

Documentation

Table of contents

User Table of Contents

Installation

Application Window Overview (Application Window Overview)

Views

- Database Navigator (Database Navigator)

 - Filter Database Objects (Filter Database Objects)

 - Configure Filters (Configure Filters)

- Projects

 - Project Explorer (Project Explorer)

Database Connections (Database Connections)

- Connecting to Databases (Connecting to Databases)

- Editing Connections (Editing Connections)

Database Object Editor (Database Object Editor)

- Properties Tab (Properties Tab)

- Data Tab (Data Tab)

- ER Diagram Tab (ER Diagram Tab)

Data editor

- Filters (Data-filter)

- Working with XML (Working with XML)

SQL Editor

- Client side commands (Client-side-scripting)

- SQL Templates (SQL-Editor-Templates)

Database drivers (Database drivers)

SQL databases

NoSQL databases

- MongoDB

Cassandra (NoSQL-Cassandra)

Redis (NoSQL-Redis)

Optional extensions

Office

SVG format

Miscellaneous

Diagrams (ERD)

Query Manager

Data transfer (export/import) (Data-transfer)

Data/metadata search (Database-Search)

Database compare (Database-compare)

Command-Line

Enterprise Edition (Enterprise Edition)

Projects Management

Admin Guide

Managing connections

Managing drivers

Development

Develop in Eclipse

Features structure

Troubleshooting

Reset workspace

Posting issues

Log files

Thread dump

Installation

The installation process depends on the distribution type and your Operational System - see <https://dbeaver.jkiss.org/download/> (<https://dbeaver.jkiss.org/download/>)

Windows / MacOS Installer

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

NOTE:

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

ZIP Archive

When installing DBeaver manually, without using an installer:

1. Install Java 1.8
(<http://www.oracle.com/technetwork/java/javase/downloads/jre-downloads-2133155.html>) as a separate step.

2. Extract the contents of the archive.

NOTE: Do not unzip the archive over a previous DBeaver version. If you already have any version of DBeaver extracted in the same location - remove it before unzipping the new

version.

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

3. Run the **dbeaver** executable.

Debian Package

To install DBeaver using a Debian package:

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

RPM Package

To install DBeaver using an RPM package:

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

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

MacOS DMG

You could try DMG or ZIP archive if you can't use PKG installer for some reason.

To install DBeaver on MacOS, just drag-and-drop the DMG archive to your disk.

NOTE: DBeaver requires Java 1.8 to be installed, so you need to install JDK 1.8+

(<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>) prior to installing DