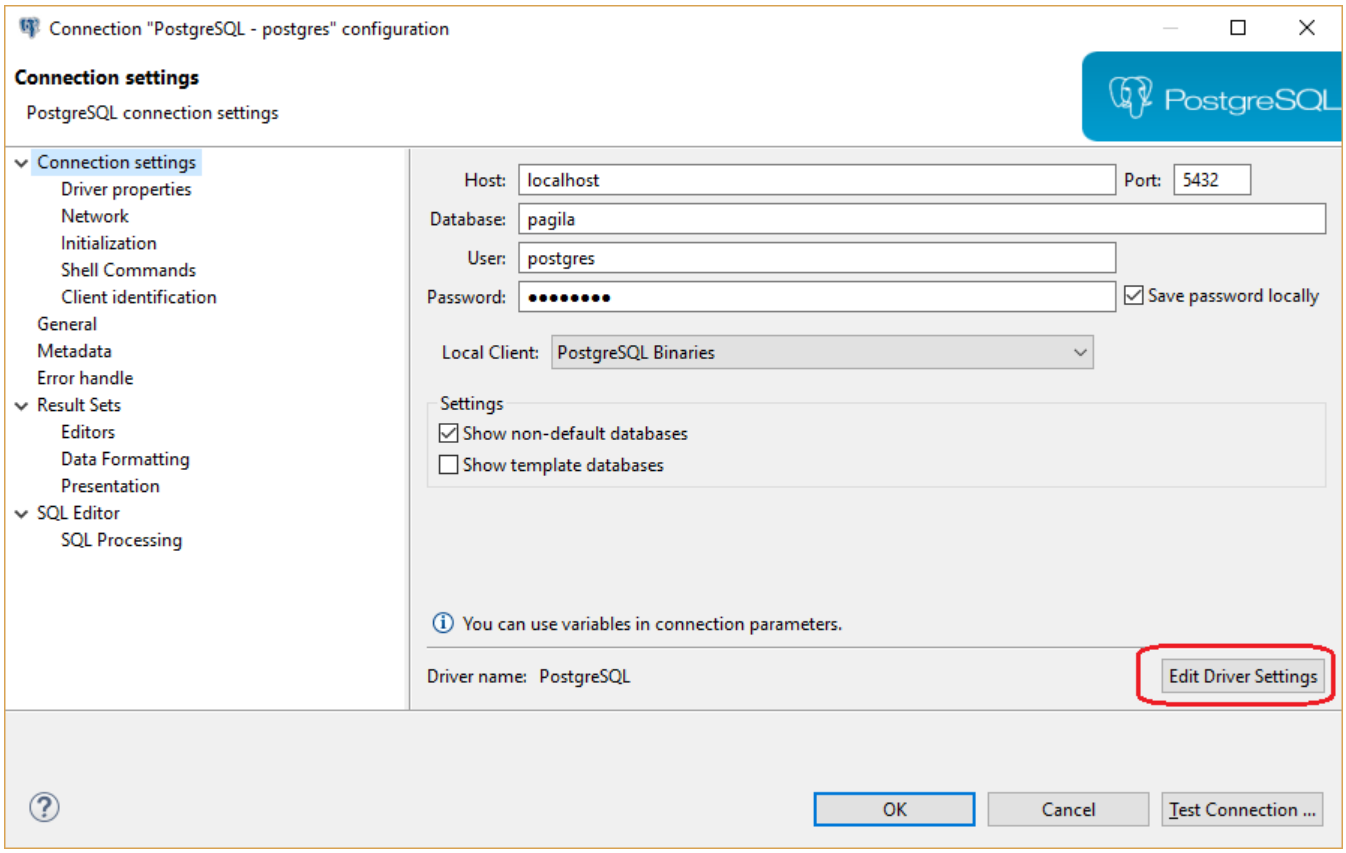
DBeaver can download driver jars directly from the Maven repository (it is a global public repository of Java libraries, usually an open-source). If your database driver is published on some public repository you can use this feature. Maven artifacts are better than plain jar files because you can see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

for additional information look How to add additional artifacts to the driver

Saving driver, adding connection

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

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



URL Templates

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

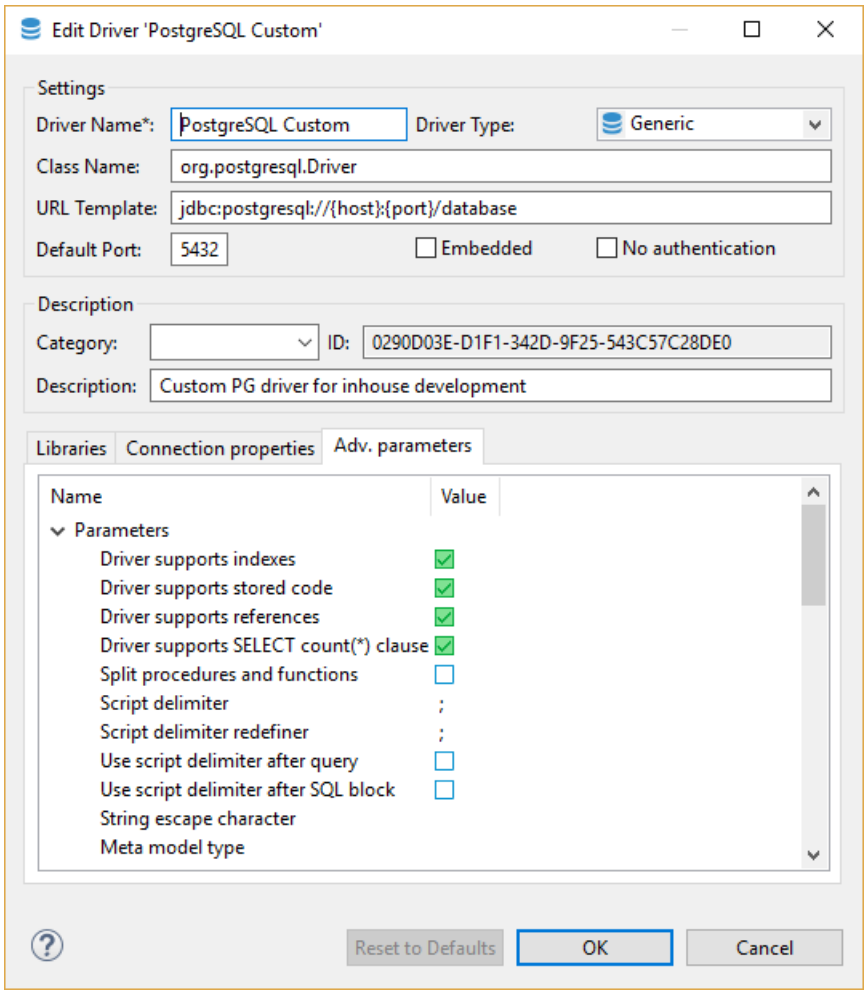
For example above the URL template is: jdbc:postgresql://{host}:{port}/{database}
Host, port and database are parameters which you will need to enter on the connection configuration page.

Supported URL variables:

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

Advanced settings

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



Main parameters

Parameter	Description
Driver supports indexes	Driver supports table indexes
Driver supports stored code	Whether this driver supports stored code (procedures, functions, packages, etc)
Driver supports references	Driver supports table references (foreign keys)
Driver supports SELECT count(*) clause	Driver supports SELECT count(*) clause
Driver supports views	Driver supports table views
Split procedures and functions	Show procedures and functions in different folders
Script delimiter	Literal for SQL queries separation in scripts
Script delimiter redefiner	SQL clause which redefines script delimiter value
Use script delimiter after query	Keep SQL script delimiter after each SQL query
Use script delimiter after SQL block	Keep SQL script delimiter after SQL script blocks (BEGIN/END)
String escape character	Character used to escape special symbols in strings
Meta model type	Type of metadata reading model - standard or indexed
All Objects Pattern	SQL pattern for all metadata objects
Omit catalog(s)	Do not read and use catalog (aka database) information
Omit single catalog	Hide catalog (database) if there is only one catalog on server
Omit schema(s)	Do not read and use schemas information

Parameter	Description
Omit single schema	Hide schema if there is only one schema on the server
Use schema filters	Use JDBC schema filters when the database does not support catalogs. Otherwise just read all database schemas and filter on client-side
Omit type cache	Do not use data types provided by driver
Shutdown parameter	Database shutdown URL parameter
Create database parameter	Database create URL parameter
Driver supports multiple results	Driver supports multiple results for a single query
Driver supports result set limit	Driver supports multiple result set limit (max rows)
Driver supports structure cache	Driver supports structure cache reading. Enables schema columns, keys, etc
Driver supports TRUNCATE operation	Driver supports TRUNCATE command. It is much faster than DELETE without criteria

Queries (Custom driver queries)

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

DDL (DDL generation options)

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

Formatting (SQL values formats)

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

Summary

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

ticket (in any suitable form).

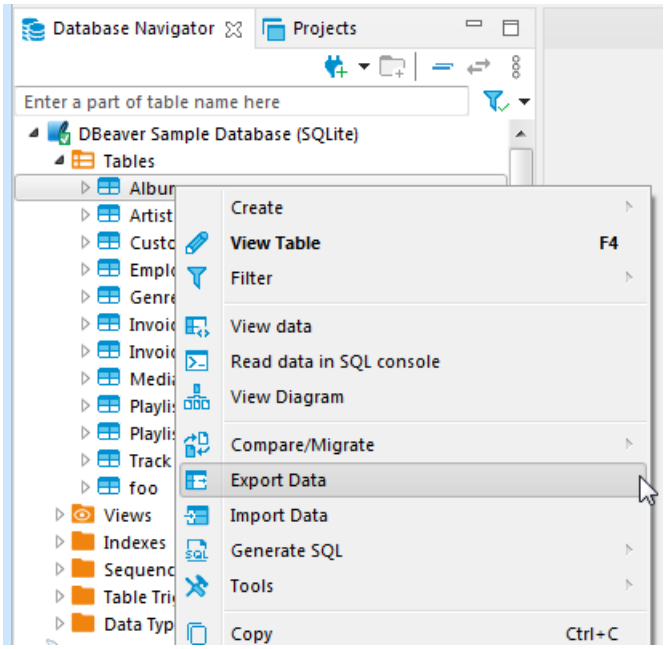
Thank you.

Data export/import

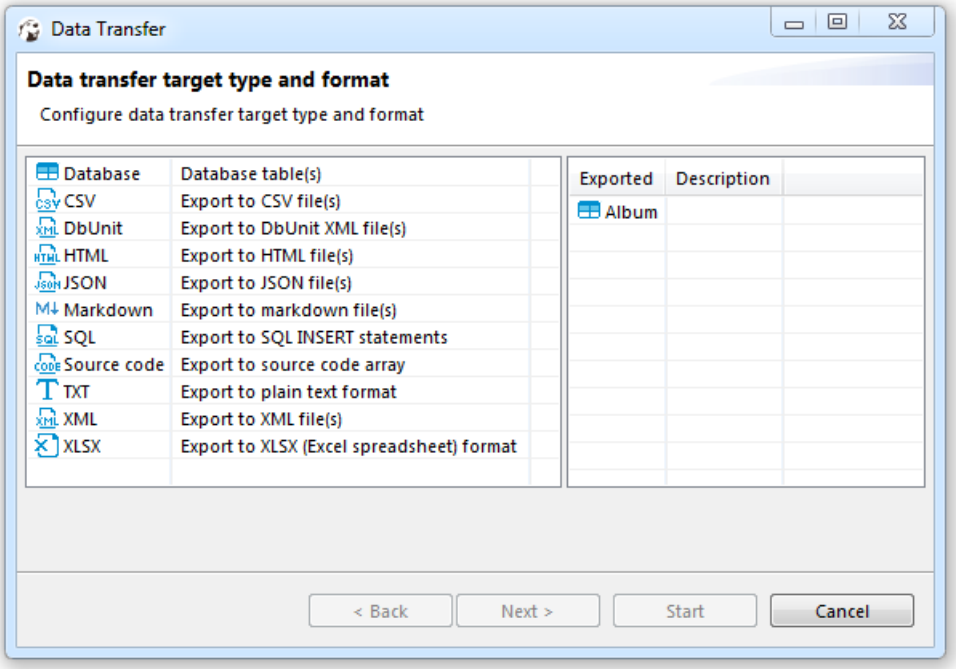
You can perform data export/import or migration for database table(s). We will describe most typically used cases.

Exporting table data to CSV format

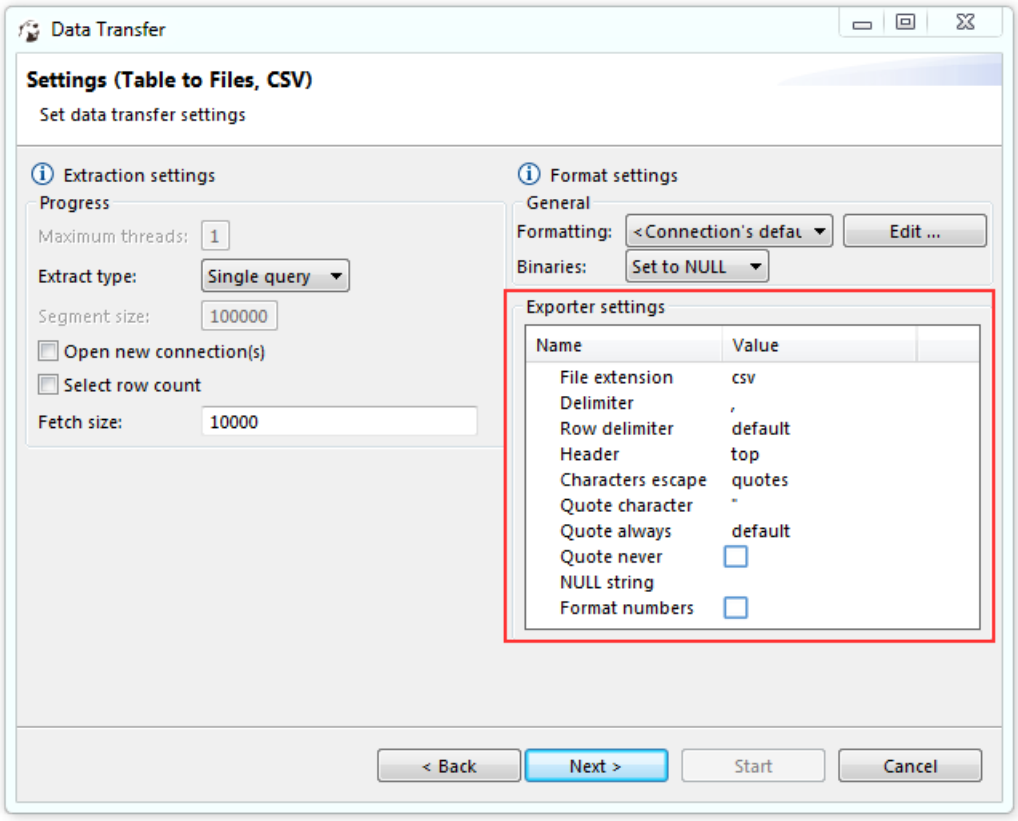
1. Select a table or tables you want to export. In the context menu choose "Export Data".
   (Note: you can also export data from custom SQL queries results. To do that, choose "Export results" in the results context menu).



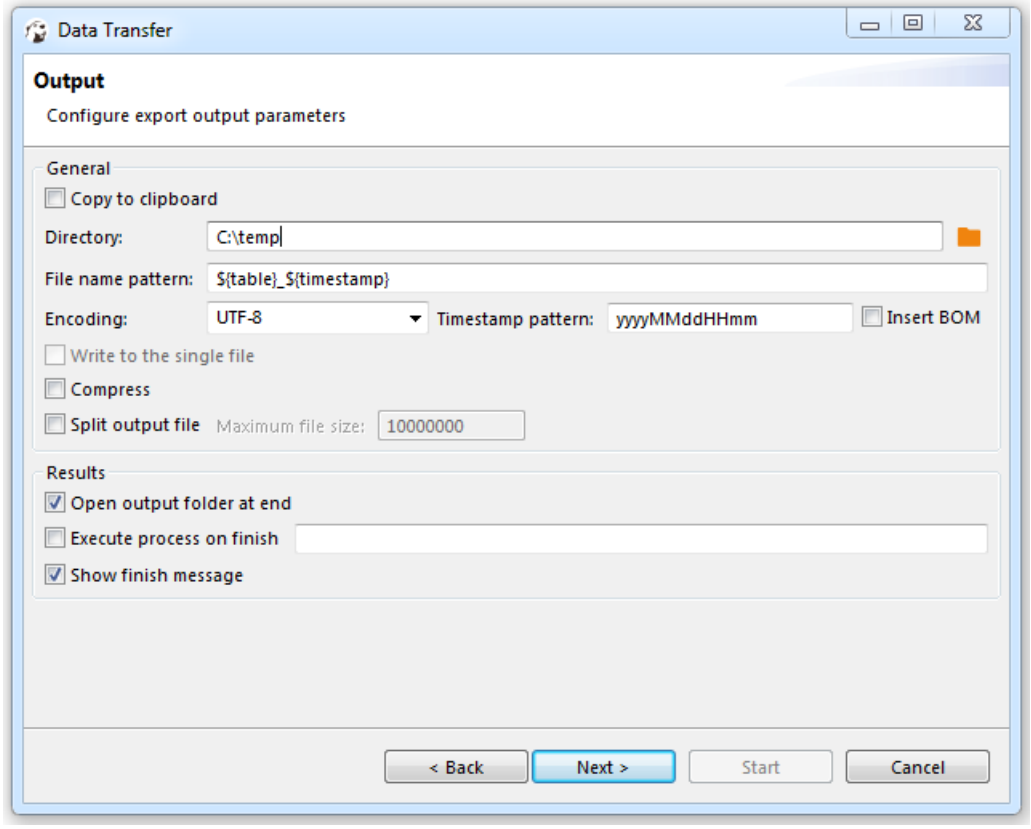
2. Choose export format. DBeaver supports many different output formats including CSV, HTML, XLSX, etc:



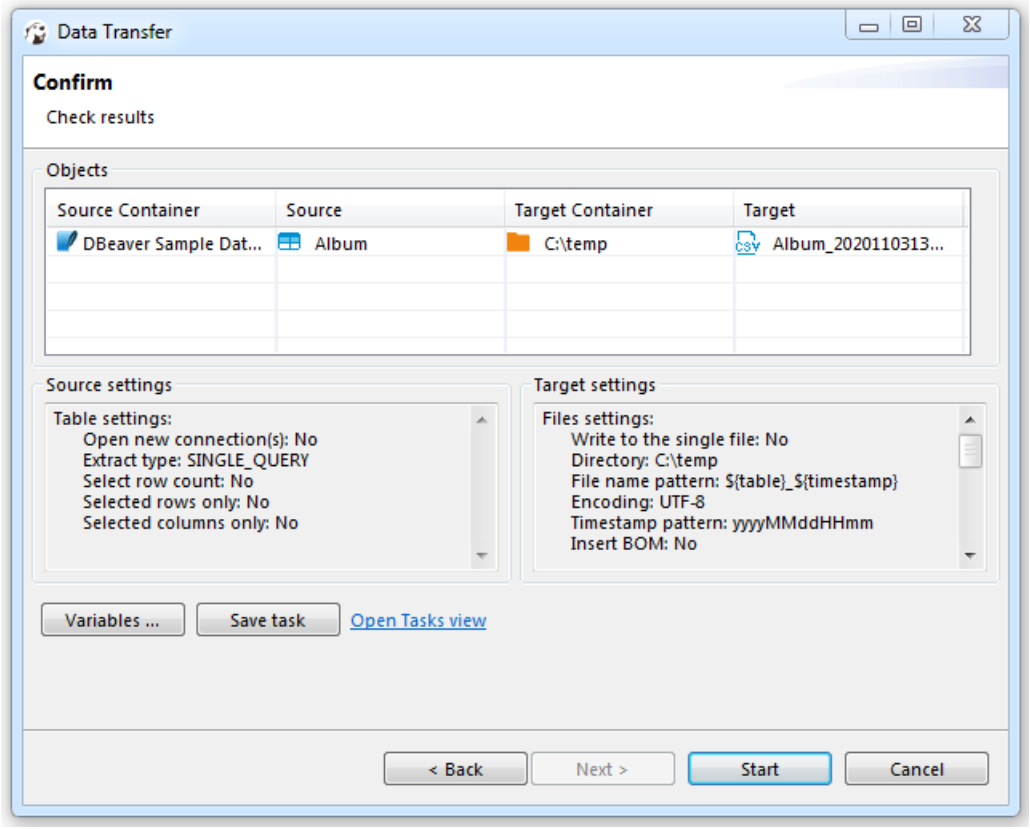
3. Set data extraction options (how the data will be read from the tables). This may affect the extraction's performance. And set export format option. They are specific to the data format you chose on step 2:

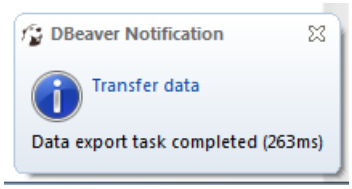


4. Set options for output files or clipboard. Note: Timestamp pattern is used here to target the file name pattern:



5. Review what you want to format and into which format you will export it. You can also save all your settings as a task in this step or change the task variables:

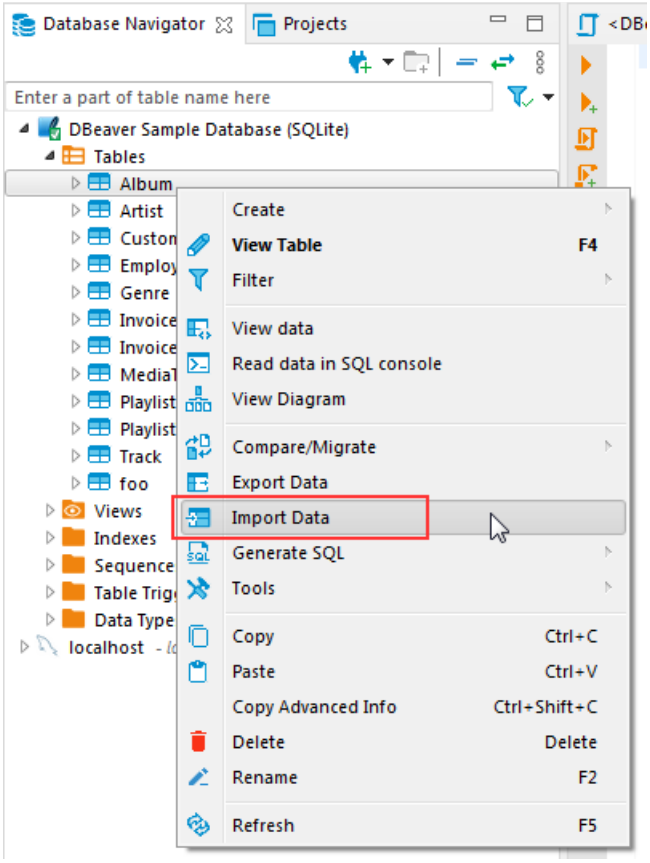


6. Press finish. See extraction progress. You can keep working with your database during the export process as the extraction will be performed in the background. Note: avoid changing data in tables you have selected to be exported while the exporting is in progress. In the end you will see status message: 

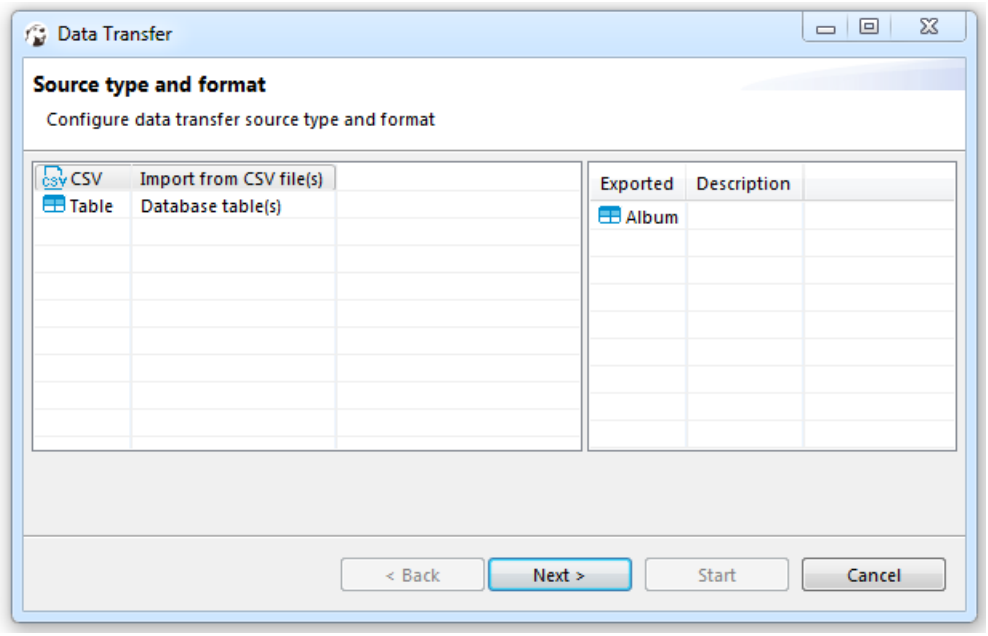
Importing data from CSV format

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

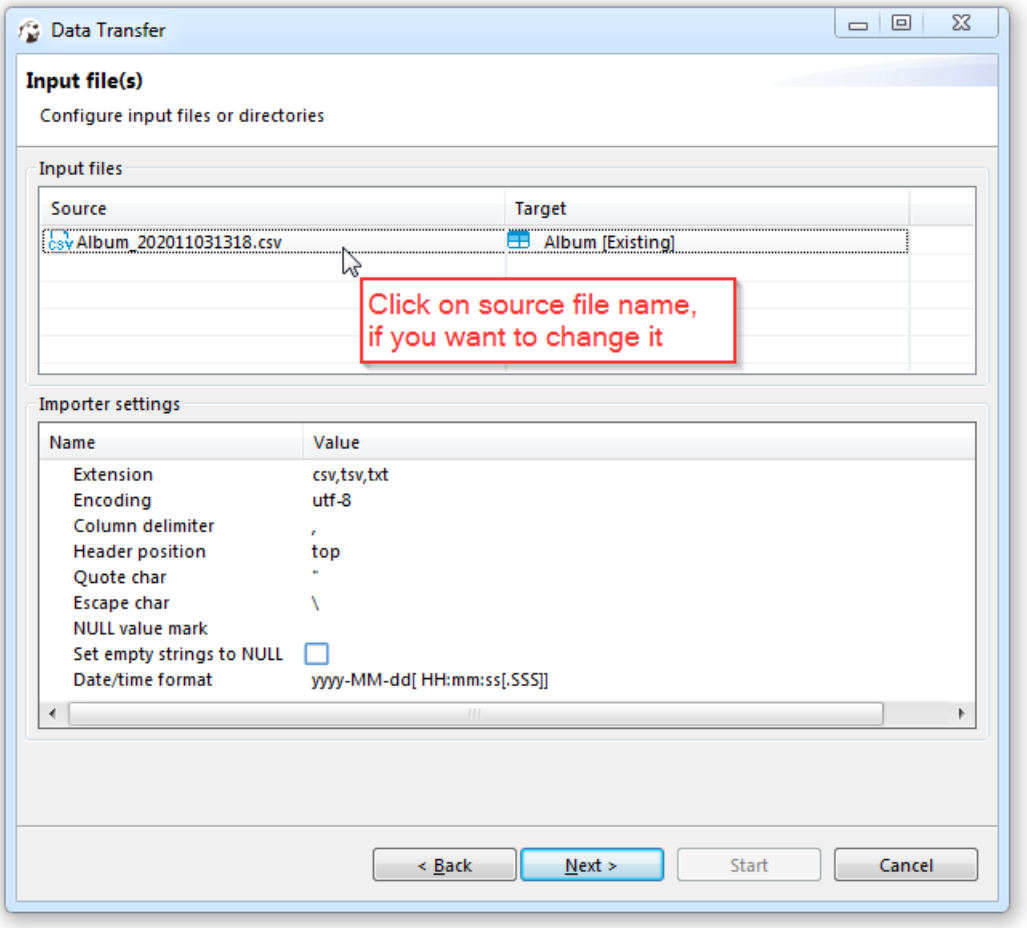
1. Select a table(s) to which you want to import data. In the context menu choose "Import Data":



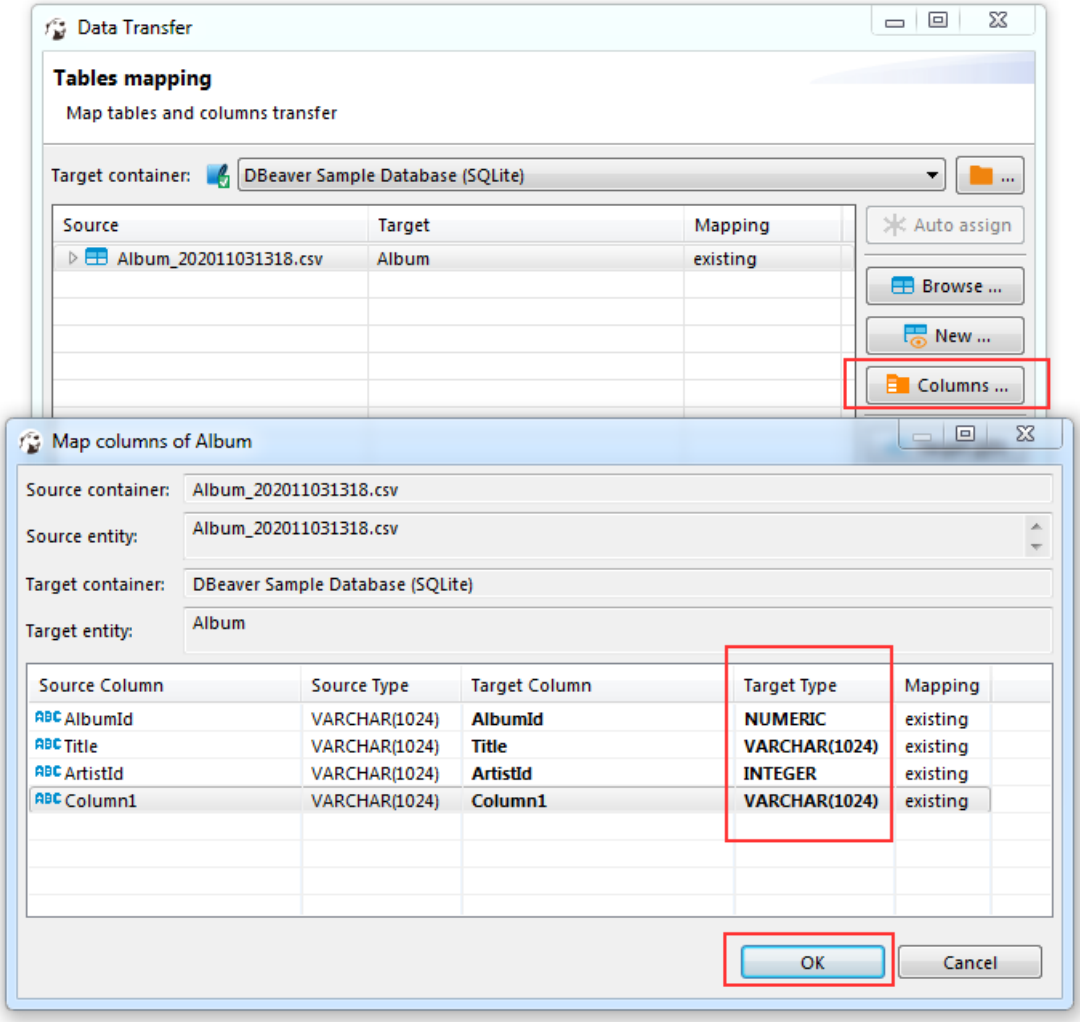
2. Choose import format (CSV):



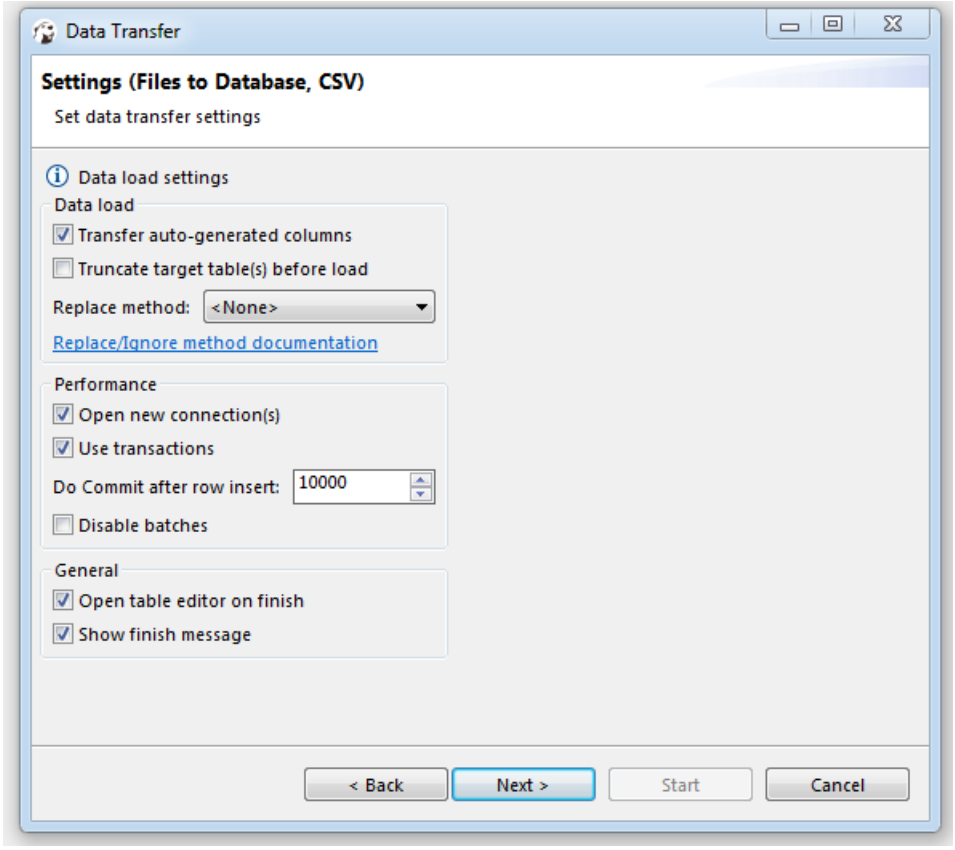
3. Select input CSV file for each table you want to import and you can change the Importer settings (format specific) at this step:



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

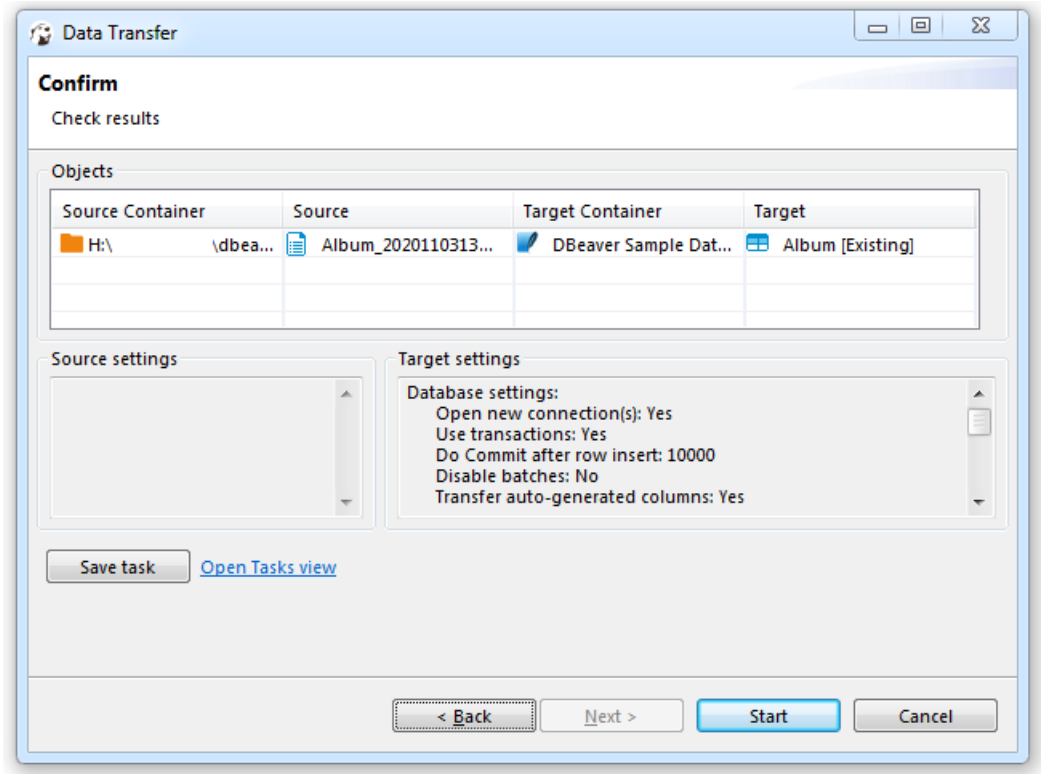


5. Set options for loading data in the database. These options may affect the loading's performance:

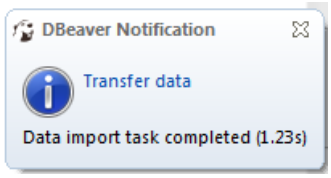


About the replacing method option, you can read here.

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



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



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

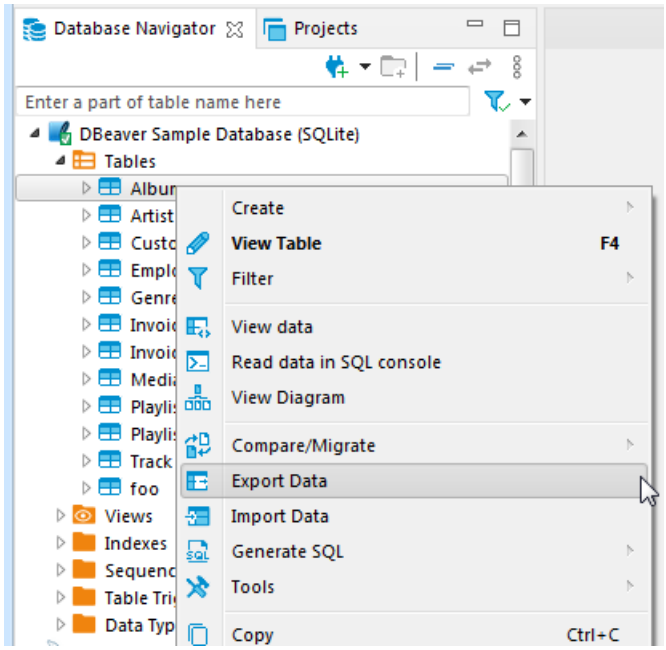
Data migration

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

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

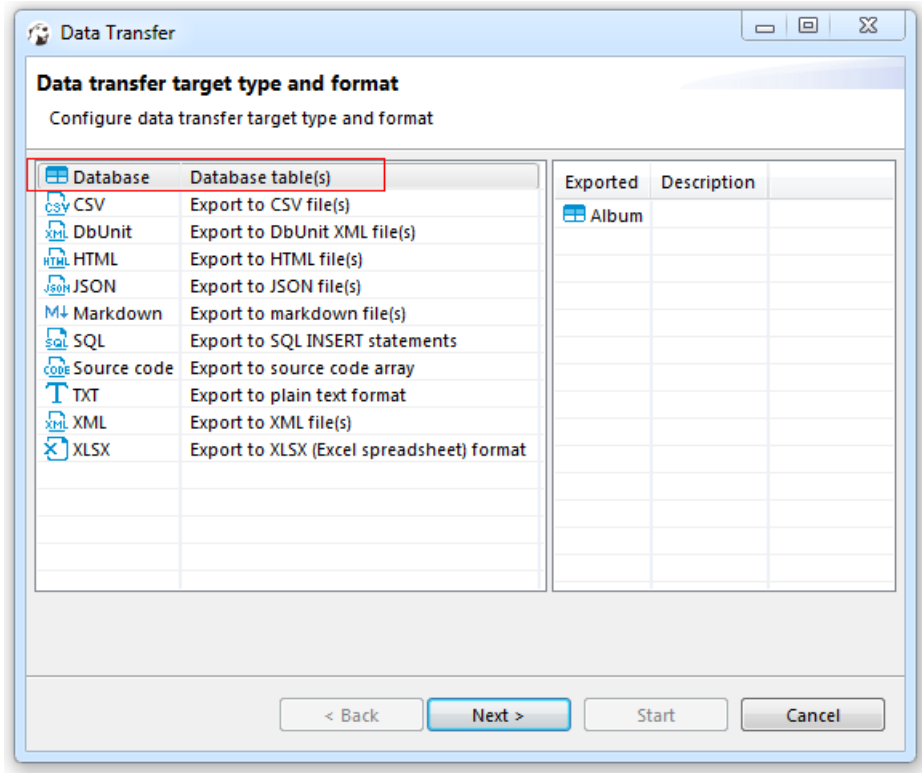
Step 1: Define the data source

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



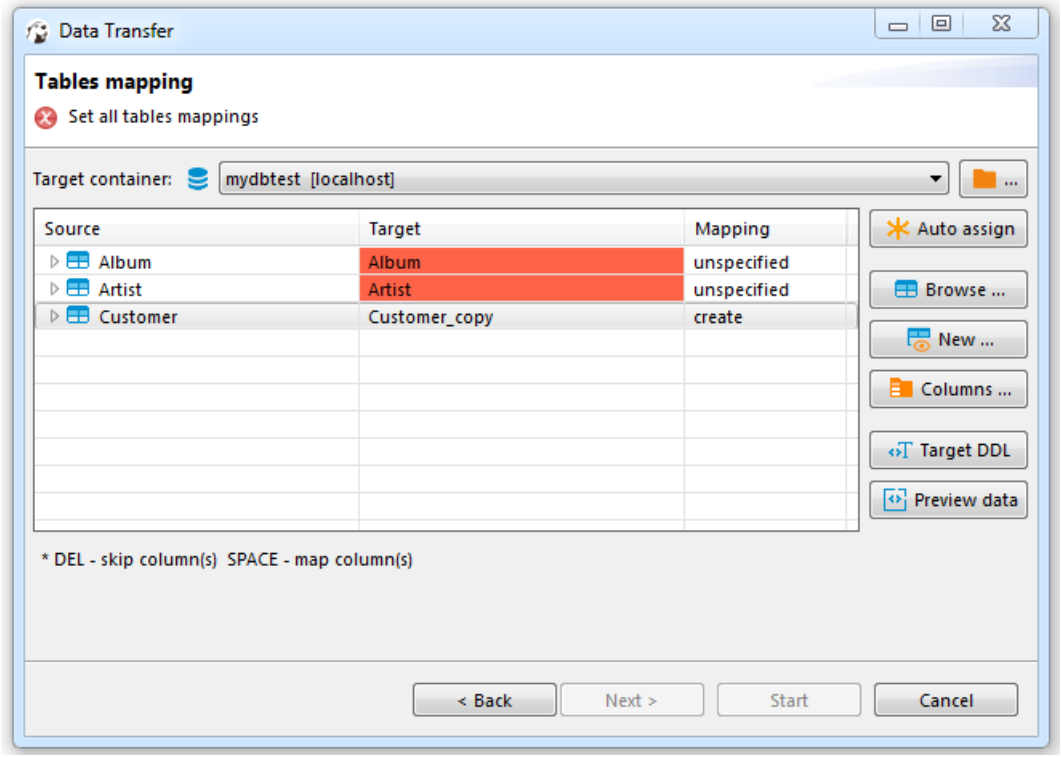
Step 2: Define data transfer target type

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

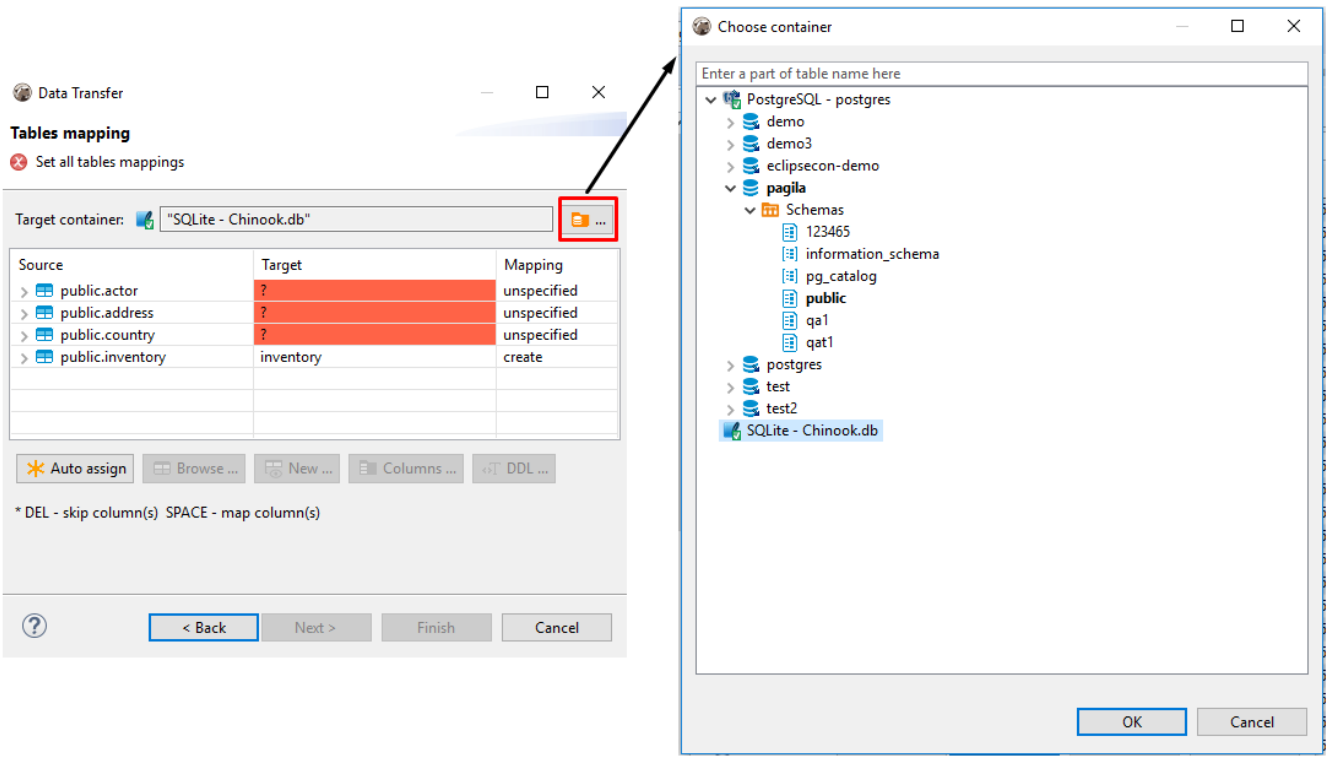


Step 3: Define data mapping

For proper table mapping, the following options are available:



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

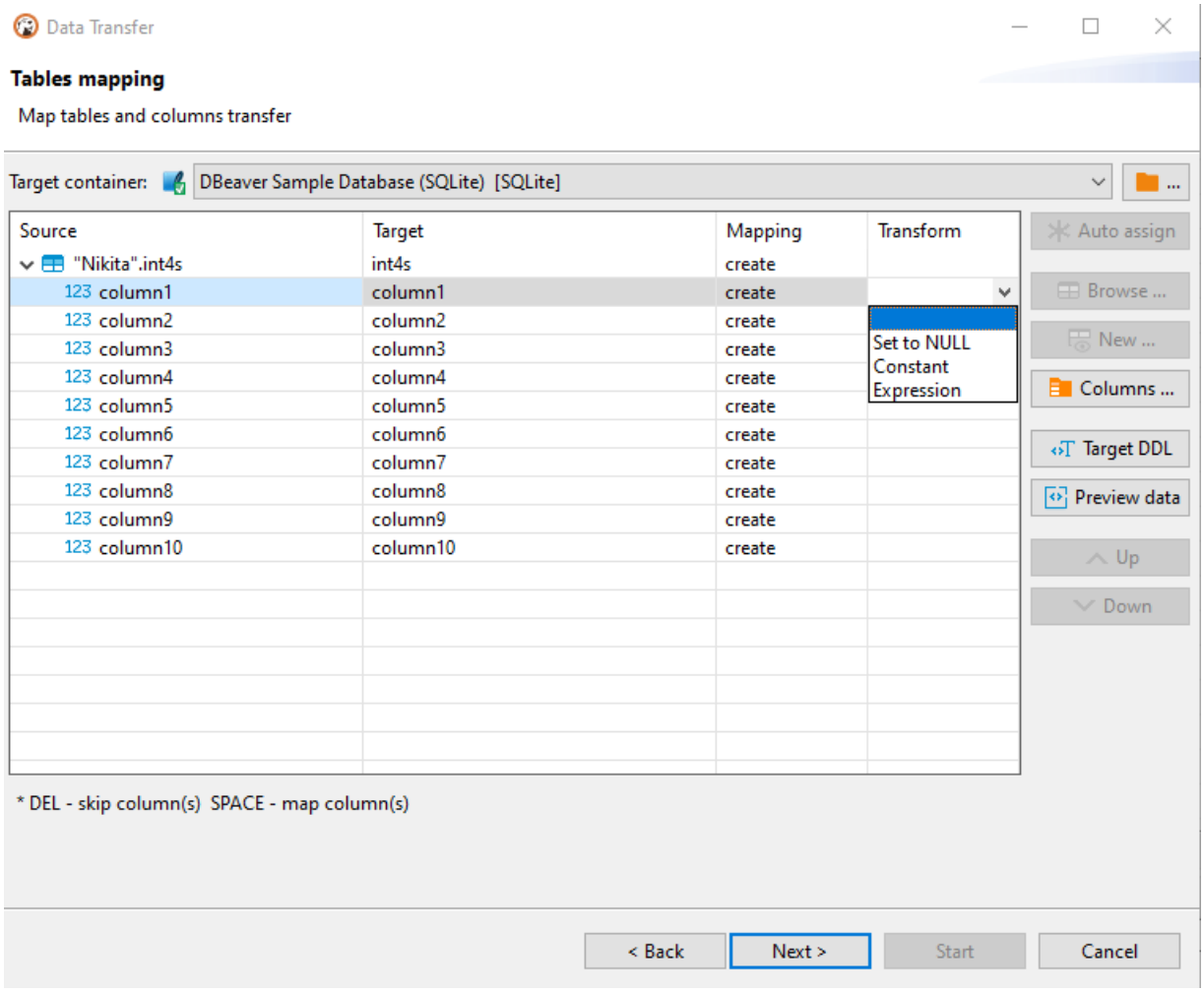


- **Source** - contains names of all the tables selected at step 1. You can also see the list of columns existing in the source table by pressing ›
- **Target** - contains names of the tables where the data will be transferred to.
- **Mapping** - contains the list of actions to be applied to the source 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.

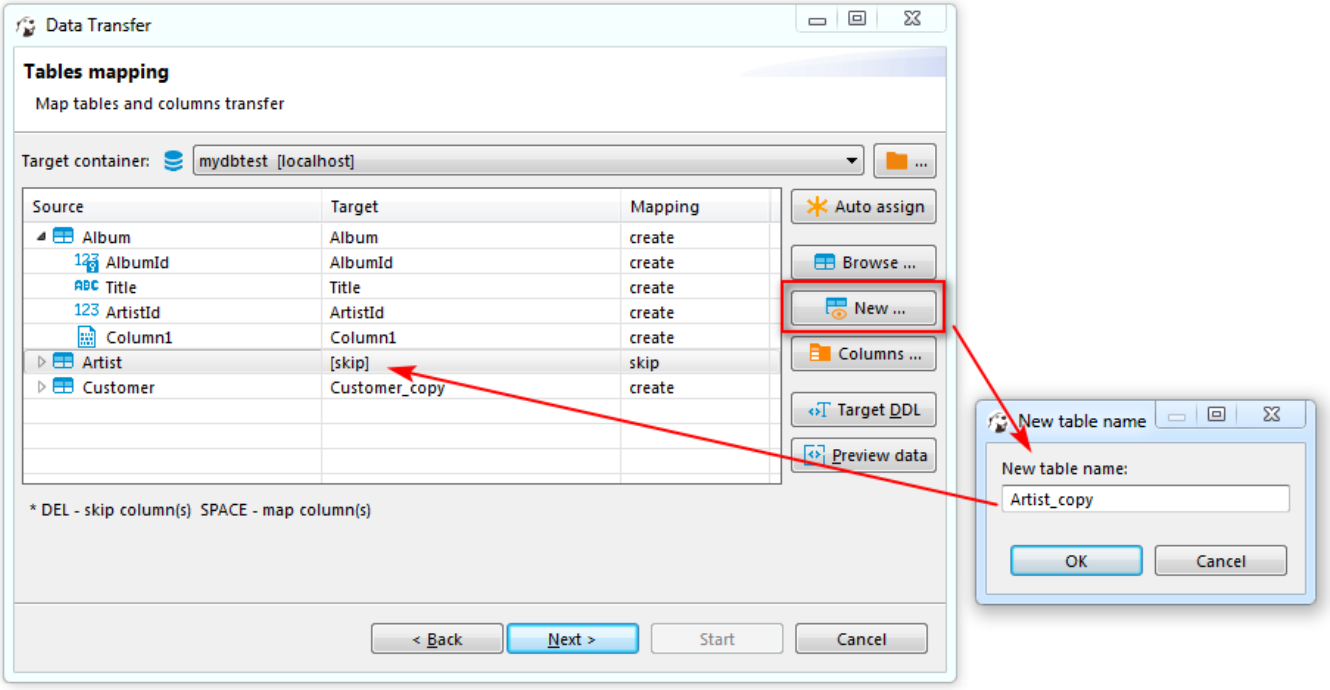
If the cells are marked with ?, it means that in the target table there are no columns with matching names, otherwise the names will be filled in automatically.

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

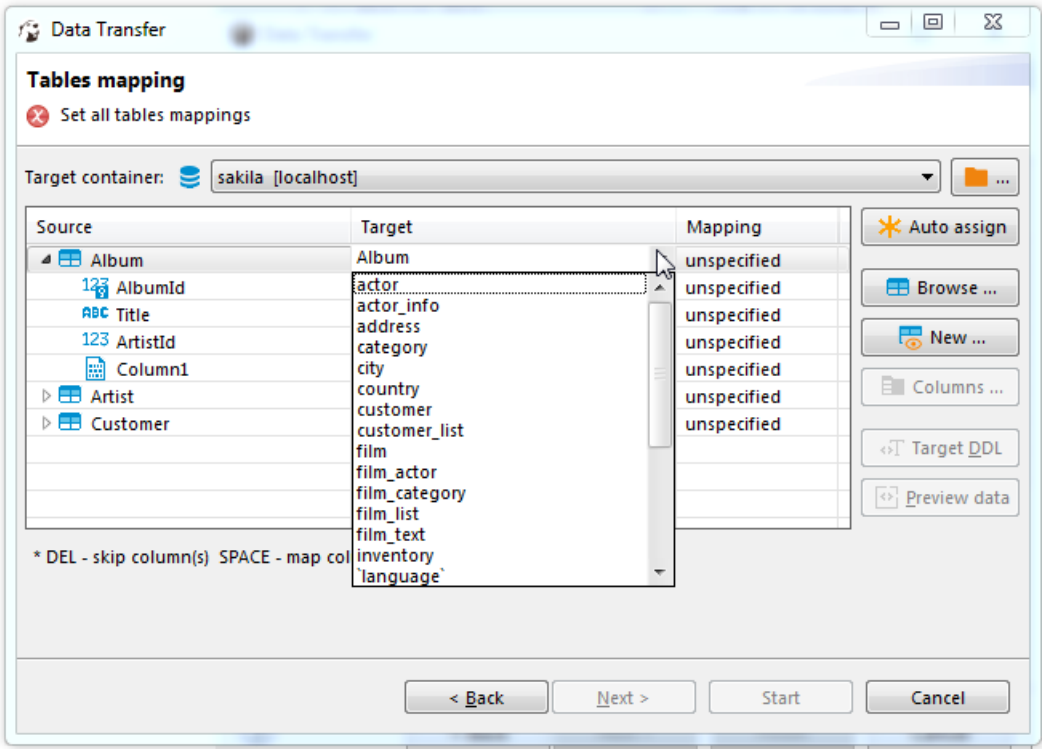


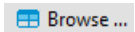
- *Set to NULL*. All values in the corresponding column will be set to null.
- *Constant*. All values in the corresponding column will be set to the specified constant.
- *Expression*. This transformer uses expressions (namely, JEXL expressions) to calculate the cell's value. You can use basic arithmetic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

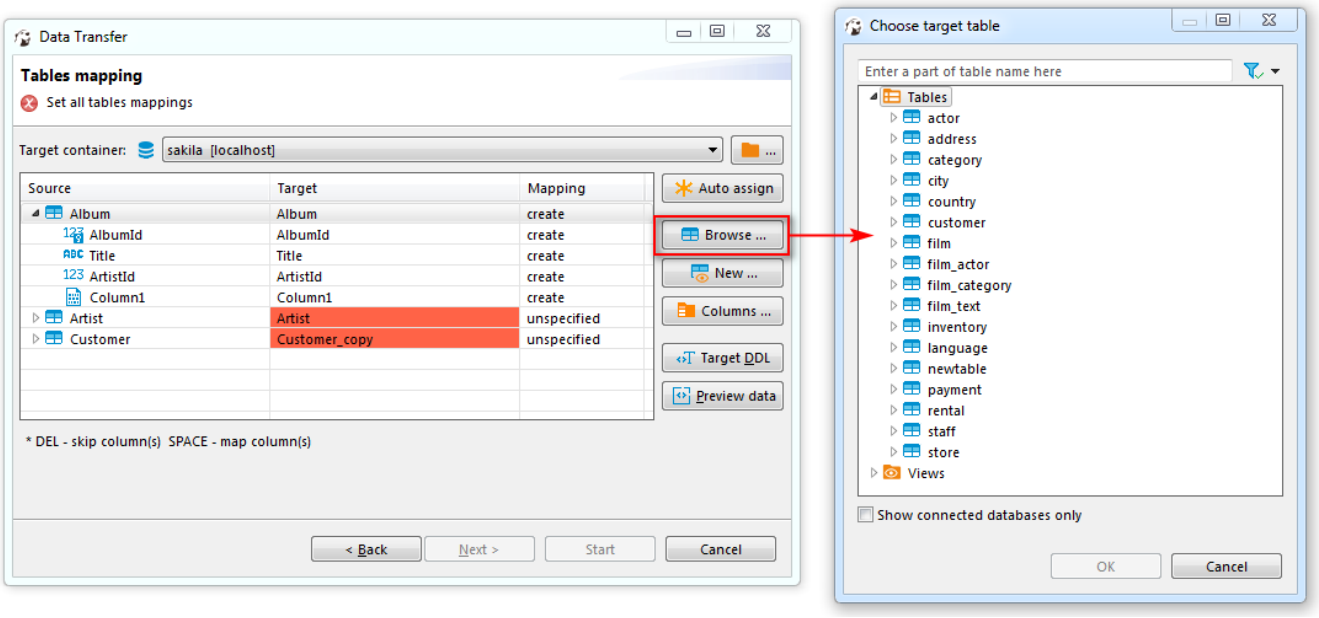
You can define a target table by clicking on a cell in the **Target** column and entering its name, or press the **New** button New ... and enter a new name in the opened dialog box.



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



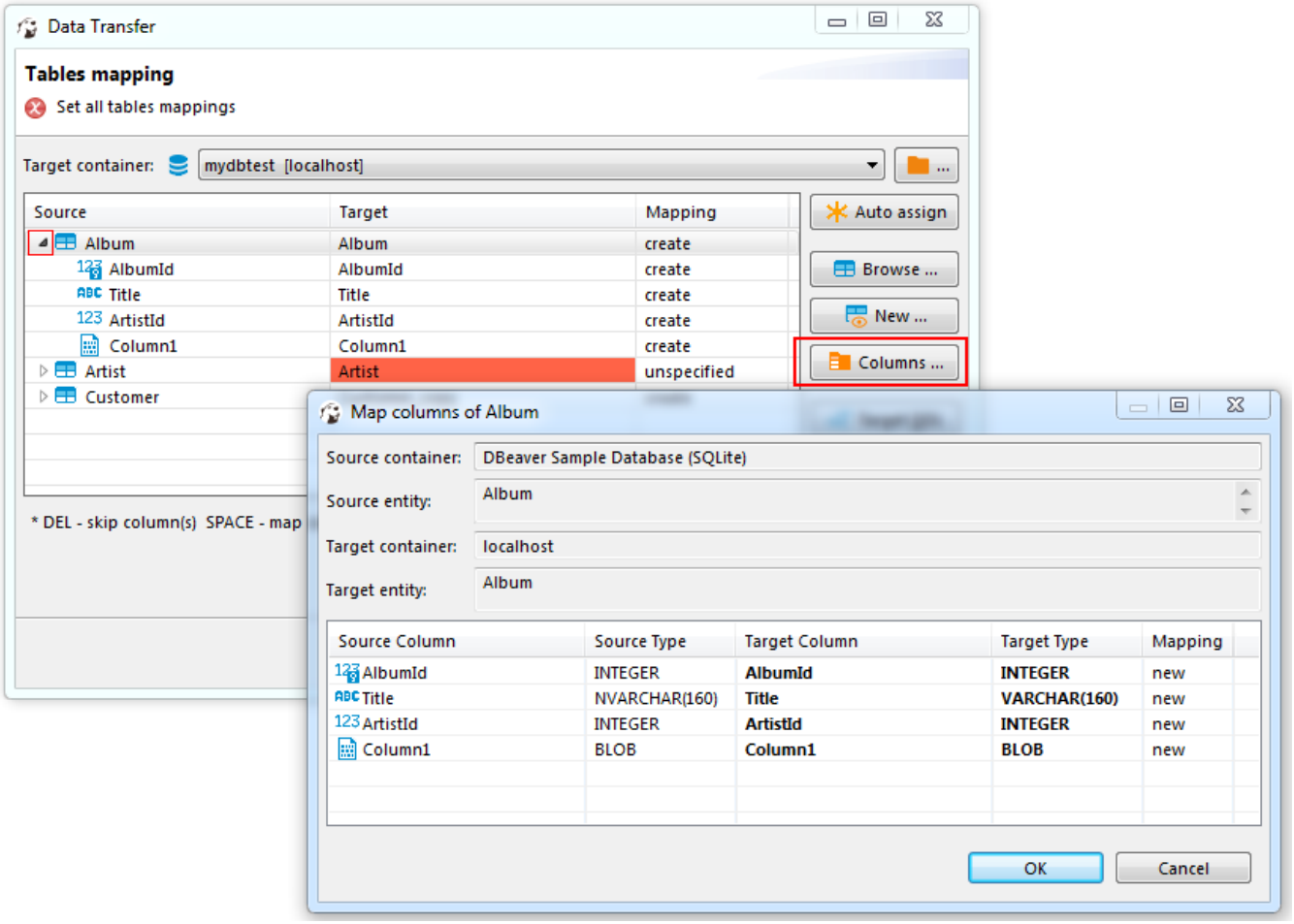
Or select a table from the existing tables in the target container by pressing the **Browse** button 



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

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

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



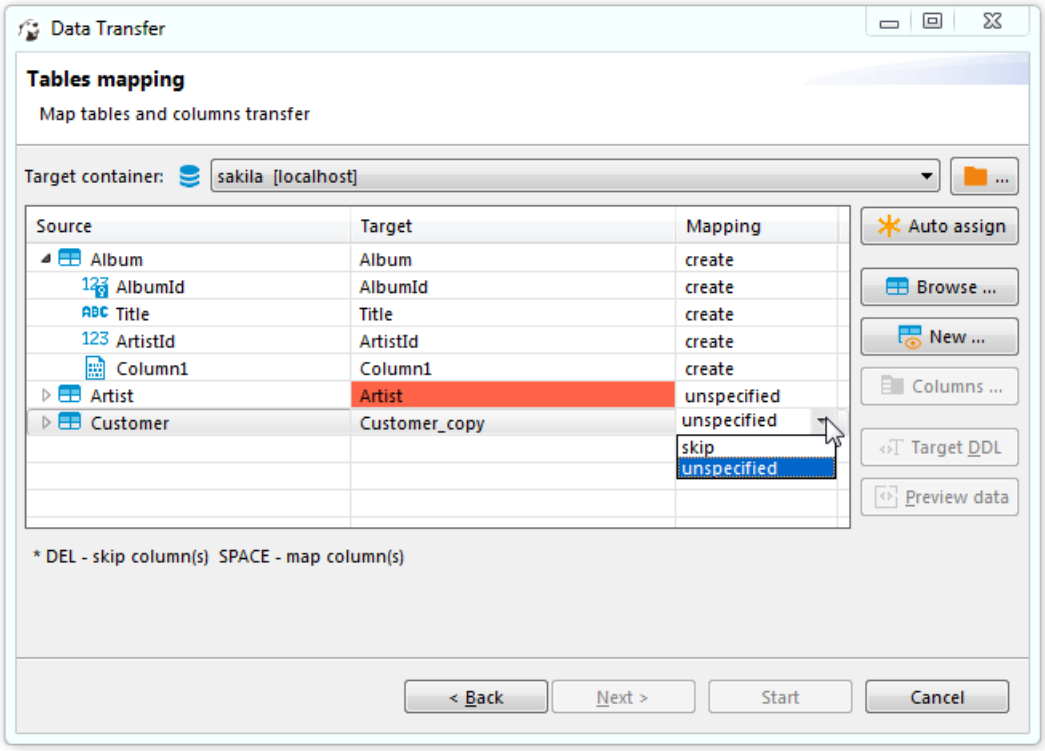
The following elements are available here:

- **Source column** -this column contains names of columns existing in the selected source table;
- **Source type** - this column contains the list of data types assigned to the columns in the selected source table;

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

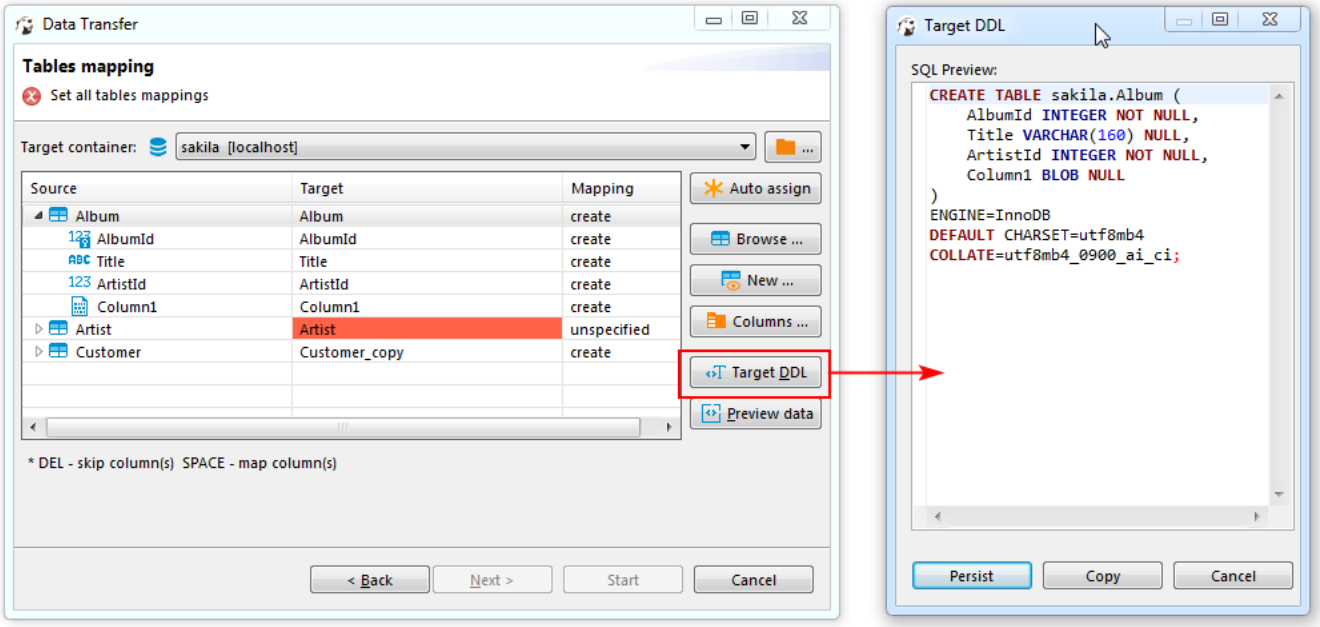
  IMPORTANT: Sometimes data types that are supported on the source database are not supported on the target or vice versa. To set a data type for a column in a target table, click the cell in the **Target Type** column and select one from the dropdown list of data types supported on the target.
- **Mapping**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

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

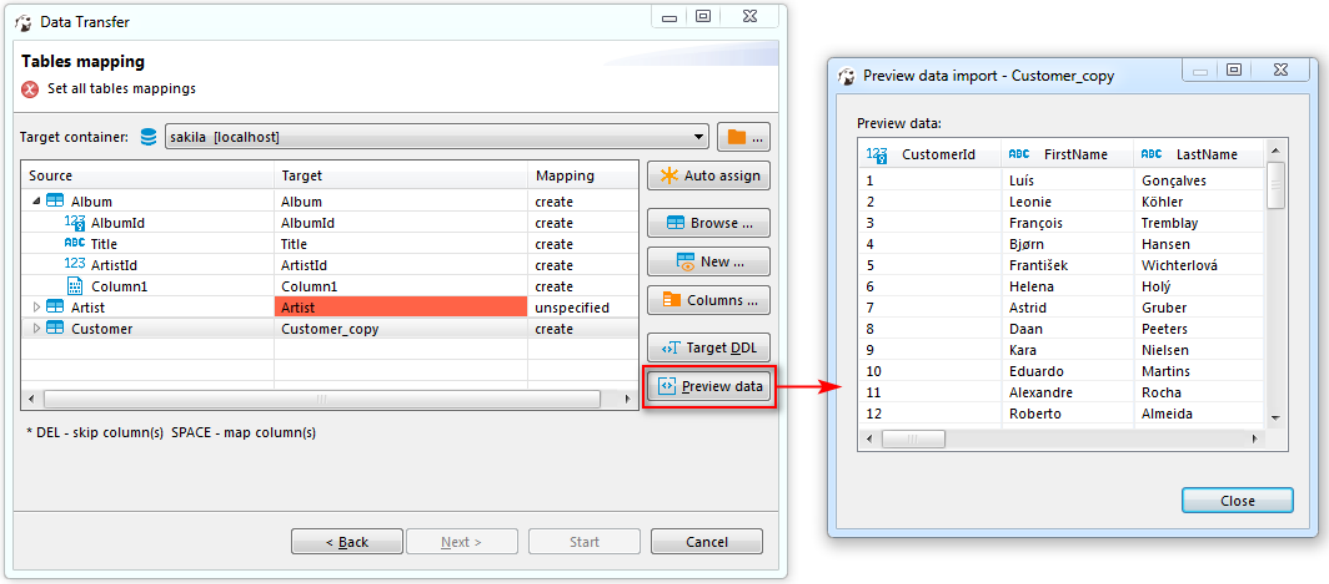


You can also view the SQL script that will be run on data transfer by pressing the **Target DDL** button 

.



If you want to see a preview of the imported data, you can select the **Preview data** button



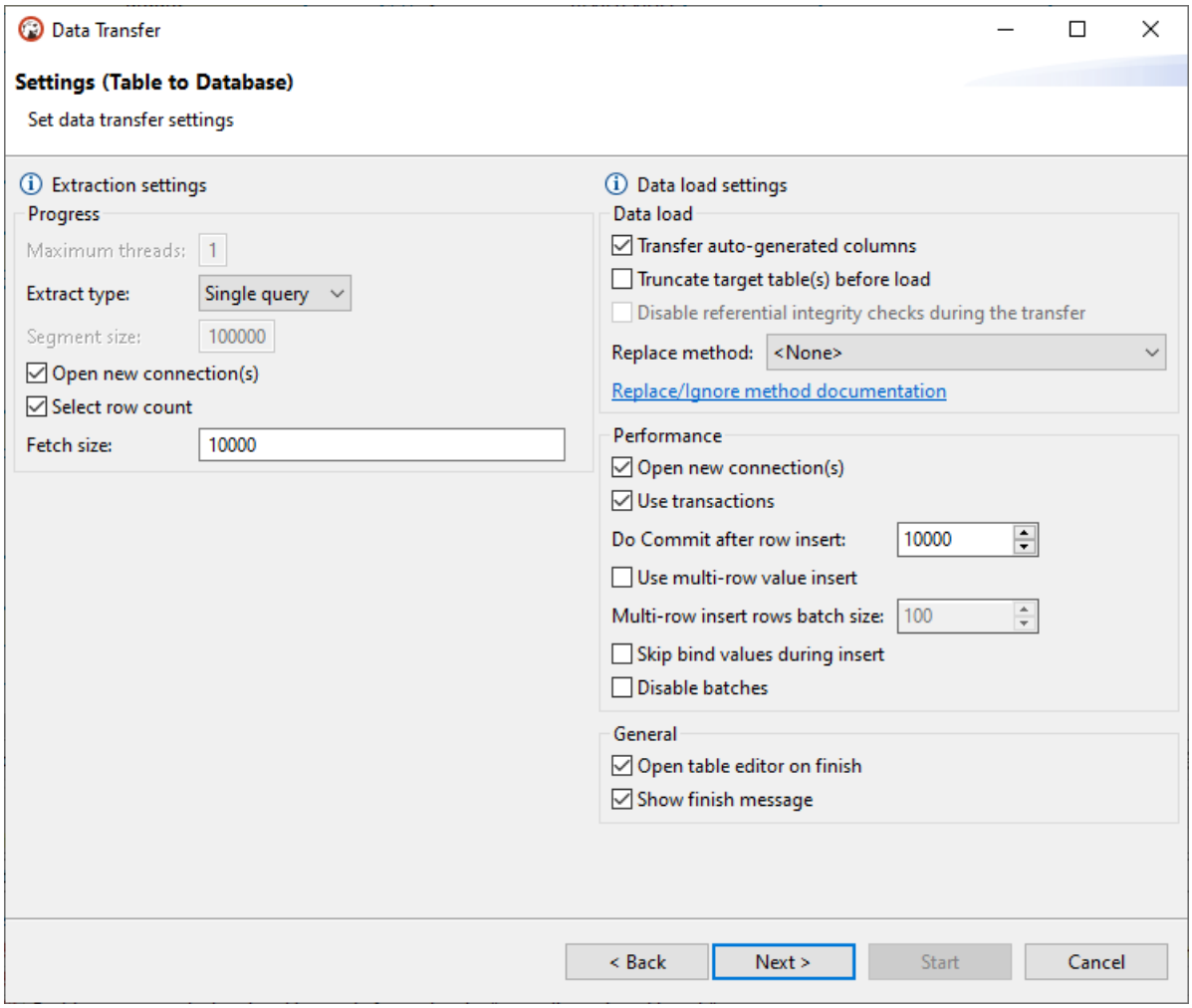
The following keyboard shortcuts for easy navigation within the mapping table area of **Table mapping** screen are supported:

Shortcut	Action
Up	Moves one row up.
Down	Moves one row down.
Right	Releases the list of source table columns.
Left	Swaps the list of source table columns.
Space	Auto-assigns the target.
Del	Sets mapping type to skip.

Configure data mapping and press **Next**.

Step 4: Define export settings

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



Extraction Settings

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

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

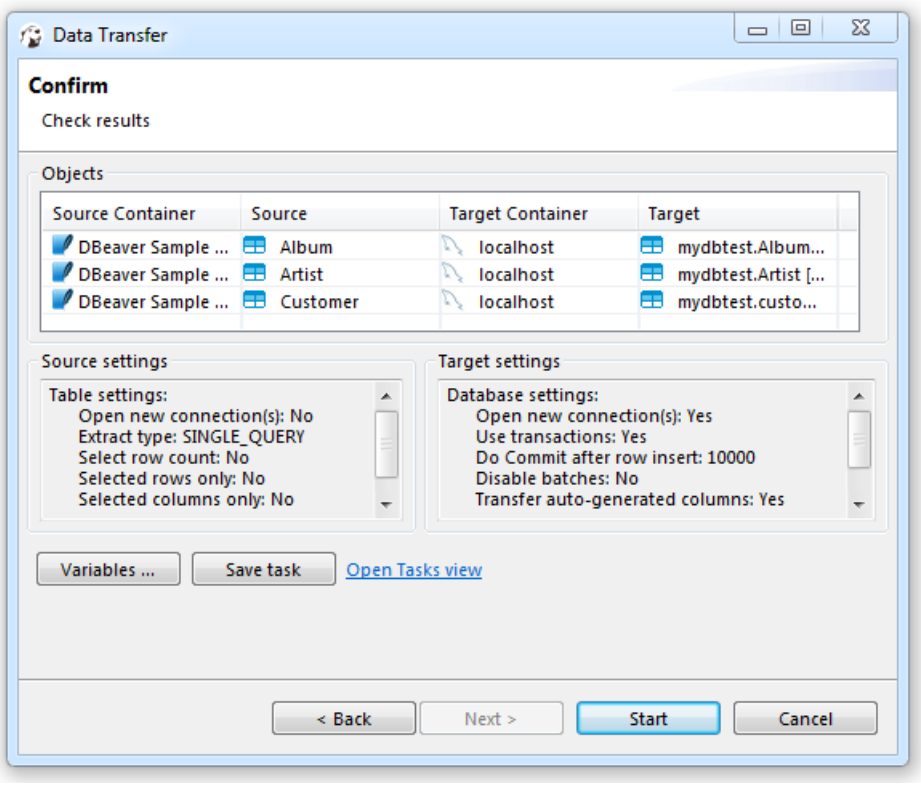
Data load settings

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

Option	Description
Truncate data load table before load	Select this check-box only if you want all the data be cleared from the target table. Be very careful with this option!
Replace method	Select this drop-down list if you want to change the import behavior in case of a duplicate primary key value. Data Import and Replace page
Open new connections	Use this option to speed up data transfer. If selected, a new connection will be opened and the data transfer will not interfere with other calls to the database where data is being transferred to.

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

Step 5: Confirm



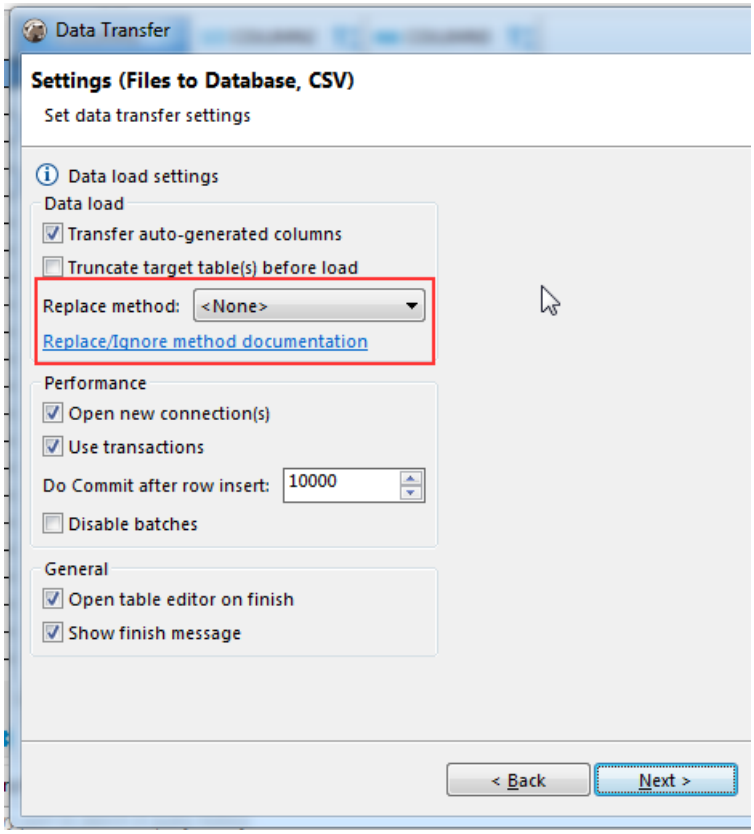
Check out the data transfer settings and press **Start** or save as task.

Data Import and Replace

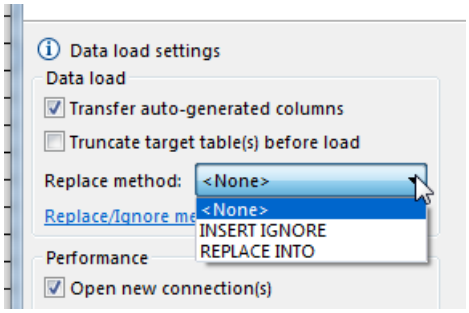
Data Import and Replace.

Sometimes there are situations when you want to ignore the current primary key value when importing into a table. Some databases have syntax constructs in addition to the INSERT INTO that may help.

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



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



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

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

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

```
CREATE TABLE language_insert (
language_id tinyint unsigned NOT NULL,
name char(20) NOT NULL,
last_update timestamp NOT NULL,
PRIMARY KEY (language_id)
);
```

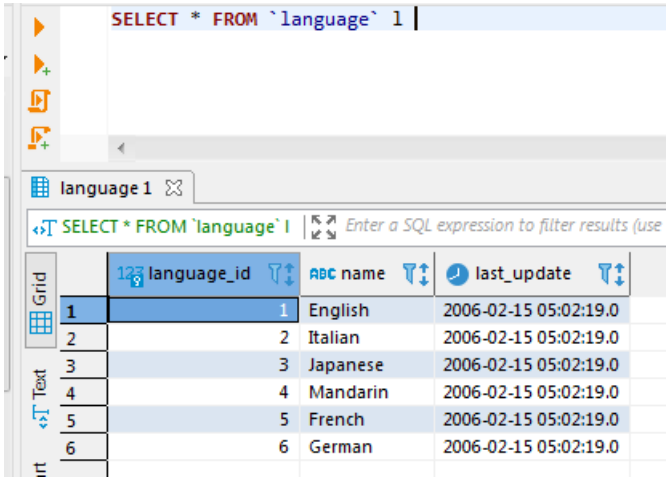
```
INSERT INTO sakila.language_insert (language_id,name,last_update) VALUES
(1,'English','2006-02-15 05:02:19.0'),
(2,'Italian','2006-02-15 05:02:19.0'),
(3,'Japanese','2006-02-15 05:02:19.0');
```

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

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

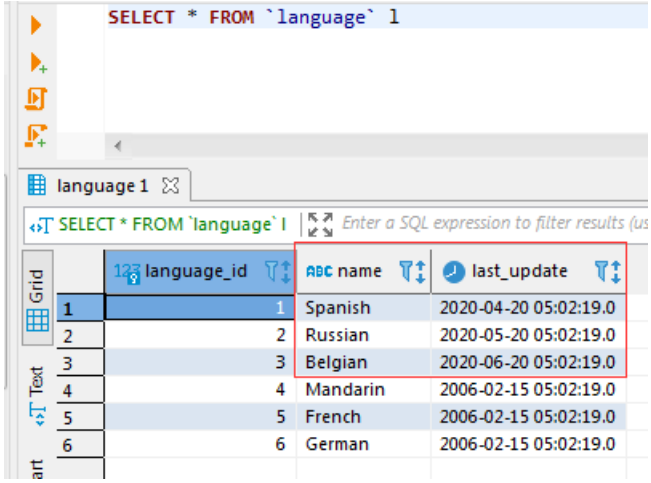
```
"language_id","name","last_update"
1,Spanish,"2020-04-20 05:02:19.0"
2,Russian,"2020-05-20 05:02:19.0"
3,Belgian,"2020-06-20 05:02:19.0"
4,Mandarin,"2006-02-15 05:02:19.0"
5,French,"2006-02-15 05:02:19.0"
6,German,"2006-02-15 05:02:19.0"
```

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



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

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



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

Which databases support replace/insert methods?

MySQL

INSERT IGNORE and REPLACE INTO

Insert examples:

"INSERT IGNORE"

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

"REPLACE INTO"

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

SQLite

Documentation
"INSERT OR IGNORE" and "INSERT OR REPLACE"

Insert examples:

"INSERT OR IGNORE"

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

"INSERT OR REPLACE"

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

PostgreSQL

Available for PostgreSQL version 9.5.
"ON CONFLICT DO NOTHING" and "ON CONFLICT DO UPDATE SET"

Insert examples:

"ON CONFLICT DO NOTHING"

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

"ON CONFLICT DO UPDATE SET"

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

FireBird

Available for FireBird version 2.1.
"UPDATE OR INSERT INTO"

Insert examples:

"UPDATE OR INSERT INTO"

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

Oracle

Available for Oracle version 11.2.
"INSERT IGNORE ROW INDEX"

Insert examples:

"INSERT IGNORE ROW INDEX"

```
INSERT /*+ IGNORE_ROW_ON_DUPKEY_INDEX(LANGUAGE_INSERT, LANGUAGE_INSERT_PK) */
INTO LANGUAGE_INSERT(LANGUAGE_ID, NAME, LAST_UPDATE) VALUES(1, 'English', TIMESTAMP '2006-02-15 05:02:19.0');
```

Database backup/restore

NB: This feature is not available in the Lite edition.

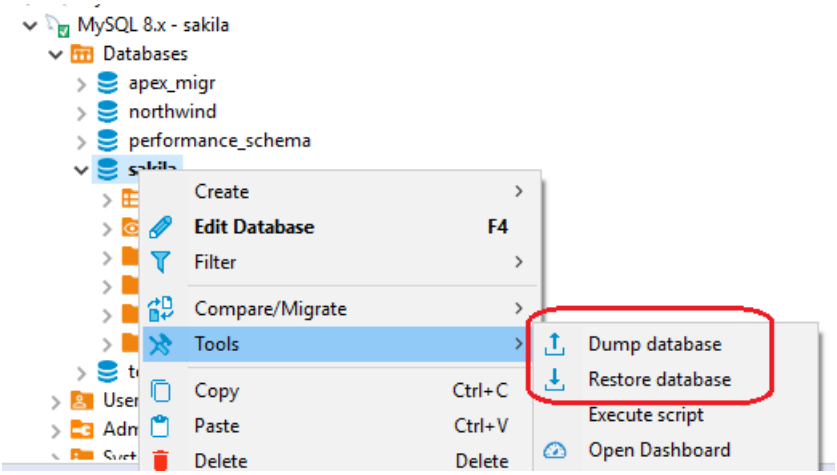
Database Backup/restore

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

- PostgreSQL
- MySQL

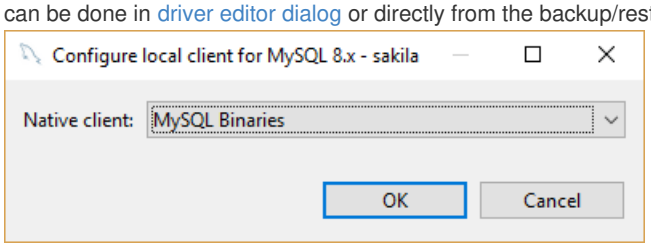
The native backup restore differs from the standard DBeaver data transfer feature. It uses database native dump formats and it may work much faster as it uses special utilities for the direct high-performance database access.

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

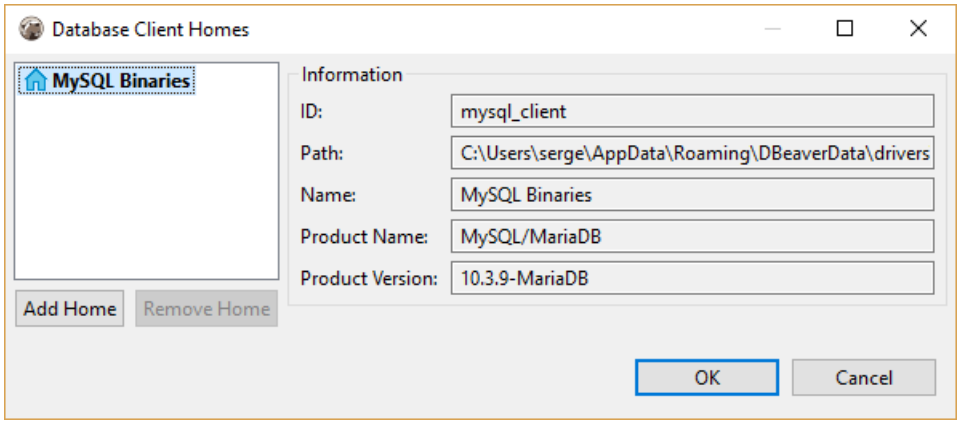


Native client configuration

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

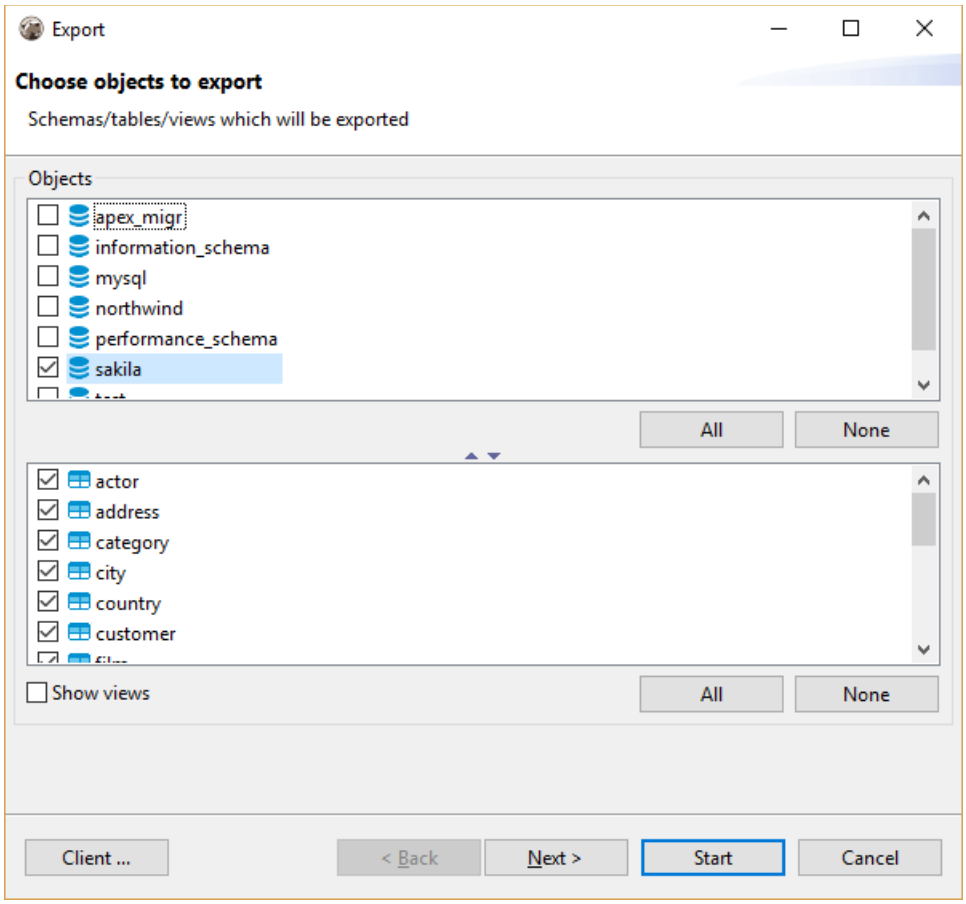


To configure a new client location, choose the `Browse ...` item and add a new client in the following dialog:



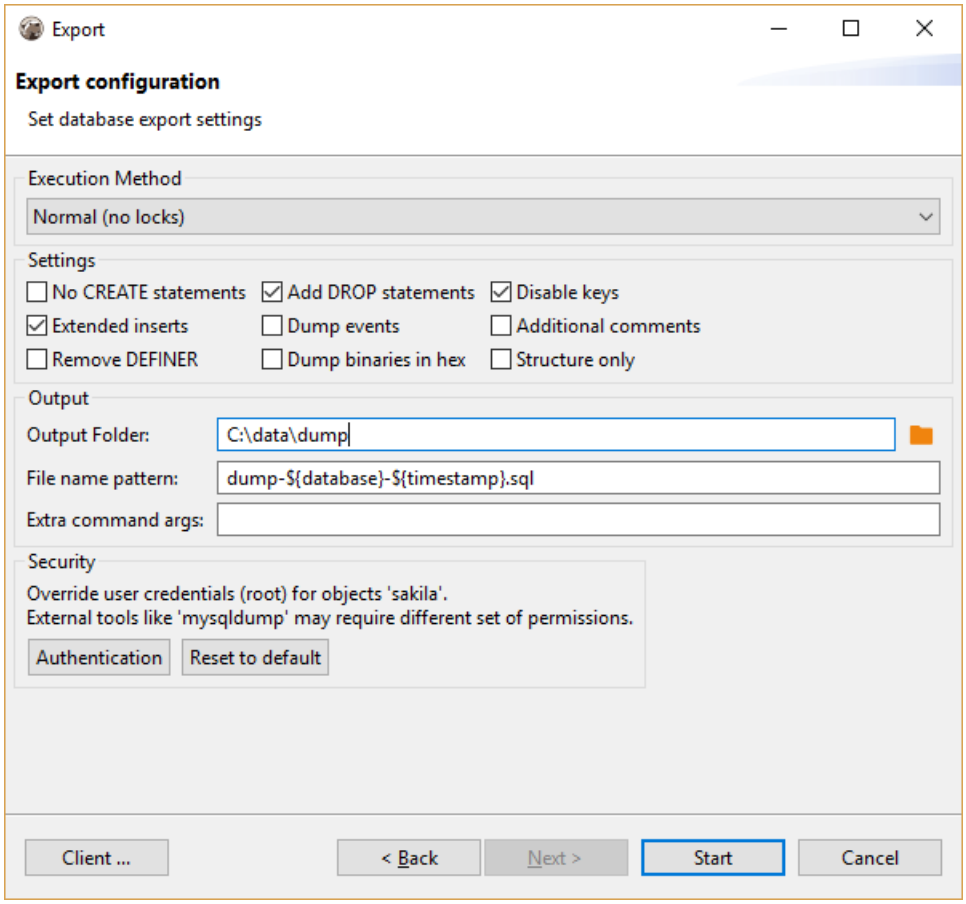
Database dump object selector

You can choose what schemas/tables you want to backup/dump:



Database native tool configuration

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



Task management

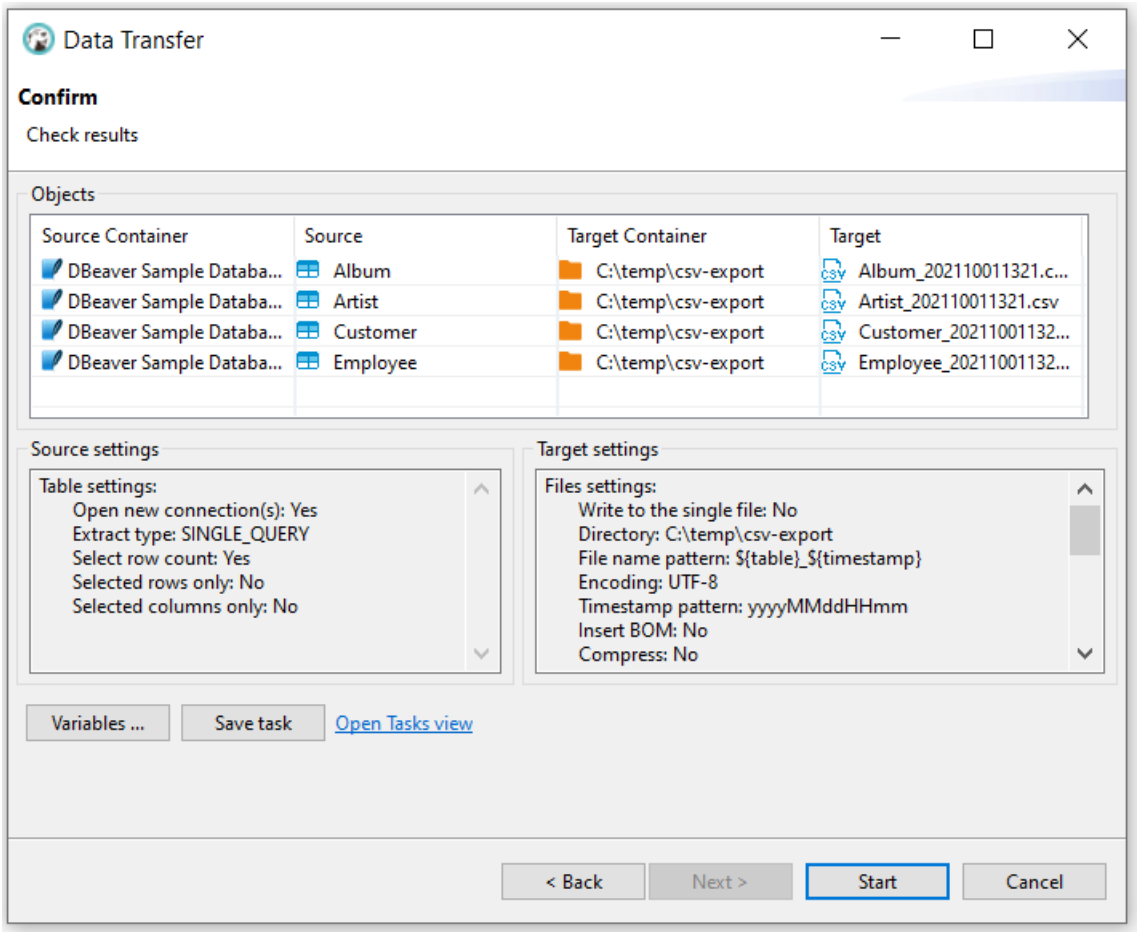
NB: This feature is not available in the Lite edition.

Creating tasks

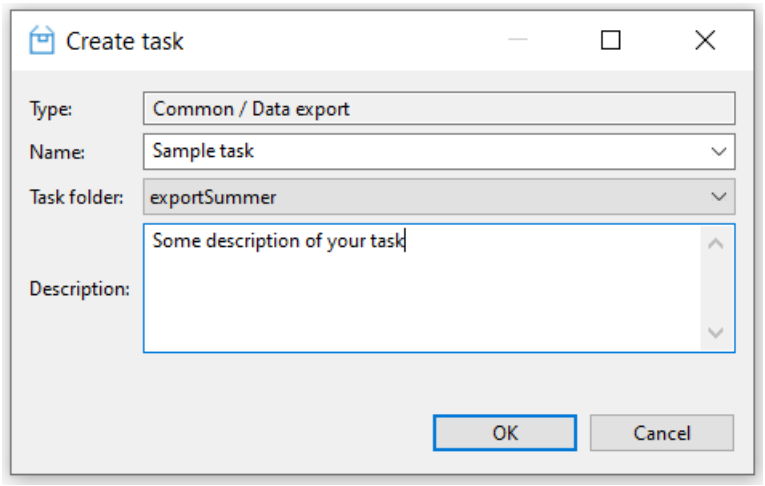
The task is a saved configuration of a database tool. It can be started from the task management view or from the menu by a single click. You can create tasks for frequently used tools. Also, tasks can be scheduled for regular execution.

Create a task from tool configuration

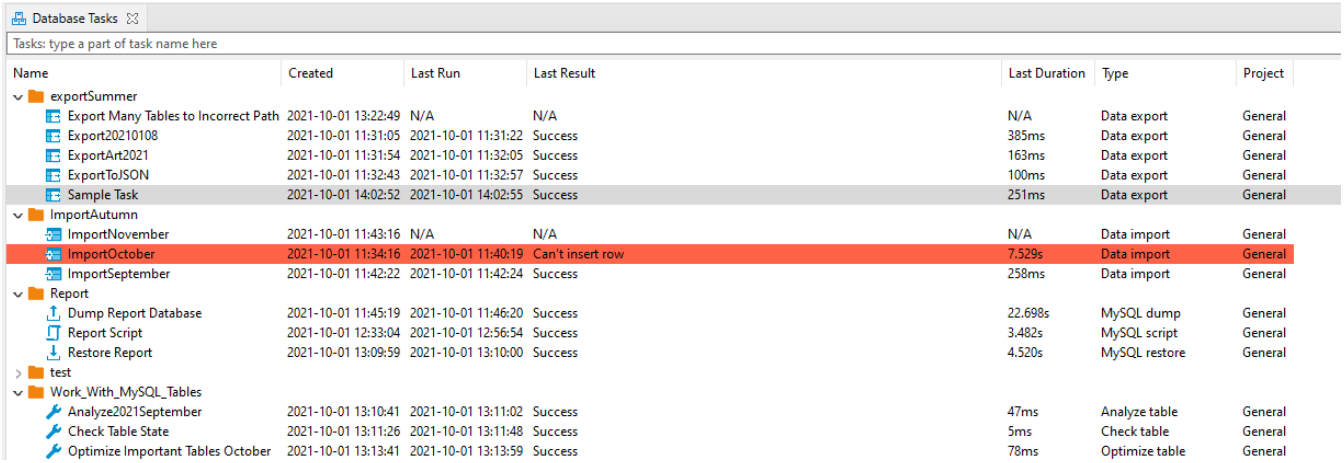
You can save the tool configuration into a task and run your task later with a single click. For example, you can start the Data Transfer wizard and configure the data export from several tables in the SQLite database into CSV files:



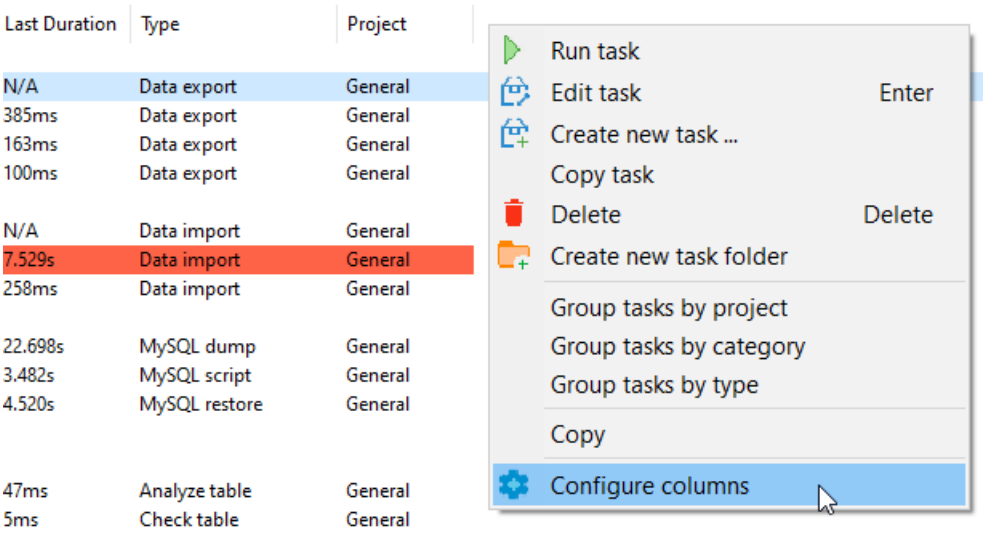
Click on the `Save configuration as task` button and fill the task properties:

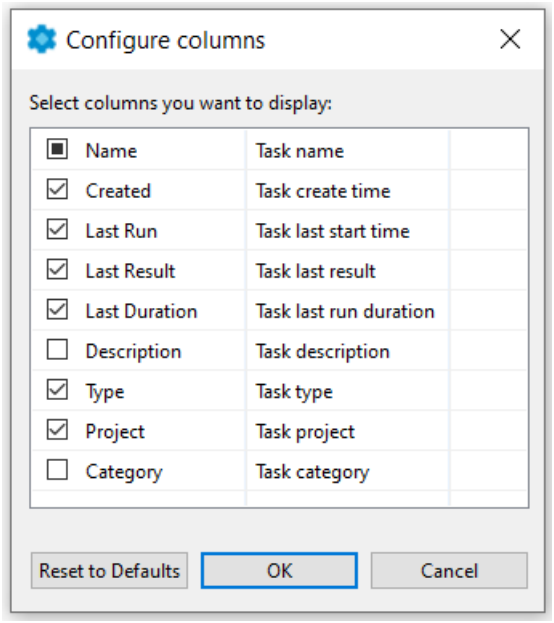


Now click on the `Open Tasks view` link to open the task list:

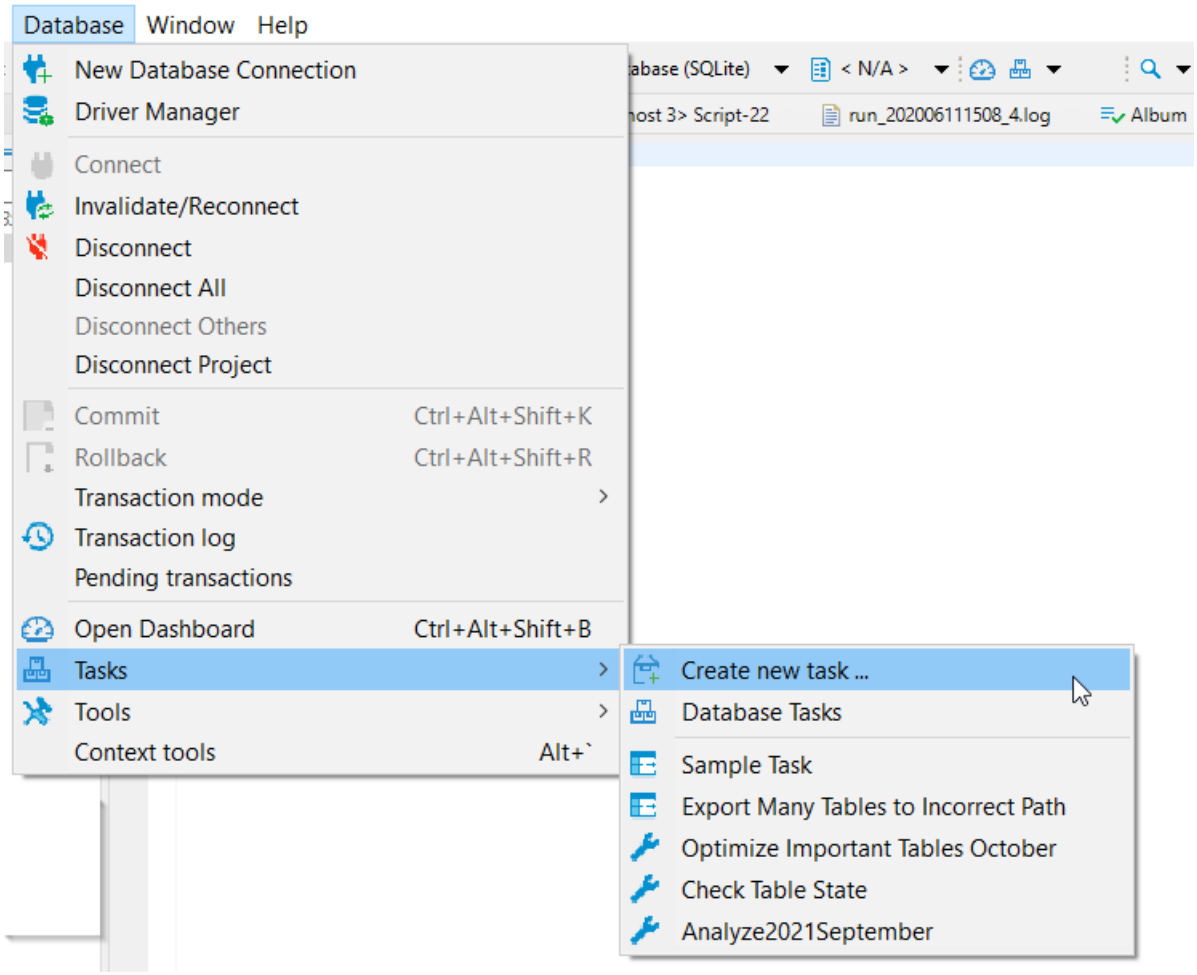


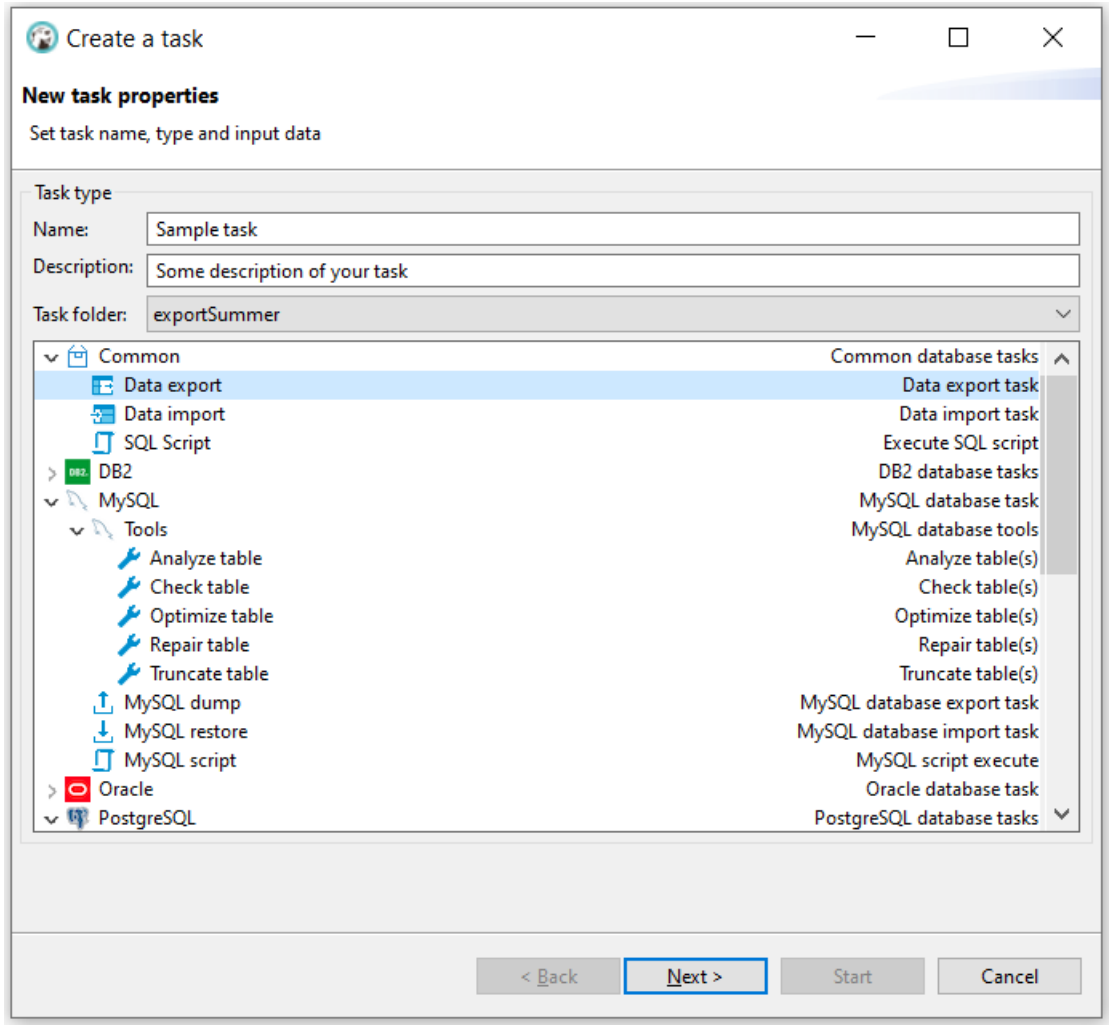
You can configure the Database Tasks View to see more or fewer View columns. Right-click inside the Database Tasks tab and choose the `Configure columns` button from the menu.





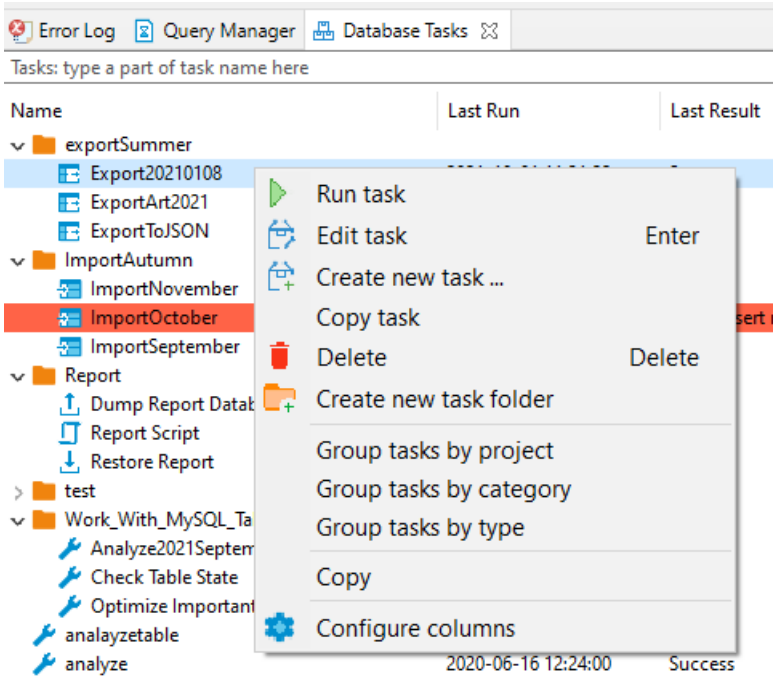
Also, you can create a new task from the main menu `Database -> Tasks -> Create new task...` .



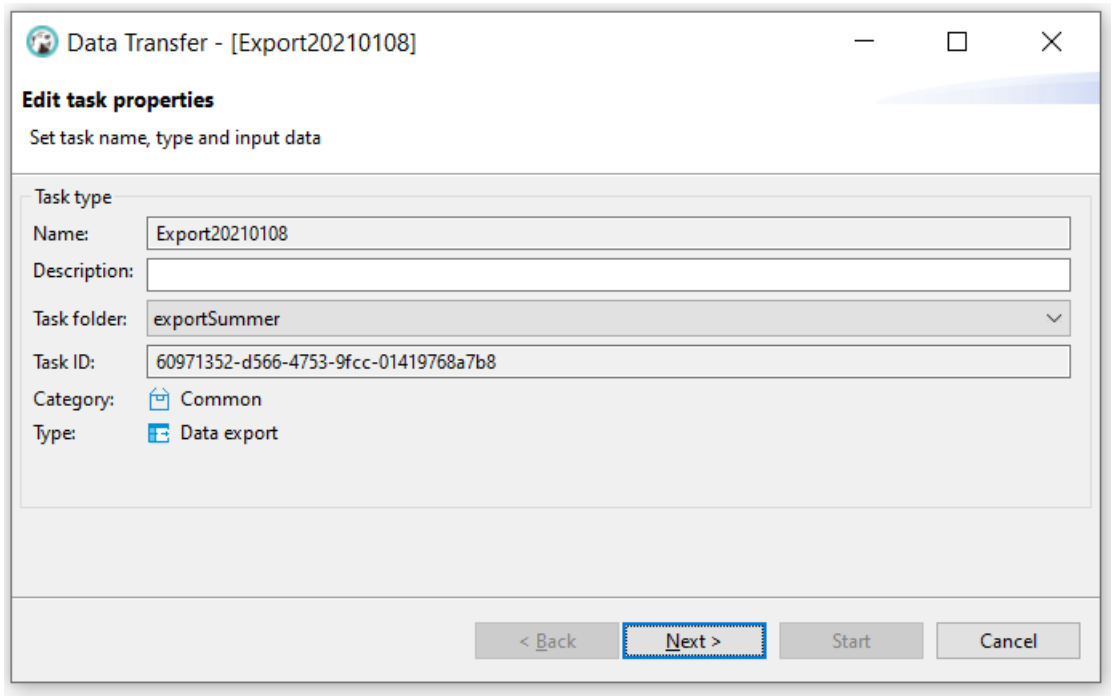


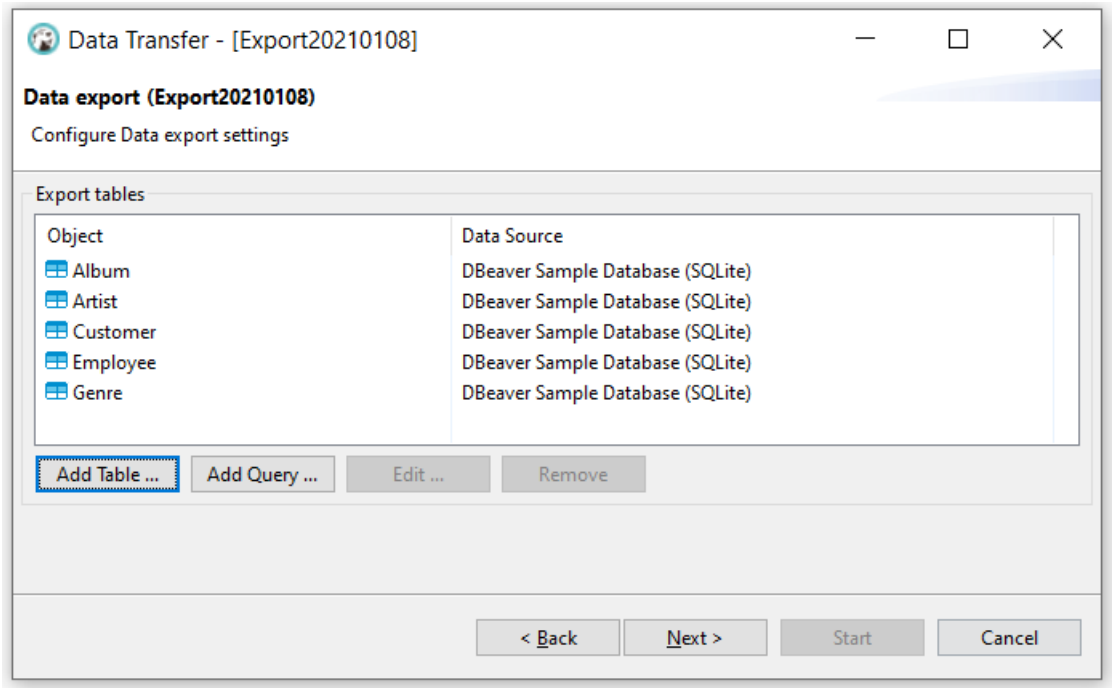
Editing/running tasks

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



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



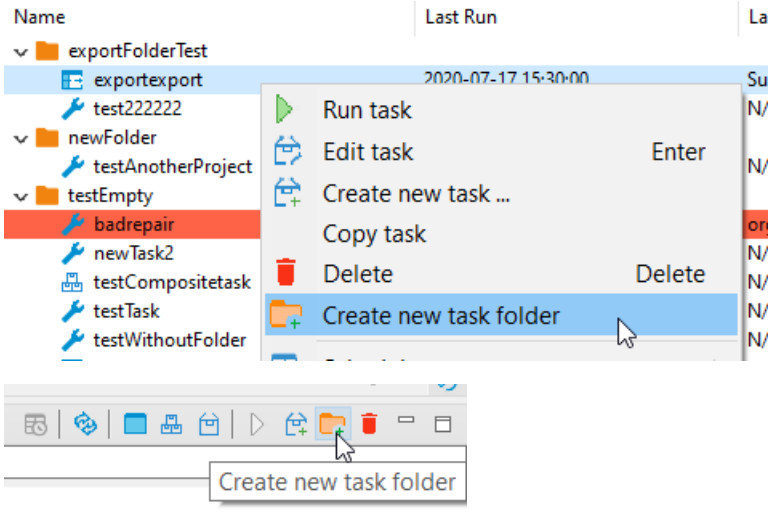


Create a task from task management view

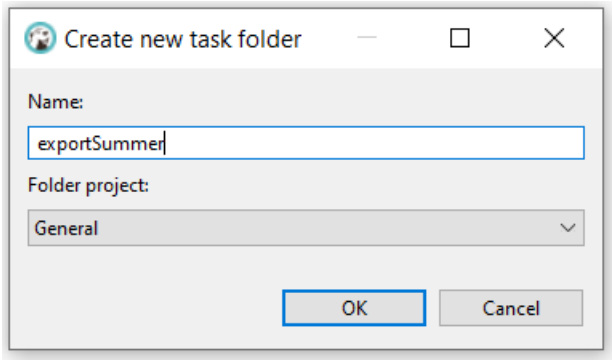
You can create a task from scratch using the tasks view. Open tasks view and click on the `Create new task` button in the View toolbar or in the context menu. In the task wizard, you can choose the task category, task type, task folder, and name. On the next wizard pages, actual tool configuration pages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

Tasks folders

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



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



You can change the task folder for an existing task in the task Edit dialog (use button `Back`). Or you can drag your task to another task folder in the View.

The task folder as a task can be deleted with the button `Delete`.

Scheduling tasks

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

Task scheduler

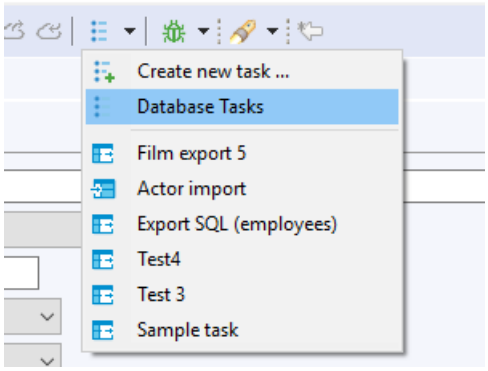
Note: This functionality is available in Enterprise and Ultimate editions only.

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

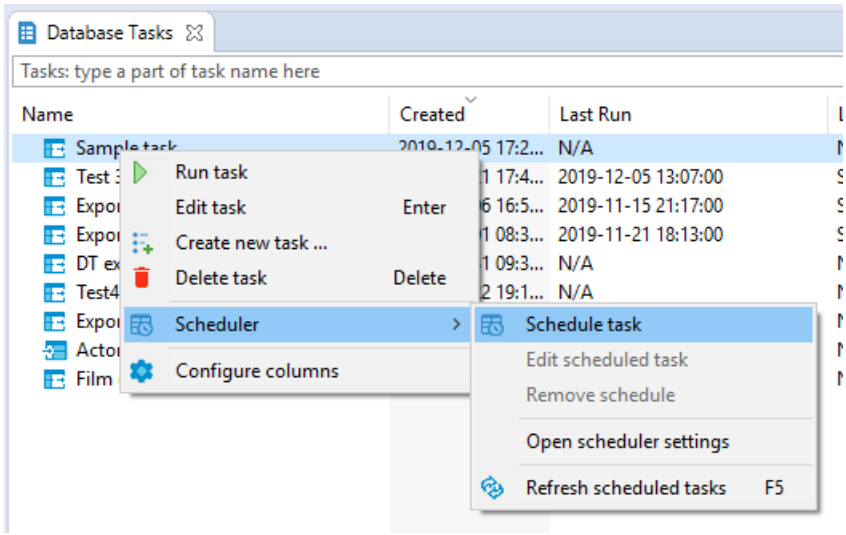
Scheduling tasks from the Tasks view

Windows

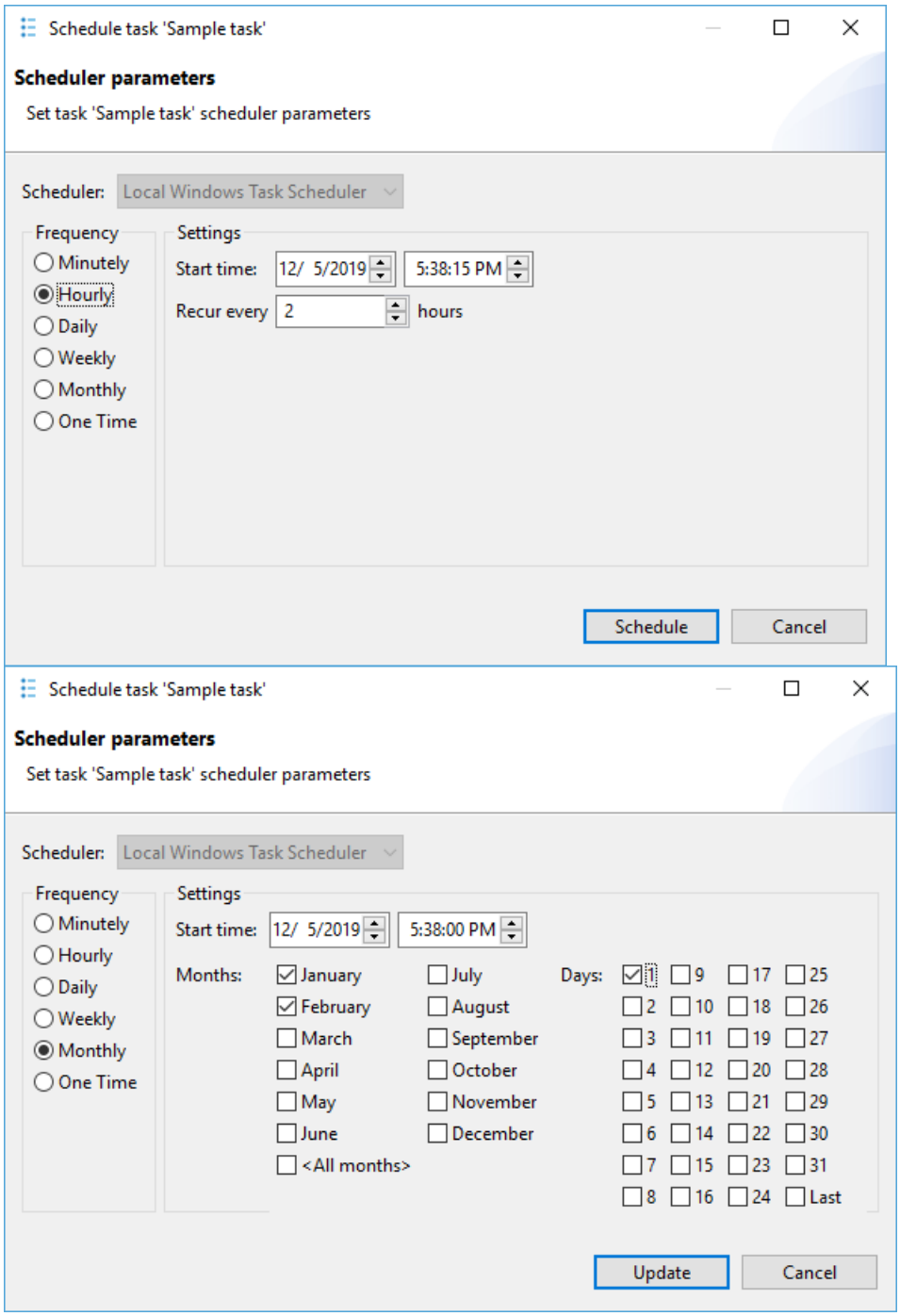
You can open the tasks view from the main toolbar:



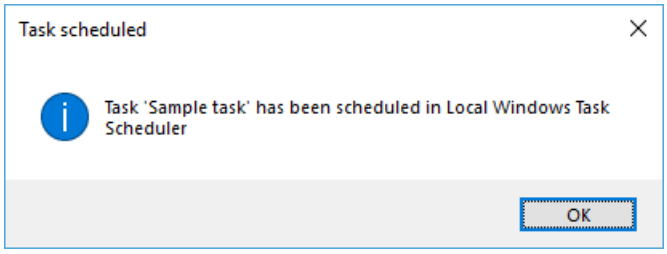
or from the main menu Window. Select a task that you want to schedule in the tasks view and open the context menu:



DBeaver will open the scheduler configuration dialog. You can configure task frequency, recurrence period and start time there:



To schedule the task, click on the Schedule button. If everything is OK, you will see the confirmation dialog:



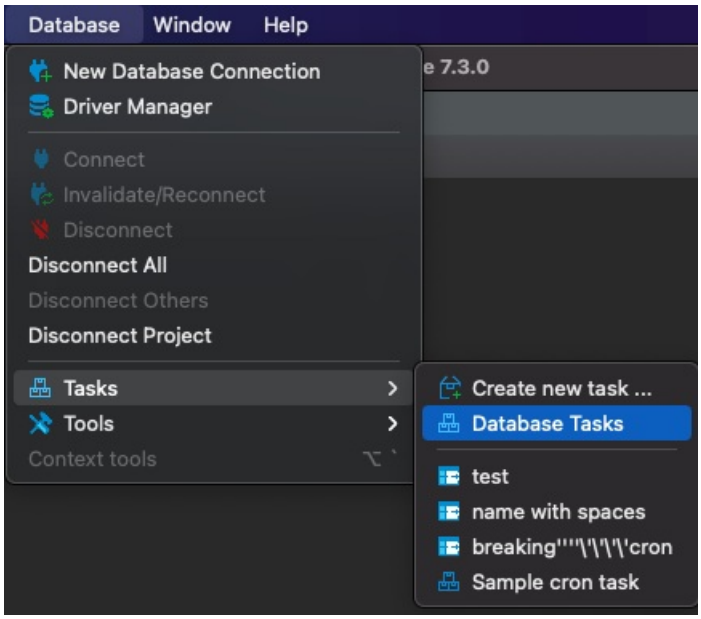
If anything goes wrong, you will see an error message dialog. You can view error details in the Error Log view.

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

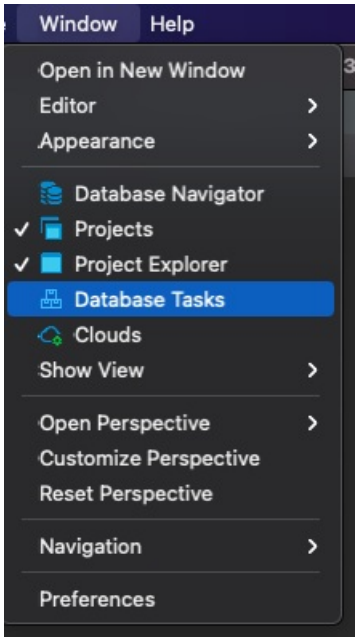
macOS or GNU/Linux

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

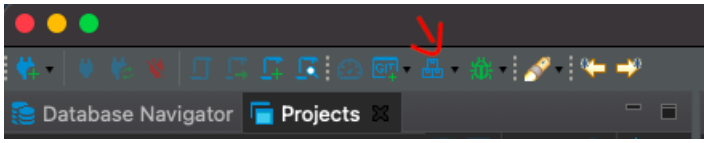
1. Database -> Tasks -> Database Tasks



2. Window -> Database Tasks

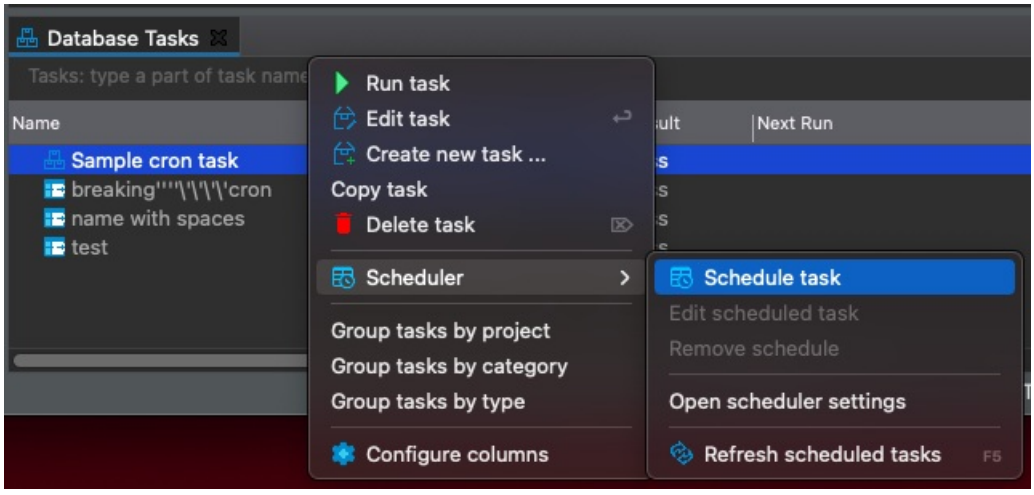


3. Click on 'Show View (Database Tasks)' icon



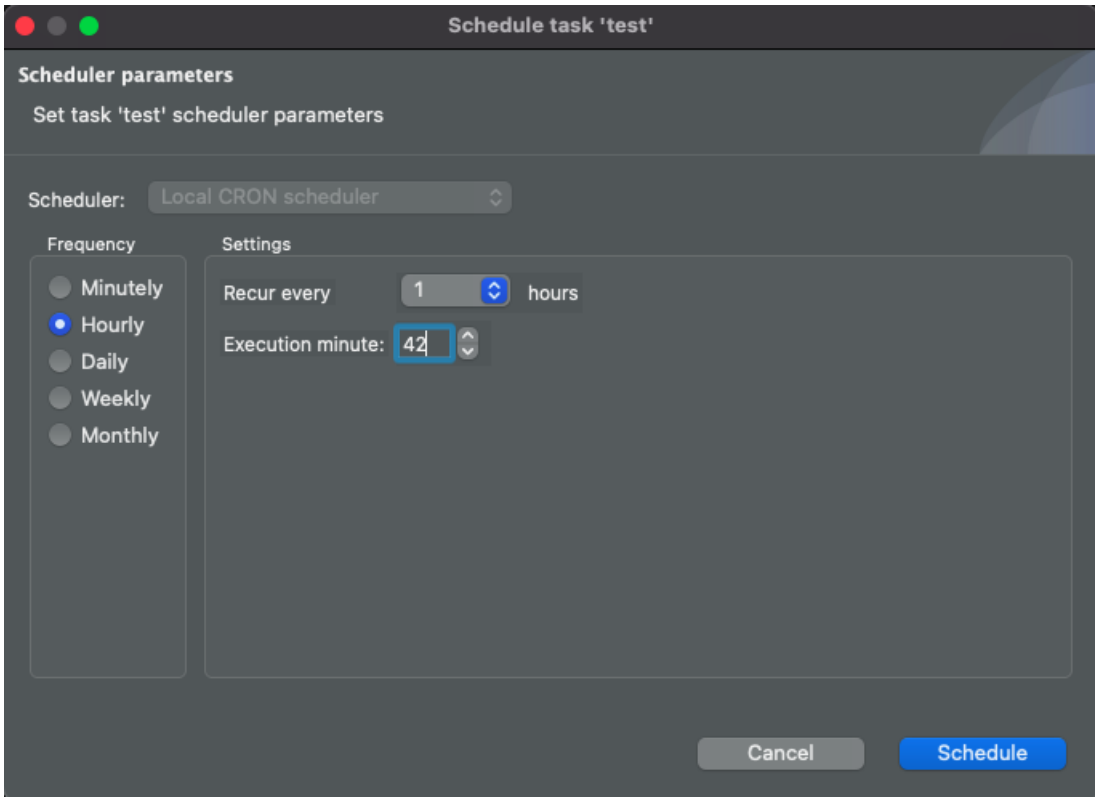
Select a task you want to schedule in the tasks view. To open the scheduler dialog, either:

1. Open the context menu with right-click -> Scheduler -> Schedule task



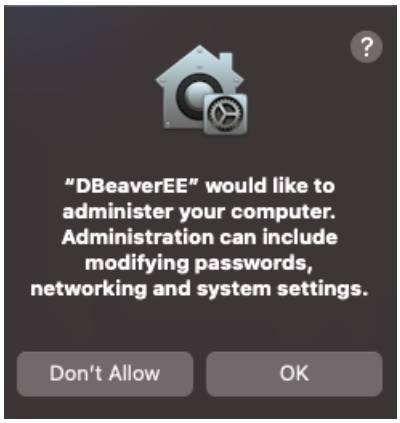
2. or click on the 'Schedule task' icon

DBeaver will open the scheduler dialog. It has a lot of similarities with the corresponding dialog in Windows, but unfortunately, there are fewer settings on macOS and GNU/Linux due to the limitations of `cron`. For instance, when configuring an hourly task, you can only choose the minute at which the scheduler executes the task. In the example below, the task executes at 1:42 PM, 2:42 PM, 3:42 PM, and so on:



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

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



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

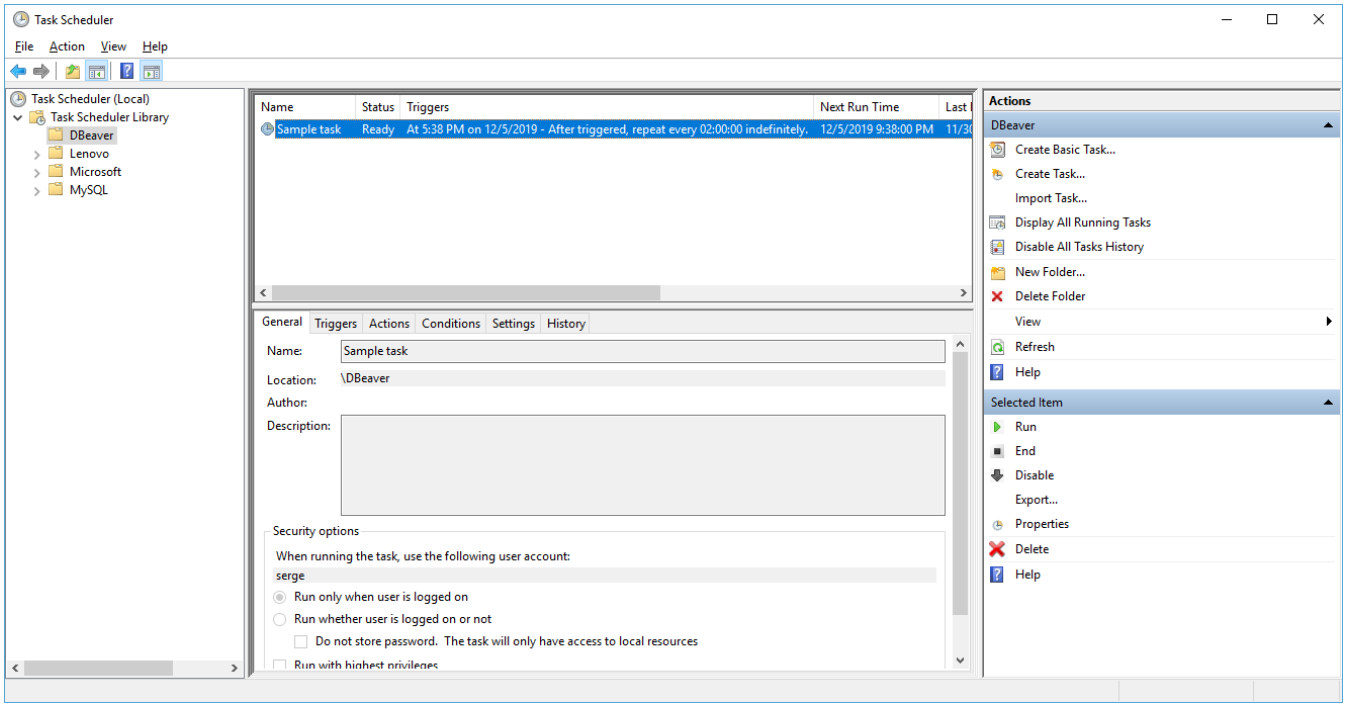
After that, you will see the confirmation message. Just like in Windows, you can change the scheduler settings at any moment by choosing the 'Edit scheduled task' command from the context menu, or cancel the schedule by clicking on 'Remove schedule'.

See schedule details

Windows

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

the task view context menu:



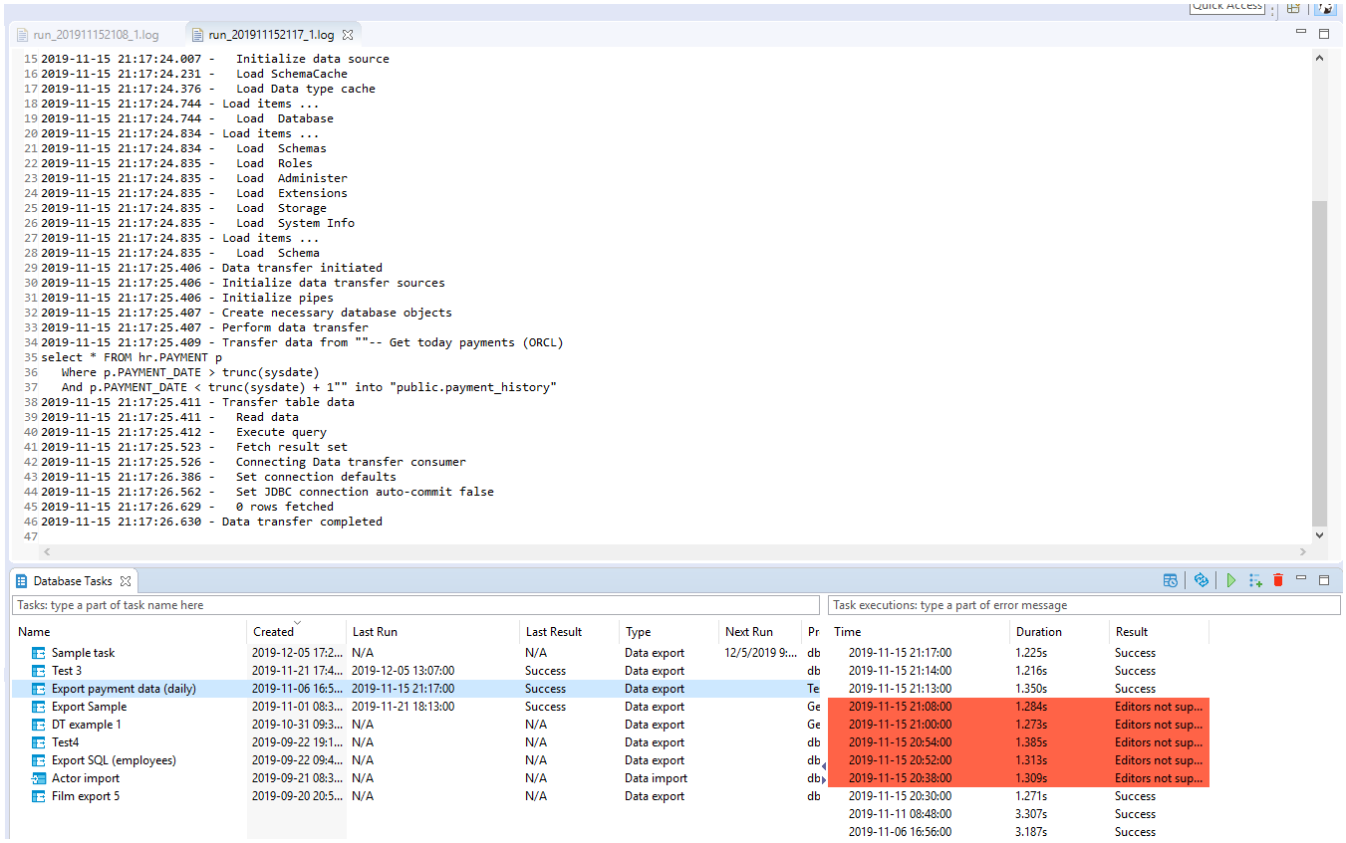
DBeaver stores all tasks in a folder called `DBeaver`.

macOS or GNU/Linux

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

Monitoring for task execution (any OS)

You can look through the task execution logs on the right side of the tasks view. By double-clicking on a task run item, you can see the full log with all details, errors, and warnings:



DBeaver keeps the task run logs in the workspace directory, subfolder .metadata/task-stats.

Running tasks from the command line (any OS)

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

Troubleshooting

Windows scheduler overview

There are two implementations of the Windows scheduler present:

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

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

Windows Task Scheduler: COM exception

Non-legacy scheduler only

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

do the following:

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

Windows Task Scheduler: incorrect date format

Legacy scheduler only

If you encounter an error in Windows which looks like this: `ERROR: Invalid Start Date (Date should be in %some_format% format).`,

do the following:

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

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

macOS 10.15+: Unable to read or write to crontab

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

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

9. Close the lock.

Tasks from password-protected projects can't be run

You need to pass a password for one or more projects via the command-line interface.

To do so, you need to set the `dbeaver.project.password` parameter in the external configuration file like so:

```
# You can specify a single password for all projects:
dbeaver.project.password=p4$$w0rd

# Otherwise, you can specify a list of passwords for given projects:
dbeaver.project.password=@General:p4$$w0rd,@Other:12345
```

The syntax for a single entry is `@ <name of the project> : <password of the project>`; others are separated by the `,` symbol.

Please note that `@` and `:` symbols are mandatory.

Composite tasks

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

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

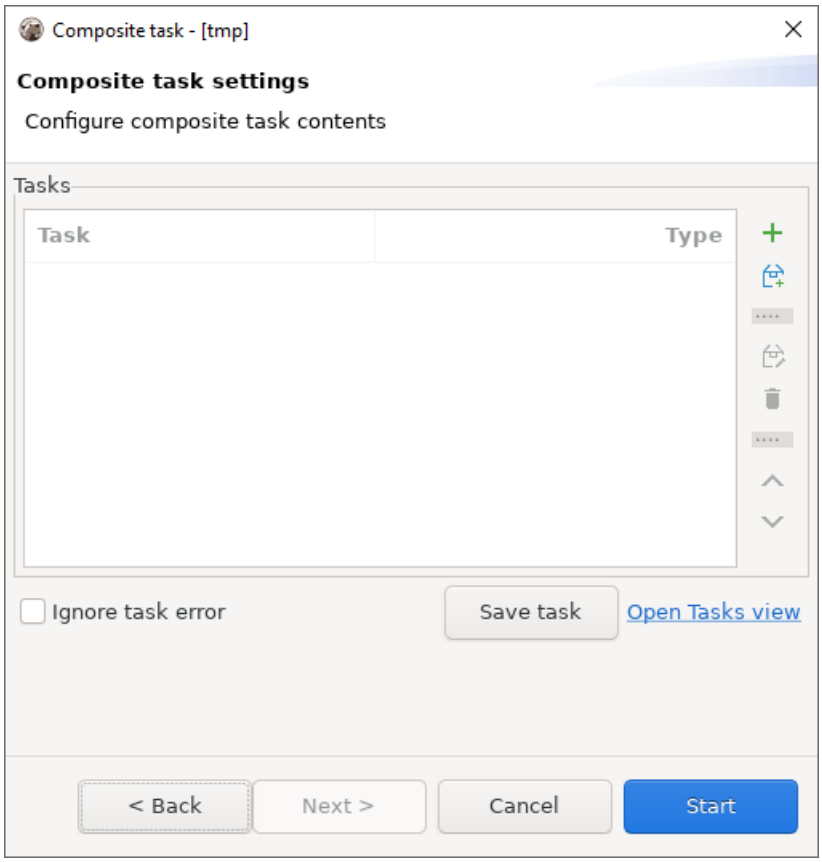
Creating a *composite task*

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

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

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

You will be presented with the following dialog:



Setting up a *composite task*

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

This can be done:

1. By adding an existing task. To do that, click the button with the plus sign
2. By creating a new task and adding it simultaneously. To do that, click the button below the aforementioned button with the plus sign
3. By drag-and-dropping a task from the _Database tasks panel.

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

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

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

Sending results by e-mail

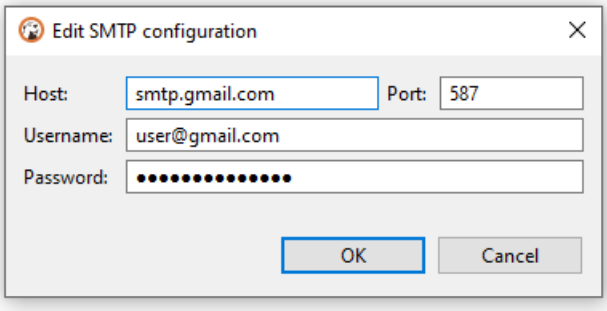
Note: The following feature is only available in Enterprise and Ultimate editions.

DBeaver offers a way to send data exported via Data Transfer by email.

SMTP profile configuration

First, you will need to add an SMTP profile to send the email from. Go to `Window` ⇒ `Preferences` ⇒ `General` ⇒ `Mail` and create a new profile.

Parameters `Host` and `Port` may depend on the mail service you use. If the service offers both SSL and TLS ports, use the latter one. Gmail, for example, uses host `smtp.gmail.com` and port `587`. An example of a configured profile:

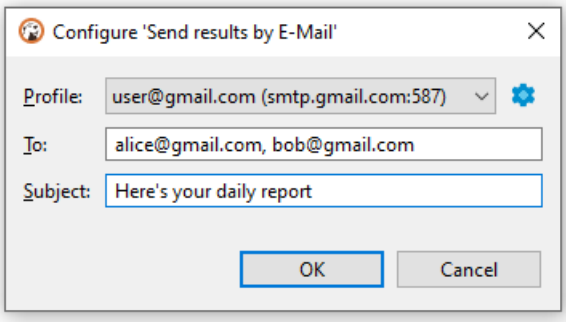


Then you can use the `Test connection` button to verify that the host and credentials are valid.

See the troubleshooting section for more information on resolving common authorization problems.

Setting up data transfer

When at least one profile is present, you can actually set up email sending. Create a regular export task, go to the `Output` page and make sure the `Send results by E-Mail` option is enabled. By pressing the `Configure` label near it, you can specify several recipients and the subject for your mail:



That's it. After the task is completed, the specified recipients will receive an email containing the exported file in a specified format during the data transfer.

Authorization troubleshooting

You may face various problems while setting up a new SMTP profile.

Several common errors when using Gmail and solutions for them are described below:

- `535-5.7.8 Username and Password not accepted`. Check that the username and password are correct. If you are certain that you have entered valid credentials, then try allowing less secure apps.
  Read more at https://support.google.com/accounts/answer/6010255
- `534-5.7.9 Application-specific password required`. You have two-factor authorization enabled. You will need to generate a special password for DBeaver to use.
  Read more at https://support.google.com/accounts/answer/185833

There were also several cases when antivirus' would block DBeaver from sending an email.

Cloud Explorer

Overview

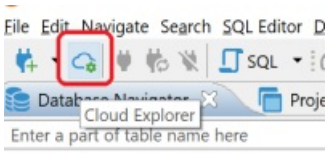
Cloud Explorer provides a deep integration with classic cloud service providers such as Amazon, Google and Azure.

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

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

Cloud configuration

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

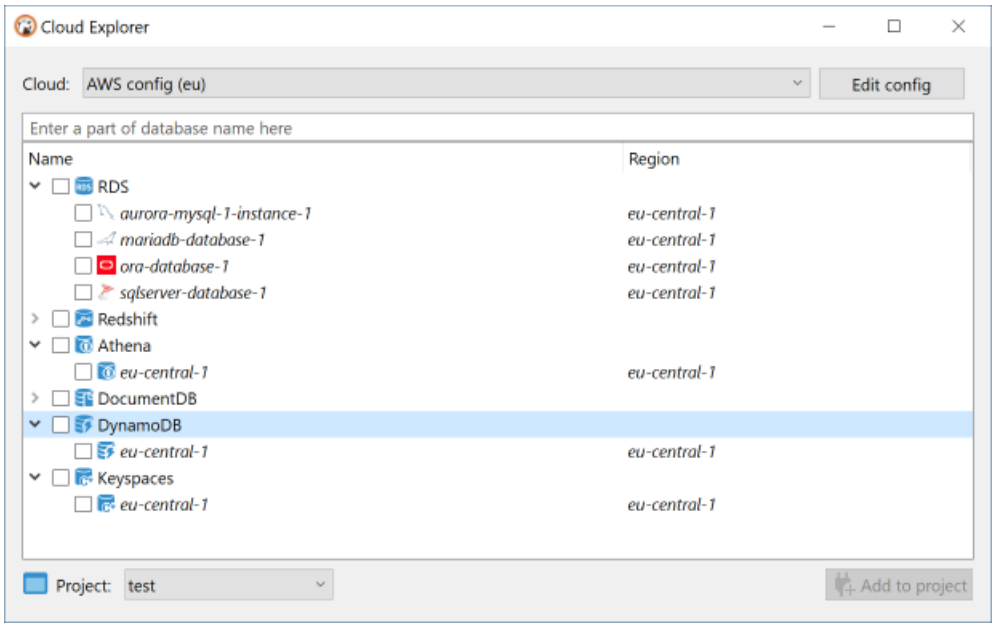


Configuring AWS cloud

Explorer

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

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

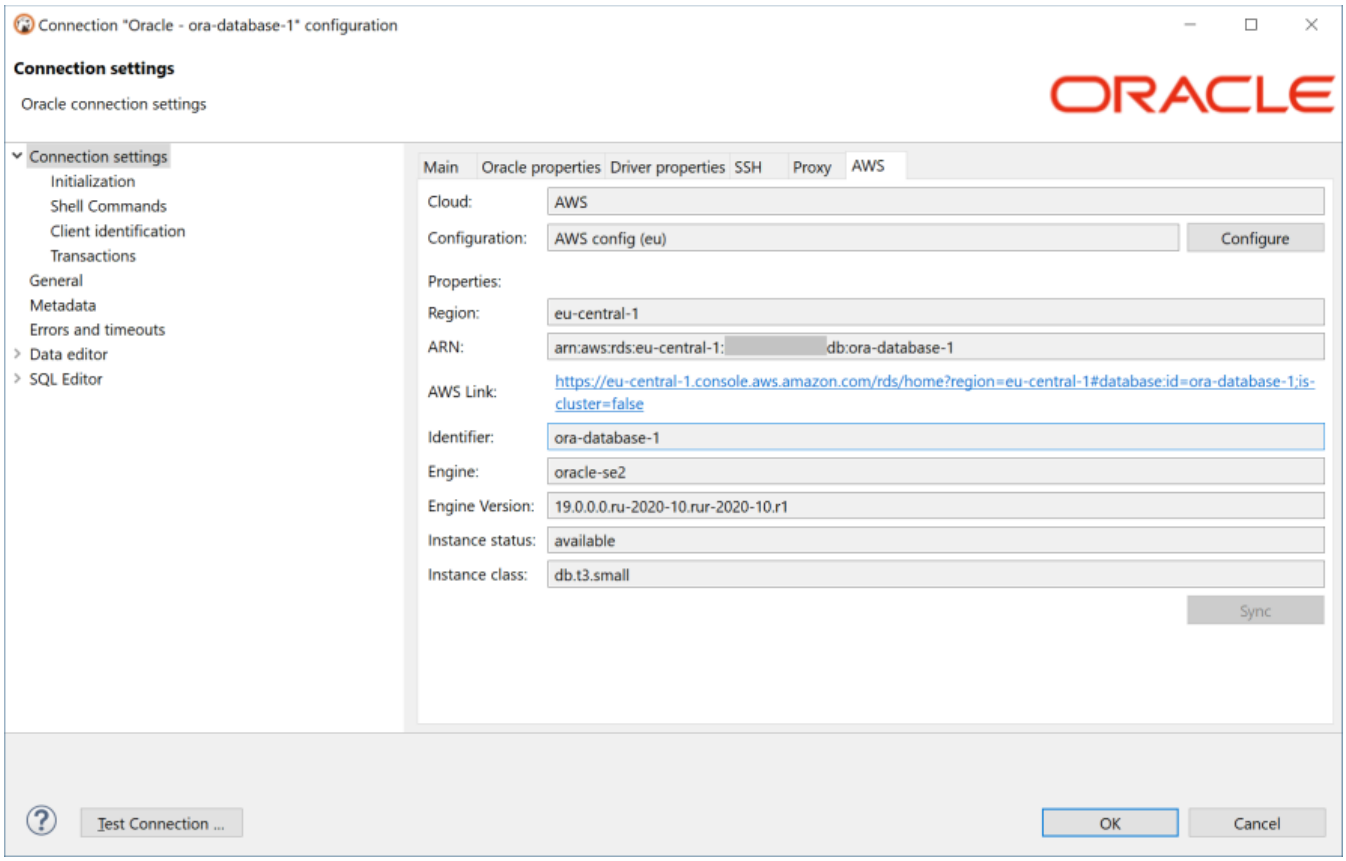


If you have a large number of databases in your cloud, you can search or filter them using filter text above the cloud navigator.

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

Database cloud information

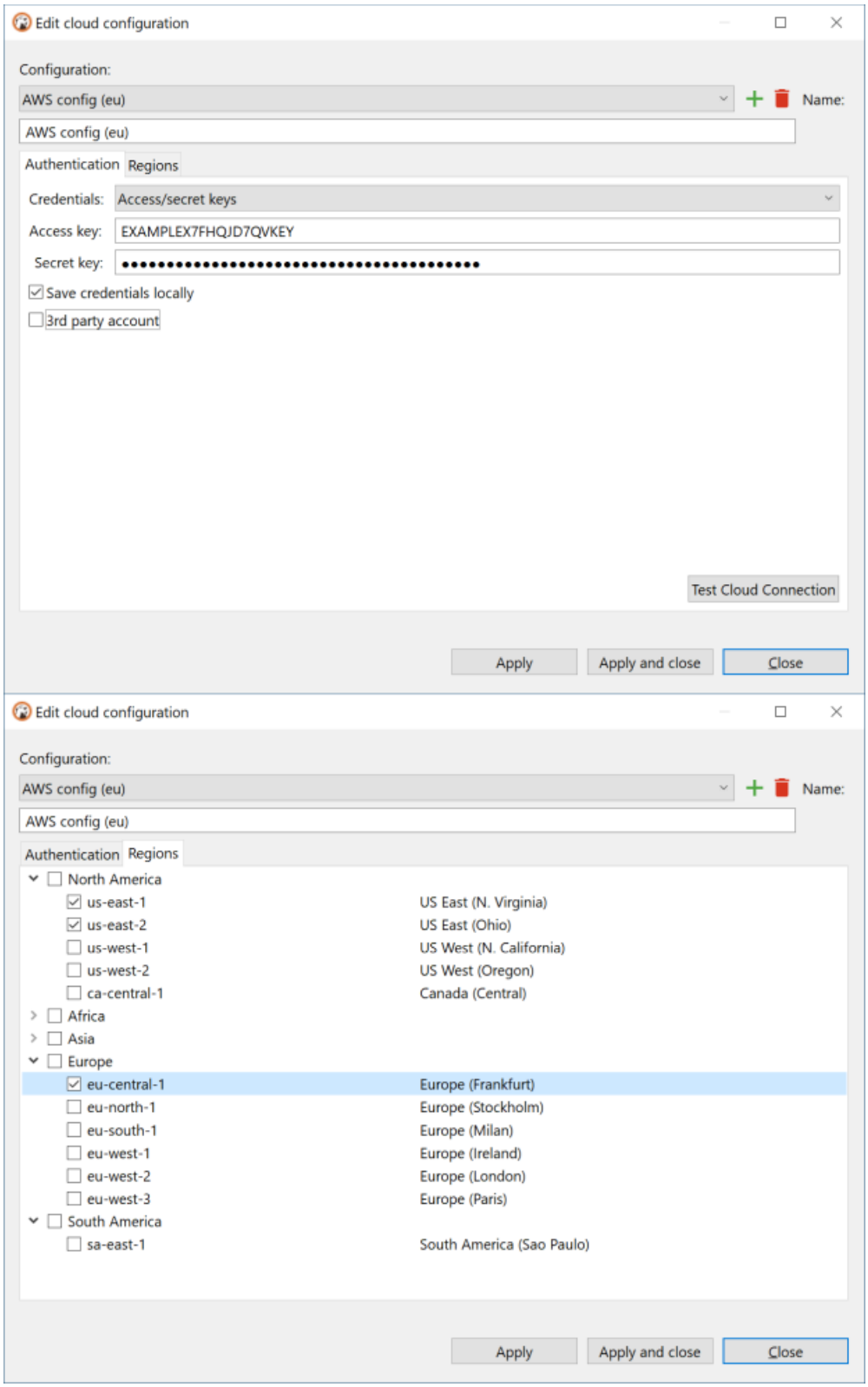
You can always see your cloud database configuration in a special tab in the connection settings dialog. This information depends on both cloud and database type. You can also click on the external link to open your database configuration in the cloud provider web console.


Connection "Oracle - ora-database-1" configuration
Connection settings
Oracle connection settings
ORACLE
Connection settings
Initialization
Shell Commands
Client identification
Transactions
General
Metadata
Errors and timeouts
Data editor
SQL Editor
Main
Oracle properties
Driver properties
SSH
Proxy
AWS
Cloud:
AWS
Configuration:
AWS config (eu)
Configure
Properties:
Region:
eu-central-1
ARN:
arn:aws:rds:eu-central-1:
db:ora-database-1
AWS Link:
https://eu-central-1.console.aws.amazon.com/rds/home?region=eu-central-1#database:id=ora-database-1;is-cluster=false
Identifier:
ora-database-1
Engine:
oracle-se2
Engine Version:
19.0.0.0.ru-2020-10.rur-2020-10.r1
Instance status:
available
Instance class:
db.t3.small
Sync
Test Connection ...
OK
Cancel

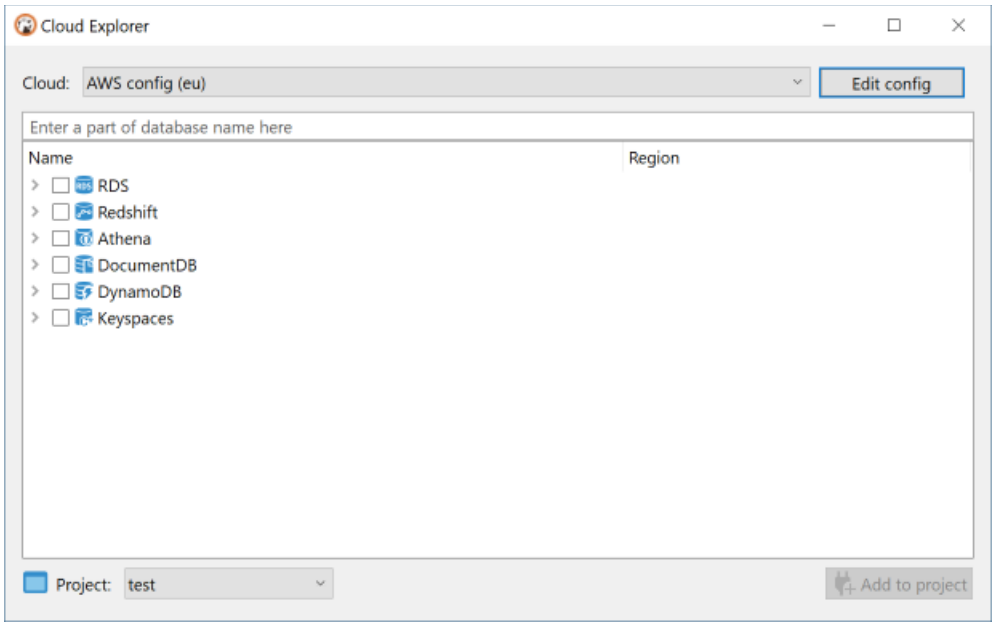
AWS

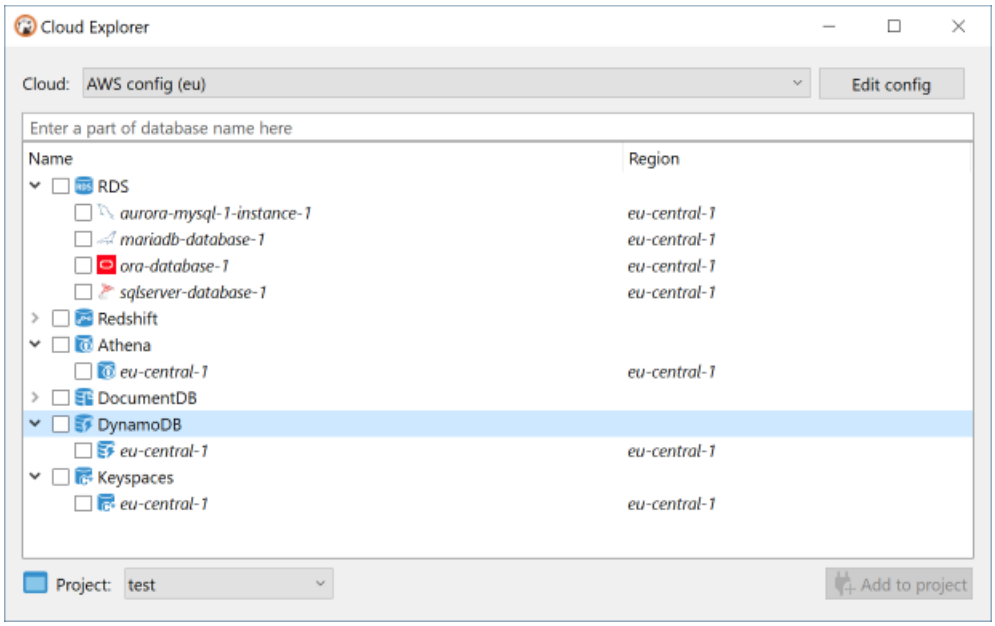
AWS (Amazon Web Services) Cloud Explorer

Cloud configuration



Supported cloud databases


Cloud Explorer
Cloud: AWS config (eu)
Edit config
Enter a part of database name here
Name
Region
RDS
Redshift
Athena
DocumentDB
DynamoDB
Keyspaces
Project: test
Add to project


Cloud Explorer
Cloud: AWS config (eu)
Edit config
Enter a part of database name here
Name
Region
RDS
aurora-mysql-1-instance-1
eu-central-1
mariadb-database-1
eu-central-1
ora-database-1
eu-central-1
sqlserver-database-1
eu-central-1
Redshift
Athena
eu-central-1
eu-central-1
DocumentDB
DynamoDB
eu-central-1
eu-central-1
Keyspaces
eu-central-1
eu-central-1
Project: test
Add to project

AWS Credentials

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

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

Default credentials

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

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

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

Please read the AWS credentials documentation for a detailed explanation.

Access keys

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

Official AWS instructions: Managing access keys for IAM users

AWS Profiles

Similar to default credentials but you can also choose which credentials profile you want to use.
The official AWS instructions can be found at credentials config files.

Single Sign On

If your AWS account has a configured SSO portal, you can use a web-based SSO authorization. SSO support can be enabled for Default and Profile-based AWS authorization types.
You need to turn on the "Enable SSO" option.

AWS Secrets Manager

If you have a configured AWS Secret you can use it to access you database. Secrets can be used for a RDS databases and Redshift. Instructions on how to create AWS Secret can be found here. Password field is required.

Note

Secret needs to be in the same region as the database.

AWS SSO

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

AWS Single Sign-On is a cloud-based single sign-on (SSO) service that makes it easy to centrally manage SSO access to AWS resources.
You do not need to specify any user credentials explicitly in DBeaver connections configuration. All authorization is performed in a web browser in a 3rd party SSO provider, e.g. Google workspace, Microsoft AD portal, Facebook, etc.

AWS CLI

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

AWS CLI version 2.2 is recommended.

AWS SSO configuration

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

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

Connection configuration

In the DBeaver database connection dialog you need to:

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

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

GCP Credentials

DBeaver is integrated with Goofle Cloud IAM authentication.
Thus it provides the possibility to authenticate in GCP to access your cloud databases.

Default credentials

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

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

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

Please read the GCP authentication documentation for a detailed explanation.

Access key file

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

You can read more about User and Service authentication here

Web browser or Single Sign On

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

GCP SSO

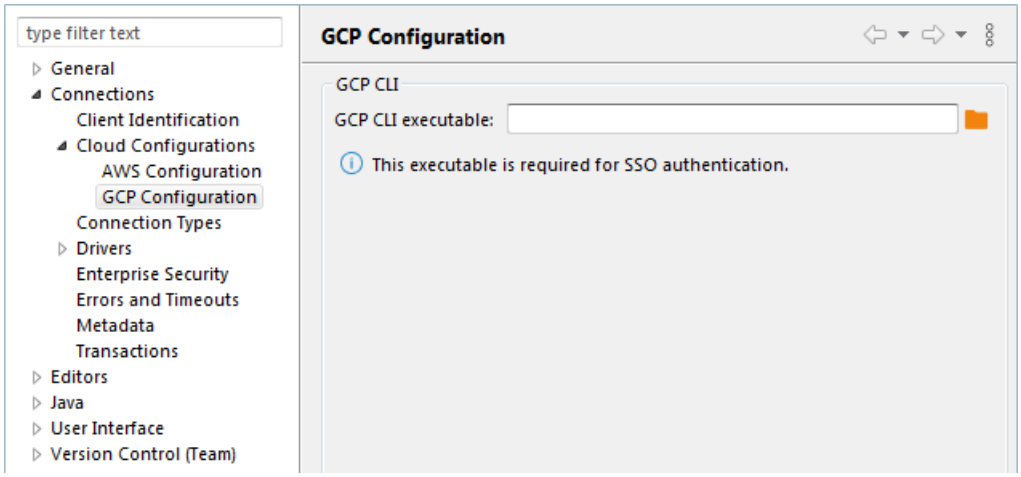
NB: This feature is available in Lite, Enterprise, and Ultimate editions only.

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

GCP CLI

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

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



GCP web browser-based authentication configuration

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

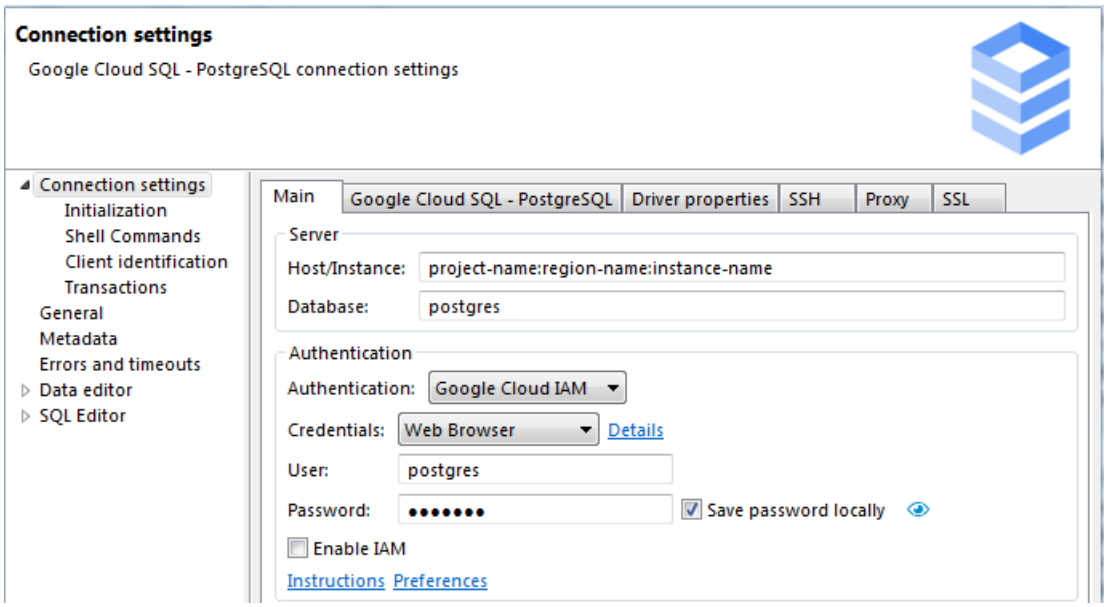
Connection configuration

In the DBeaver database connection dialog, you need to:

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

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

As example:


Connection settings
Google Cloud SQL - PostgreSQL connection settings
Connection settings
Initialization
Shell Commands
Client identification
Transactions
General
Metadata
Errors and timeouts
Data editor
SQL Editor
Main
Google Cloud SQL - PostgreSQL
Driver properties
SSH
Proxy
SSL
Server
Host/Instance:
project-name:region-name:instance-name
Database:
postgres
Authentication
Authentication:
Google Cloud IAM
Credentials:
Web Browser
Details
User:
postgres
Password:
Save password locally
Enable IAM
Instructions
Preferences

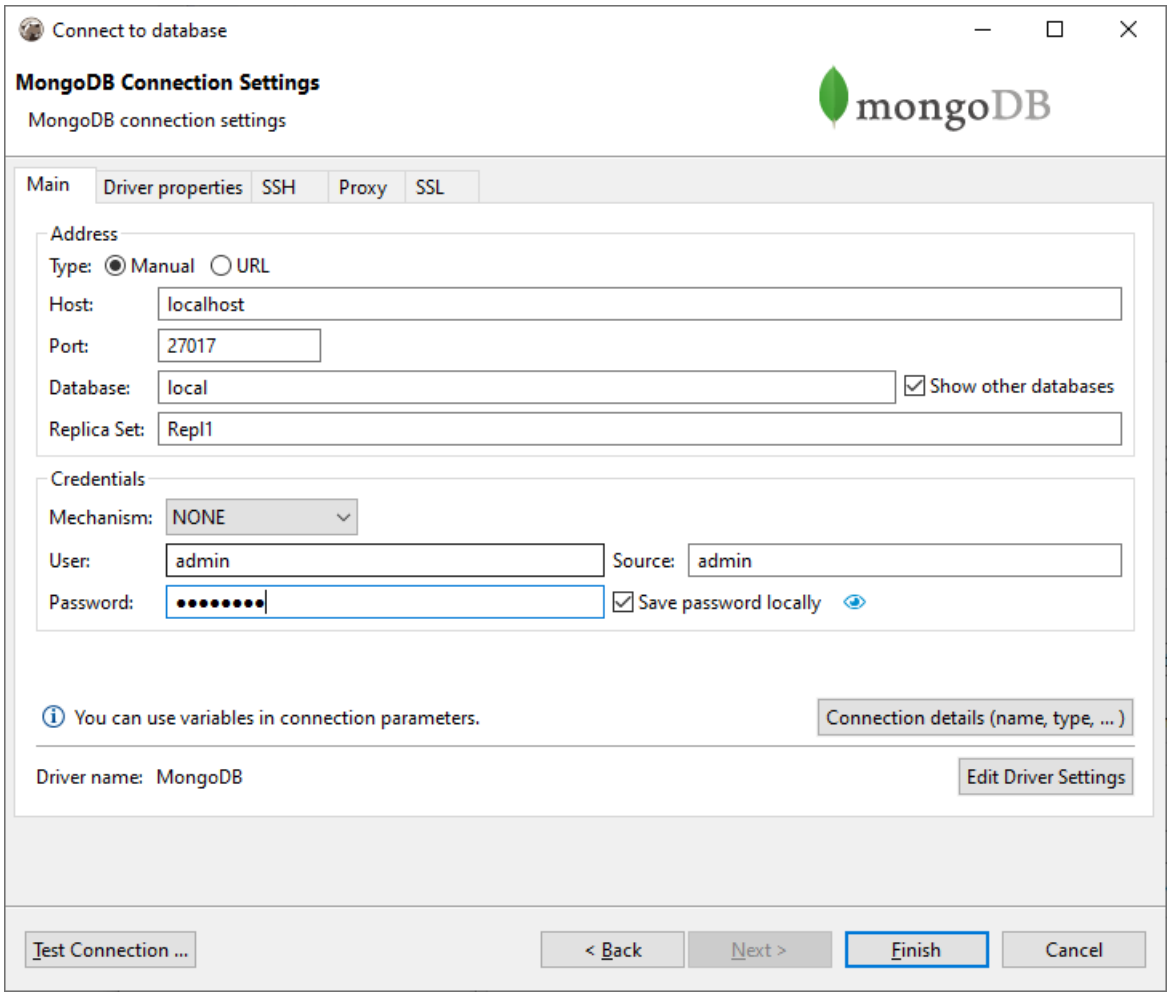
MongoDB

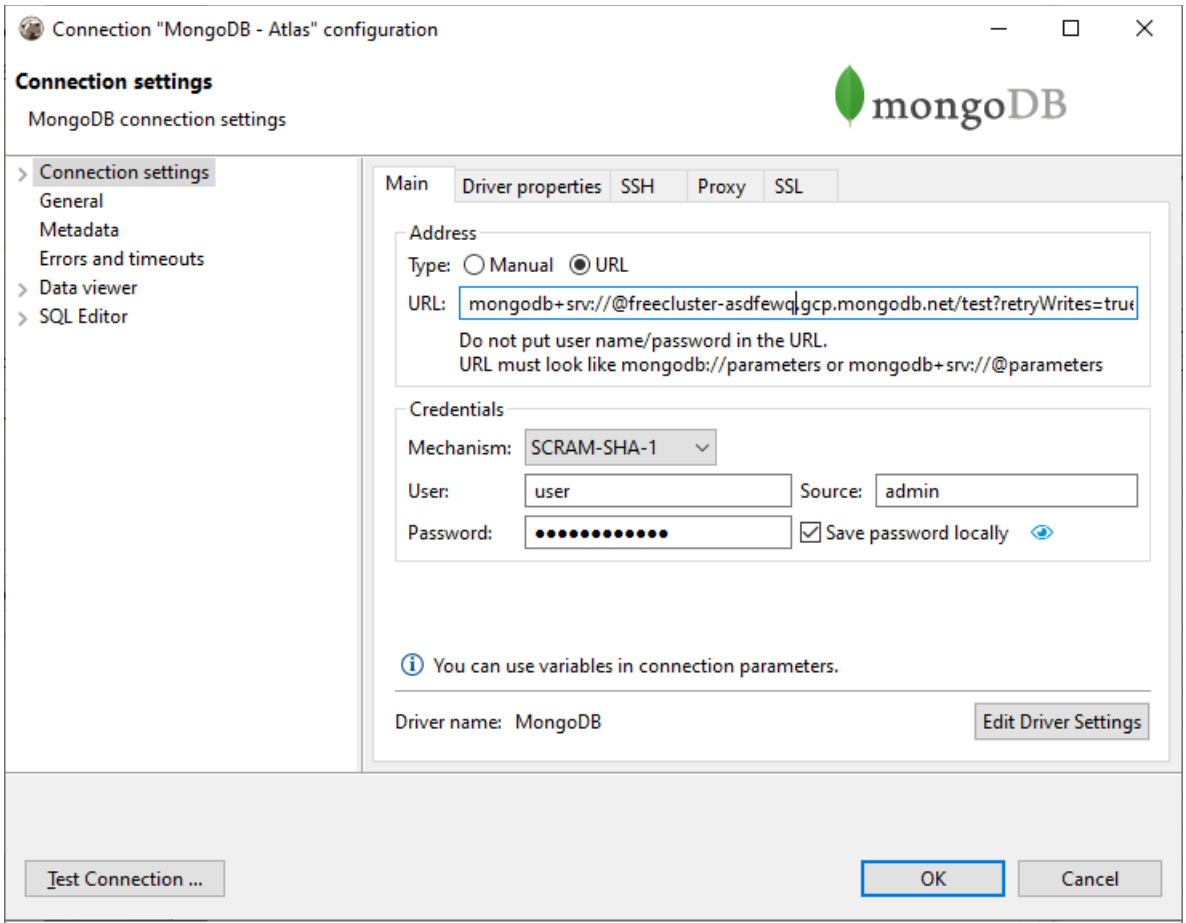
Overview

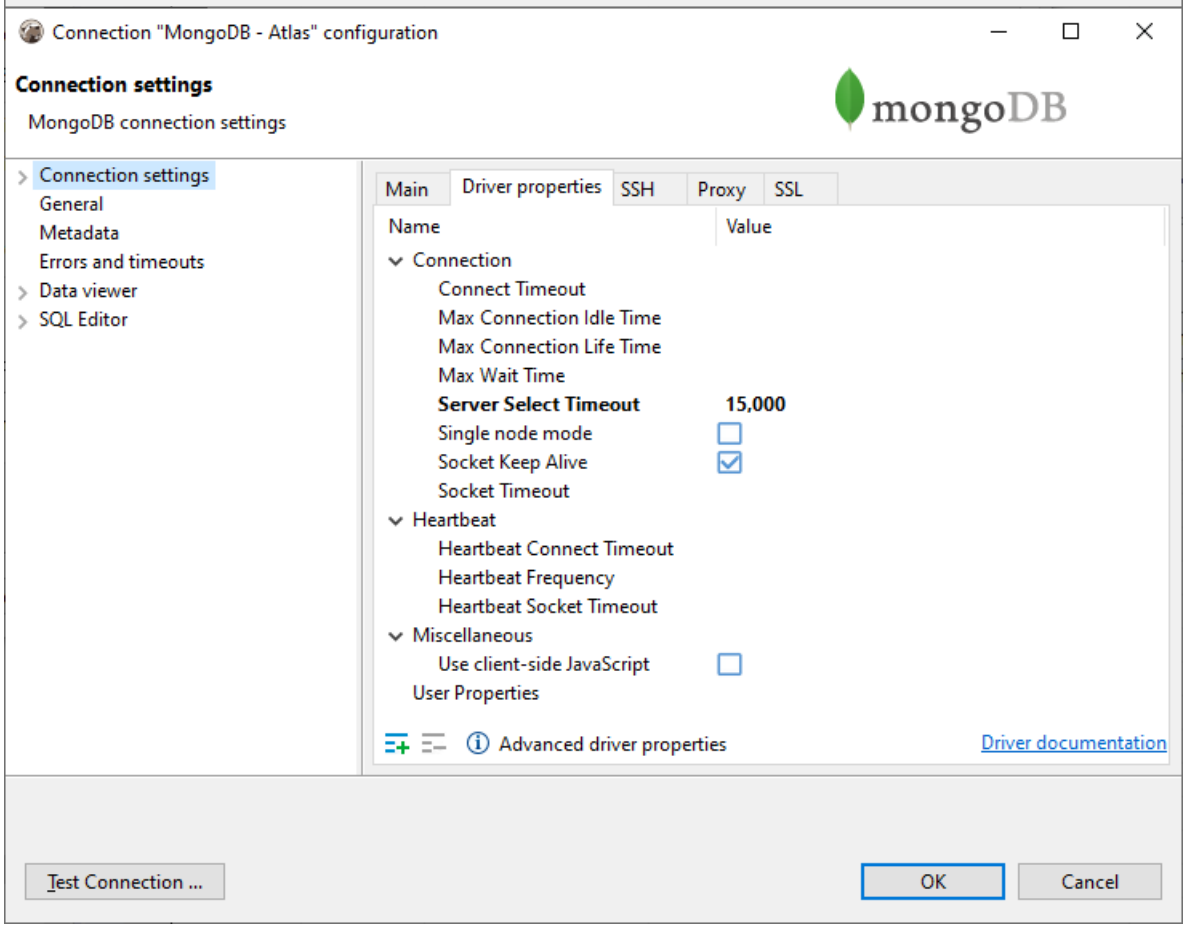
DBeaver EE supports MongoDB schema browser, data viewer, SQL and JavaScript queries execution. It also supports various administrative tools (like server sessions manager).
DBeaver uses MongoDB Java driver 3.8.0 to operate with a server. It supports MongoDB servers from 2.x to 4.x.

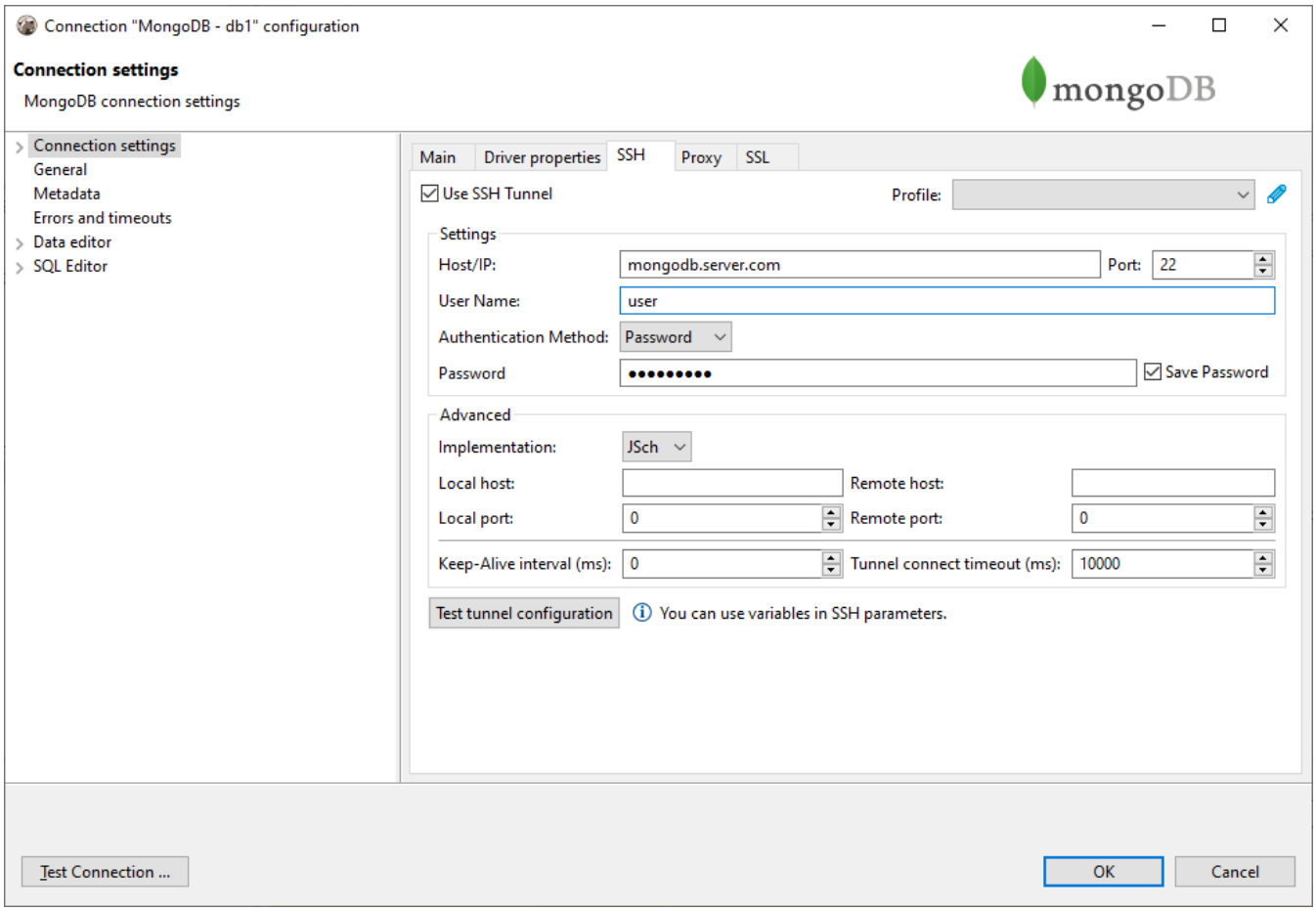
Connecting to MongoDB Server

You can connect directly to a server or use SSH tunneling or SOCKS proxy.
You can specify server address as a host/port/database configuration or you can enter the target database URL with all necessary parameters:



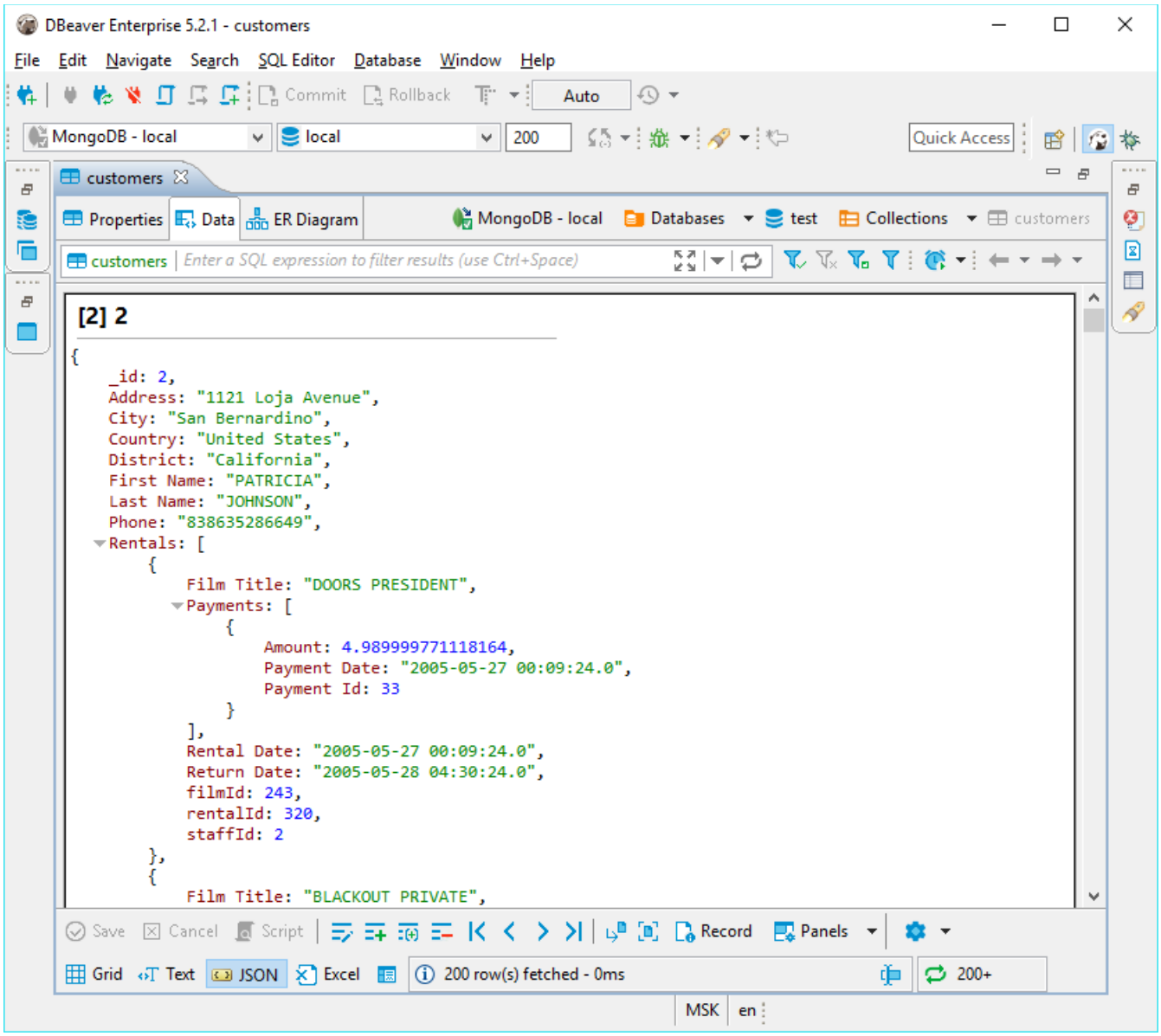

Connection "MongoDB - Atlas" configuration
Connection settings
MongoDB connection settings
mongoDB
Connection settings
General
Metadata
Errors and timeouts
Data viewer
SQL Editor
Main
Driver properties
SSH
Proxy
SSL
Address
Type: Manual URL
URL: mongodb+srv://@freecluster-asdfewq.gcp.mongodb.net/test?retryWrites=true
Do not put user name/password in the URL.
URL must look like mongodb://parameters or mongodb+srv://@parameters
Credentials
Mechanism: SCRAM-SHA-1
User: user
Source: admin
Password: ••••••••••••
Save password locally
You can use variables in connection parameters.
Driver name: MongoDB
Edit Driver Settings
Test Connection ...
OK
Cancel

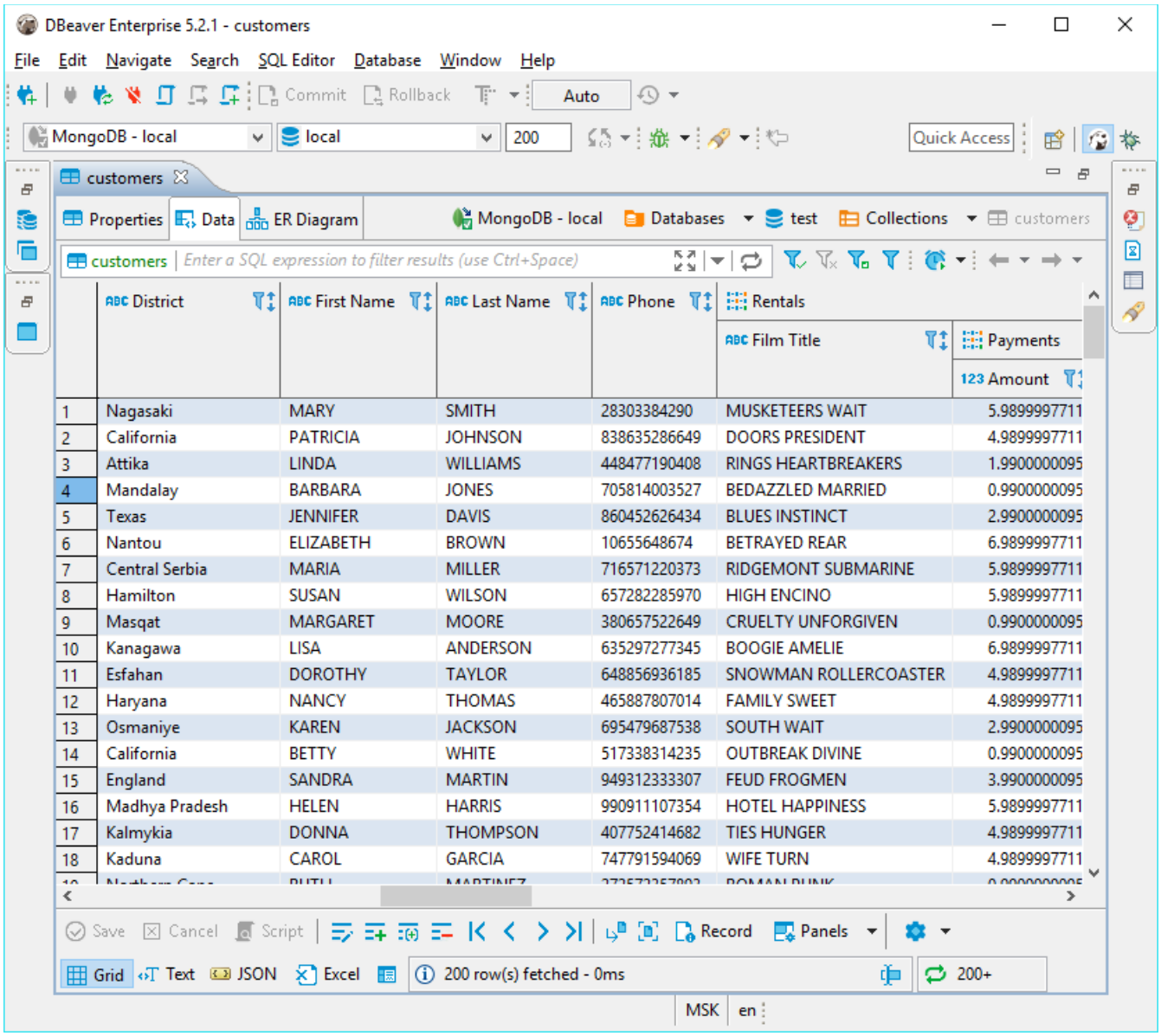

Connection "MongoDB - Atlas" configuration
Connection settings
MongoDB connection settings
mongoDB
Connection settings
General
Metadata
Errors and timeouts
Data viewer
SQL Editor
Main
Driver properties
SSH
Proxy
SSL
Name
Value
Connection
Connect Timeout
Max Connection Idle Time
Max Connection Life Time
Max Wait Time
Server Select Timeout 15,000
Single node mode
Socket Keep Alive
Socket Timeout
Heartbeat
Heartbeat Connect Timeout
Heartbeat Frequency
Heartbeat Socket Timeout
Miscellaneous
Use client-side JavaScript
User Properties
Advanced driver properties
Driver documentation
Test Connection ...
OK
Cancel

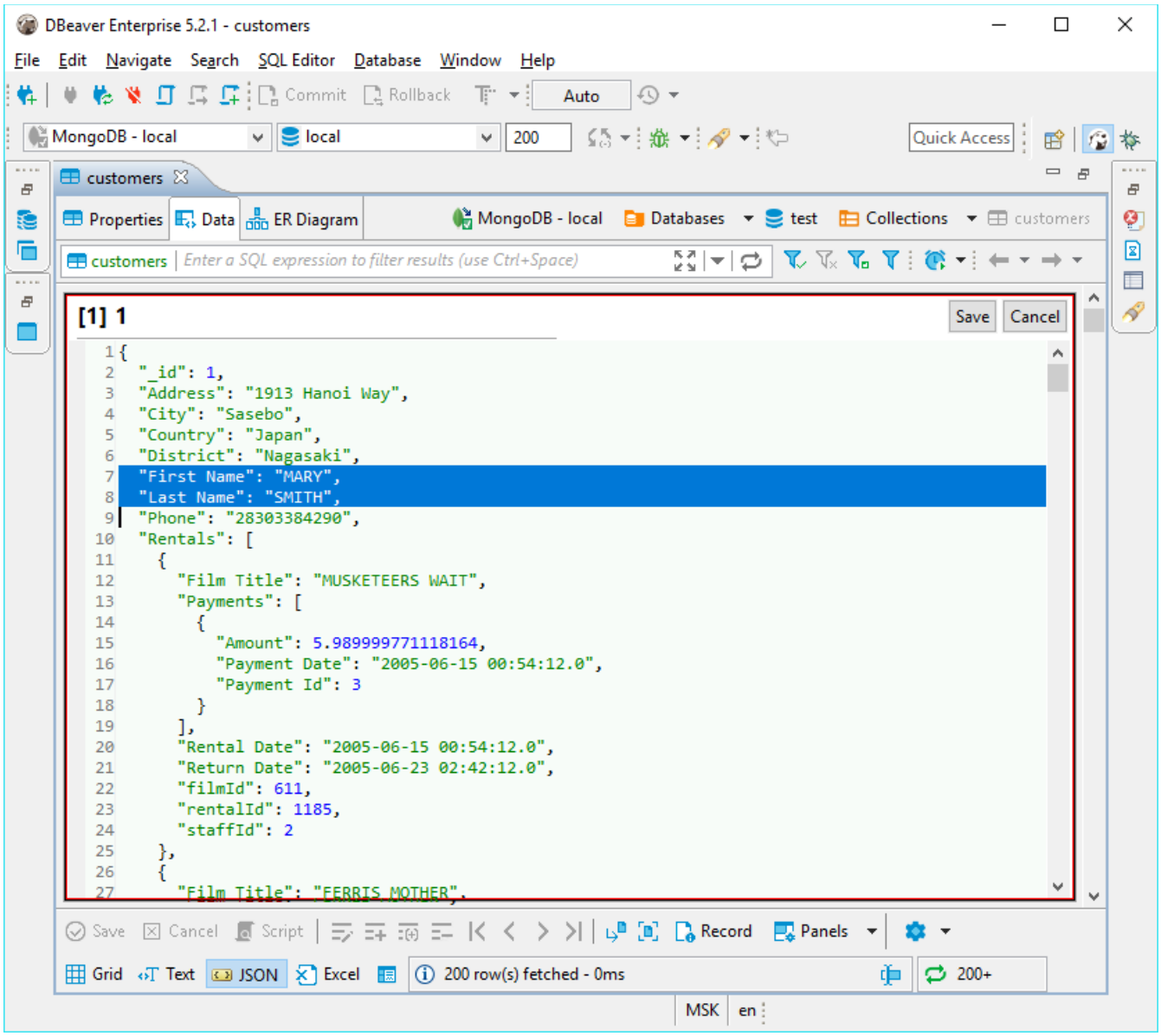


Browsing Mongo collections

You can view/edit MongoDB collections content as standard relational tables (grid/plain text presentations) or as JSON documents. The presentation can be switched in the Results Viewer toolbar.
In a grid, DBeaver will try to unify all documents in some particular collection (as they have the same structure/the same set of properties).


DBeaver Enterprise 5.2.1 - customers
File Edit Navigate Search SQL Editor Database Window Help
Commit Rollback Auto
MongoDB - local local 200 Quick Access
customers
Properties Data ER Diagram MongoDB - local Databases test Collections customers
customers Enter a SQL expression to filter results (use Ctrl+Space)
[2] 2
{
_id: 2,
Address: "1121 Loja Avenue",
City: "San Bernardino",
Country: "United States",
District: "California",
First Name: "PATRICIA",
Last Name: "JOHNSON",
Phone: "838635286649",
Rentals: [
{
Film Title: "DOORS PRESIDENT",
Payments: [
{
Amount: 4.989999771118164,
Payment Date: "2005-05-27 00:09:24.0",
Payment Id: 33
}
],
Rental Date: "2005-05-27 00:09:24.0",
Return Date: "2005-05-28 04:30:24.0",
filmId: 243,
rentalId: 320,
staffId: 2
},
{
Film Title: "BLACKOUT PRIVATE",
Save Cancel Script Record Panels
Grid Text JSON Excel 200 row(s) fetched - 0ms 200+
MSK en


DBeaver Enterprise 5.2.1 - customers
File Edit Navigate Search SQL Editor Database Window Help
Commit Rollback Auto
MongoDB - local local 200 Quick Access
customers
Properties Data ER Diagram MongoDB - local Databases test Collections customers
customers Enter a SQL expression to filter results (use Ctrl+Space)
District First Name Last Name Phone Rentals
Film Title Payments
Amount
1 Nagasaki MARY SMITH 28303384290 MUSKETEERS WAIT 5.9899997711
2 California PATRICIA JOHNSON 838635286649 DOORS PRESIDENT 4.9899997711
3 Attika LINDA WILLIAMS 448477190408 RINGS HEARTBREAKERS 1.9900000095
4 Mandalay BARBARA JONES 705814003527 BEDAZZLED MARRIED 0.9900000095
5 Texas JENNIFER DAVIS 860452626434 BLUES INSTINCT 2.9900000095
6 Nantou ELIZABETH BROWN 10655648674 BETRAYED REAR 6.9899997711
7 Central Serbia MARIA MILLER 716571220373 RIDGEMONT SUBMARINE 5.9899997711
8 Hamilton SUSAN WILSON 657282285970 HIGH ENCINO 5.9899997711
9 Masqat MARGARET MOORE 380657522649 CRUELTY UNFORGIVEN 0.9900000095
10 Kanagawa LISA ANDERSON 635297277345 BOOGIE AMELIE 6.9899997711
11 Esfahan DOROTHY TAYLOR 648856936185 SNOWMAN ROLLERCOASTER 4.9899997711
12 Haryana NANCY THOMAS 465887807014 FAMILY SWEET 4.9899997711
13 Osmaniye KAREN JACKSON 695479687538 SOUTH WAIT 2.9900000095
14 California BETTY WHITE 517338314235 OUTBREAK DIVINE 0.9900000095
15 England SANDRA MARTIN 949312333307 FEUD FROGMEN 3.9900000095
16 Madhya Pradesh HELEN HARRIS 990911107354 HOTEL HAPPINESS 5.9899997711
17 Kalmykia DONNA THOMPSON 407752414682 TIES HUNGER 4.9899997711
18 Kaduna CAROL GARCIA 747791594069 WIFE TURN 4.9899997711
Save Cancel Script Record Panels
Grid Text JSON Excel 200 row(s) fetched - 0ms 200+
MSK en



Executing JavaScript

JS statements can be executed in the SQL editor as usual. DBeaver supports all JS queries for MongoDB 2 and 3 as well as a subset of the `mongo` shell queries.

The following example creates a user in the current database.

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

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

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

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

Executing SQL

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

SELECT queries support `WHERE` , `ORDER BY` , `GROUP BY` , `JOIN` and `HAVING` clauses.

```
SELECT * FROM test_col
WHERE propName.subProp='value'

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

Conditions

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

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

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

Nested fields

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

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

Working with dates

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

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

Querying data in JavaScript:

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

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

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

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

Working with object IDs

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

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

Working with JOINs

Currently, SQL dialect for MongoDB supports `LEFT JOIN` and `INNER JOIN`:

```
SELECT
    ar.Name as Artist,
    al.Title as Album,
    SUM(tr.Milliseconds) as Duration
FROM Track tr
INNER JOIN Album al ON tr.AlbumId = al.AlbumId
INNER JOIN Artist ar ON al.ArtistId = ar.ArtistId
GROUP BY Artist, Album
ORDER BY Duration DESC
```

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

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

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

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

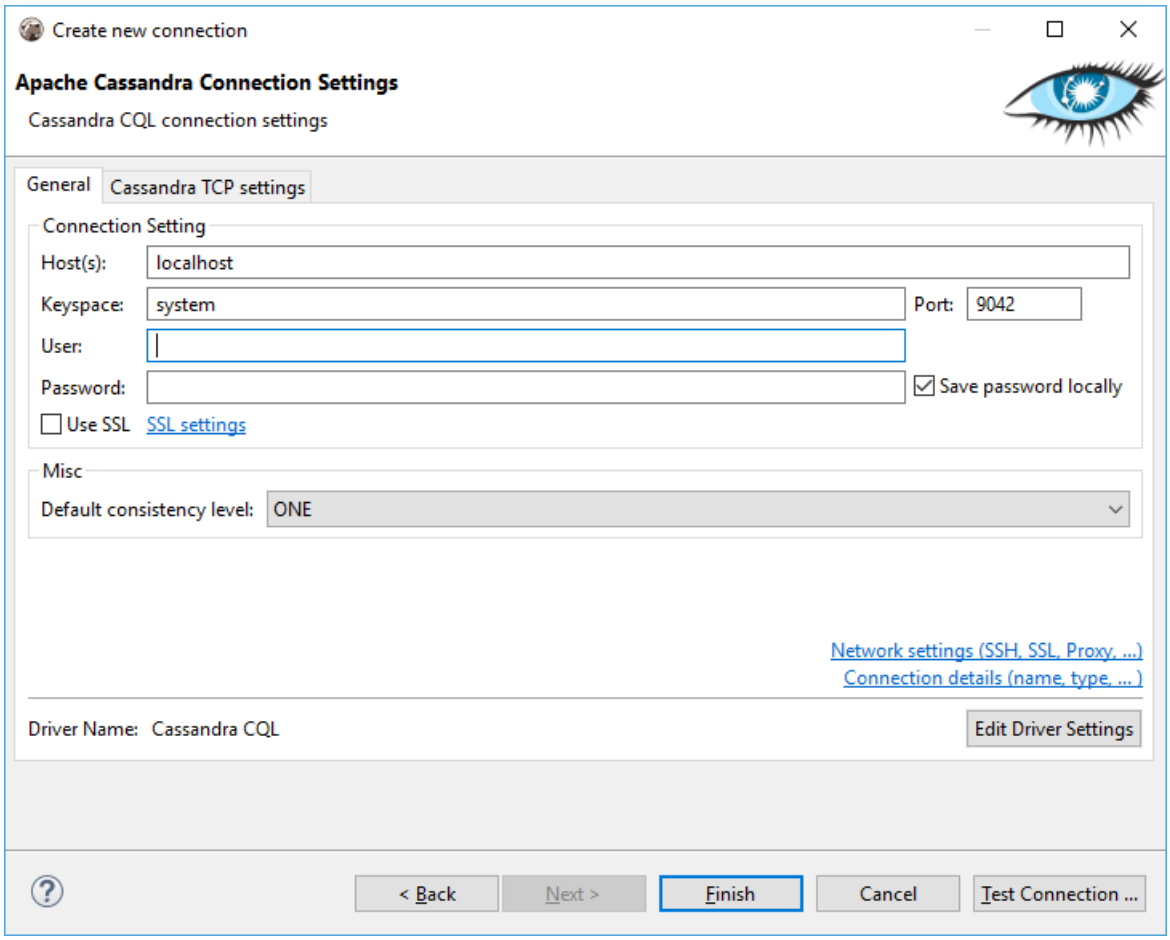
Cassandra

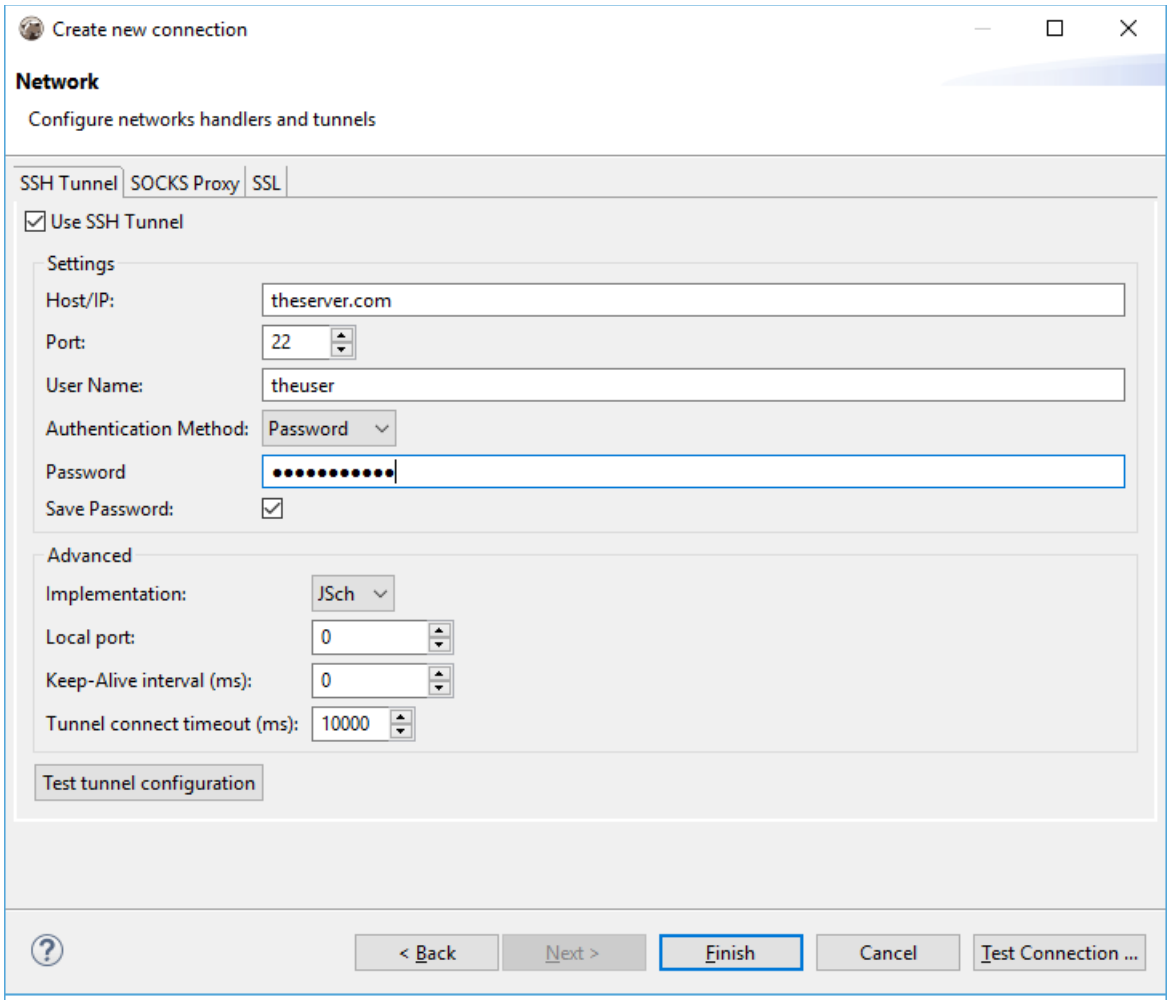
Overview

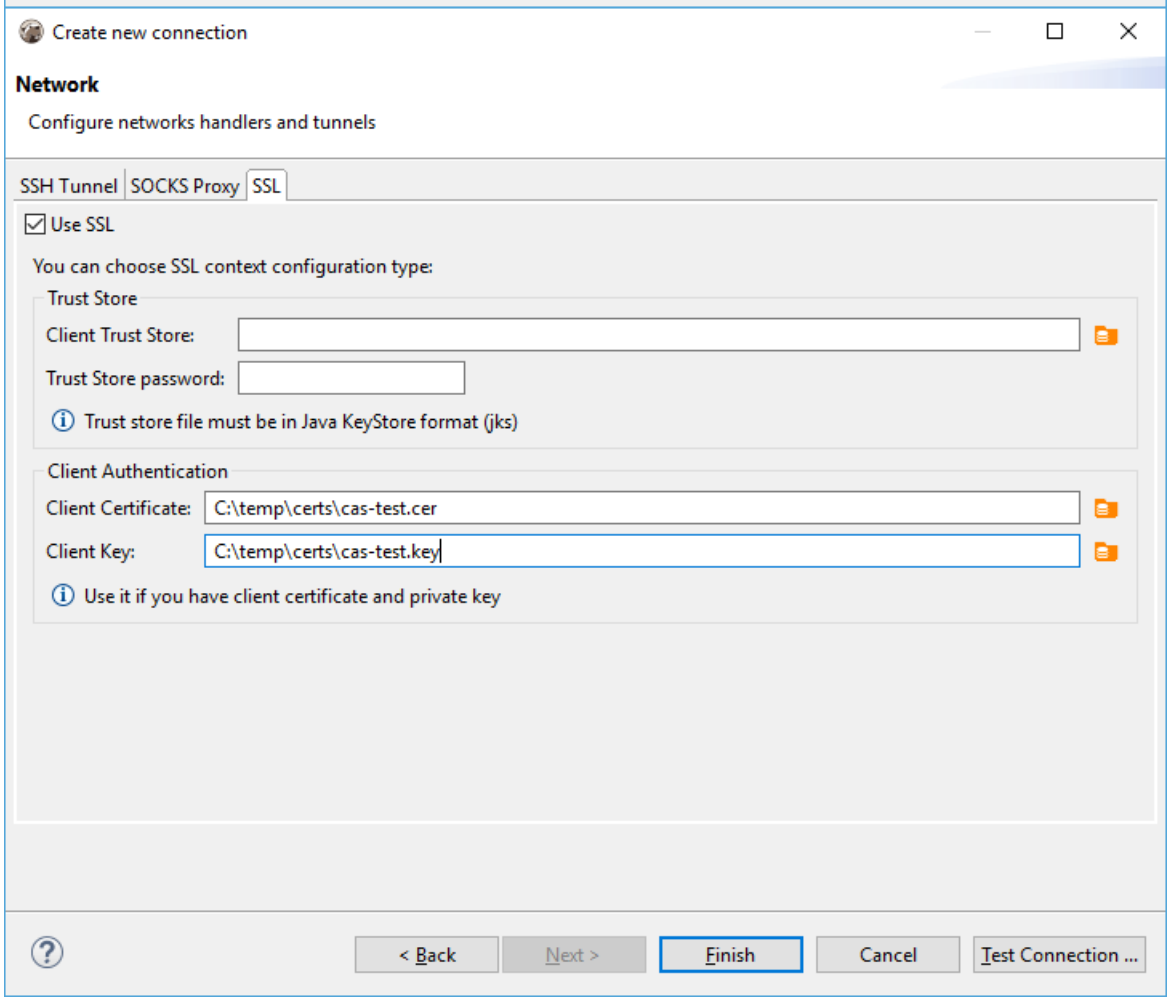
DBeaver EE supports Cassandra schema browser, data viewer and CQL queries execution. It also supports various administrative tools.

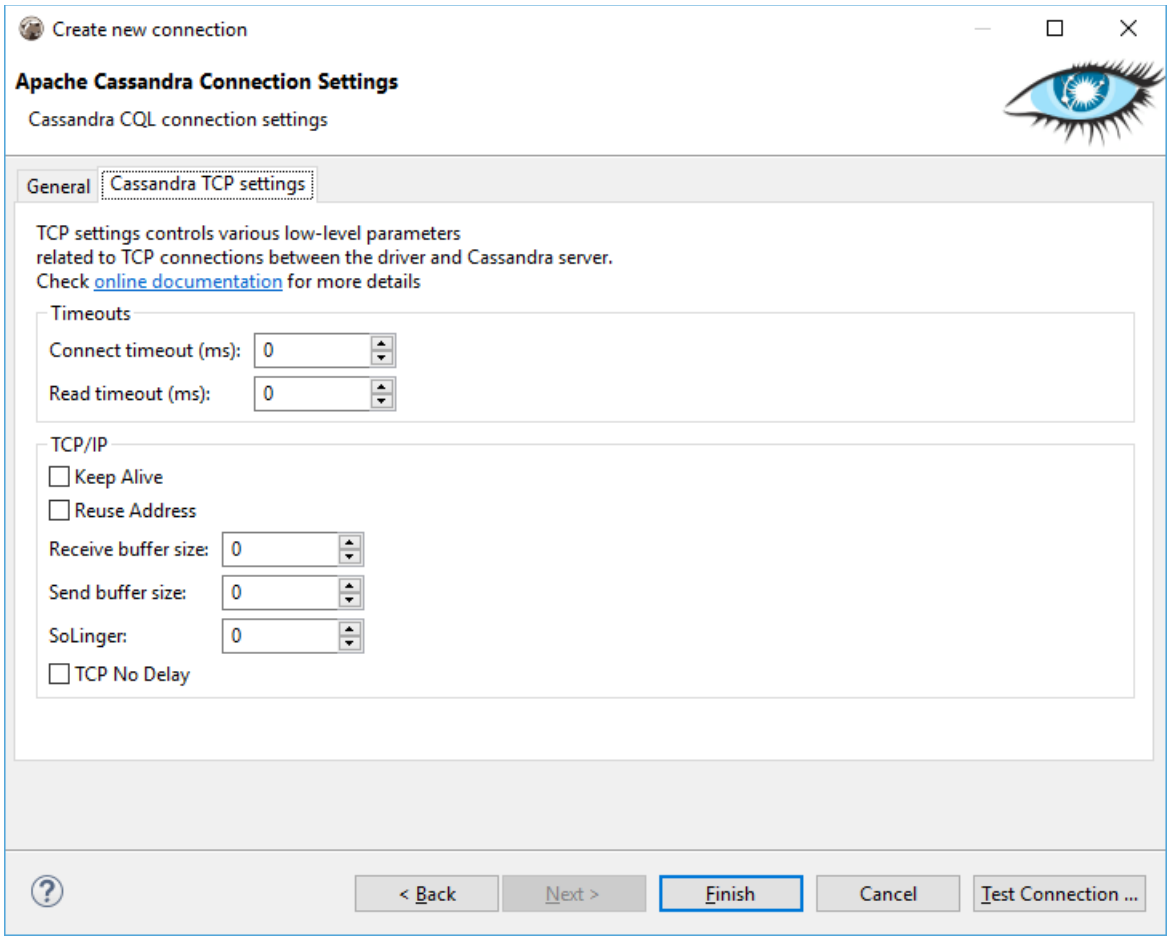
Connecting to Cassandra cluster

You can connect directly to a server or use SSH tunneling or SOCKS proxy. DBeaver uses the DataStax Java driver to operate with a server. It supports Cassandra servers 2.x, 3.x or higher.



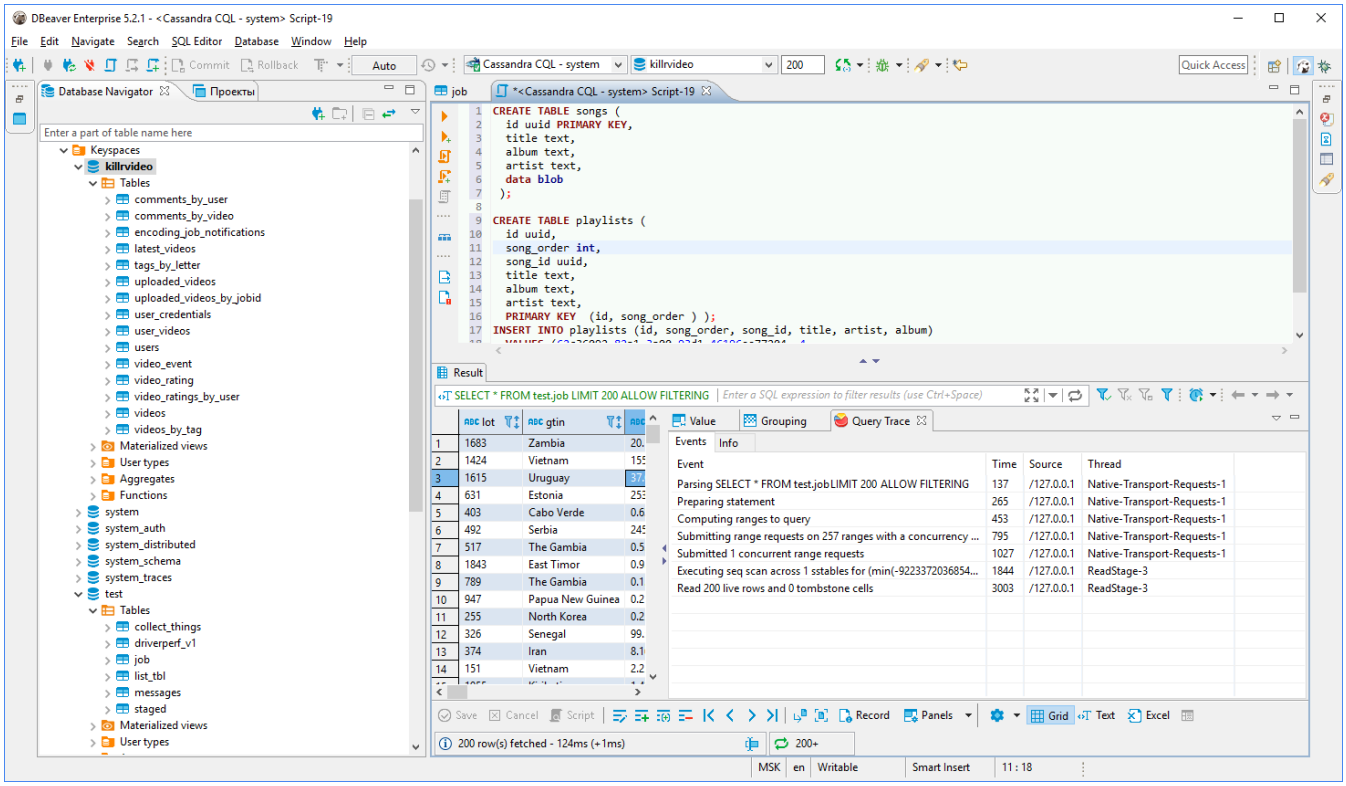

Create new connection
Network
Configure networks handlers and tunnels
SSH Tunnel
SOCKS Proxy
SSL
Use SSH Tunnel
Settings
Host/IP:
theserver.com
Port:
22
User Name:
theuser
Authentication Method:
Password
Password
Save Password:
Advanced
Implementation:
JSch
Local port:
0
Keep-Alive interval (ms):
0
Tunnel connect timeout (ms):
10000
Test tunnel configuration
< Back
Next >
Finish
Cancel
Test Connection ...

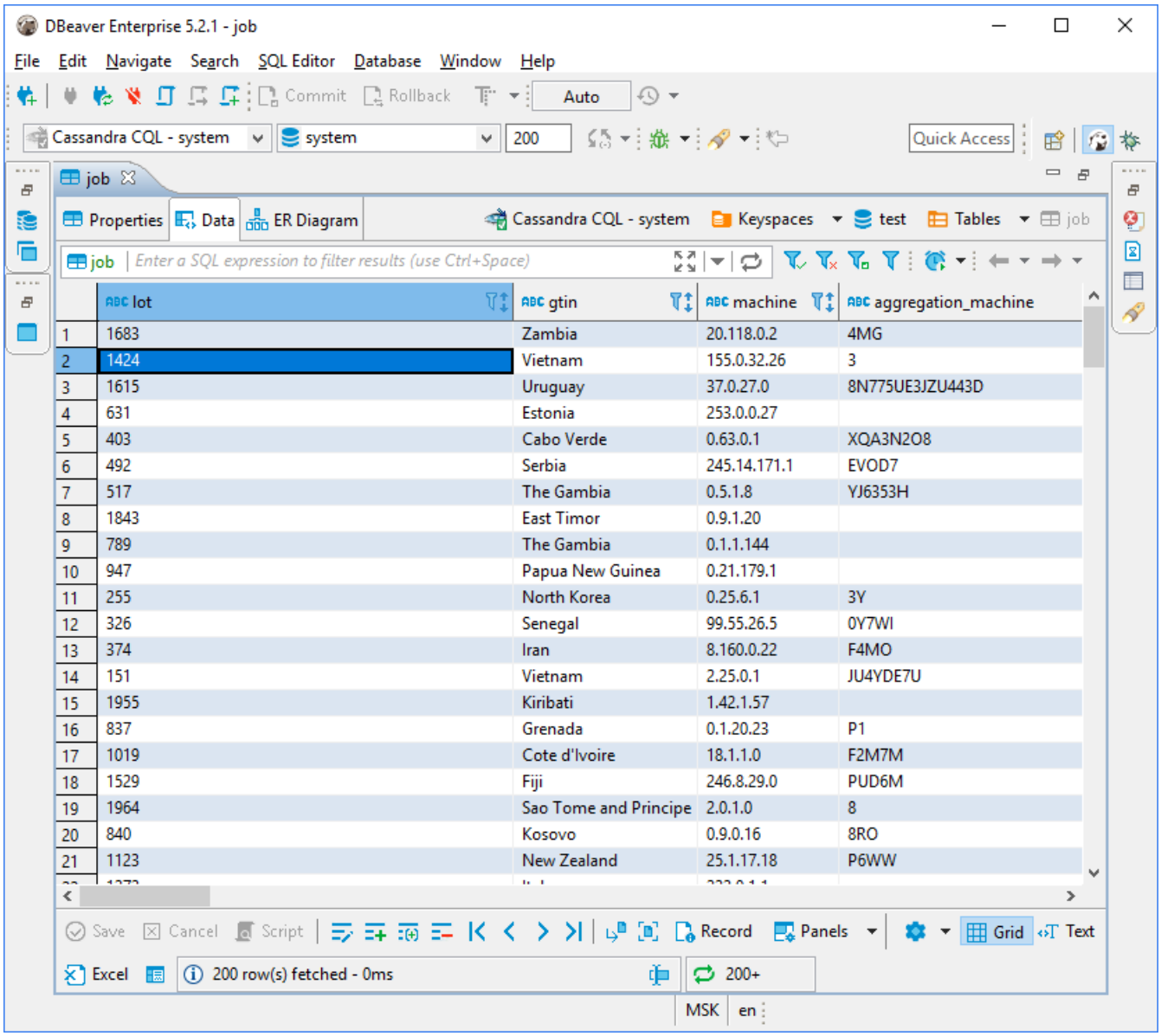

Create new connection
Network
Configure networks handlers and tunnels
SSH Tunnel
SOCKS Proxy
SSL
Use SSL
You can choose SSL context configuration type:
Trust Store
Client Trust Store:
Trust Store password:
Trust store file must be in Java KeyStore format (jks)
Client Authentication
Client Certificate:
C:\temp\certs\cas-test.cer
Client Key:
C:\temp\certs\cas-test.key
Use it if you have client certificate and private key
< Back
Next >
Finish
Cancel
Test Connection ...



Browsing Cassandra tables

You can browse, view, edit and filter Cassandra tables the same way as with regular (relational) tables. However, being a distributed key-value database, Cassandra does not support any kind of referential integrity. There are no foreign keys, references, etc.
You should note that Cassandra has a very advanced (comparing to relational databases) data type system. Each column may be a collection, map, or set of values (with very big number of values). In some cases this makes browsing data in the "Grid" mode inconvenient.

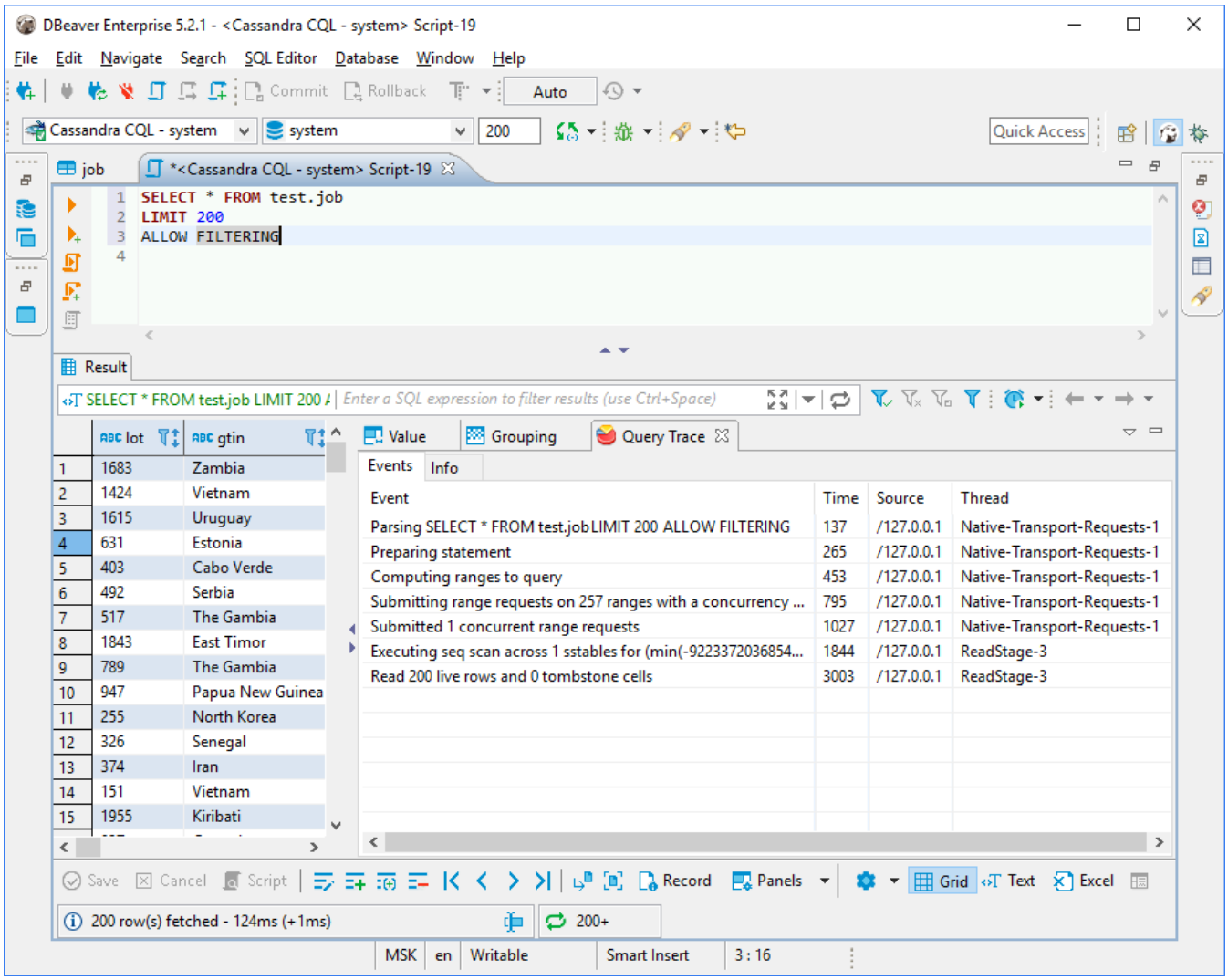




Executing CQL

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

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



ERD

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

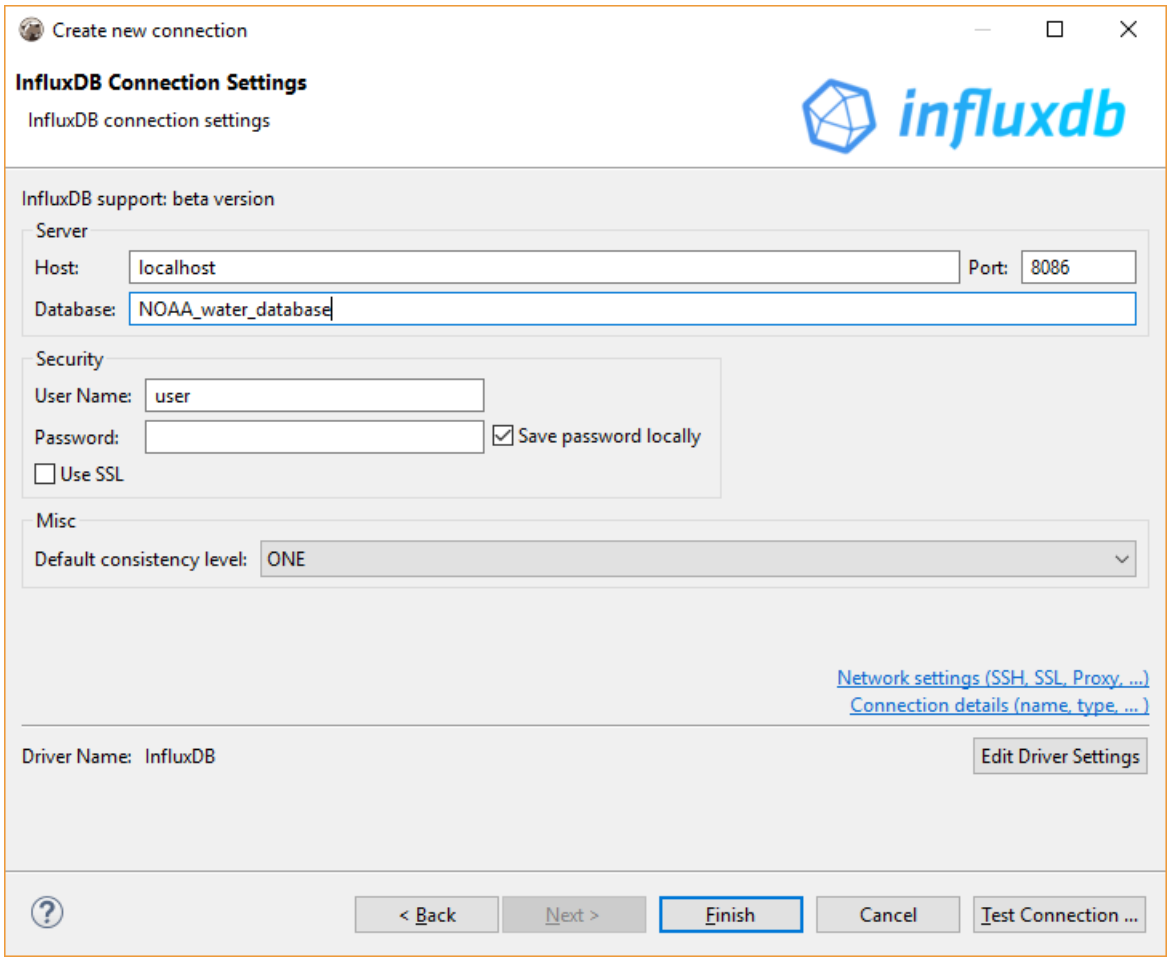
InfluxDB

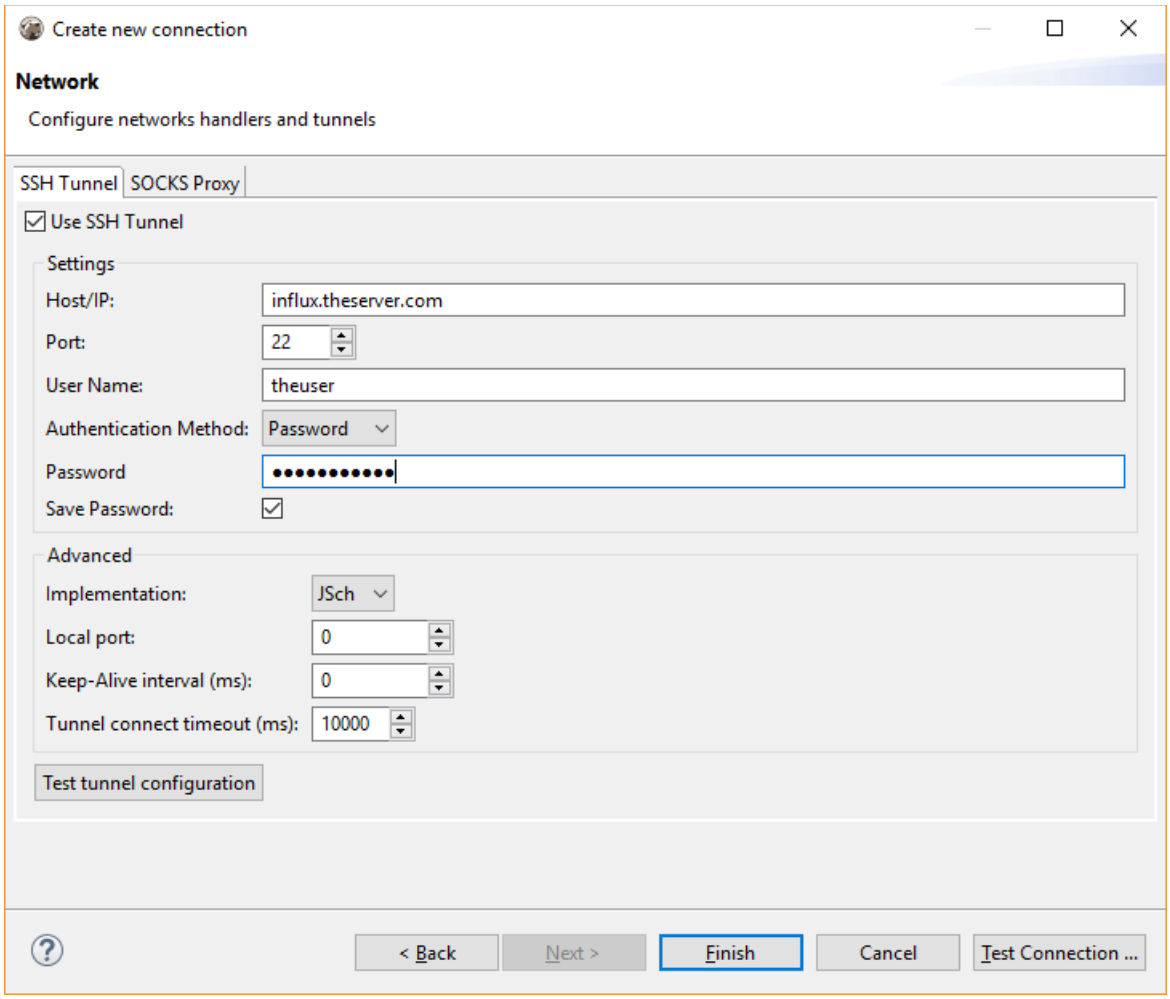
Overview

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

Connecting to Influx Server

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

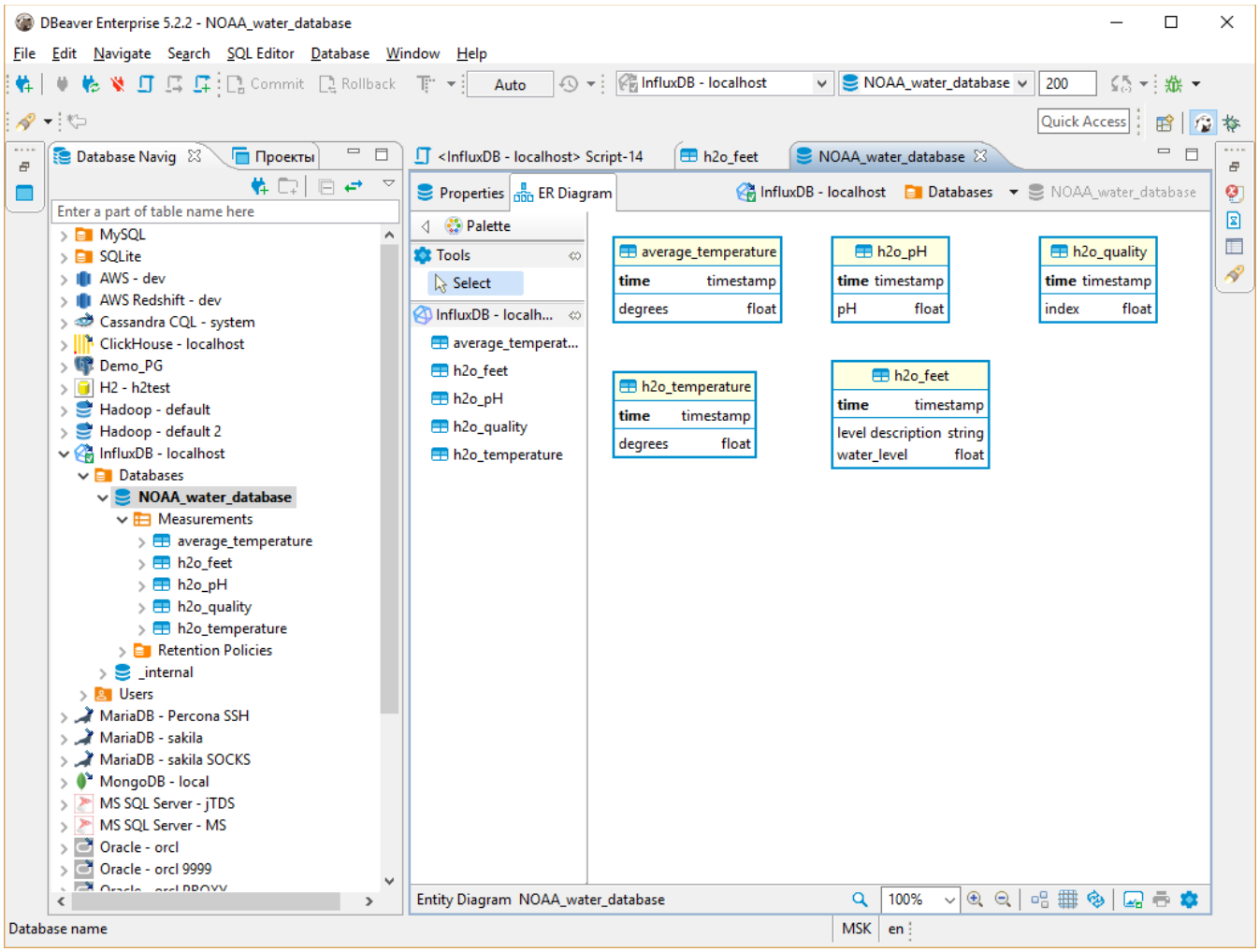




Browsing InfluxDB schema

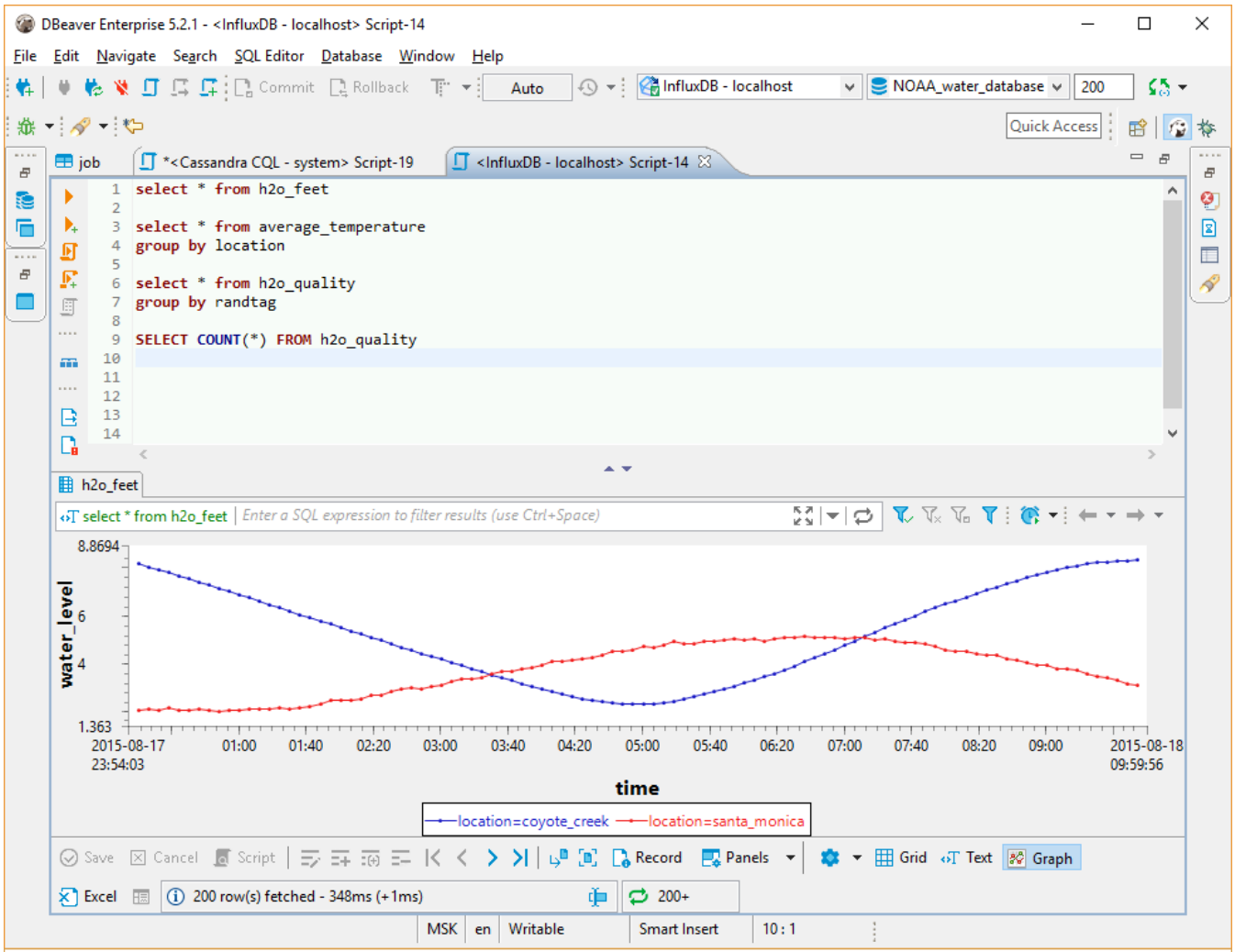
InfluxDB is TimeSeries database, it does not support tables, foreign keys and other relational entities.

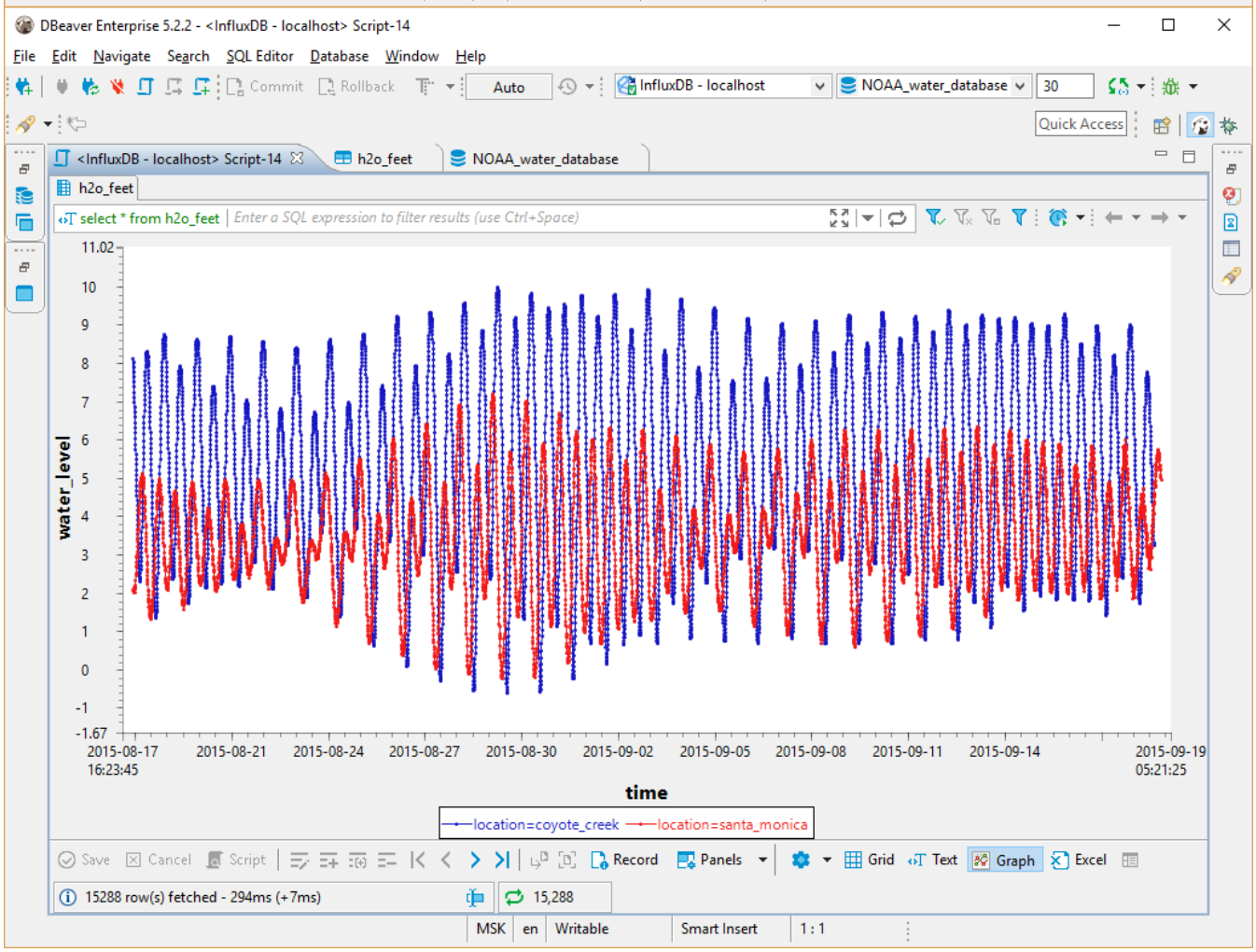
DBeaver does not support data insert/update in InfluxDB. Database is basically a in read-only state for DBeaver. You can browse schema and view/analyse data.
While data itself is loaded by various sensors/data collectors in real time.
Instead of tables InfluxDB has measurements. Instead of columns it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



Executing InfluxQL

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


DBeaver Enterprise 5.2.1 - <InfluxDB - localhost> Script-14
File Edit Navigate Search SQL Editor Database Window Help
Commit Rollback Auto InfluxDB - localhost NOAA_water_database 200
Quick Access
job *<Cassandra CQL - system> Script-19 <InfluxDB - localhost> Script-14
1 select * from h2o_feet
3 select * from average_temperature
4 group by location
6 select * from h2o_quality
7 group by randtag
9 SELECT COUNT(*) FROM h2o_quality
h2o_feet
select * from h2o_feet Enter a SQL expression to filter results (use Ctrl+Space)
8.8694
6
4
1.363
water_level
2015-08-17 23:54:03
01:00
01:40
02:20
03:00
03:40
04:20
05:00
05:40
06:20
07:00
07:40
08:20
09:00
2015-08-18 09:59:56
time
location=coyote_creek
location=santa_monica
Save Cancel Script Record Panels Grid Text Graph
Excel 200 row(s) fetched - 348ms (+1ms) 200+
MSK en Writable Smart Insert 10 : 1


DBeaver Enterprise 5.2.2 - <InfluxDB - localhost> Script-14
File Edit Navigate Search SQL Editor Database Window Help
Commit Rollback Auto InfluxDB - localhost NOAA_water_database 30
Quick Access
<InfluxDB - localhost> Script-14 h2o_feet NOAA_water_database
h2o_feet
select * from h2o_feet Enter a SQL expression to filter results (use Ctrl+Space)
11.02
10
9
8
7
6
5
4
3
2
1
0
-1
-1.67
water_level
2015-08-17 16:23:45
2015-08-21
2015-08-24
2015-08-27
2015-08-30
2015-09-02
2015-09-05
2015-09-08
2015-09-11
2015-09-14
2015-09-19 05:21:25
time
location=coyote_creek
location=santa_monica
Save Cancel Script Record Panels Grid Text Graph Excel
15288 row(s) fetched - 294ms (+7ms) 15,288
MSK en Writable Smart Insert 1 : 1

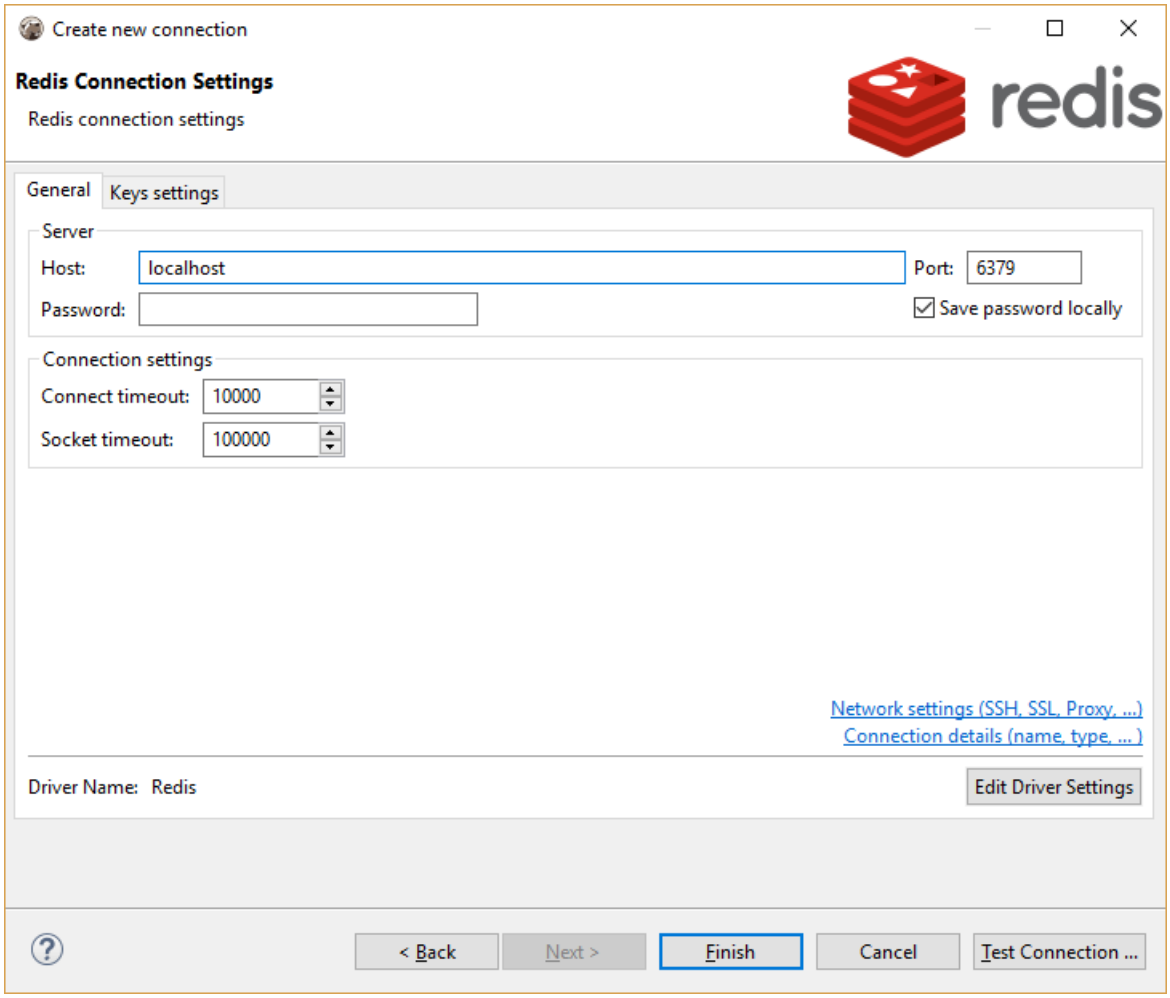
Redis

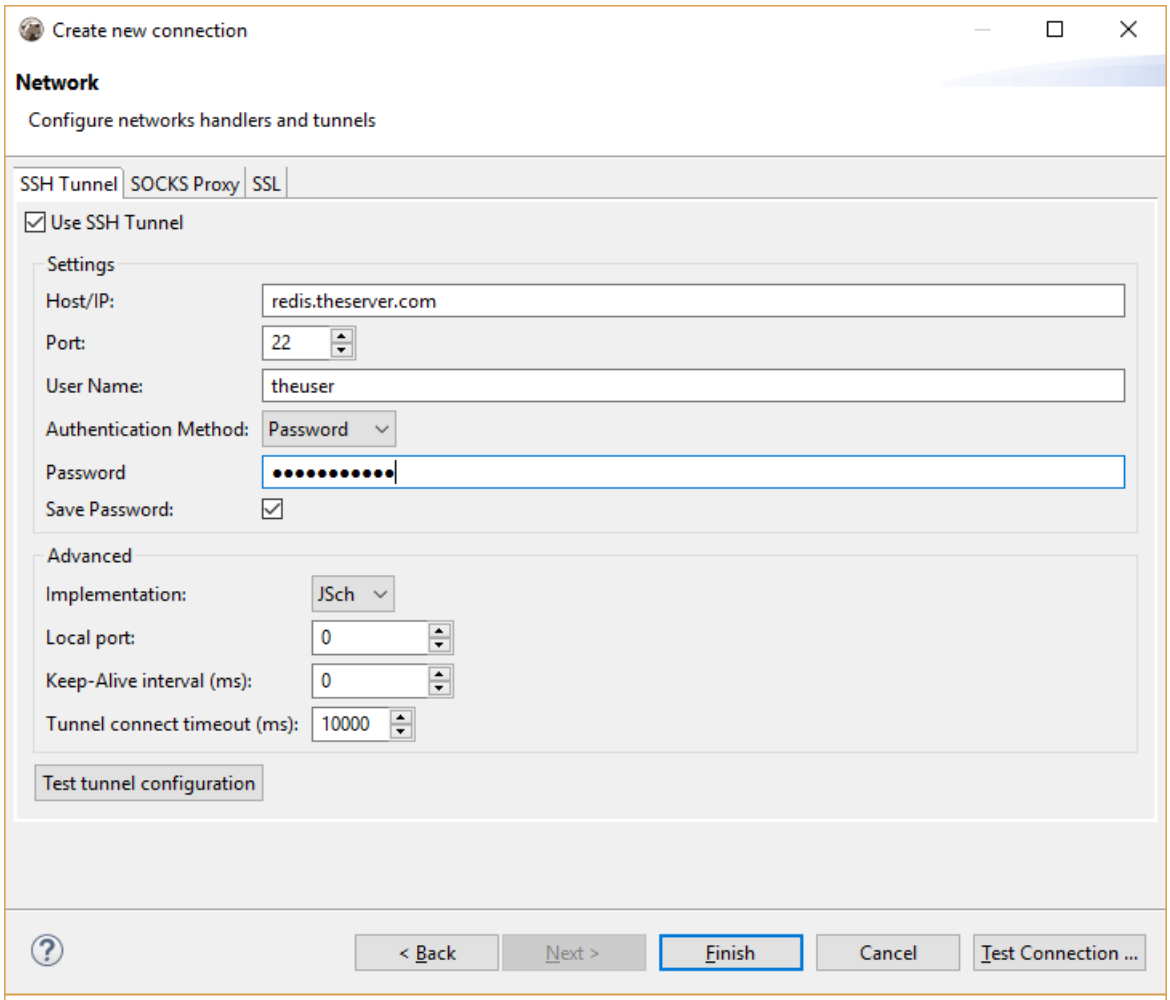
Overview

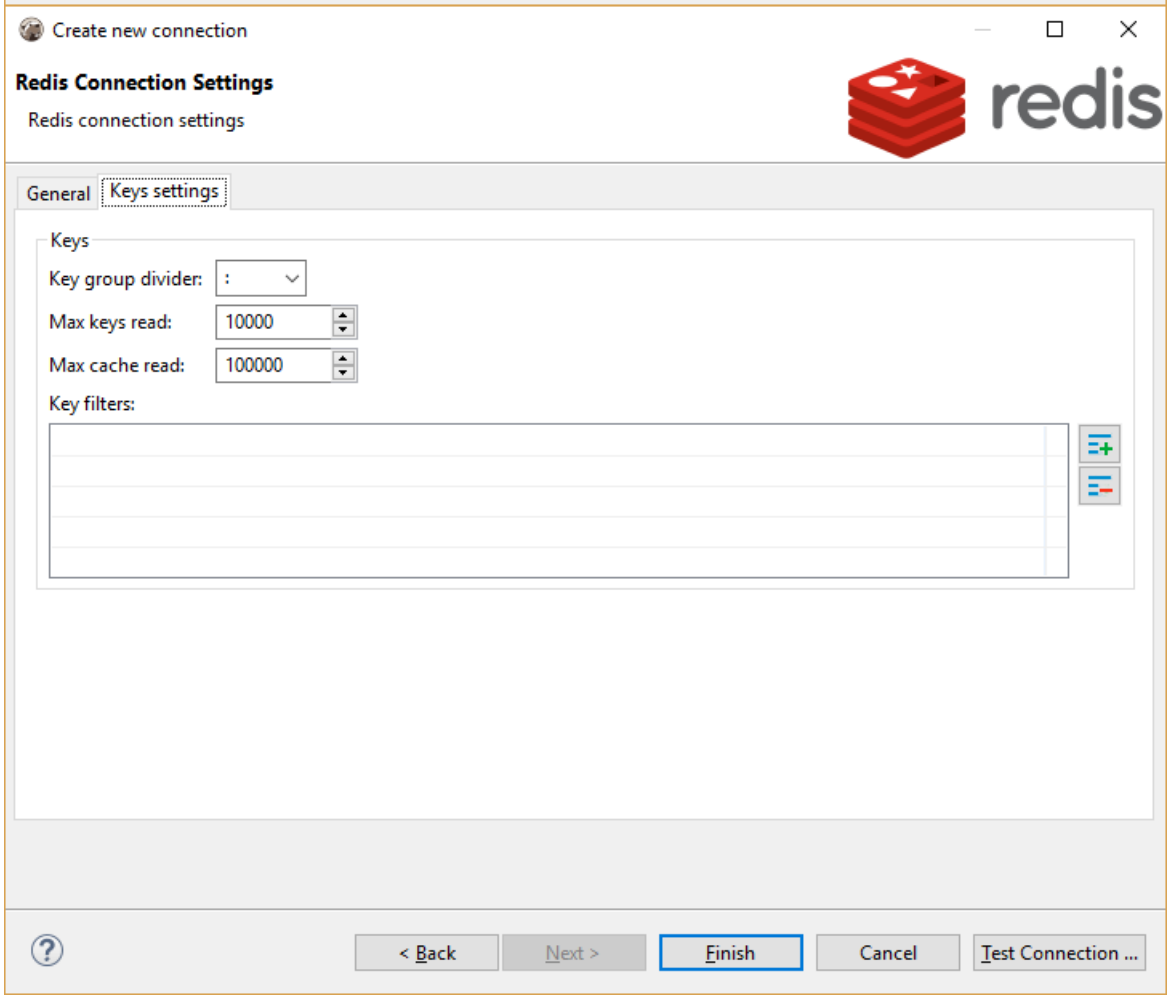
DBeaver EE supports Redis key browser, key value viewer and Redis commands shell.
DBeaver uses Jedis driver 2.9.0 to operate with Redis server. It supports Redis servers of any version.

Connecting to Redis Server

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





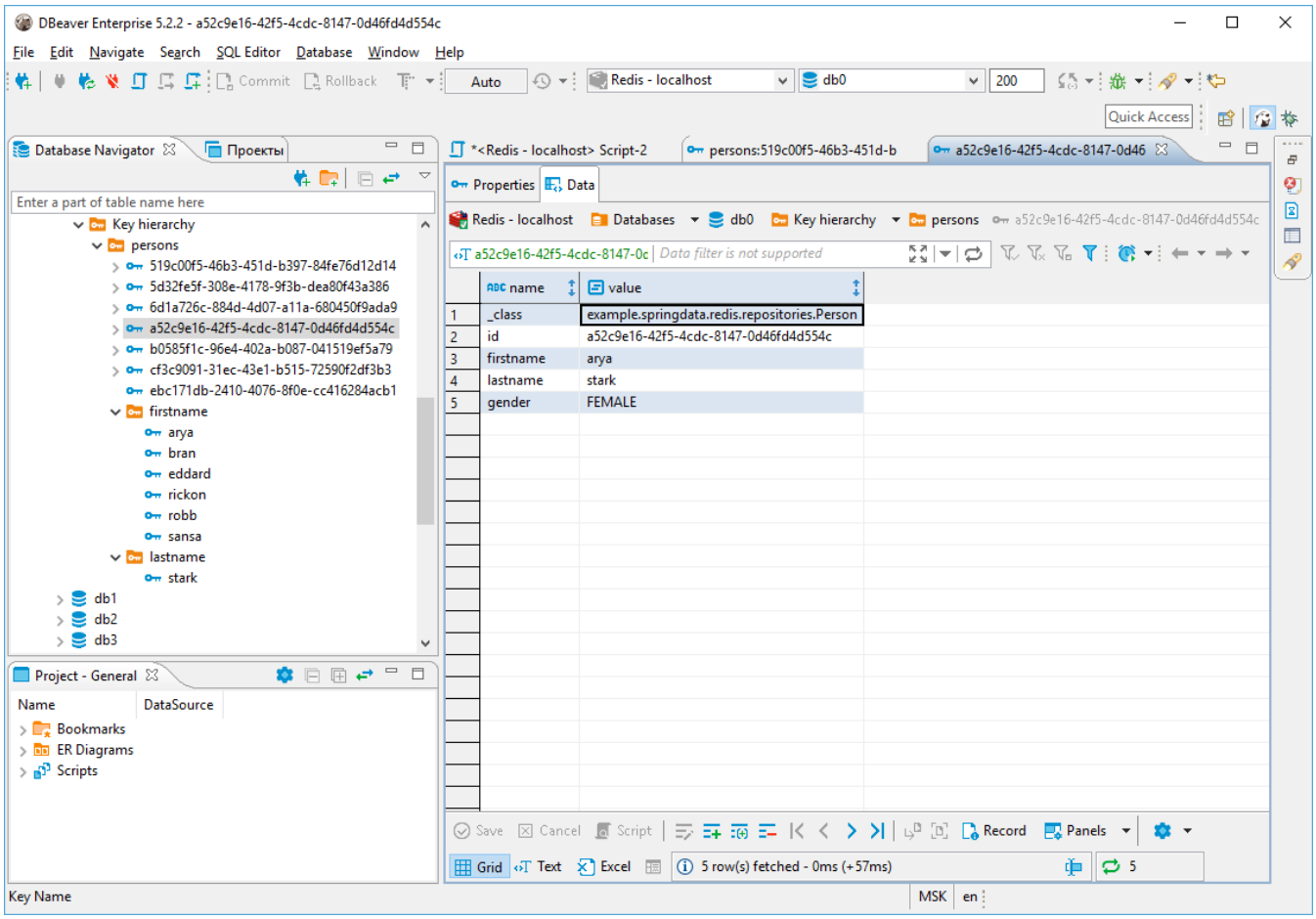


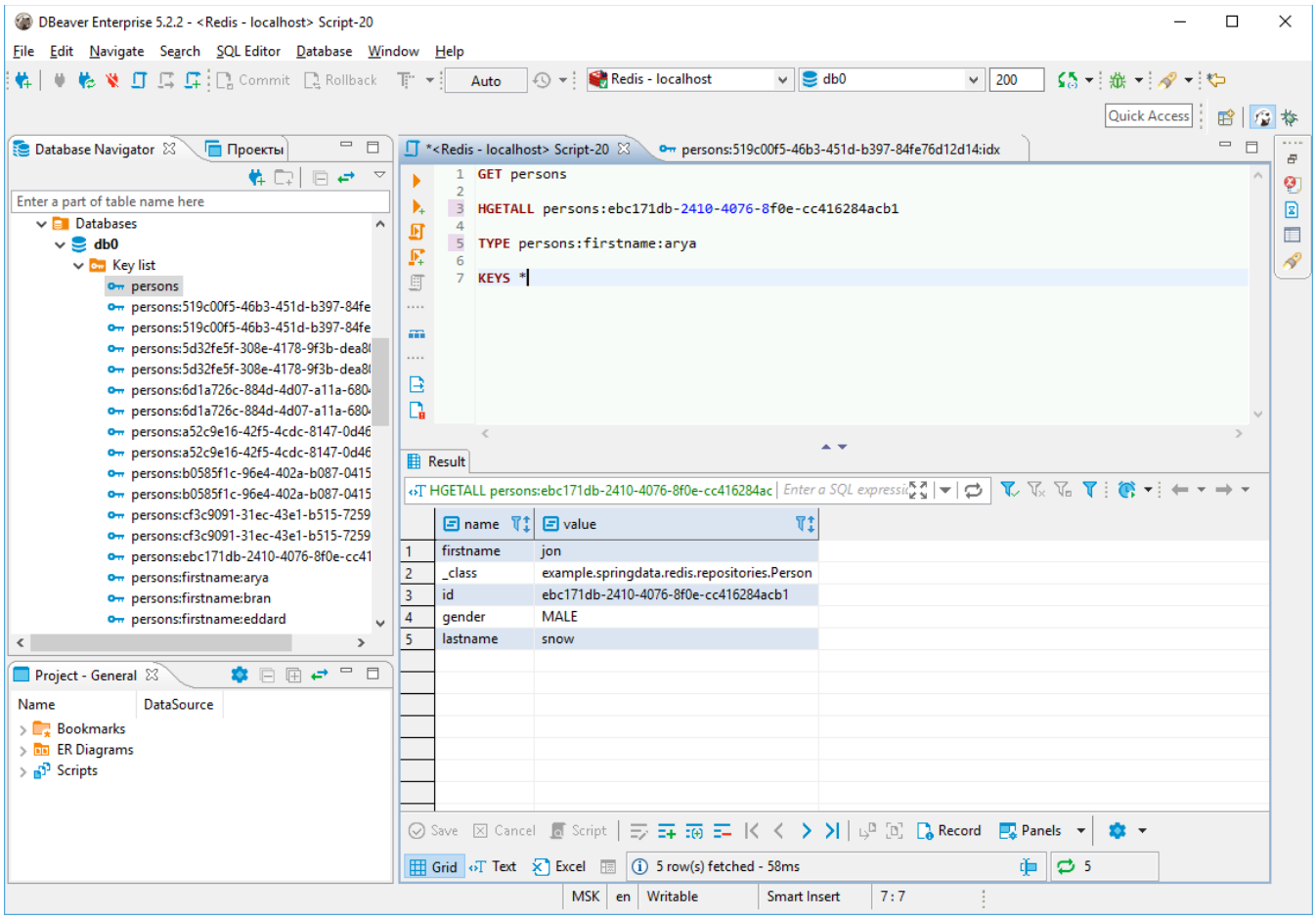
Browsing Redis keys

You can view/edit Redis keys as a plain list. However the Redis database usually contains a lot of keys (millions or even billions) and

using list presentation is not convenient (or it is not possible).
DBeaver supports a hierarchy presentation of keys. Internally Redis does not support hierarchies but application level key names may be divided into groups using a character (e.g. coma, dash or colon). DBeaver uses this pattern to show hierarchy. Group separator can be configured in connection properties.

Key browser may be convenient in some cases but in the case of big databases it is very difficult to find your key in the navigator, so the SQL editor should be used instead. Redis commands is the most flexible way to operate with keys.





Executing Redis commands

Redis does not support SQL or any other query language. Instead, it supports build-in commands and LUA scripts.

Redis commands can be executed in the same way as in a Redis command line shell: `COMMAND ARG1 ARG2 ... ARGN`

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

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

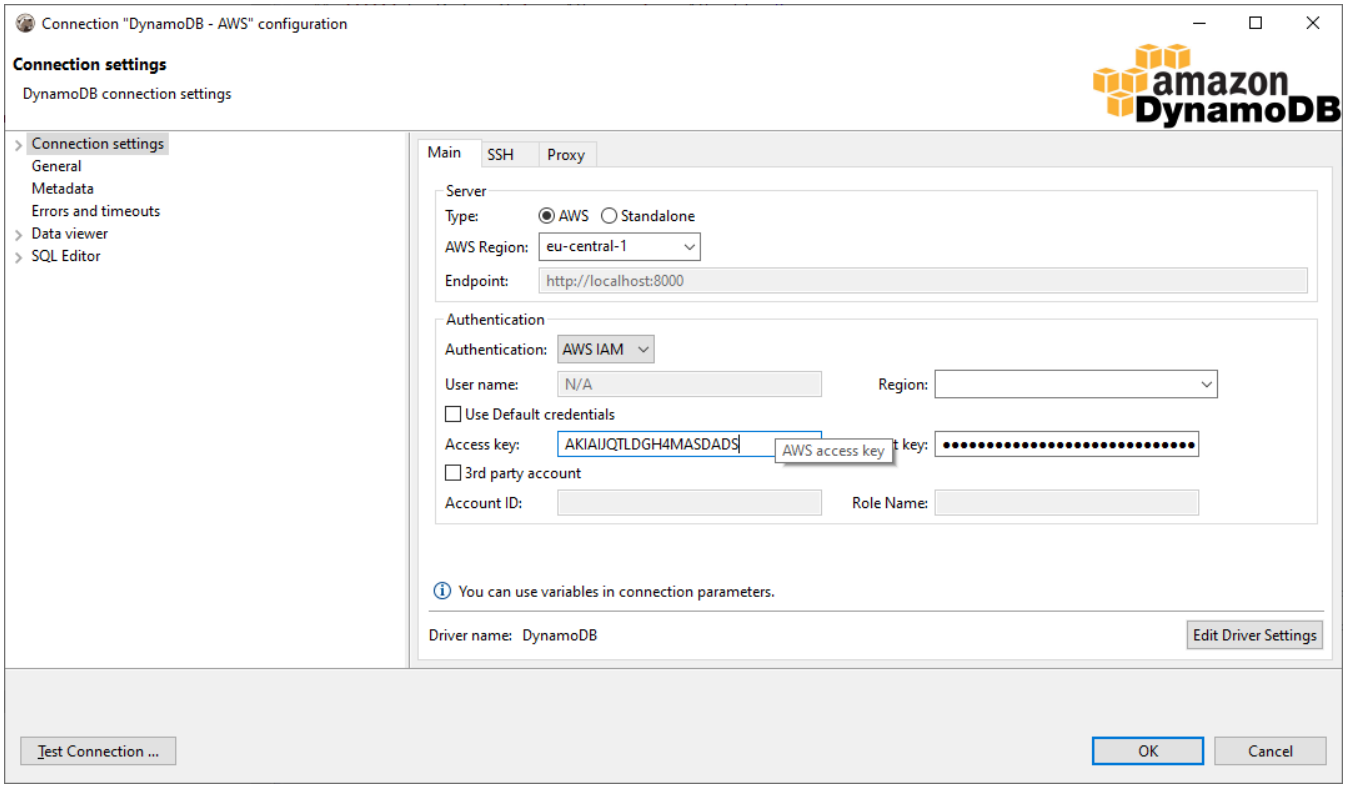
```
{
    return {1,2,{3,'Hello World!'}}
}
```

DynamoDB

Supported features:

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

DynamoDB connection



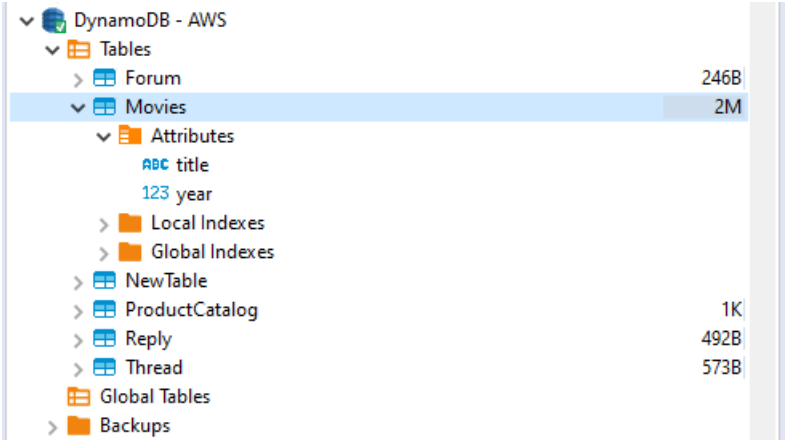
DBeaver supports AWS Cloud and Standalone versions of DynamoDB.
For standalone server you need to enter endpoint (http or https URL).
For cloud server you must enter the AWS region. DynamoDB exists in all available regions in your AWS account but the tables are different.

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

Press "Test Connection" to validate your connection settings.

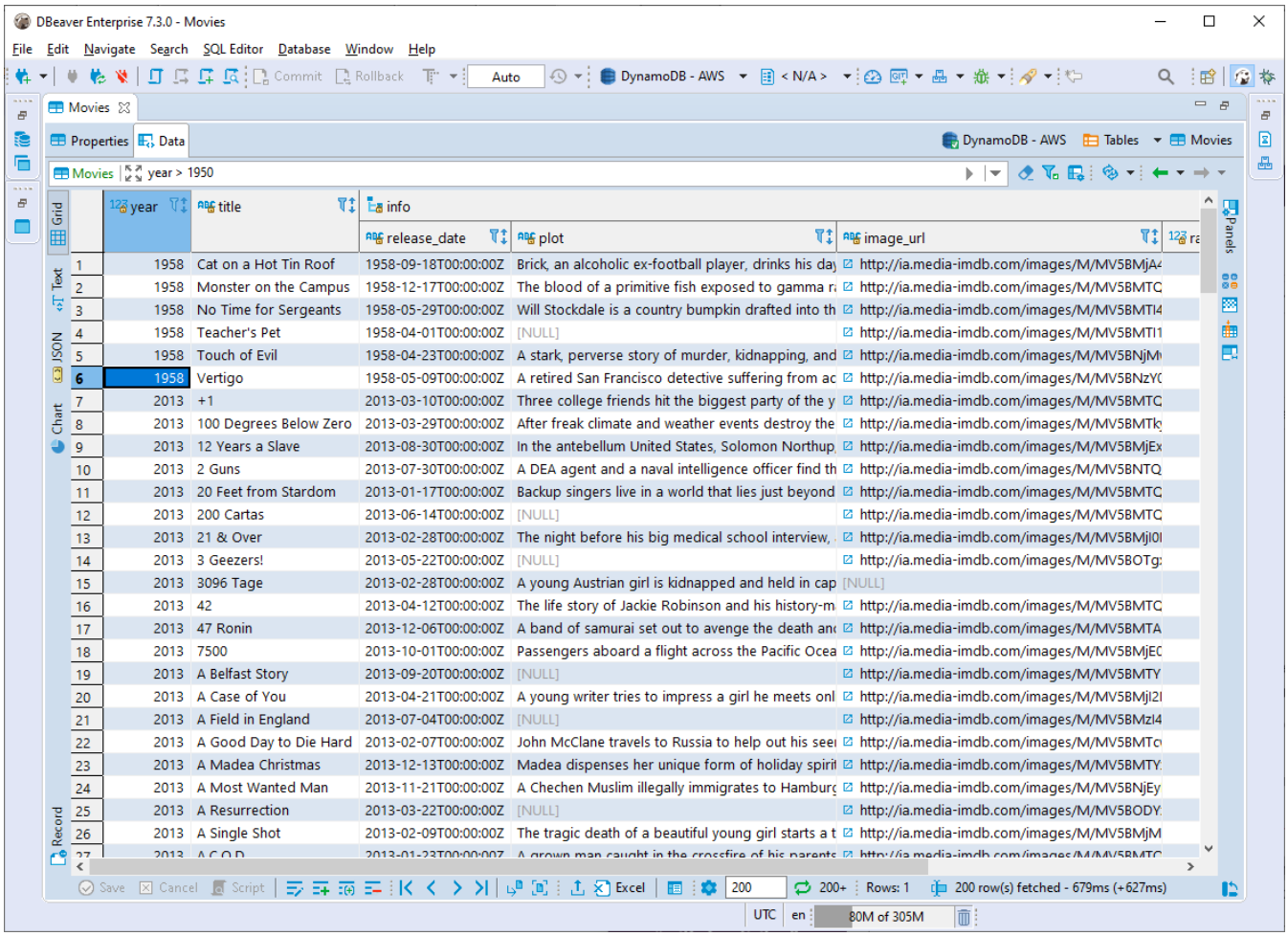
Database navigation

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



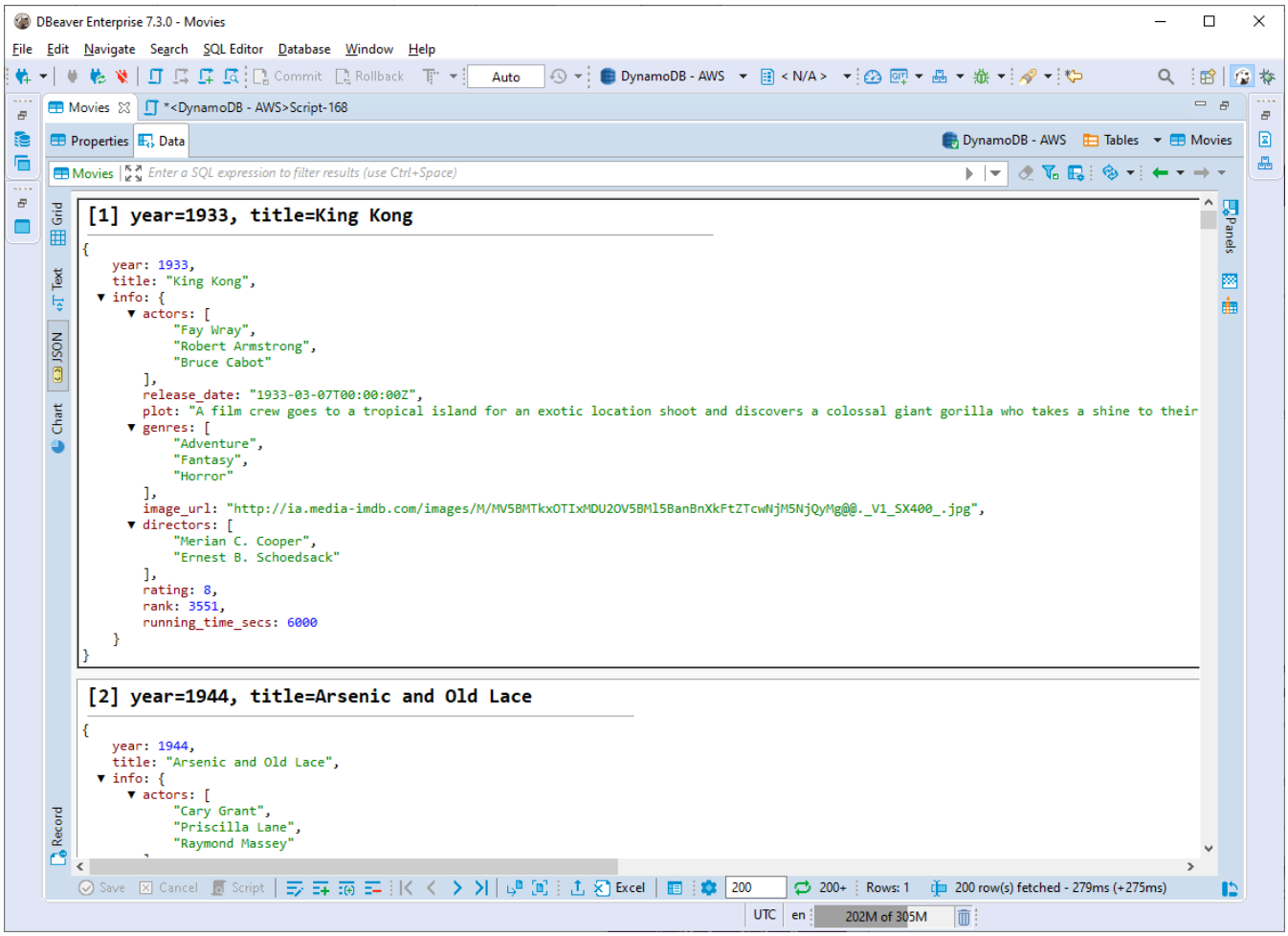
Viewing table data

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



Viewing data in JSON document format

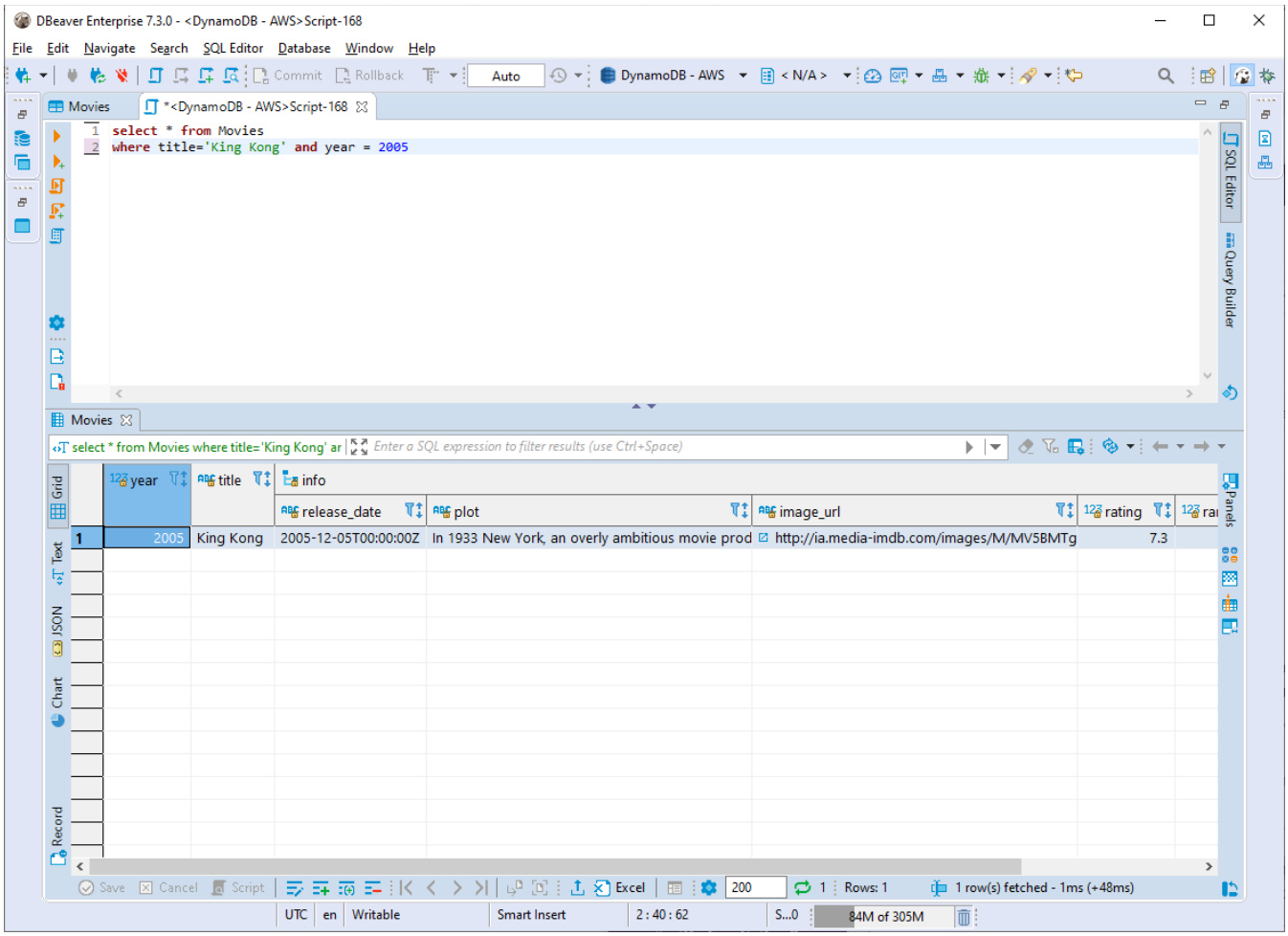
You view, search and edit JSON documents. Double-click on a document to activate the editor.



Executing queries

DBeaver supports simple SQL dialect for DynamoDB.
You can use the WHERE clause in the same fashion as in regular SQL in order to find or filter documents.

You can also use JSON requests syntax to query documents. See Amazon DynamoDB query reference.



Exporting and importing data

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

DocumentDB

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

Connections

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

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

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

Queries

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

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

Keyspaces

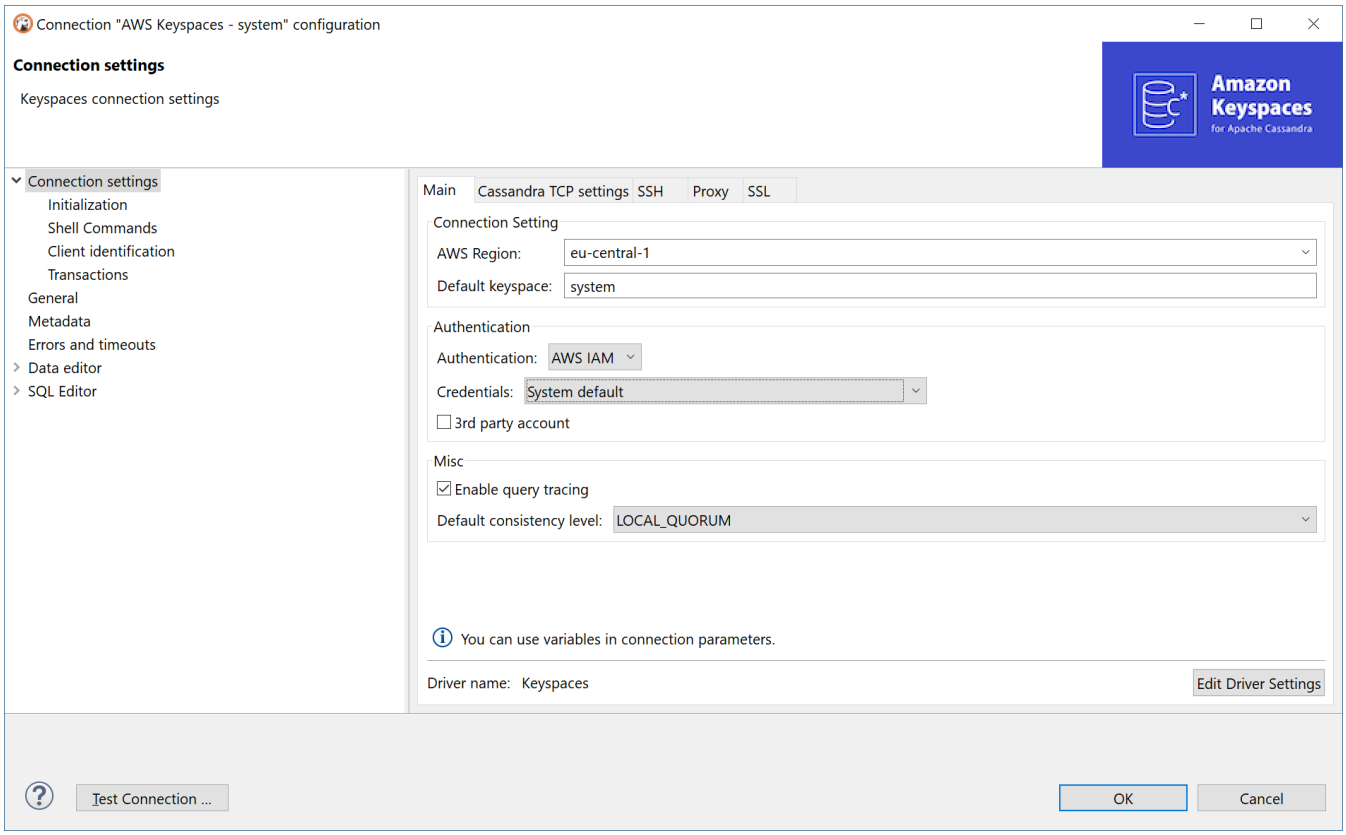
Overview

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

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

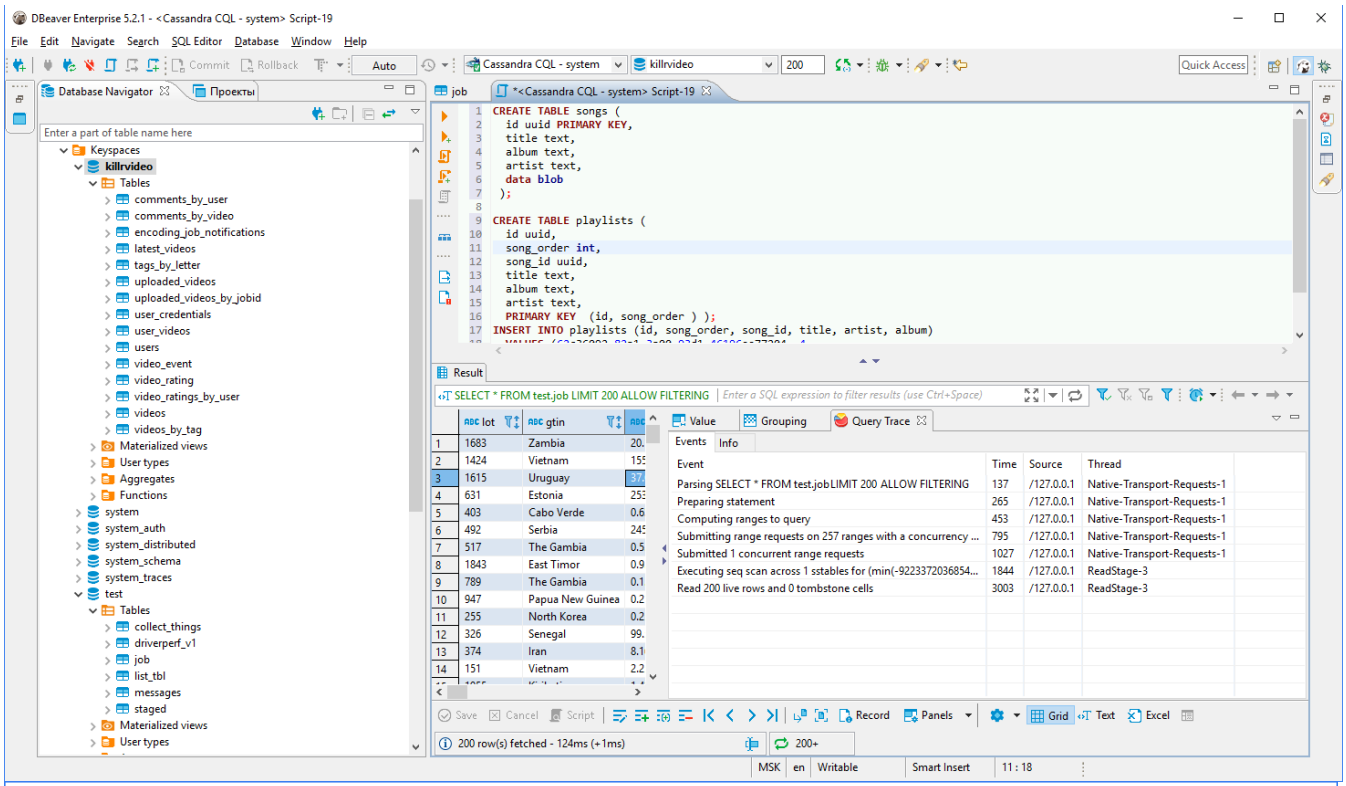
Connecting to Keyspaces

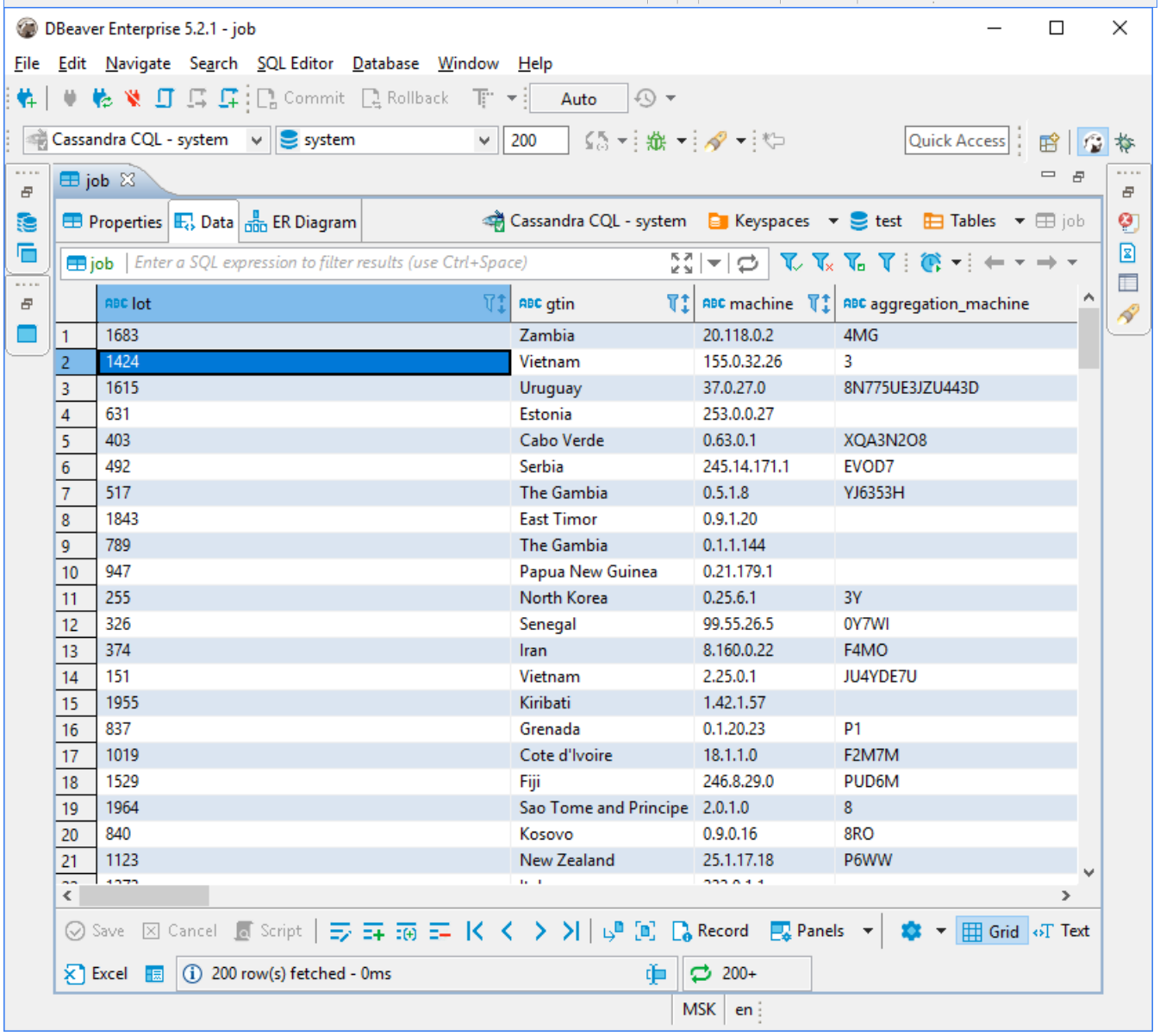
AWS Keyspaces uses AWS IAM authentication.
You need to specify your IAM credentials and AWS region. There is no specific endpoint - there is only one cluster per account per AWS region.
There is also no need to configure SSH or SSL - DBeaver uses default AWS settings to access the Keyspace cluster.



Browsing Keyspaces tables

You can browse, view, edit and filter Cassandra tables the same way as with regular (relational) tables. However, being a distributed key-value database, Keyspaces does not support any kind of referential integrity. There are no foreign keys, references, etc.
You should note that Cassandra has a very advanced (comparing to relational databases) data type system. Each column may be a collection, map, or set of values (with a very big number of values). In some cases this makes browsing data in the "Grid" mode inconvenient.





Executing CQL

CQL Cassandra Query Language is a very simple kind of SQL language dialect.

It supports simple SELECT queries, DDL statements (like CREATE TABLE) and some other query types.

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

ERD

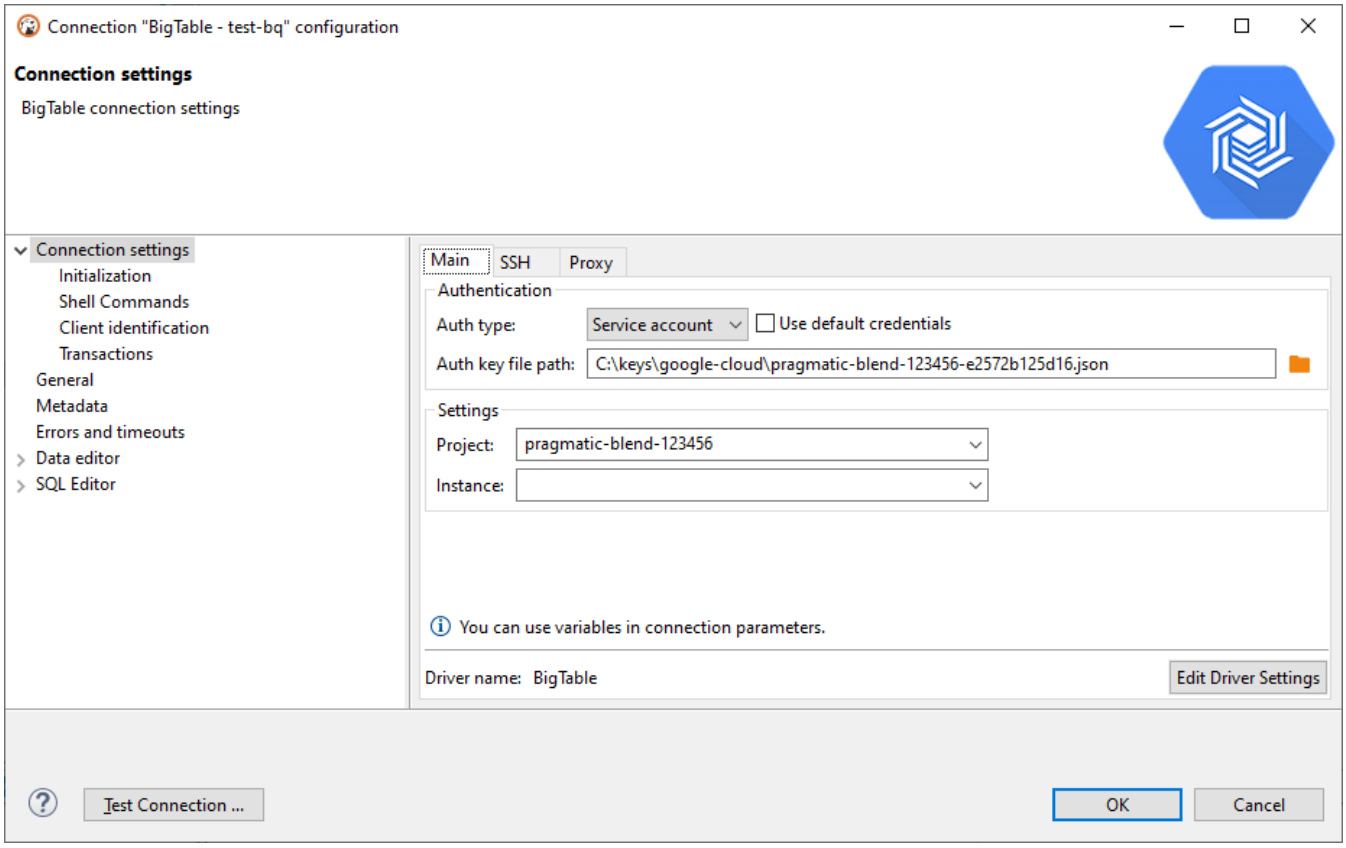
Physical ERD (Entity Relation Diagram) does not make much sense for Keyspaces as there are no foreign keys. However, you can make you own custom ERD and connect Keyspaces tables with each other using logical associations.

Bigtable

Supported features:

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

Bigtable connection

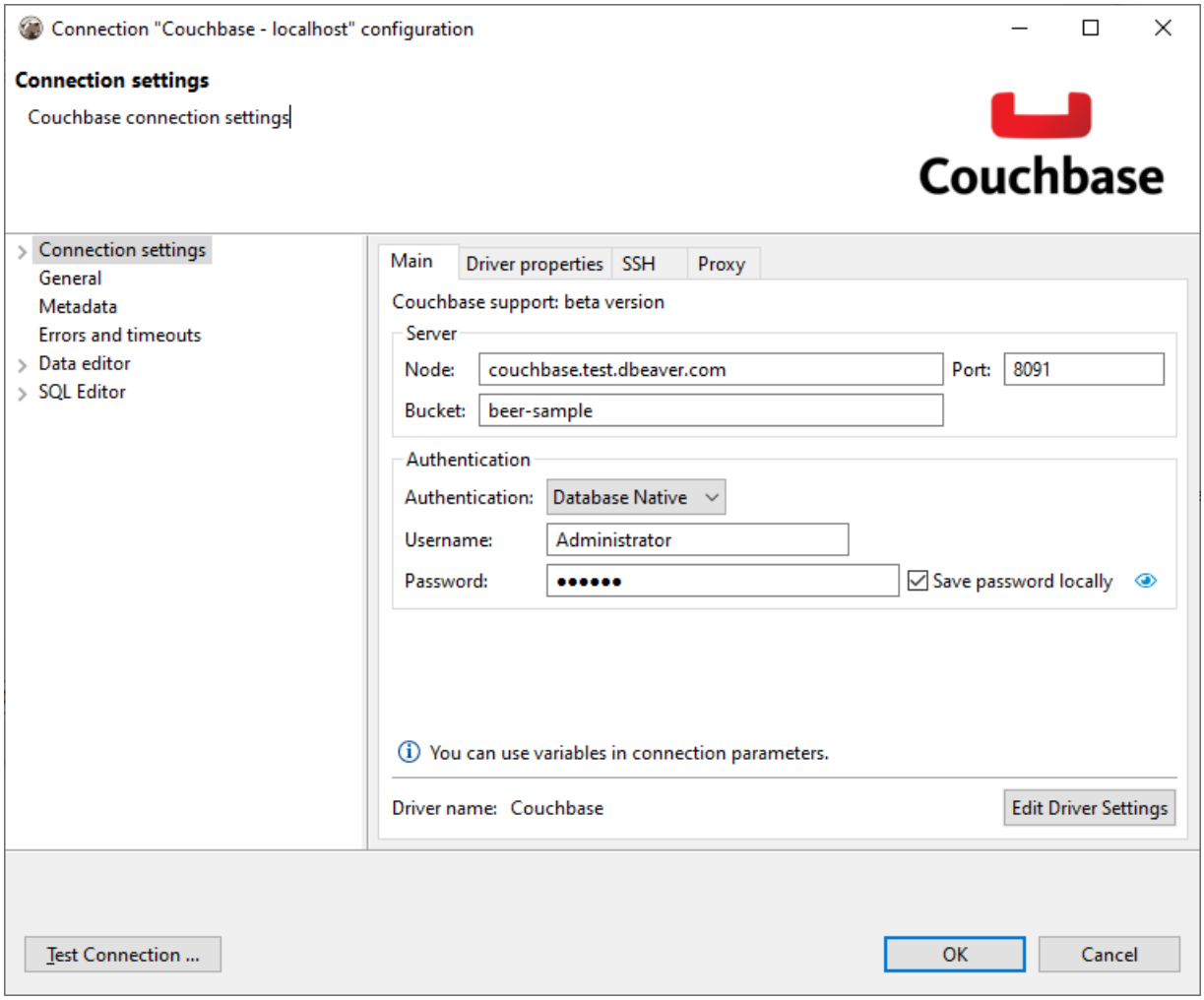


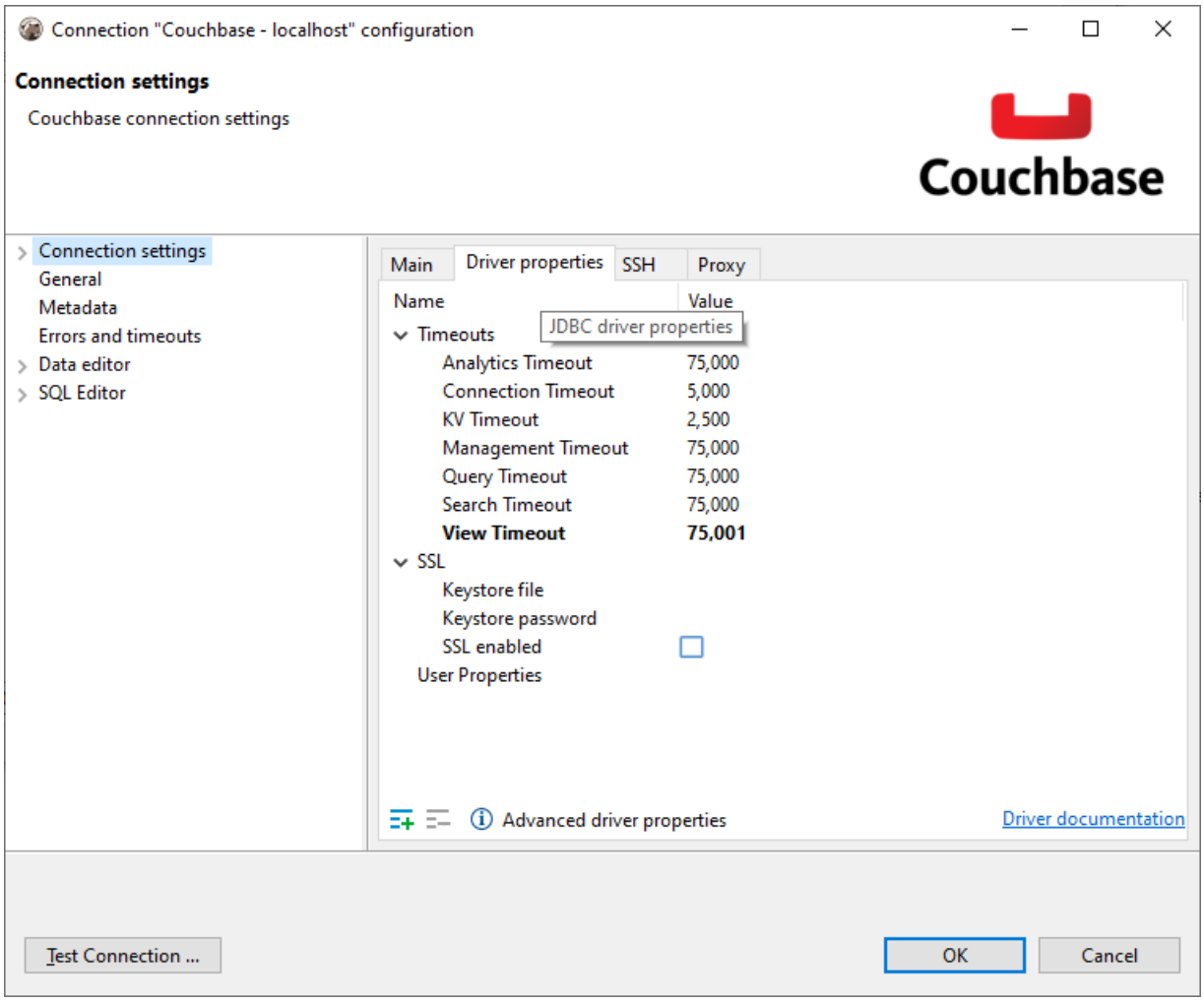
Couchbase

Connecting to Couchbase server

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

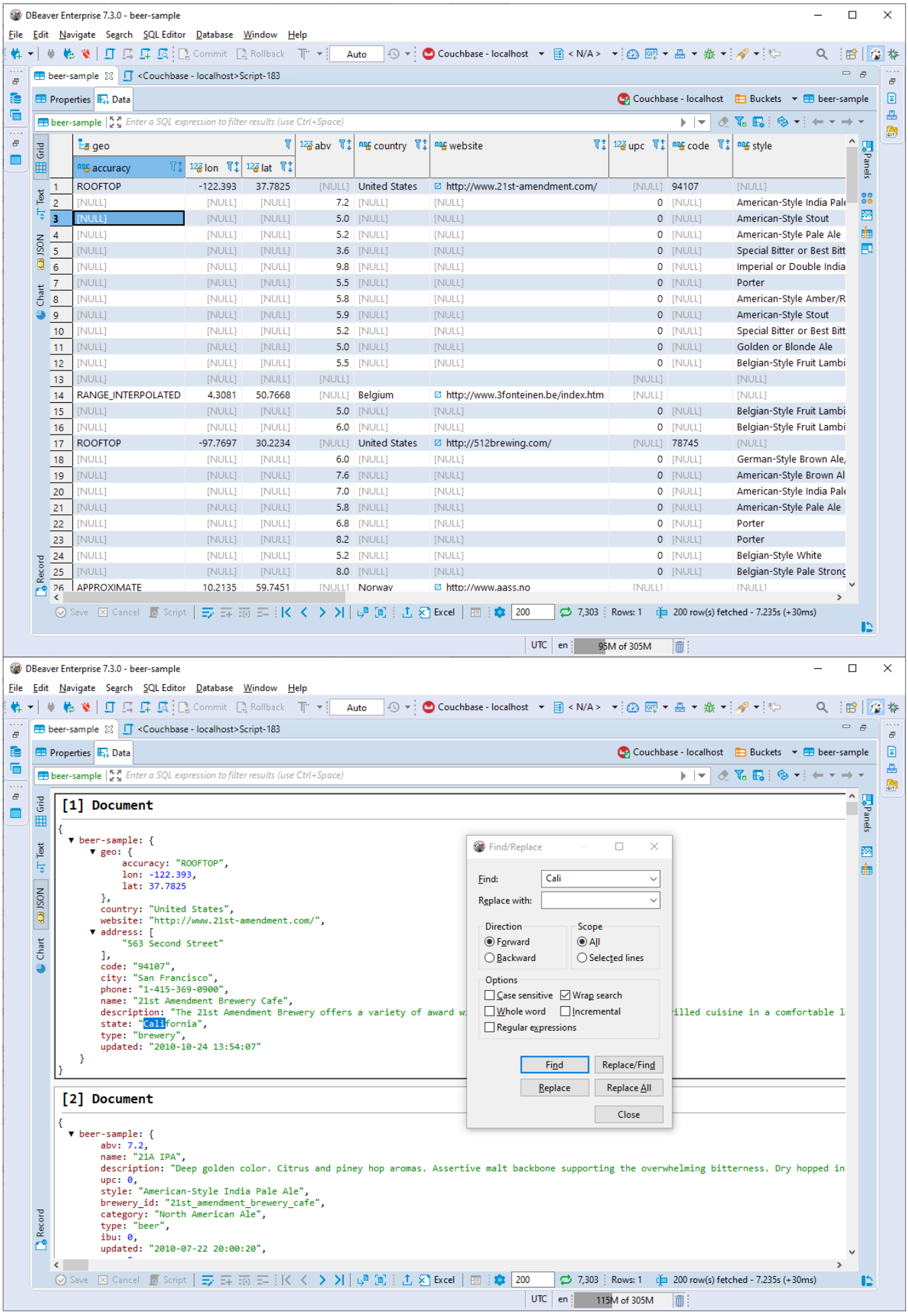
Couchbase client uses multiple ports to connect to a cluster (8091-8096, 9140, etc). Some of these ports are dynamic (i.e. depend on server settings) and cannot be overwritten. It makes SSH tunnelling impossible. Thus, if you work with a remote Couchbase deployed behind a firewall, you will need to setup a VPN connection or SOCKS proxy.





Viewing and editing Couchbase tables

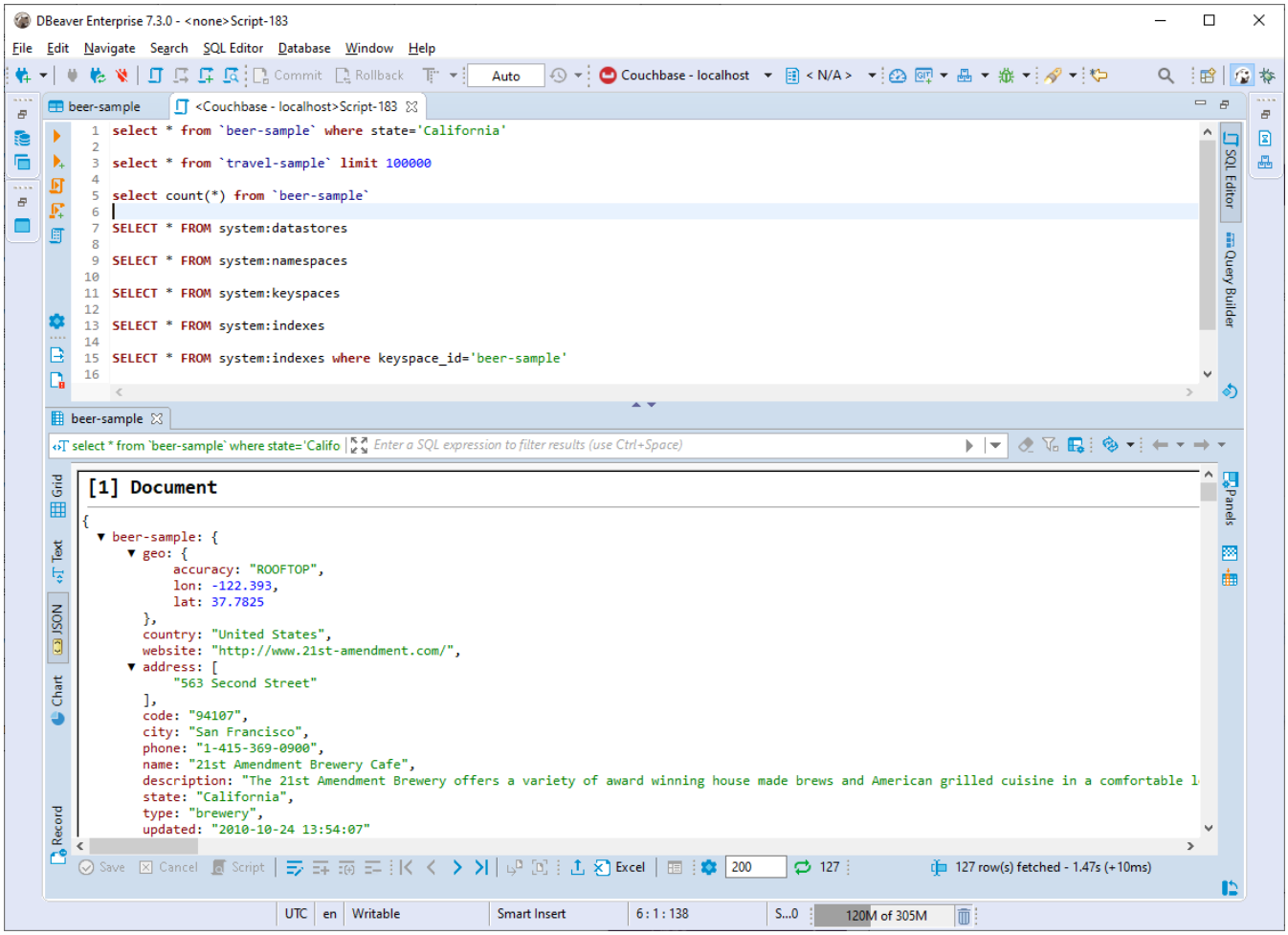
Couchbase is a document-oriented database. It means that all documents may have different structures.
You can view/edit buckets content, such as standard relational tables (grid/plain text presentations) or JSON documents.



Executing Couchbase queries

Couchbase uses N1QL language for queries. It is very similar to the standard SQL language.

```
SELECT country FROM `travel-sample` WHERE name = "Excel Airways";
```



Apache Hive/Spark/Impala

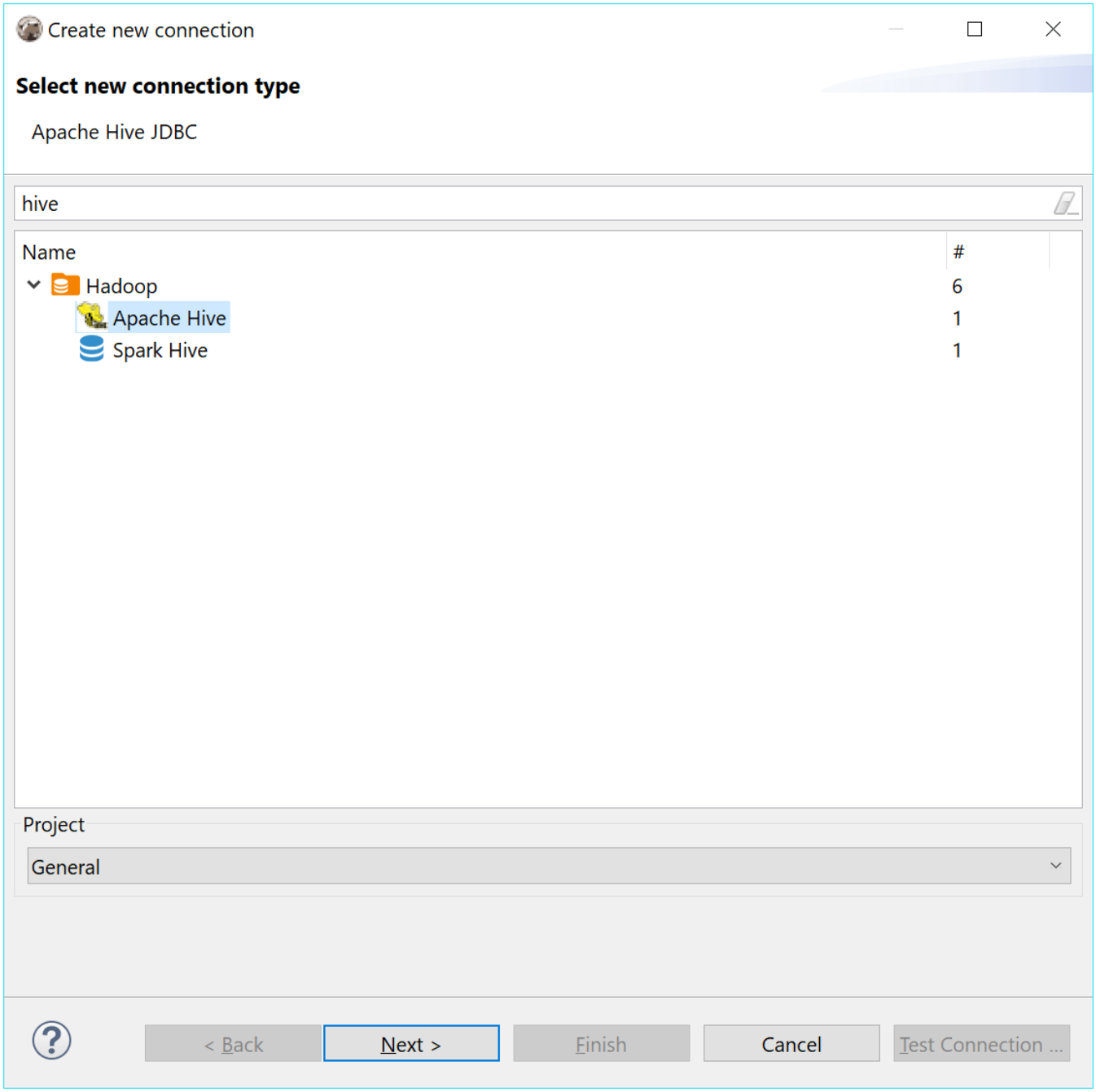
Apache Hive

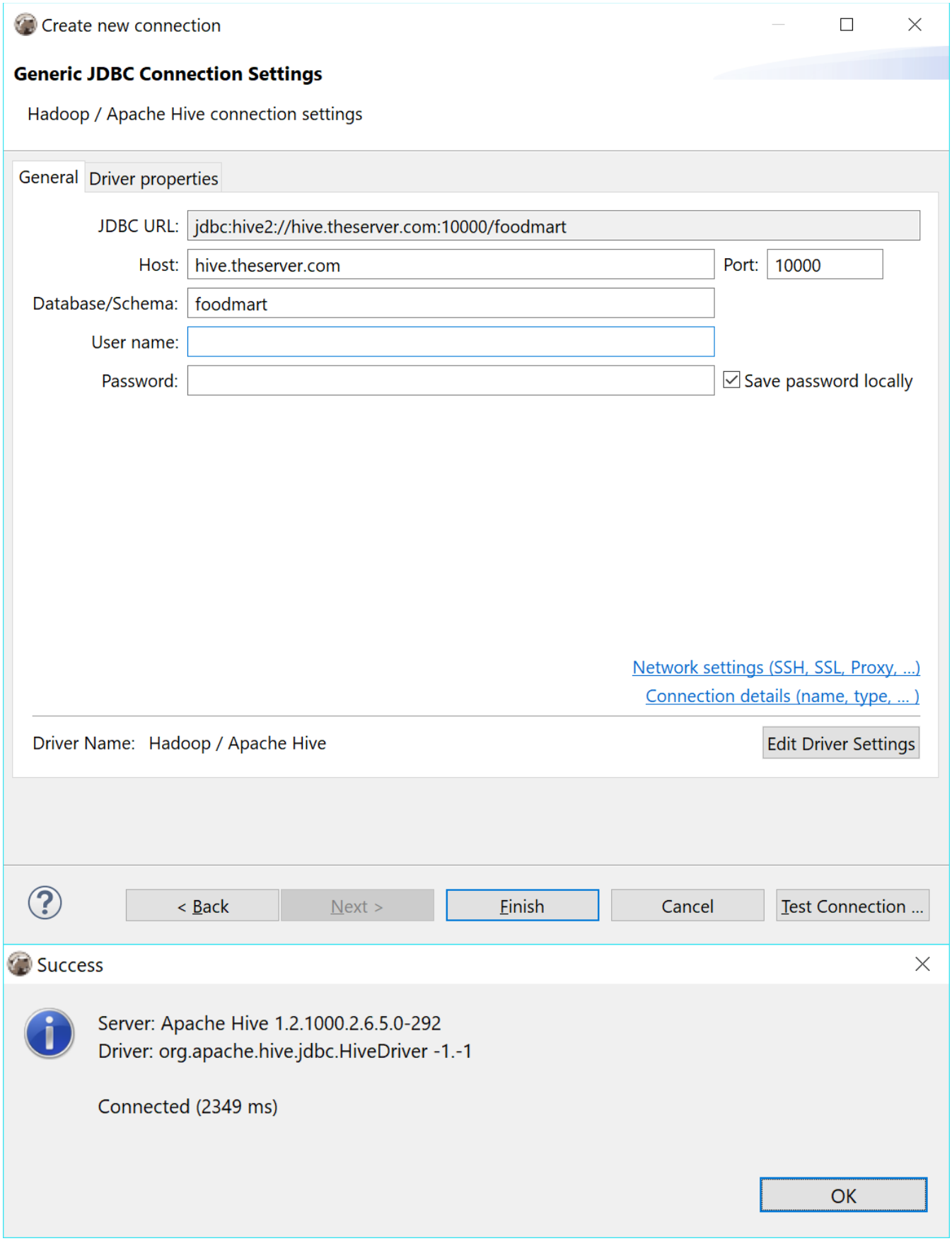
Hive is a Hadoop-based storage system. Hive uses a special SQL dialect (HiveQL) to operate with data and metadata. Generally, it is quite similar to SQL.

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

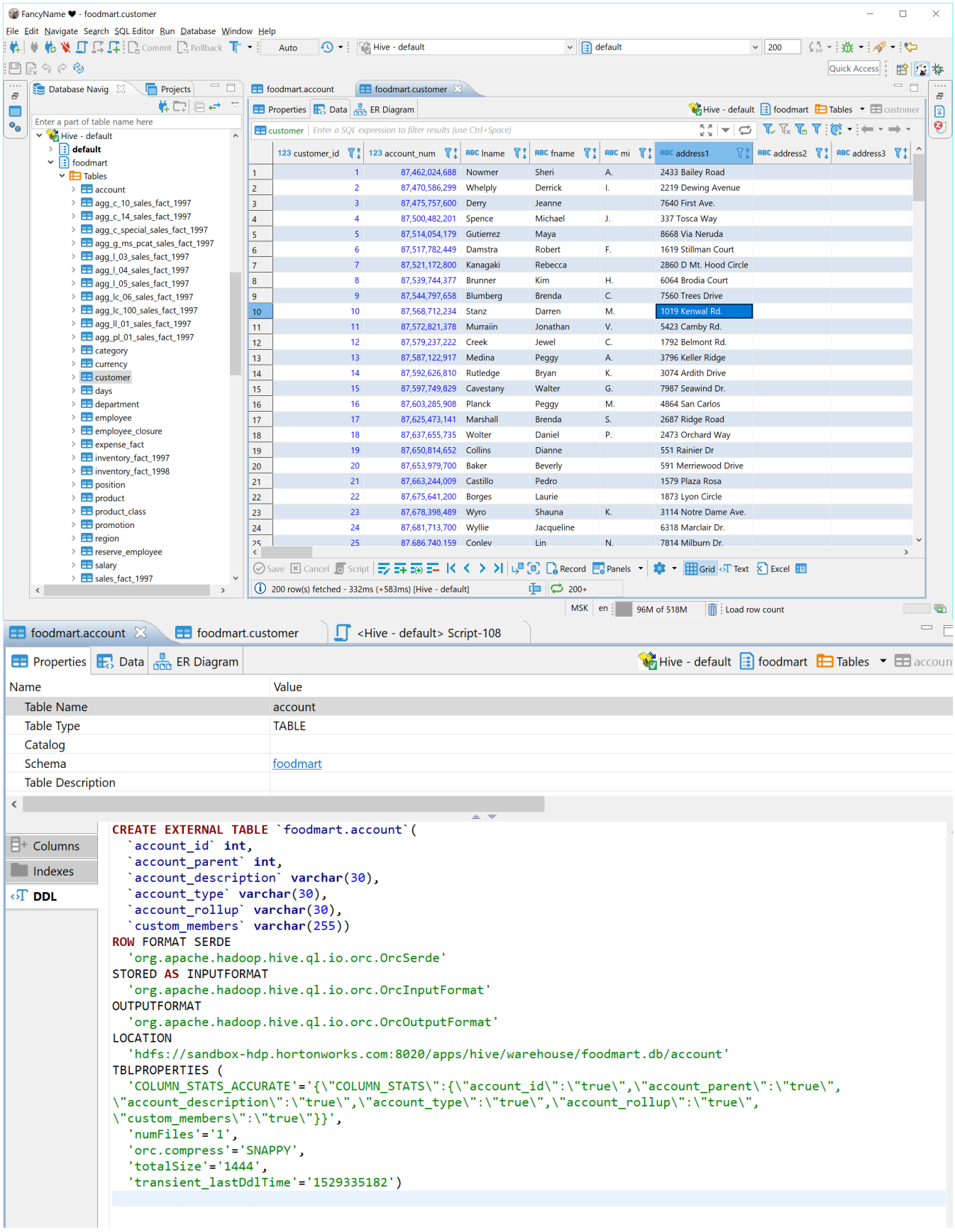
DBeaver uses a so-called Hive JDBC Uber Jar driver (https://github.com/timveil/hive-jdbc-uber-jar) which includes all necessary dependencies. You do not need to download anything - DBeaver will download everything automatically (if you have internet access).

Connection setup





Schema/data browser



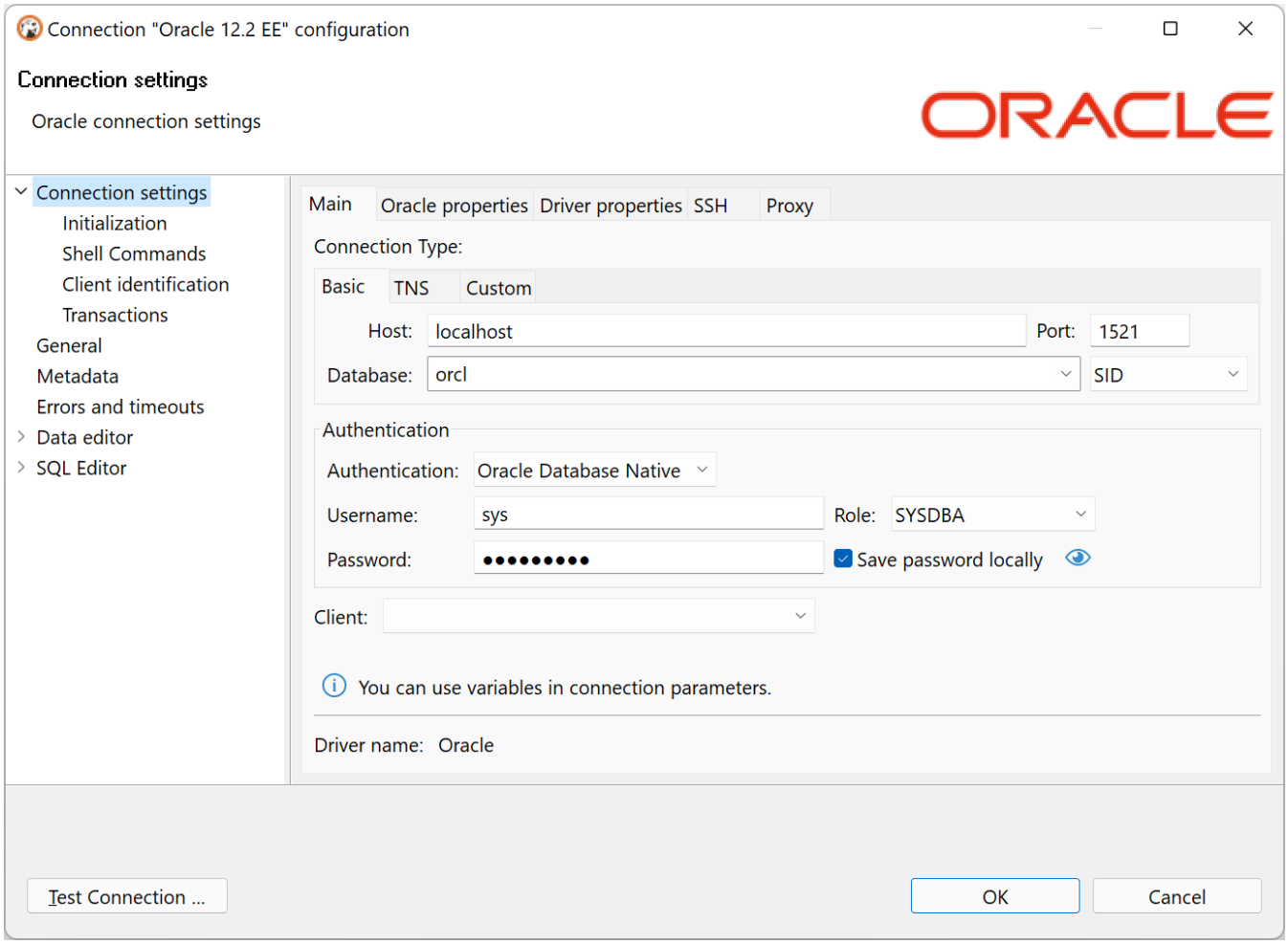
Limitations

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

Oracle

Connecting to Oracle databases

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



Configuration types

Basic connections

Host/port based configuration

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

TNS

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

Parameter	Description	Example
Network Alias	Name of configuration from tnsnames.ora	ORCL1

Parameter	Description	Example
TNS names path	Path to `tnsnames.ora` file. By default, it is got from the TNS_ADMIN environment variable or from Windows' registry	c:\oracle\network\admin

Custom URL

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

Sample URL (Oracle Cloud):

```
jdbc:oracle:thin:@(description= (retry_count=20)(retry_delay=3)(address=(protocol=tcps)(port=1522)(host=adb.us-ashburn-1.oraclecloud.com))(connect_data=(service_name=xxxxxxxxxxxxxxxxx_high.adb.oraclecloud.com))(security=(ssl_server_cert_dn="CN=adwc.uscom-east-1.oraclecloud.com, OU=Oracle BMCS US, O=Oracle Corporation, L=Redwood City, ST=California, C=US")))
```

Authentication

Database

Parameter	Description	Example
User name	Database user name	SYS
Password	Database user password	
Role	Role for connection. Roles SYSDBA and SYSOPER are needed for some administrative operations	Normal
Save password	Saves the user/password information in the local DBeaver configuration	SID

OS authentication

The Oracle driver gets user information from the current OS user.
You do not need to explicitly specify any credentials.

Oracle Wallet

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

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

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

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

Kerberos

Kerberos is the most complicated authentication in Oracle.

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

Oracle Cloud connections

DBeaver supports Oracle Cloud Autonomous databases connectivity.
There are two ways to authenticate:

Plain URL connection

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

Oracle Wallet connection

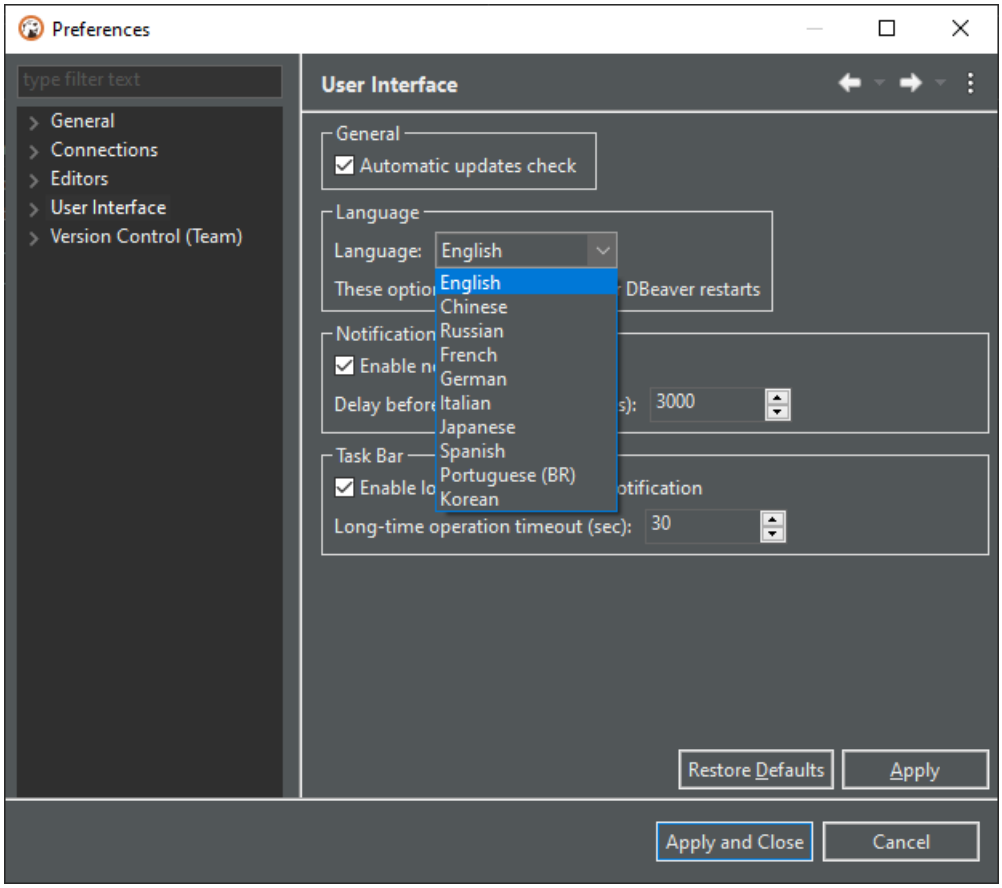
It is the default authentication type for the Oracle Cloud.
.

- Download Wallet from the Oracle Cloud website
- Expand the wallet archive to a folder
- Set TNS configuration type
- Set the `TNS path` to the wallet location directory
- Choose the proper `Network Alias` from the drop-down menu
- Set `Authentication` to the Oracle Wallet
- Set the database user name and password (you can get them from the Oracle Cloud database information page)
- Set the `Wallet location` to the wallet location directory

Changing interface language

Changing interface language in Preferences

Go to Preferences->User Interface:



Select your language in the drop-down list and click the "Apply and Close" button.
If DBeaver is installed in a read-only directory, the automatic language change is not possible. In this case, try to edit the configuration file (see below).

Changing interface language in configuration file

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

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

```
-nl
XX
```

where XX is two-letter language code:

Language	Code
English	en
Chinese	zh
French	fr
Italian	it
Japanese	jp
German	de
Korean	ko
Portuguese (BR)	pt_BR
Russian	ru

Language	Code
Spanish	es

Installing extensions - Themes, version control, etc

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

DBeaver-specific extensions

- Office formats support (XLSX)
- Vector graphics support (SVG)
- SSHJ and advanced cryptography (since version 21 it is included in the base distribution)
- Git support - Git version control integration
- SQL debugger

Popular 3rd party extensions for Eclipse and DBeaver

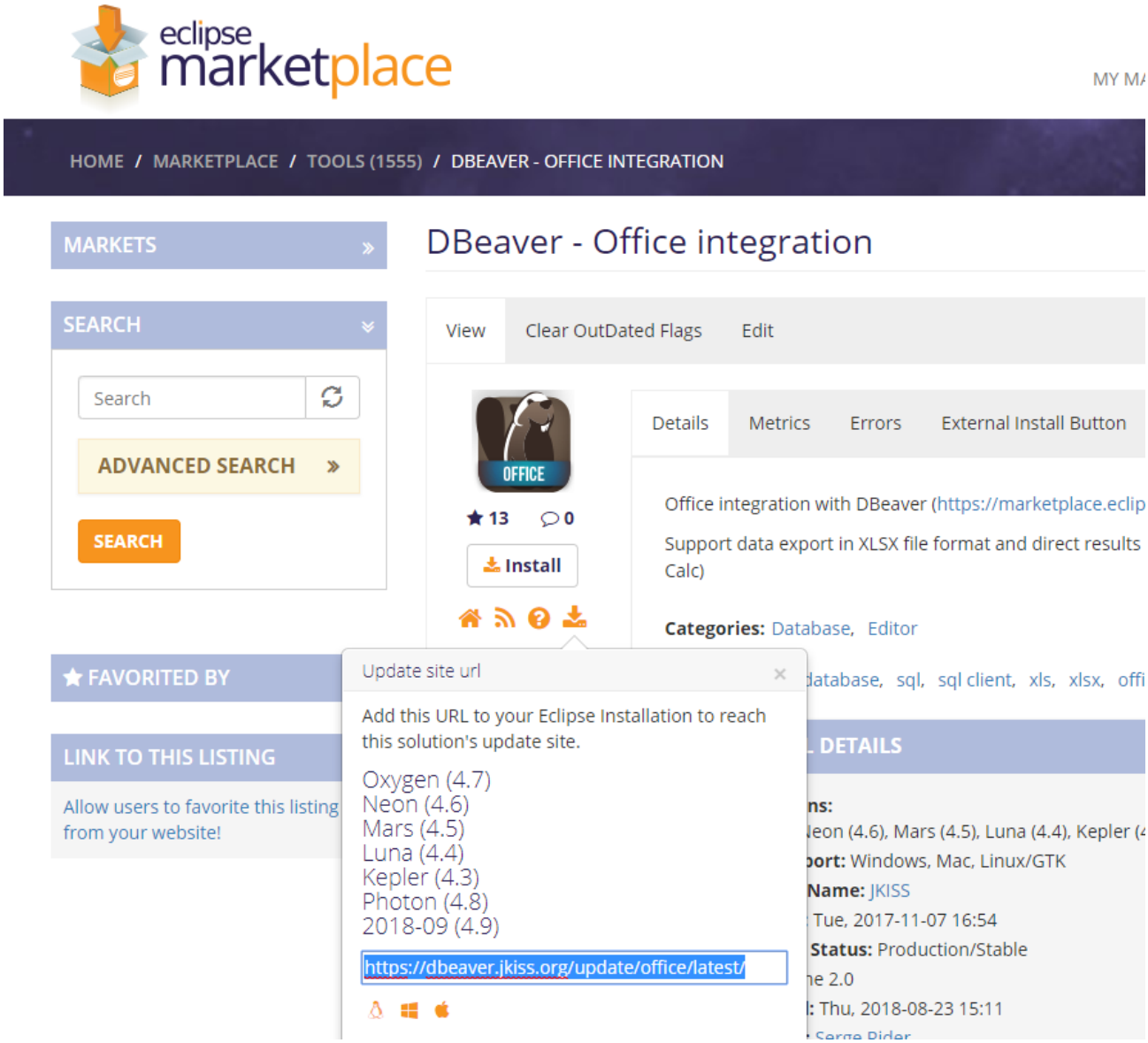
- Darkest Dark theme - the best Dark theme for DBeaver
- Eclipse Color Theme - if for some reason you do not like the Darkest Dark theme, you can use this one
- Subversion support - Subversion integration
- Embedded Shell - Allows you to run shell commands directly from DBeaver
- Editor vertical indents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

Install Process

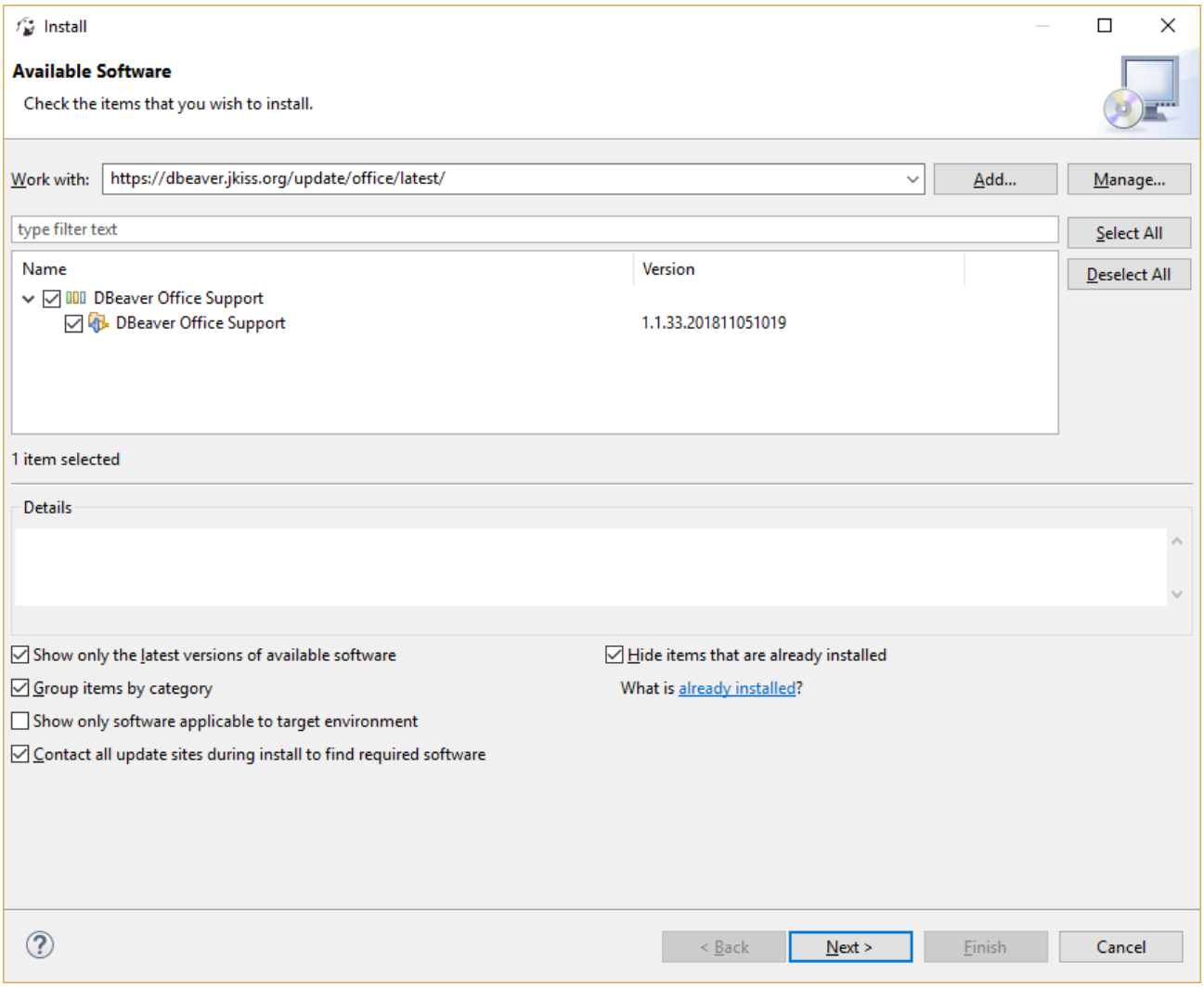
In DBeaver EE you can use drag-n-drop from the Marketplace web site (see button `Install`) in the DBeaver main window. This will launch the Marketplace installation wizard automatically. In the DBeaver Community or other DBeaver-based products which do not include marketplace clients, you can use the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

Extension installation in CE version:

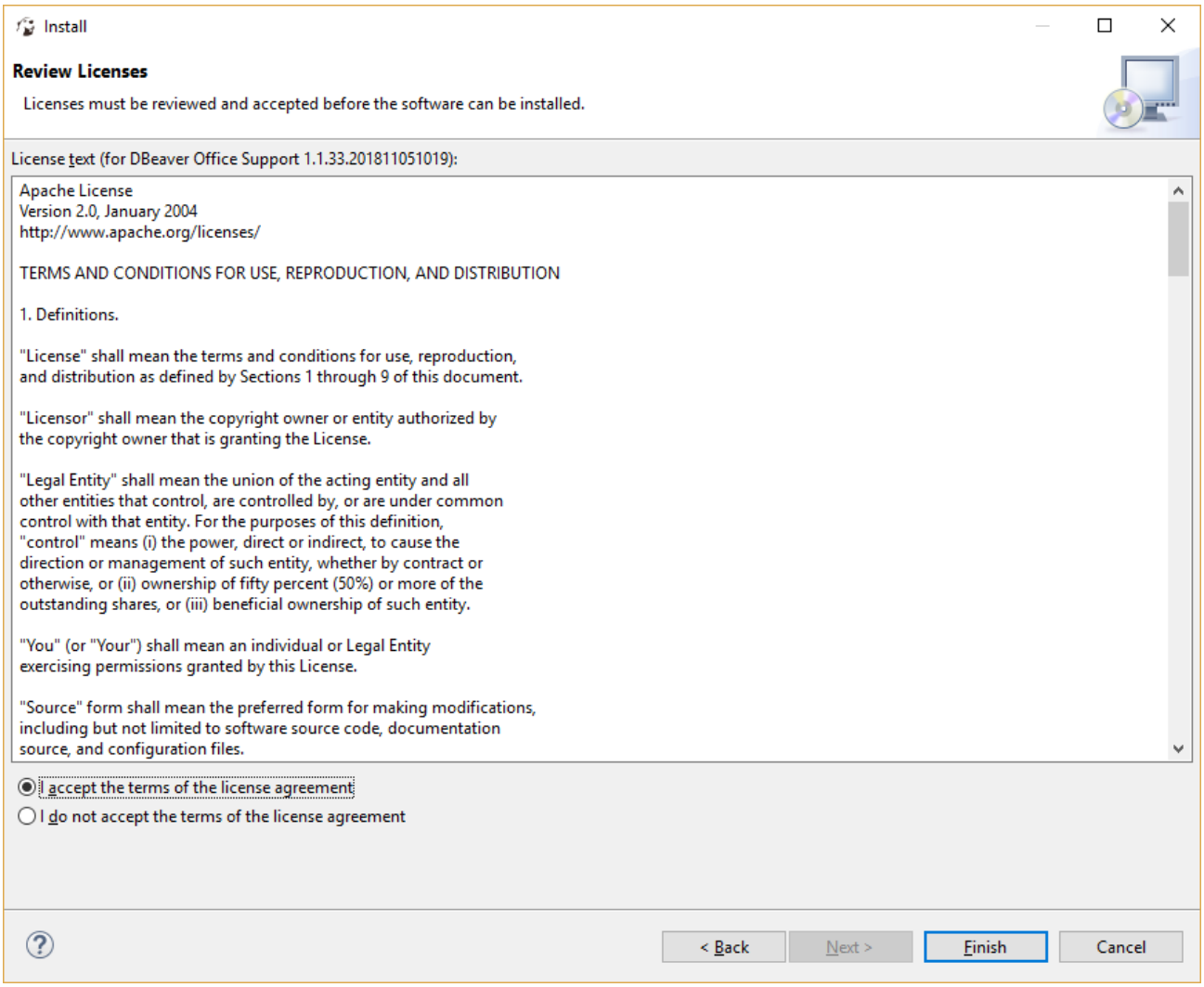
1. Copy URL of extension update site:



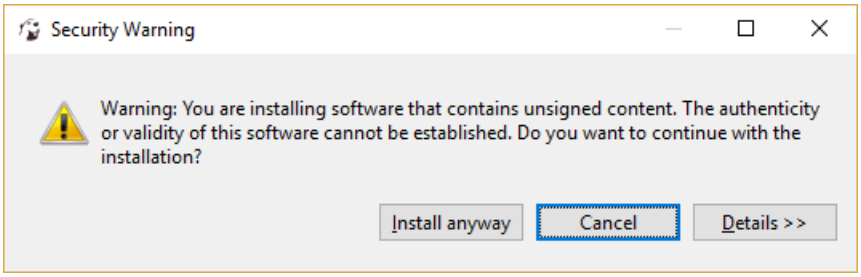
2. In the DBeaver main menu open `Help -> Install New Software`
3. Paste update site URL into `Work with` field and press `Enter`
4. Check items you wish to install (in most cases just all items)



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



6. Some extensions may contain unsigned bundles. Only install such extensions if you really trust the author.



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

Enterprise Edition

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

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

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

Command Line

Command line parameters

Command line parameters might be passed directly to dbeaver[.exe] executable.
In Windows, you can use `dbeaver-cli.exe` executable (it does not spawn a new window so you can see the output messages).

Also, you can add parameters in the `dbeaver.ini` configuration file. You need to write them to the beginning of the file, and each parameter has to be on its line.

DBeaver control

Name	Value	Example
-help	Prints help message	
-stop	Quits DBeaver	
-dump	Prints DBeaver thread dump	
-f	Opens the file in DBeaver UI, if the command has -con argument, connects it to datasource	`-f c:\some-path\some-file.sql`
-con	Opens database connection in DBeaver UI	See connection parameters table
-closeTabs	Closes all open editor tabs	
-disconnectAll	Closes all open connections	
-reuseWorkspace	Forces reuse of single workspace by multiple DBeaver instances	
-newInstance	Forces new DBeaver instance creation (do not try to reuse already running one)	
-bringToFront	Brings the DBeaver window on top of other applications	
-var	Customs variables for runTask. You can change existing variables in the task. You cannot add new task variables with this parameter. You can add several parameters at once to the command line, each starting with "-var". Used right before -runTask. Template: `-var variableName=variableValue`	`-var film=sakila.film` `-var actor=sakila.actor` `-runTask "exportFromSakila"` EE version only.
-vars	Path to a property file with variables	`-vars c:\path\to\file.properties` For more information see the main article
-runTask	Executes specified task	`-runTask "@projectName:taskName"`. EE version only. See task scheduler.
-license	Path to the EE license file	`-license "/etc/licenses/dbeaver.txt"`. EE version only.

System parameters

Name	Value	Example
-nl	Locale	en_US
-data	Workspace path	c:\ProgramData\MyWorkspace
-nosplash	Omits splash screen	true
-clean	Clears all Eclipse caches. Use it if DBeaver fails to start after it upgrades.	
-vmargs	VM parameters	See VM arguments table

VM arguments

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

Parameters supported by all JVMs:

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

Connection parameters

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

Name	Description	Example
name	Connection name	`Test connection`
driver	Driver name or ID	`driver=sqlite` , `driver=mysql` , etc
url	Connection URL. Optional (JDBC URL may be constructed by a driver from other parameters)	`url=jdbc:sqlite:C:\db\SQLite\Chinook.db`
host	Database host name (optional)	`host=localhost`
port	Database port number (optional)	`port=1534`
server	Database server name (optional)	`server=myserver`
database	Database name or path (optional)	`database=db-name`
user	User name (optional)	`user=root`
password	User password (optional)	`password=mysecret`
auth	Authentication model ID. See Auth models	`auth=postgres_pgpass`
authProp.propName	Custom authentication parameters (depends on the driver and auth model)	`authProp.oracle.net.wallet_location=C:/temp/ora-wallet`
savePassword	Does not ask user for a password on connection	`savePassword=true`
showSystemObjects	Shows/Hides system schemas, tables ,etc	`showSystemObjects=true`
showUtilityObjects	Shows/Hides utility schemas, tables ,etc	`showUtilityObjects=true`
folder	Puts a new connection in a folder	`folder=FolderName`
autoCommit	Sets connection auto commit flag (default value depends on driver)	`autoCommit=true`
prop.propName	Advanced connection parameters (depend on driver)	`prop.connectTimeout=30`
id	Connection id	`oracle_thin-16a88e815bd-70598e648cedd28c` (useful in conjunction with `create=false`)
connect	Connects to this database	`connect=false`
openConsole	Opens the SQL console for this database (sets `connect` to true)	`openConsole=true`
create	Creates new connection	`create=false` (true by default). If it is set as false, then an existing connection configuration will be used. The name or id parameter must be specified.

Name	Description	Example
save	Saves new connection	When `create=true`, then `save=false` (default) makes new connection temporary, `save=true` means that new connection will be saved and accessible between DBeaver launches.

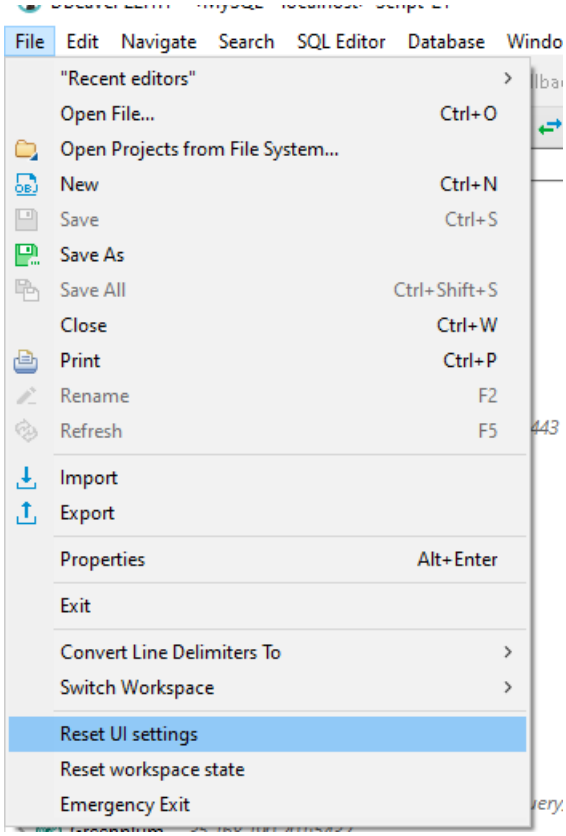
Declare external variables in a file

See the main article

Reset UI settings

After multiple version and/or upgrades/incorrect shutdowns, DBeaver UI may become corrupted. You could experience glitches such as extra toolbar elements might appear, menu items might go missing, keyboard shortcuts and localization strings could be broken, etc.

To reset DBeaver UI press `Reset UI settings` button in the `File` menu:



Also, you can do this operation manually. Just delete the file `workbench.xmi` in DBeaver workspace/.metadata. By default, workbench.xmi file locations is:

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

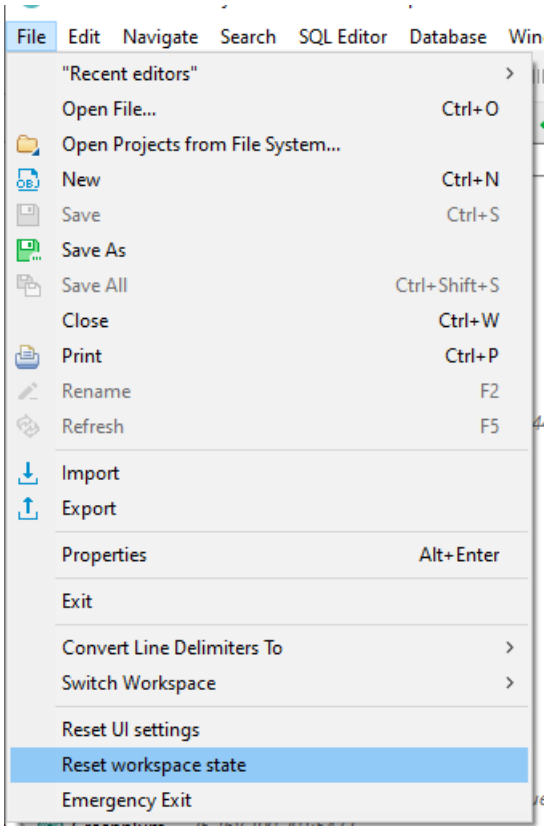
To reset the settings:

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

Reset workspace

Sometimes (especially after multiple DBeaver versions upgrade) the workspace can become messy.
Some keyboard shortcuts may stop working, toolbar layouts may be broken, etc, etc.

Press `Reset workspace state` in the `File` menu to reset all UI settings (this includes menus, shortcuts, view and toolbar layouts):



Also, you can do it manually:

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

If that does not help then you can try to remove the `.metadata/.plugins/org.eclipse.core.resources` folder.

If that does not help then remove the `.metadata` folder. It will erase all your UI settings (but all connections, settings and scripts will remain as is).

That is it.

Troubleshooting system issues

Linux

GTK

DBeaver is an Eclipse RCP application. Therefore, it may have issues with GTK. Here are known workarounds regarding various issues which may arise.

Fixing screen flickering.

- Add `export GTK_IM_MODULE=ibus` to `~/.profile`.

Parts of DBeaver are white/dark on dark/white theme

To fix the issue, we recommended having a graphical theme which is similar to system one in terms of the color palette.

`GTK-WARNING xxx:Theme parsing error`

This issue requires you to change the GTK-program-style `system settings-appearance-program style` (the setting location may vary for different systems).

`gtk_box_gadget_distribute: assertion size 'size >=0 0' failed in GtkScrollbar`

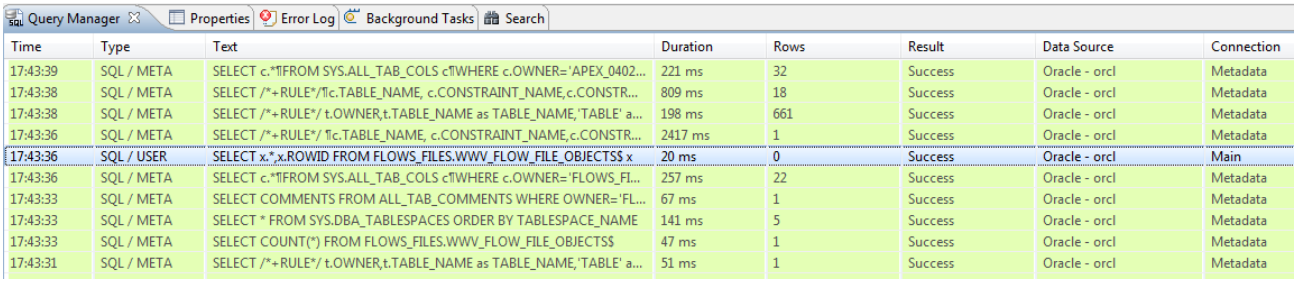
To fix this issue, you need to remove the overlay scrollbars in `~/.config/gtk-3.0/settings.ini`.

```
[Settings]
gtk-overlay-scrolling = false
```

Posting issues

A few tips.

- Check existing issues for your issue (including closed ones). Duplicating an issue is slower for both parties so search through the open and closed issues to see if what you are running into has already been addressed.
- Be clear about what your problem is: what was the expected outcome, what happened instead? Detail how someone else can recreate the problem.
- If you posting a bug report check "Error Log" view. If there are any errors related to your bug then post a complete stacktrace. Sometimes there are no errors in Error Log - if so, try to find them in log files.
- If your issue is related to database data or metadata management - check the Query Manager view. It contains information about all queries DBeaver executes (explicitly or implicitly). To see more detailed information you can configure Query Manager in Preferences.



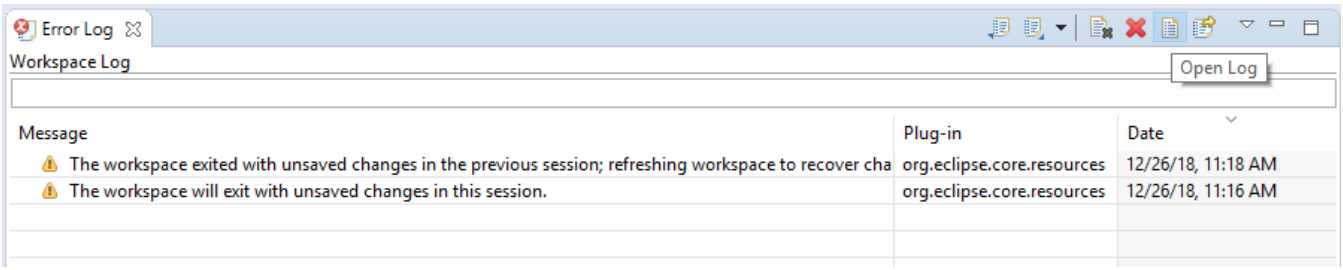
Time	Type	Text	Duration	Rows	Result	Data Source	Connection
17:43:39	SQL / META	SELECT c.*¶FROM SYS.ALL_TAB_COLS c¶WHERE c.OWNER='APEX_0402...	221 ms	32	Success	Oracle - orcl	Metadata
17:43:38	SQL / META	SELECT /*+RULE*/¶c.TABLE_NAME, c.CONSTRAINT_NAME,c.CONSTR...	809 ms	18	Success	Oracle - orcl	Metadata
17:43:38	SQL / META	SELECT /*+RULE*/ t.OWNER,t.TABLE_NAME as TABLE_NAME,'TABLE' a...	198 ms	661	Success	Oracle - orcl	Metadata
17:43:36	SQL / META	SELECT /*+RULE*/ ¶c.TABLE_NAME, c.CONSTRAINT_NAME,c.CONSTR...	2417 ms	1	Success	Oracle - orcl	Metadata
17:43:36	SQL / USER	SELECT x.*,x.ROWID FROM FLOWS_FILES.WWV_FLOW_FILE_OBJECTS$ x	20 ms	0	Success	Oracle - orcl	Main
17:43:36	SQL / META	SELECT c.*¶FROM SYS.ALL_TAB_COLS c¶WHERE c.OWNER='FLOWS_FI...	257 ms	22	Success	Oracle - orcl	Metadata
17:43:33	SQL / META	SELECT COMMENTS FROM ALL_TAB_COMMENTS WHERE OWNER='FL...	67 ms	1	Success	Oracle - orcl	Metadata
17:43:33	SQL / META	SELECT * FROM SYS.DBA_TABLESPACES ORDER BY TABLESPACE_NAME	141 ms	5	Success	Oracle - orcl	Metadata
17:43:33	SQL / META	SELECT COUNT(*) FROM FLOWS_FILES.WWV_FLOW_FILE_OBJECTS$	47 ms	1	Success	Oracle - orcl	Metadata
17:43:31	SQL / META	SELECT /*+RULE*/ t.OWNER,t.TABLE_NAME as TABLE_NAME,'TABLE' a...	51 ms	1	Success	Oracle - orcl	Metadata

- Depending on the nature of your bug report provide information about:
  - Operating system
  - Window manager (for Linux)
  - Database (name and version)
  - Database driver (name and version)
- Do not write issue type in the issue title (like Feature Request:, Bug: etc). We will review your issue and assign a corresponding label.

Log files

Error Log view

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



Log files

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

- In Windows open Explorer and paste path `%APPDATA%\DBeaverData\workspace6\.metadata`.
- In Linux just type `cd $XDG_DATA_HOME/DBeaverData/workspace6/.metadata`
- In MacOS open path `~/Library/DBeaverData/workspace6/.metadata` in Finder.
    - To view hidden folders press `Cmd+Shift+.` in the folder view.

Two standard log files:

- workspace/workspace6/.metadata/.log - all warnings and errors which happen during normal work
- workspace/workspace6/.metadata/dbeaver-debug.log - the same as `.log` plus debug information

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

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

If you are reporting an error, please attach the applicable part of the log - not the complete file.
Logs are very useful. Many errors cannot be reproduced and fixed without a full error stacktrace (all the details).

Java fatal logs

On the rare occasion that the DBeaver process dies, it does not leave any valuable logs. This is caused by a Java VM crash.
JVM creates a fatal log file for each crash (log gile `hs_err_PID.log`). This log usually resides in the same directory where the DBeaver launcher is (e.g. dbeaver.exe).
But in some cases it is a write-protected directory and the log file will be created in other folder.
Instructions on how to find the Java fatal log file: https://docs.oracle.com/javase/9/troubleshoot/fatal-error-log.htm

JDBC trace

In some cases, custom JDBC drivers work incorrectly in DBeaver - they show the wrong metadata like table columns, constraints or foreign keys.
It usually happens because the driver is not compliant with the JDBC API specification and DBeaver cannot correctly interpret the metadata provided by the driver.

To understand what is going on inside the driver, you can enable JDBC tracing:

1. Find `dbeaver.ini` file (it is located in the same folder where DBeaver is installed)
2. Add line `-Ddbeaver.jdbc.trace=true` in the end of `dbeaver.ini`
3. Restart DBeaver
4. Connect to your database and browse the metadata in the database navigator/object editors.
5. In DBeaver Workspace go to `.metadata` folder
6. File `jdbc-api-trace.log` contains all JDBC API invocations and all queries with results.

Analyzing contents of `jdbc-api-trace.log` you can understand what is wrong with the metadata. Attach the piece of the trace file in the GitHub ticket if you think that something is wrong on DBeaver's side.

WARNING: disable JDBC tracing in your regular work. Enable it only for debugging. The trace generation decreases application performance and may produce huge log files.

Thread dump

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

Mac and Linux

Run the following on your terminal:

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

Windows

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

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

in the Command Prompt.

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

Managing connections

This guide describes how to manage/secure the DBeaver database connections. It is designed for System administrators. Regular users should check this guide.

Provide predefined connections

DBeaver keeps connections information in the project folder. By default, all projects reside in the workspace.
The default project folder is workspace\workspace6\General.

DBeaver 6.1.3+

DBeaver keeps information about project connections in the `.dbeaver/data-sources.json`. file.
All secured information (user name, password, secret keys, etc) is stored in the encrypted file, `.dbeaver/credentials-config.json`.

DBeaver can load multiple connection files. Any files in the project folder matching the `.dbeaver/data-sources*.json` pattern will be loaded on the startup. So you can create a file, for example, `.dbeaver/data-sources-2.json` in the project folder and DBeaver will see it.

DBeaver < 6.1.3 (Legacy)

DBeaver keeps information about project connections in the `.dbeaver-data-sources.xml`. file.

DBeaver can load multiple connection files. Any files in the project folder matching the `.dbeaver-data-sources*.xml` pattern will be loaded on the startup. So you can create a file, for example, `.dbeaver-data-sources-2.xml` in the project folder and DBeaver will see it.

Importing connections from CSV/XML

You can import a connection from CSV or XML files.

The CSV file must have a header row (first line of file) with column names (see list of supported columns below). The XML file should contain a top-level element and a set of nested elements. The connections config must be specified in the attributes of the nested elements. Attribute names are the same as the CSV column names.

Supported names:

	Name	Meaning	
	name	Connection name	
	url	JDBC URL	
	host	Database server host name	
	port	Database server port	
	database	Database/schema name	
	user	User name	
	password	User password	

You can only specify the URL or the host/port/etc setting.
User name/password are optional.

Sample CSV

```
name,host,port,server,database,url,user,password,type
Postgre Import XML 1,localhost,5432,,postgres,jdbc:postgresql://localhost:5432/postgres,postgres,postgres,dev
Postgre Import XML 2,localhost,5432,,postgres2,jdbc:postgresql://localhost:5432/postgres2,postgres2,postgres2,prod
```

Sample XML

```
<connections>
    <connection name="Postgre Import XML 1" host="localhost" port="5432" server="" database="postgres" url="jdbc:postgresq
l://localhost:5432/postgres" user="postgres" password="postgres" type="dev"/>
    <connection name="Postgre Import XML 2" host="localhost" port="5432" server="" database="postgres" url="jdbc:postgresq
l://localhost:5432/postgres2" user="postgres2" password="postgres2" type="prod"/>
</connections>
```

Secure connections from editing

It is possible to set the connection settings as read-only (protected by password)

- Generate MD5 hash of your password. You can do it from the command line using Linux utility md5sum (`md5sum <<<"your password"`) or you can do it online - just google "MD5 hash online".
- Add field `lockPassword` in the connection descriptor (in `.dbeaver/data-sources.json` in `connections` element. So it will look like this:

```
"postgres-jdbc-161537836e8-3e0957d039995715": {
   "provider": "postgresql",
   "driver": "postgres-jdbc",
   "name": "PostgreSQL - postgres",
   "save-password": true,
   "show-system-objects": true,
   "read-only": false,
   "folder": "PG",
   "lockPassword": "2ba81a47c5512d9e23c435c1f29373cb"
...
}
```

- If the user will try to change connection settings now, he/she will be asked for a password.

Using environment variables

You can use references on environment variables in most of connection configuration properties.
For example:

```
"postgres-jdbc-161537836e8-3e0957d039995715": {
   "provider": "postgresql",
   "driver": "postgres-jdbc",
   "name": "PostgreSQL - postgres",
   "user": "${dbeaver.default-user}",
...
}
```

Managing variables

Variables

Disclaimer: this article does NOT cover variables used in SQL editor.

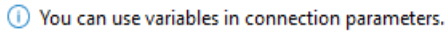
What are variables

A variable is a special template which is replaced with an associated value. Variables help keep your configuration clean and tidy by avoiding unnecessary repetitions: instead of manually replacing each occurrence of some value, you can replace the value of a variable just once and its occurrences will be kept intact.

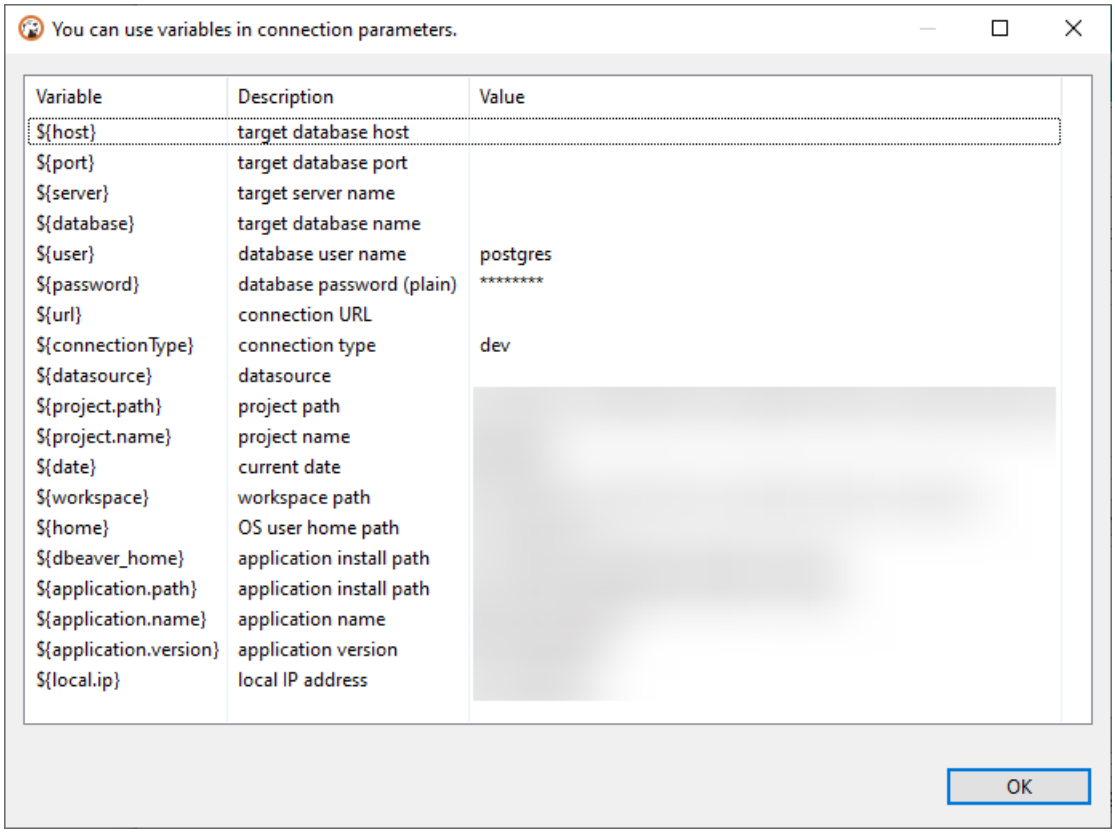
Variables are available in a lot of places. You might find them in:

1. The Connection settings
2. The SSH tunnel settings
3. Data Transfer and in other tasks
4. The Command Line Interface

Wherever you find variables available in the user interface, a hint will usually be present:



By clicking on it, a separate window will open showing all the available variables with their descriptions and values:



Using variables

To use the variables you need to utilize a special syntax:

```
${variable_name}
```

Where `variable_name` is a variable name which you want to use. The window shown above already contains `${` и `}` anchors so you do not need to specify them multiple times.

Variables may be a part of something bigger: e.g., a part of a path. For example, if some file lies under the user's root folder, you may want to use variable `home`: `${home}`: `${home}/path/to/file.txt`.

The Variable resolution

A value of a particular variable may be resolved from different places depending on its origin.

The resolution is performed in the following order (from the highest to the lowest priority):

1. Dialog, where a specific variable is used
2. Datasource-specific variables (`host` , `port` , `server` , `database` , etc.)
3. Application-specific variables (`application.name` , `application.version` , `home` , etc.)
4. Environment variables. For additional information see Wikipedia.
5. External configuration (see the below)

If you have an environment variable called `home` , it will be resolved in the application-specific variable because it has higher priority.

Declare external variables in a file

You can create a file and fill it with pairs of named values and pass it to DBeaver using the `-vars` command-line argument.

Variables from this file can be accessed by other command-line arguments, in the data transfer wizard, and in other places that support variable resolutions.

For example, you may want to put your credentials in that file to avoid showing them to everyone else:

```
# Lines that start with the `#` symbol are comments and therefore ignored.
sampleVar1=abc
someOtherVar=DBeaver is cool
password=P4$$w0r3
```

You can use them as such:

```
dbeaver.exe -vars C:\secrets.properties -con "driver=<xxx>|url=<xxx>|password=${password}"
```

Here, the `-con` argument has the `${password}` variable that will be replaced with `P4$$w0r3` defined in the example file from above.

Managing drivers

Configure drivers with pre-installed jars

You can customize drivers configuration in the workspace/.metadata/.plugins/org.jkiss.dbeaver.core/drivers.xml file. If you have some pre-installed jar files you can reference them in drivers.xml. Example:

```
<library type="jar" path="absolute-jar-folder-path\driver-jar.jar" custom="true"/>
```

Also in drivers.xml you can use the following variables to specify relative paths:

Variable	Meaning
drivers_home	Standard DBeaver drivers location - ($workspace/drivers by default)
dbeaver_home	DBeaver installation folder
home	User home folder
workspace	DBeaver workspace path

For instance:

```
<library type="jar" path="${workspace}\drivers\my-driver.jar" custom="true"/>
```

Full drivers.xml example:

```
<?xml version="1.0" encoding="UTF-8"?>
<drivers>

    <provider id="postgresql">
        <driver id="postgres-jdbc" custom="false" embedded="false" name="PostgreSQL" class="org.postgresql.Driver" url="jd
bc:postgresql://{host}[:{port}]/[{database}]" port="5432" description="PostgreSQL standard driver">
            <library type="jar" path="maven:/org.postgresql:postgresql:RELEASE" custom="false" version="42.2.20">
                <file id="org.postgresql:postgresql" version="42.2.20" path="${drivers_home}/maven/maven-central/org.postg
resql/postgresql-42.2.20.jar"/>
            </library>
            <library type="jar" path="maven:/net.postgis:postgis-jdbc:RELEASE" custom="false" version="2.5.0">
                <file id="net.postgis:postgis-jdbc" version="2.5.0" path="${drivers_home}/maven/maven-central/net.postgis/
postgis-jdbc-2.5.0.jar"/>
            </library>
            <library type="jar" path="maven:/net.postgis:postgis-jdbc-jtsparser:RELEASE" custom="false" version="2.5.0">
                <file id="net.postgis:postgis-jdbc-jtsparser" version="2.5.0" path="${drivers_home}/maven/maven-central/ne
t.postgis/postgis-jdbc-jtsparser-2.5.0.jar"/>
            </library>
        </driver>
    </provider>

</drivers>
```

Provide predefined drivers configuration

In some cases you may need to provide a driver's configuration or driver `jar` files for a number of DBeaver installations automatically. This can be done by adding a special parameter in the `dbeaver.ini` file: `-Ddbeaver.drivers.configuration-file=c:\some-path\dbeaver-drivers-config.xml`

This file has the same structure as `drivers.xml` file (see above) and it will be loaded *before* `drivers.xml`.

You can specify partial driver configuration. For example if you need to configure only the jar path then it may look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<drivers>
  <provider id="generic">
    <driver id="netezza">
      <library type="lib" path="X:\jdbc-drivers\netezza-jdbc.jar"/>
    </driver>
  </provider>
</drivers>
```

Windows Silent Install

It is possible to install DBeaver in silent mode using the Windows Installer command line parameters.
This might be very useful for mass install automation (SSCM and other similar systems).
Installer was improved in DBeaver 5.3.3, special thanks to the https://github.com/Drizin/NsisMultiUser team.

Parameters

Command line parameters supported by DBeaver installer:

Parameter	Description
/S	silent mode, requires /allusers or /currentuser, case-sensitive
/D=path	(installer only) set install directory, must be last parameter, without quotes, case-sensitive
/allusers	(un)install for all users, case-insensitive
/currentuser	(un)install for current user only, case-insensitive
/uninstall	(installer only) run uninstaller, requires /allusers or /currentuser, case-insensitive

In order to install with the `/allusers` parameter the current user must have the administrator's permission.

Installer return codes (decimal):

Code	Meaning
0	normal execution (no error)
1	(un)installation aborted by user (Cancel button)
2	(un)installation aborted by script
666660	invalid command-line parameters
666661	elevation is not allowed by defines
666662	uninstaller detected there is no installed version
666663	executing uninstaller from the installer failed
666666	cannot start elevated instance
other	Windows error code when trying to start elevated instance"

Snap installation

Installation

For install *dbeaver-ce* snap package with stable release (stable by default):

```
snap install dbeaver-ce --stable
```

Connect interfaces:

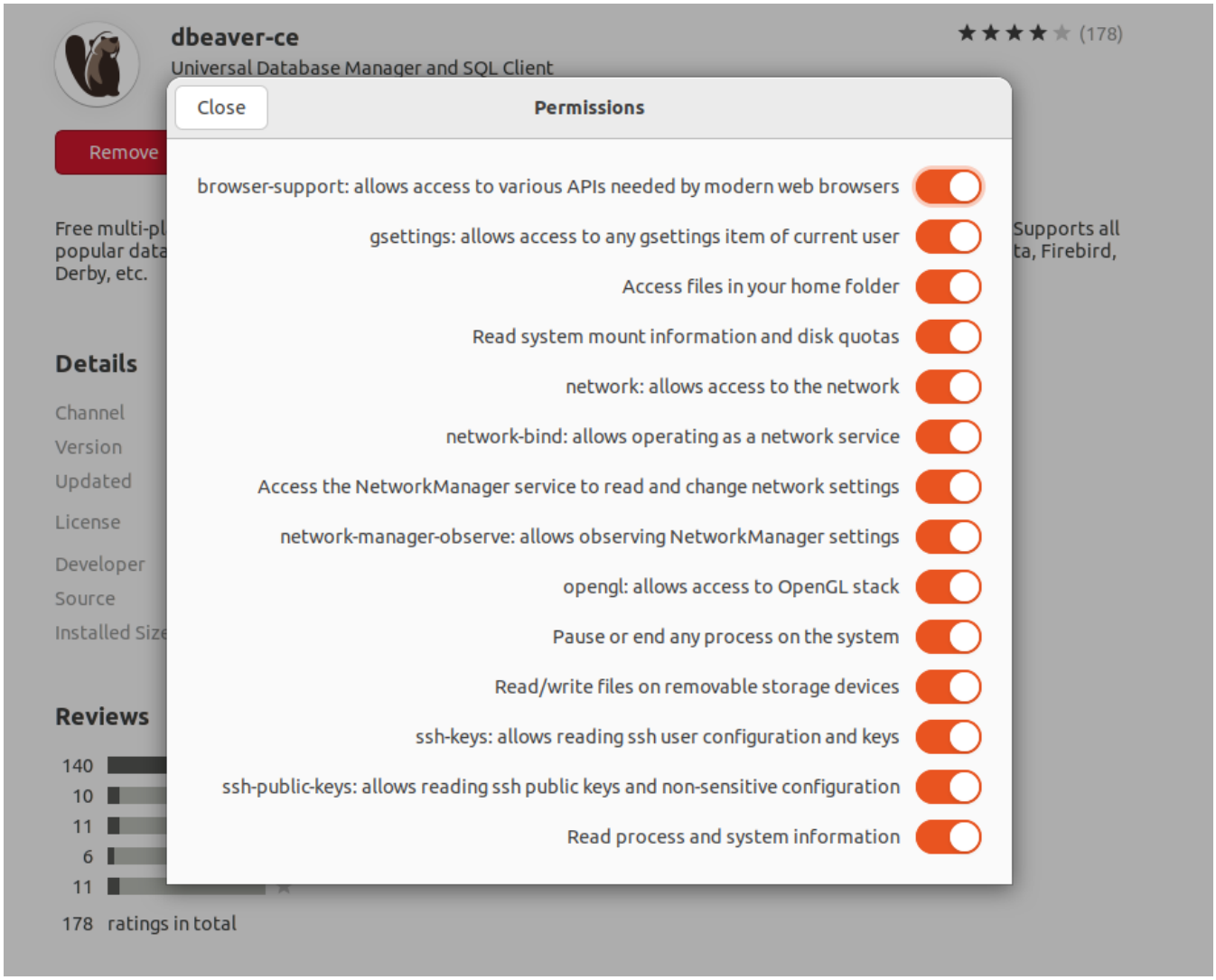
You can find list of all *dbeaver-ce* snap connections:

```
snap connections dbeaver-ce
```

And connect what you need. For example:

`snap connect dbeaver-ce:ssh-keys` for access to private keys

Or you can do it in Ubuntu Software (dbeaver-ce - Permissions):



Problems with dbeaver-ce snap package:

At the moment there is a well-known problem with opening the browser and gis maps via *dbeaver-ce* snap package. Application crashes with an error:

```
SWT WebKitGDBus: error creating DBus server Error binding to address (GUnixSocketAddress): Permission denied
SWT WebKit: error initializing DBus server, dBusServer == 0
```

Snapcraft forum topic: https://forum.snapcraft.io/t/classic-confinement-for-dbeaver-ce/27502

Workaround

While the problem is being fixed, you can use the **workaround**:

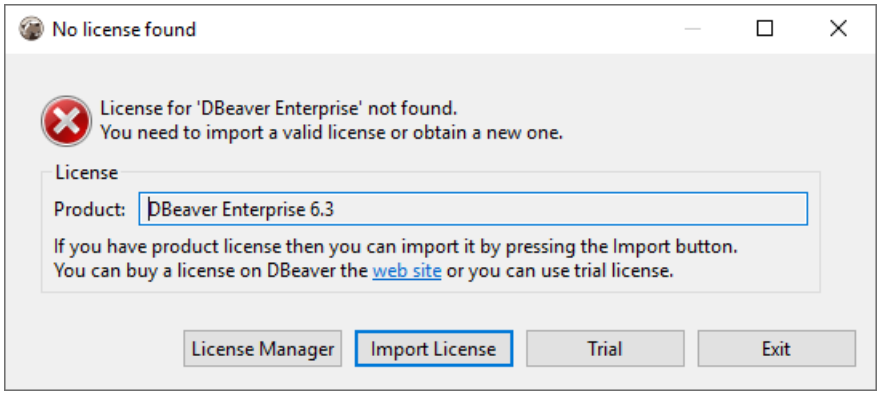
Run *dbeaver-ce* snap package from: `/snap/dbeaver-ce/current/usr/share/dbeaver-ce/dbeaver`

License Administration

Note: This feature is available in Lite, Enterprise, and Ultimate editions only.

Manual license import

Commercial versions of DBeaver ask the user to import the license file if they cannot find it locally. It is the most simple and convenient way to import the product license for individual users.



License management automation

There are several ways to automate the license management process. It makes sense for a multi-user environment.

Put the license file to the predefined locations

While DBeaver is starting up, it will look for a license file in the following locations:

- Windows
  - `%HOMEPATH%\.dbeaver-ee-license.dat`
  - `%APPDATA%\DBeaverData\workspace6\.metadata\.dbeaver-ee-license.dat`
- MacOS X
  - `~/.dbeaver-ee-license.dat`
  - `~/Library/DBeaverData/workspace6/.metadata/.dbeaver-ee-license.dat`
- Linux
  - `~/.dbeaver-ee-license.dat`
  - `$XDG_DATA_HOME/DBeaverData/workspace6/.metadata/.dbeaver-ee-license.dat`

Passing license file through command line

You can add the command line parameter `license <license-path>` to the DBeaver EE shortcut. Also, you can add this parameter to the `dbeaver.ini` config file.

Command line reference.

How to Import License

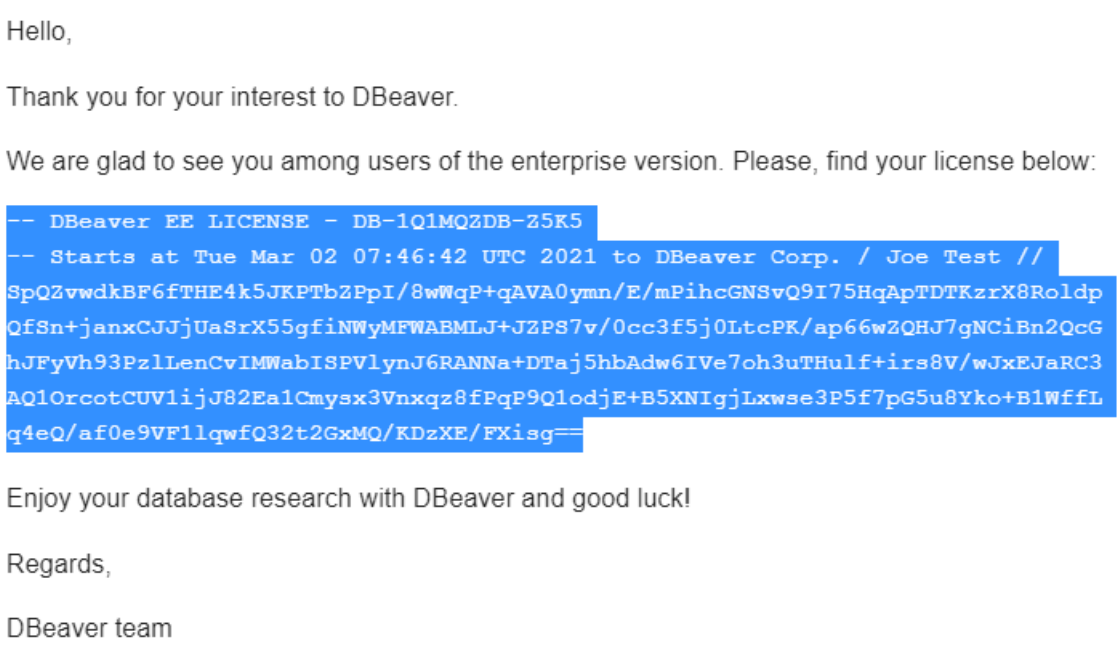
To start using commercial versions of DBeaver you can

- request a Trial license for 2 weeks;
- request an Academic license if you are a student or a teacher;
- buy a Subscription license, Standard DBeaver license or DBeaver license extension.

After purchasing the DBeaver license or getting the Trial/Academic license, you will receive a License text by email. It will also be available in your personal account on our site. This License text will contain your License ID e. g. DB-821MPZFO-ZA8W, the start date and license owner's name and company name. It is very important to import your License correctly.

Import from email

You can just copy-paste the License Key to import the license into the License Manager. Please note that you need to copy-paste the full license text (not just license ID). The license text starts with "–" and ends with "==" characters.



Hello,

Thank you for your interest to DBeaver.

We are glad to see you among users of the enterprise version. Please, find your license below:

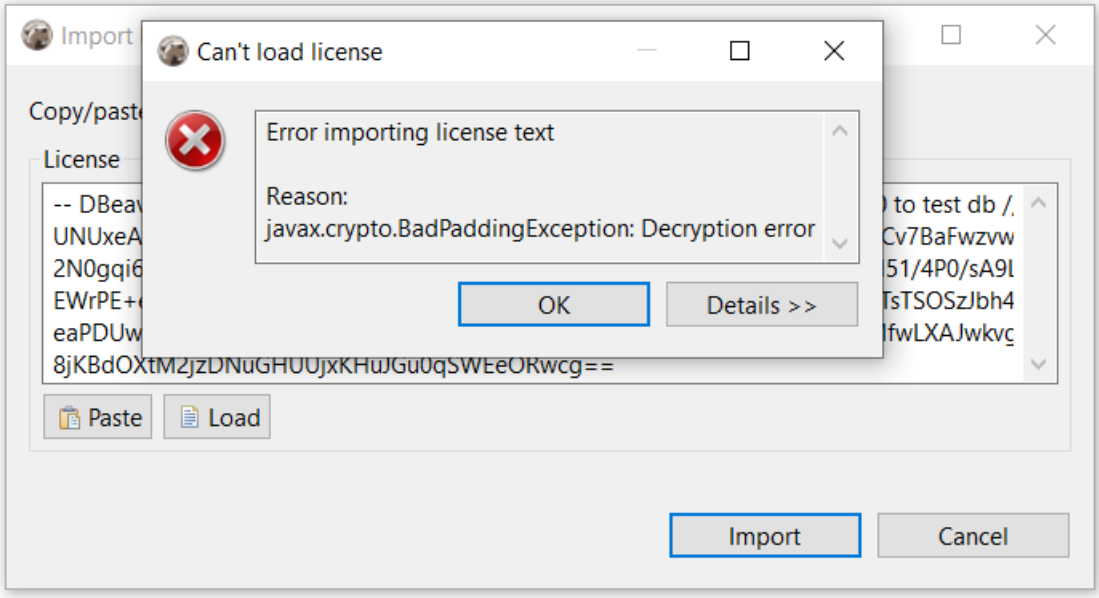
```
-- DBeaver EE LICENSE - DB-1Q1MQZDB-Z5K5
-- Starts at Tue Mar 02 07:46:42 UTC 2021 to DBeaver Corp. / Joe Test //
SpQZvwdkBF6fTHE4k5JKPTbZPpI/8wWqP+qAVA0ymn/E/mPihcGNSvQ9I75HqApTDTKzrX8Roldp
QfSn+janxCJJjUaSrX55gfiNWyMFWABMLJ+JZPS7v/0cc3f5j0LtcPK/ap66wZQHJ7gNCiBn2QcG
hJFyVh93PzlLenCvIMWabISPVlynJ6RANNa+DTaj5hbAdw6IVe7oh3uTHulf+irs8V/wJxEJaRC3
AQ1OrcotCUV1ijJ82Ea1Cmysx3Vnxqz8fPqP9Q1odjE+B5XNIgjLxwse3P5f7pG5u8Yko+B1WffL
q4eQ/af0e9VF1lqwfQ32t2GxMQ/KDzXE/FXisg==
```

Enjoy your database research with DBeaver and good luck!

Regards,

DBeaver team

Sometimes an email client can corrupt the formatting of the License Key that can cause an error.

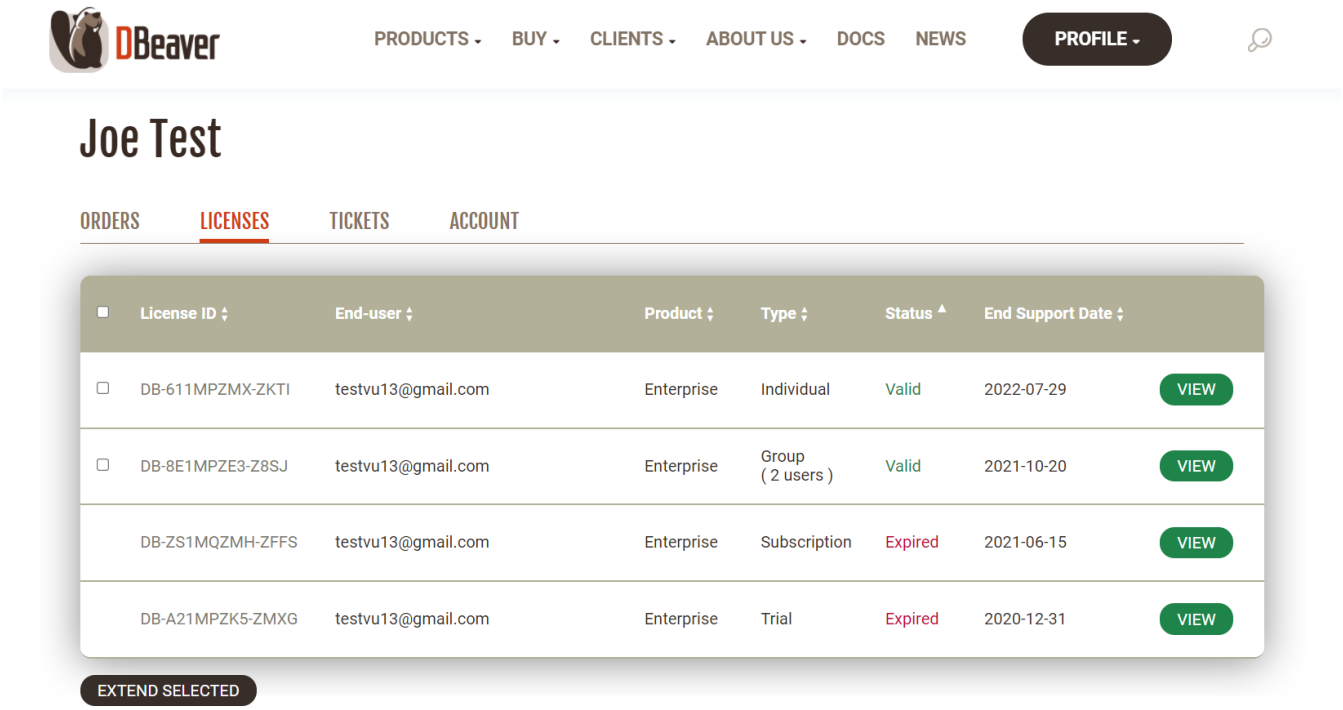


Therefore, you need to import your License Key from your personal account on our site https://dbeaver.com/.

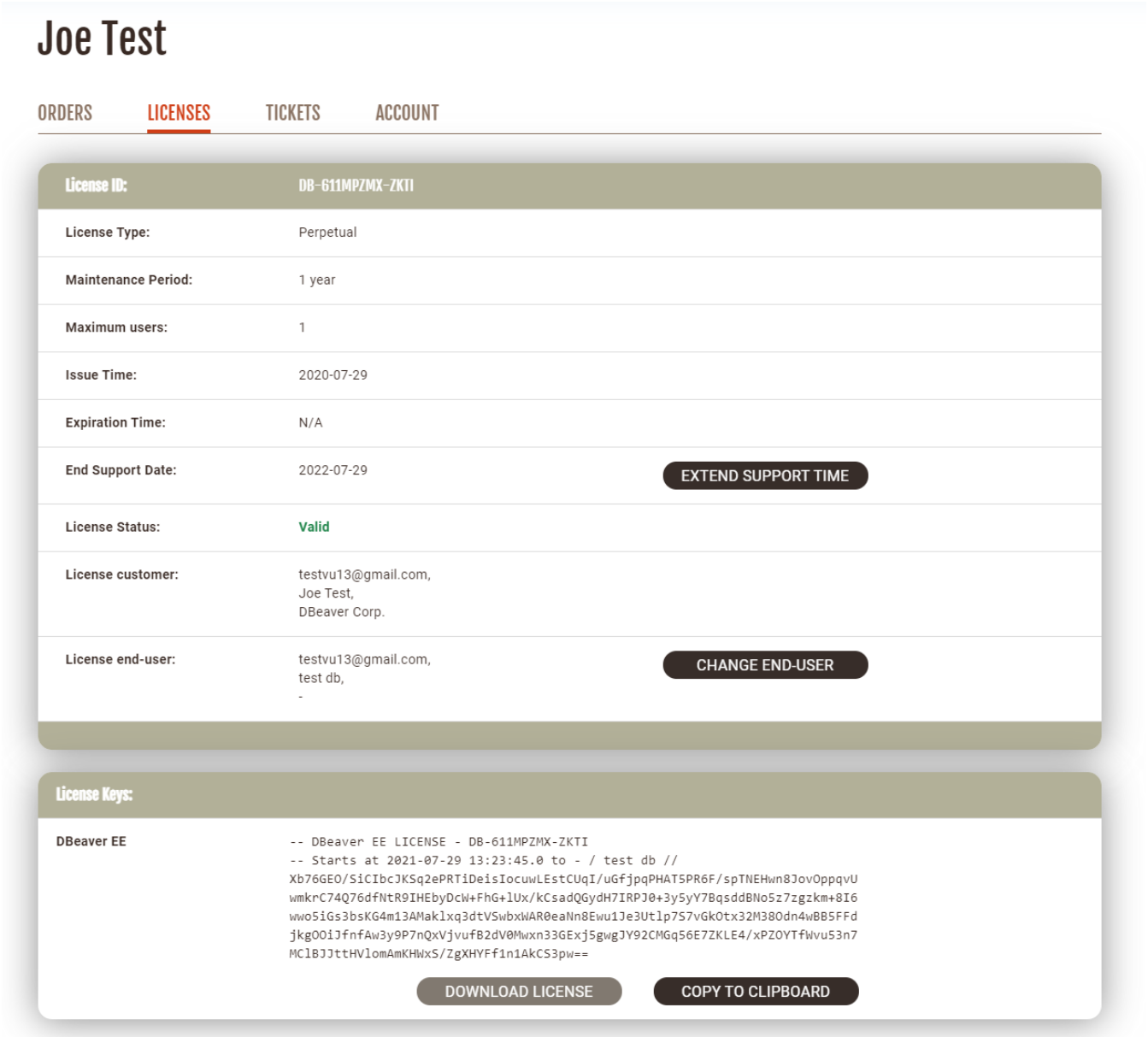
Import from the personal account

Firstly, you need to Sign in.

Secondly, you should open the **Licenses** tab, where you can find all your licenses.

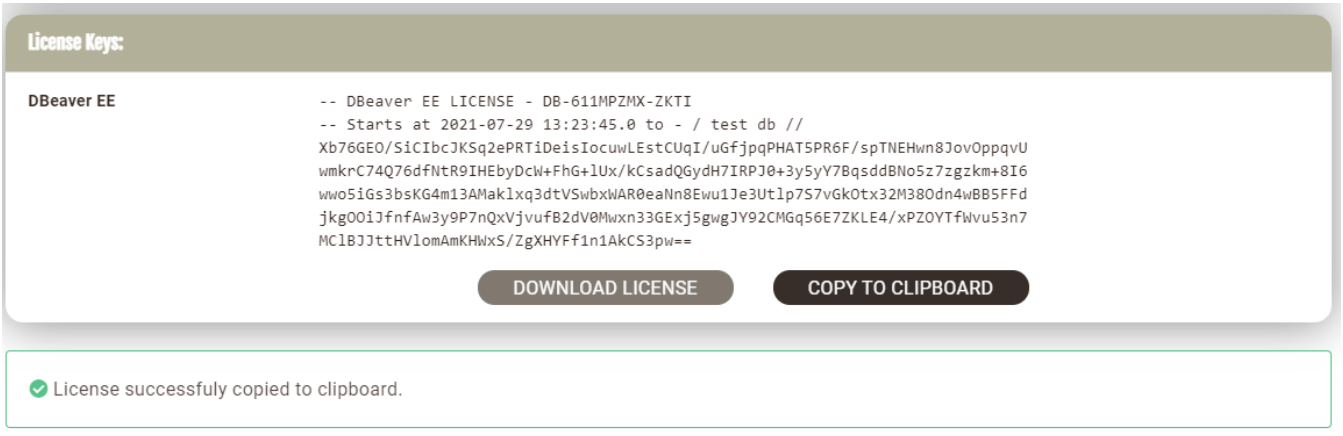


To open the License details and copy the License Key press the **View** button. You can see the License details where you can find your license status, type, maintenance period, and end support date. You can also reassign the license to another user.



At the bottom of the page you can find the License Key required to start using DBeaver. There are two options how to copy your License Key from the personal account:

1)Press the **COPY TO CLIPBOARD** button, then press OK. The license text will be copied to the clipboard.



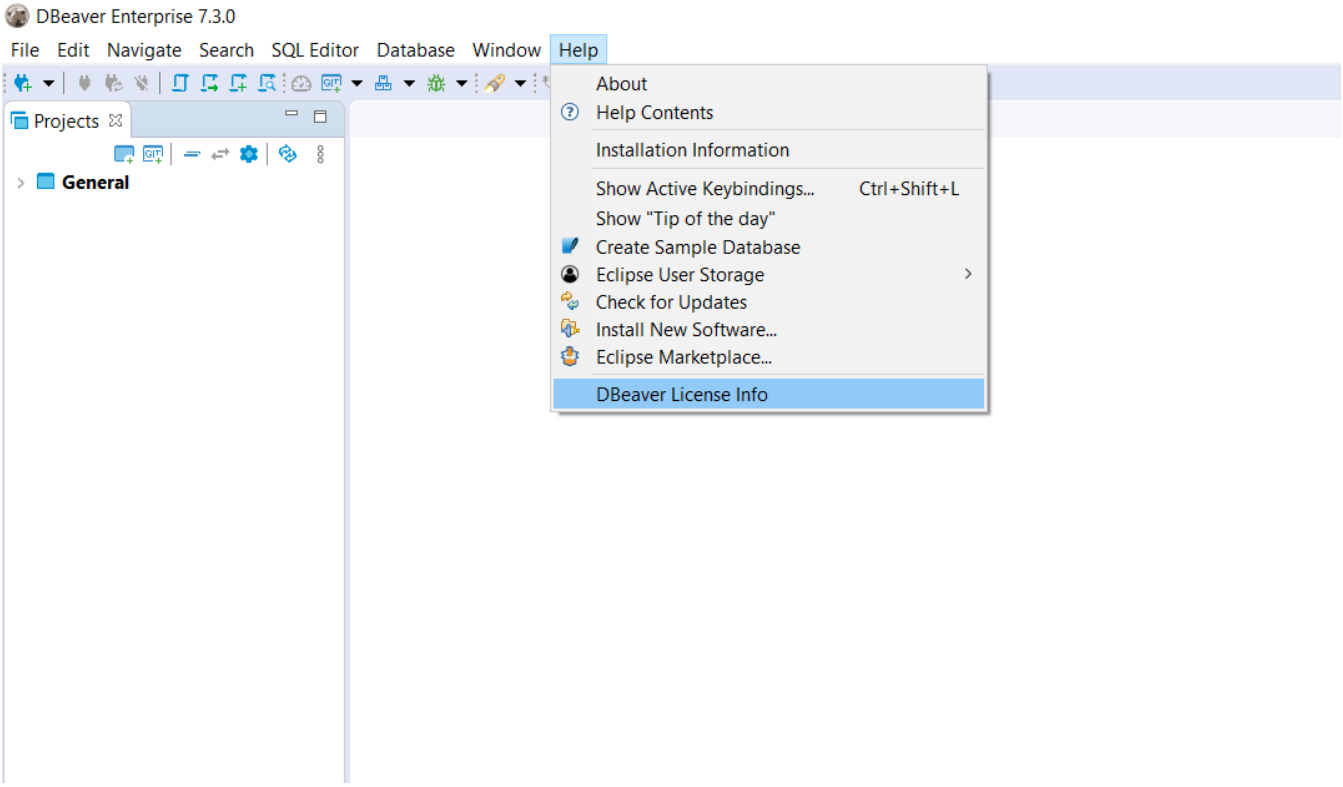
2)Press the **DOWNLOAD LICENSE** button, then press OK.

.txt file with your License Key will be downloaded to your download folder. The file name is License ID, e. g. DB-821MPZFO-ZA8W.

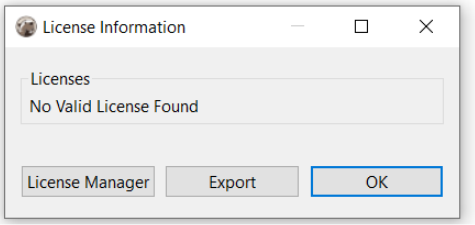
Then you need to insert the copied License Key to License Manager in DBeaver.

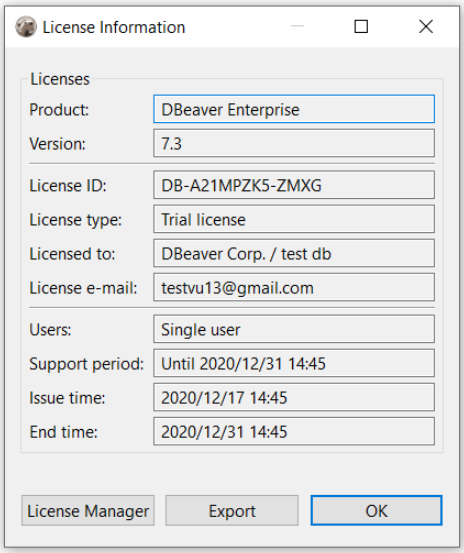
Insert the License key to License Manager

To start using commercial versions of DBeaver with your License Key you need to open License Manager in DBeaver: Help -> DBeaver License Info

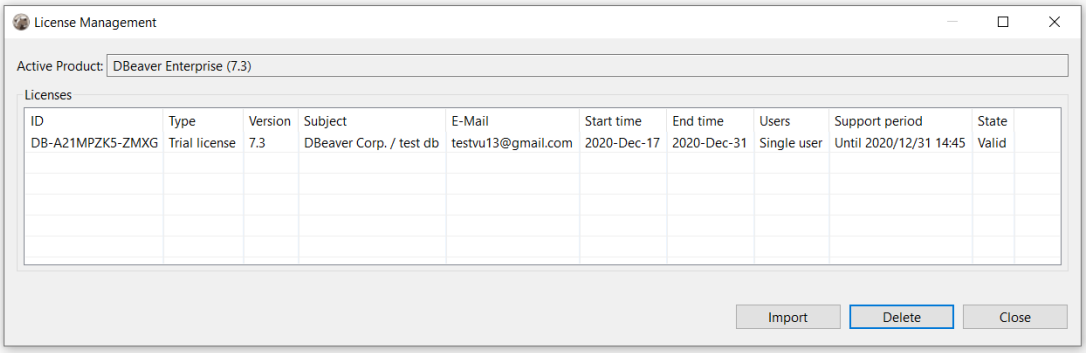


The License information window can look different depending on whether you already have a valid license or not.

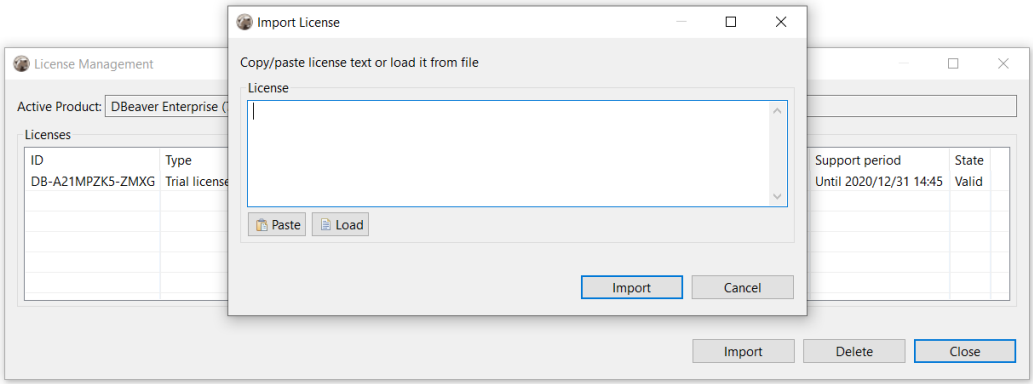




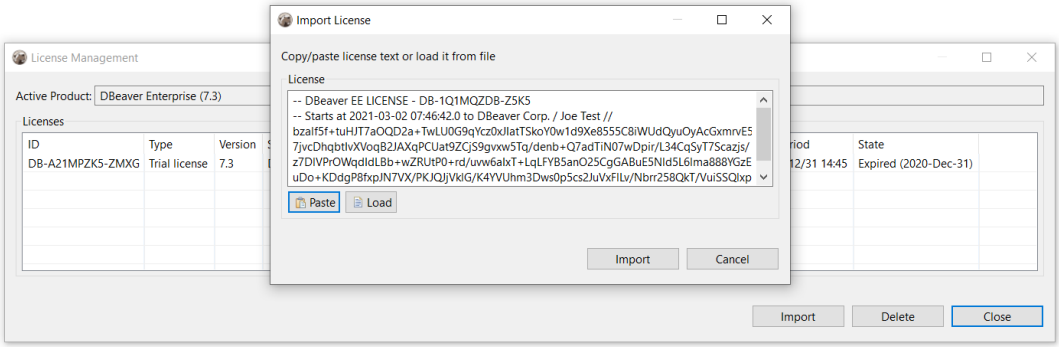
Then click **License Manager** where you can find all your imported licenses and information about them.



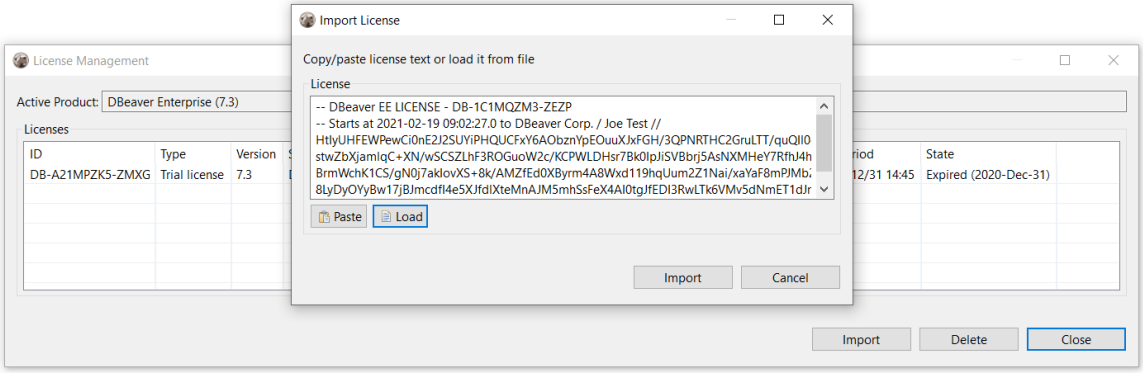
Next press the **Import** button to paste your License Key.



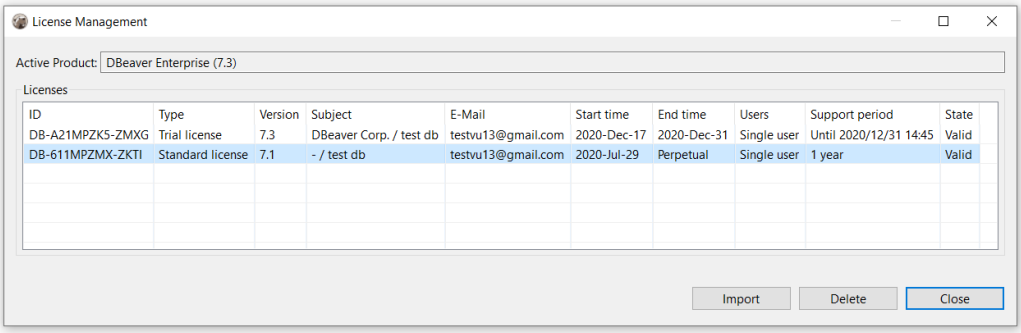
If you copied the License Key to the clipboard, press the **Paste** button and then **Import**.



If you downloaded the .txt file with the License Key, press the **Load** button and select the file from the Downloads. The License Key will be pasted automatically.



Then press the **Import** button and your license will be added to the License Manager.



You have successfully imported your license. Now you can close the License Manager and start using DBeaver.

Import of Subscription license

Subscription license requires internet access on the workstation for the first activation and each prolongation.

If you do not have an active internet connection or work behind a corporate firewall while importing the Subscription license, the following error can appear:

Invalid subscription

Can`t find the subscription information for license 'DB-821MPZFO-ZA8W'.

Check your internet connection and/or firewall settings and restart application.

In this case you need to check that DBeaver has internet access or you will need to configure your firewall.

License extension

The standard DBeaver license is a perpetual license with a limited period of support (1 year or 2 years).

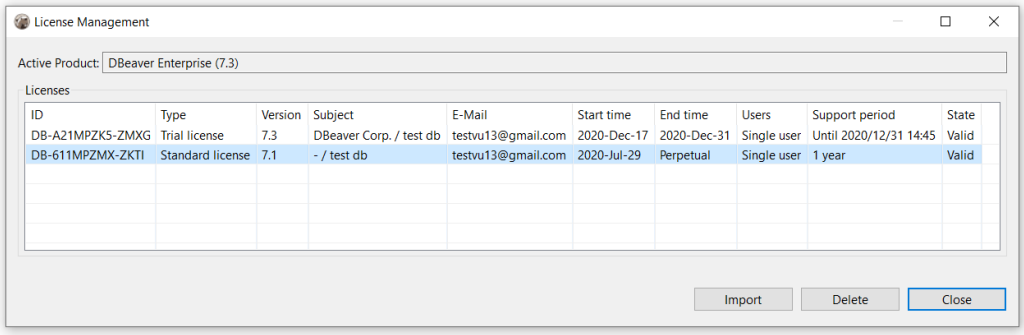
After the end of the selected support period you can continue to use commercial versions of DBeaver without support and updates or buy a license extension or a new license.

If you buy the DBeaver license extension and DBeaver has internet access, the license in DBeaver will be updated automatically. Otherwise, you have to import the license key from your personal account once again.

License Manager

License Manager provides you with the following information about your licenses:

- License ID e. g. DB-821MPZFO-ZA8W;
- License type: Trial/Academic/Subscription/Standard;
- Version;
- License owner`s name and company name;
- License owner`s email;
- Start time is the date the license was received;
- End time is the date the license expires (standard perpetual licenses do not have this)
- Number of users: single user or multiuser for group licenses;
- Support period is the period you have access to the internal support system on the site and the possibility to download new product versions;
- State: valid or expired.



How to Reassign License

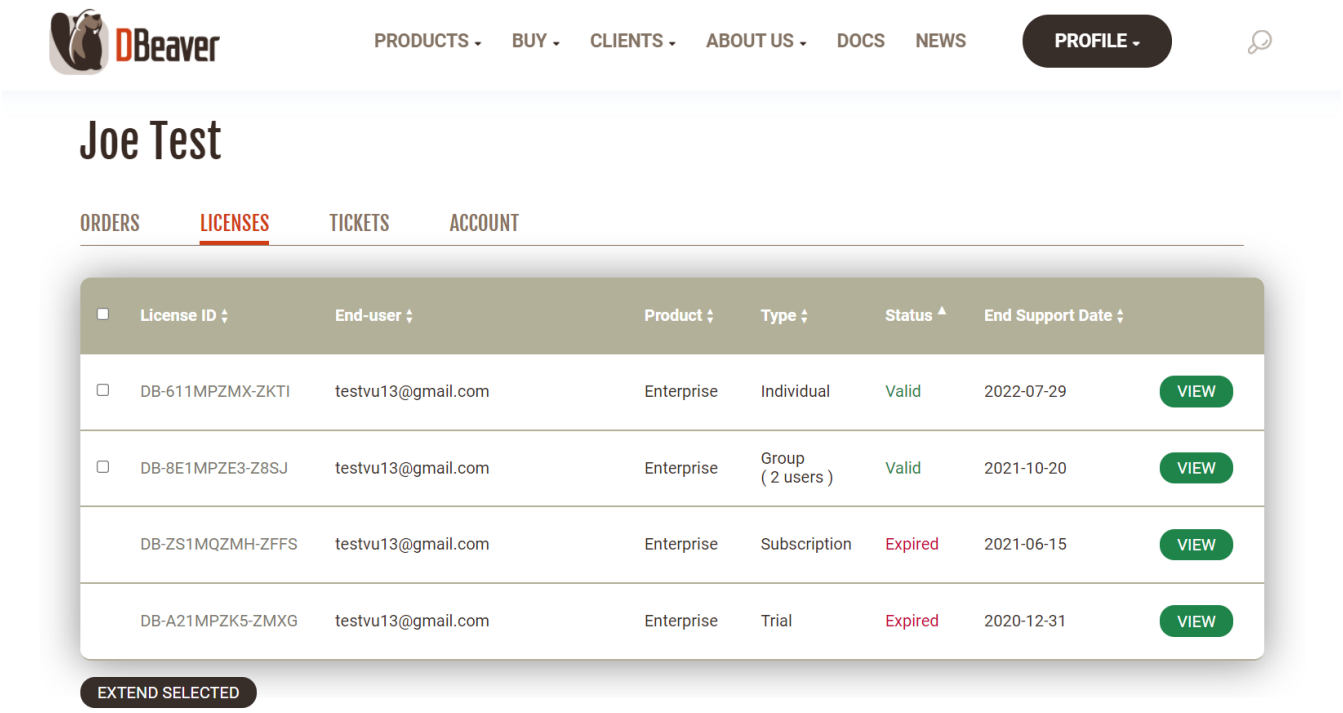
After purchasing a bunch of DBeaver EE licenses, you have to assign each license to an end user.

If an employee subsequently is leaving the company or the team that is using DBeaver, the license admin may need to reassign the license to another employee.

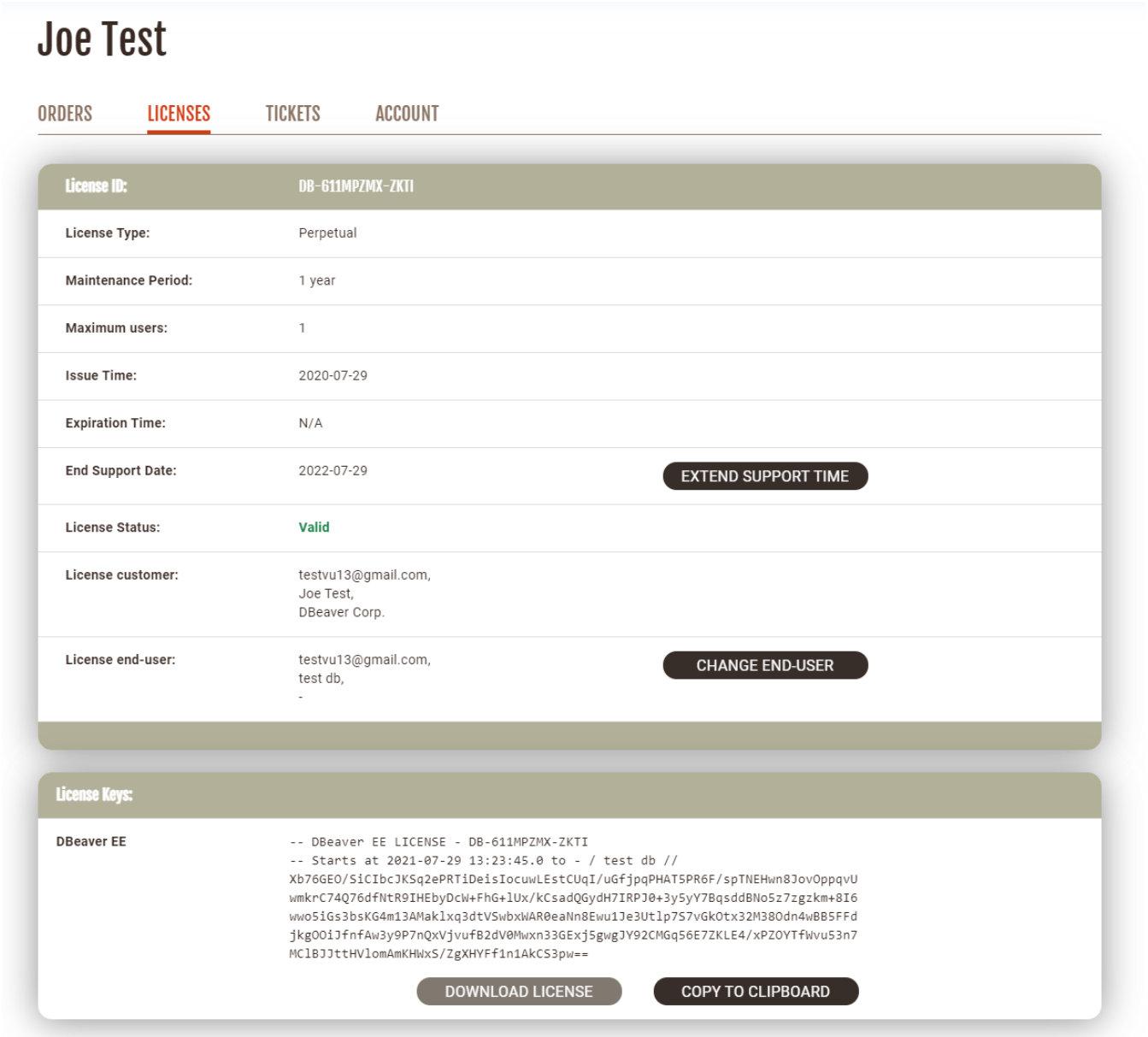
You can reassign the license to another user in your personal account.

Firstly, you need to Sign in.

Secondly, you should open the **Licenses tab**, where you can find all your licenses.

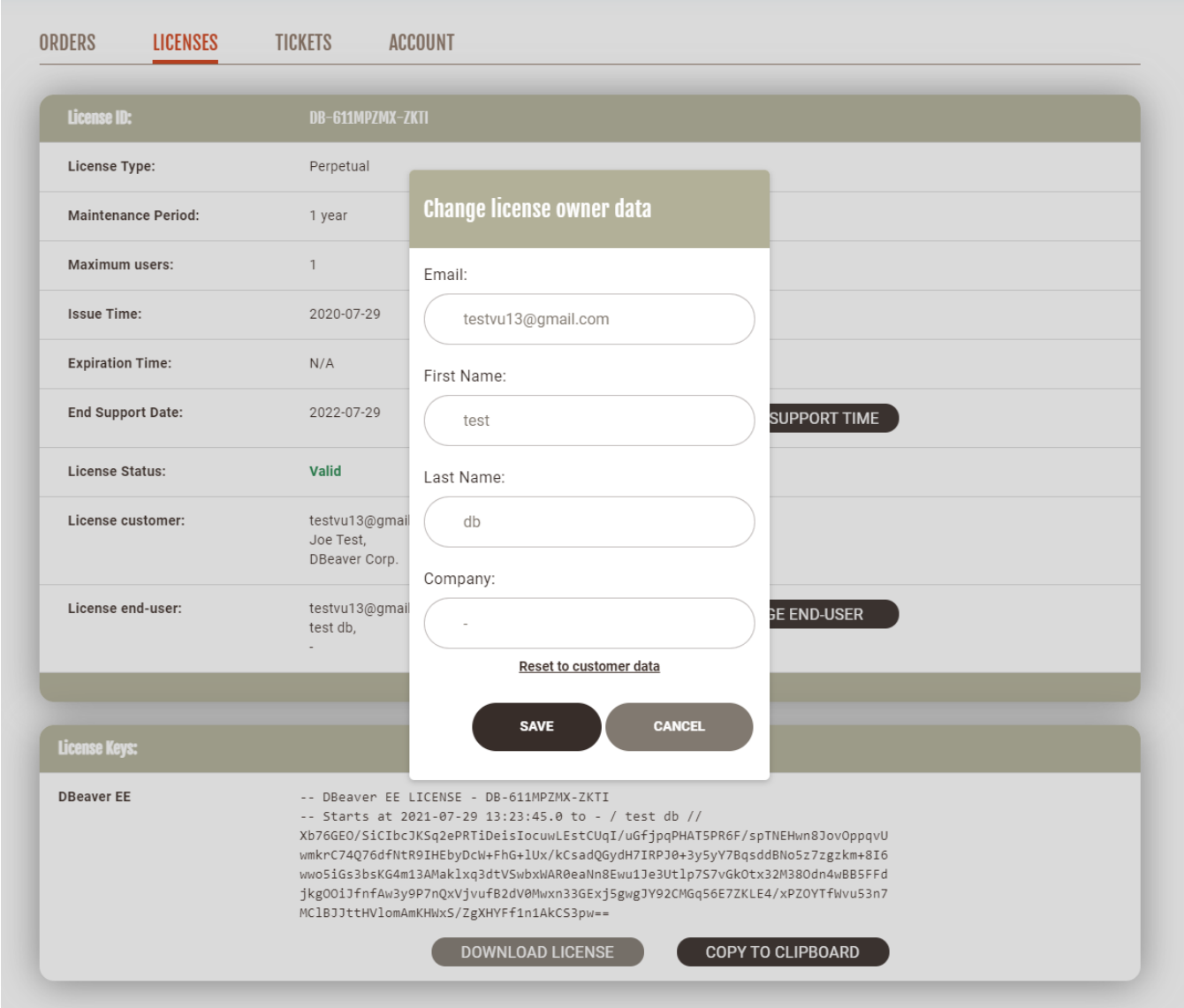


Select which license you need to reassign and press the **VIEW** button near its license ID. You can see the License details where you can find your license status, type, maintenance period, and end support date.

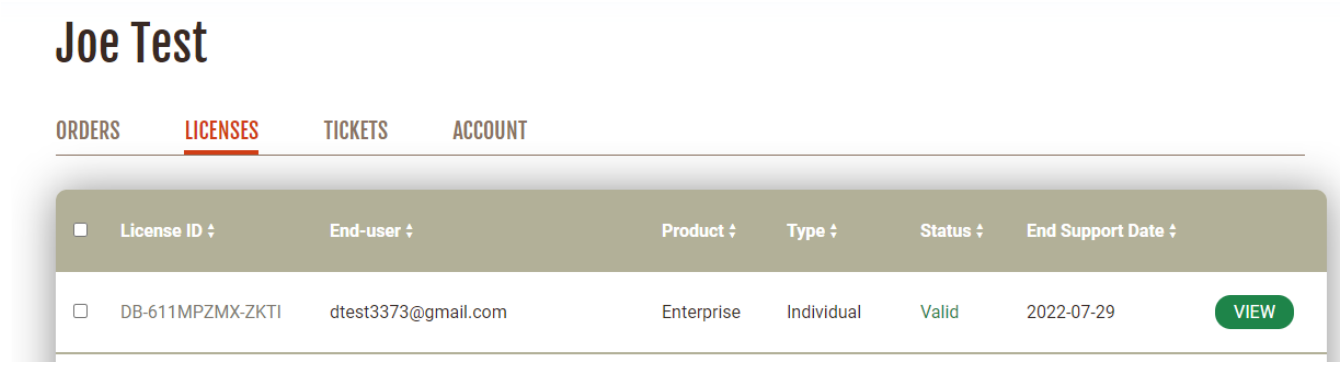


Also, there is the license end user field that contains the license end user's details. To reassign the license, you need to press the **CHANGE END USER** button near this field.

A pop-up opens, and you can enter the new license end user's data: email; first and last name; and company. After filling in the form, you need to press the **SAVE** button and the license owner will be changed. The license key that contains the license end user's name and company name will be changed too.



The license will remain in your personal account. The end user's email will be in the end user column on the Licenses tab.



Also, the new license end user will be able to find the license in the personal account and import it. There will be no CHANGE END USER button because it is only the license customer who can assign a license to the end user. If the new license end user has not been signed up on our website, the account will be created automatically. The new user will receive a welcome email with a link to set up a password.

Connecting to Oracle Database using JDBC OCI driver

This article discusses how to establish connections to an Oracle database using JDBC OCI (Type II). Please take into consideration that the proposed way uses DBeaver's Generic driver. It means that you cannot get Oracle-specific functionality this way.

Prerequisites

JDBC OCI connections require Oracle Instant Client on the local machine. Please pay attention to the Instant Client and the JDBC driver versions, as they must be identical. DBeaver uses the 12.2.0.1 version by default at the moment, so we recommend using the 12.2.0.1 version of the Instant Client.

Install the Instant Client into some folder. We will refer to this folder as ORA_HOME for the rest of the article. Append ORA_HOME to the PATH variable and restart DBeaver before proceeding.

Configuration

1. Place your *tnsnames.ora* file into ORA_HOME/network/admin directory.
2. In DBeaver, click Window -> Driver Manager -> New. This opens Create new driver dialog.
3. In the Settings tab, add a *Driver name* of your liking. Set *Class Name* to 'oracle.jdbc.OracleDriver'. Set URL Template to 'jdbc:oracle:oci:@tnsAlias', where 'tnsAlias' is an alias from your *tnsnames.ora* file. Make sure that the Driver Type is set to Generic.
4. In the Libraries tab, you need to add Maven artifacts. To do that, click Add Artifact. Paste the following XML into the text field:

```
<dependencies>
    <dependency>
        <groupId>com.oracle.database.jdbc</groupId>
        <artifactId>ojdbc8</artifactId>
        <version>12.2.0.1</version>
    </dependency>
    <dependency>
        <groupId>com.oracle.database.nls</groupId>
        <artifactId>orai18n</artifactId>
        <version>12.2.0.1</version>
    </dependency>
    <dependency>
        <groupId>com.oracle.database.xml</groupId>
        <artifactId>xdb6</artifactId>
        <version>12.2.0.1</version>
    </dependency>
    <dependency>
        <groupId>com.oracle.database.xml</groupId>
        <artifactId>xmlparserv2</artifactId>
        <version>12.2.0.1</version>
    </dependency>
</dependencies>
```

NB: Replace the versions of the artifacts if you use a different version of the Instant Client.

5. In the Driver properties tab, make right-click -> Add new property.
6. Set the property name to 'protocol' (without quotes). Set the Value to 'oci' (without quotes).
7. Close the Driver manager.
8. Create a new connection using your newly configured driver.

Importing CA Certificates into DBeaver

When working in DBeaver under a corporate network, you can encounter an error similar to the following:

```
javax.net.ssl.SSLHandshakeException:PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException:
unable to find valid certification path to requested target.
```

Here is how you can bypass it.

Import certificates from your local Java

It's possible that your system administrator has installed a local Java and imported the required certificates to its keystore. We can use them to fix the issue.

Step 1: locate your Java

Windows

Press Windows + R to open the Run window. Type `cmd` in the prompt and press OK. It will open the command prompt. In the command prompt, type the following and press Enter:

```
where java
```

macOS

Open the Terminal and execute the following command:

```
/usr/libexec/java_home -V
```

Linux

Open a terminal and execute the following command:

```
readlink -e /usr/bin/java
```

Step 2: Find the JRE in DBeaver's installation

It's pretty easy. Just find the path where you installed DBeaver and open the `jre` folder there.

Step 3: Copy the cacerts

Open the folder with the Java you found in step 1. Locate the `cacerts` files under `/lib/security`, then copy-paste it into `<PATH_FROM_STEP_2>/lib/security`, replacing the old file. Restart DBeaver, and you are ready to go.

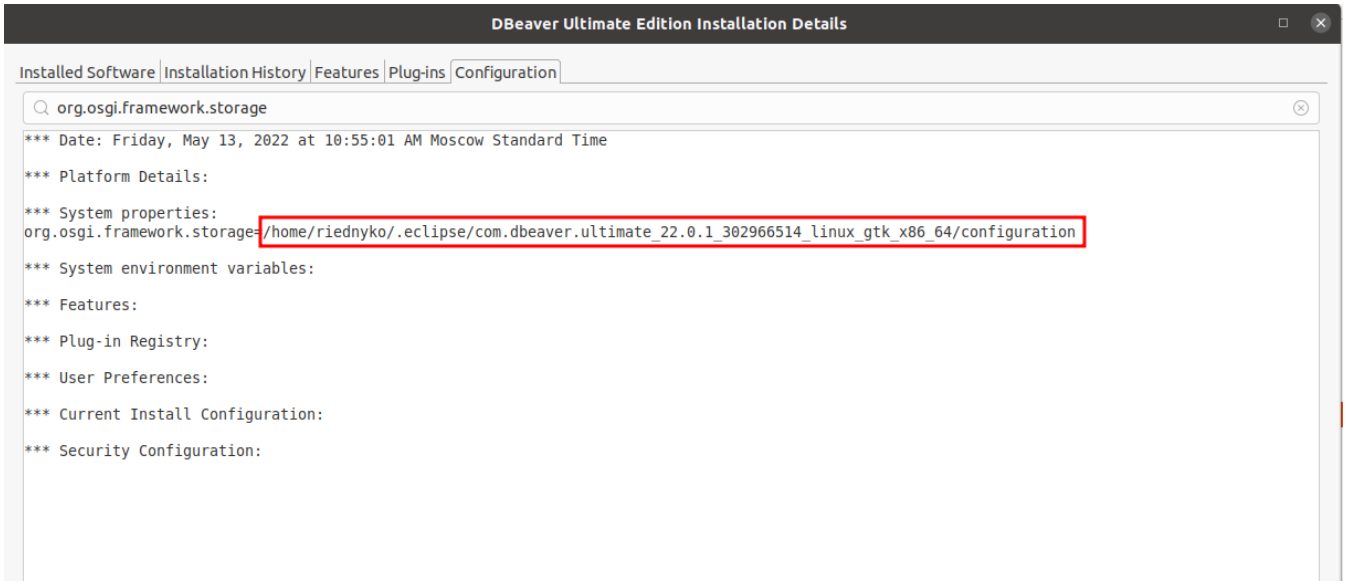
How to set a variable if dbeaver.ini is read-only

If you need to set some variable and `dbeaver.ini` have read-only permissions (this can happen, for example, in Flatpak or Snap), you can use the `config.ini`.

config.ini path:

For Linux:

You can find correct path to your configuration directory with **config.ini** when open DBeaver `Help -> Installation Information -> Configuration` then ***type filter text*** `org.osgi.framework.storage`



If **config.ini** does not exist, you can create it in the configuration directory `nano config.ini`

For Windows:

You can find correct path to your configuration directory with **config.ini** when open DBeaver `Help -> Installation Information -> Configuration` then ***type filter text*** `org.osgi.framework.storage`

Example output:

```
org.osgi.framework.storage=file:/C:/Users/user/.eclipse/org.jkiss.dbeaver.product_22.0.5_1535670467_win32_win32_x86_64
```

If **config.ini** does not exist, you can create it in the configuration directory.

config.ini example

For example, you can set system property with your custom path to point to the keystore you created.

`javax.net.ssl.trustStore=/path/to/your/cert`

Or, for example, you can change the language.

`osgi.nl=fr`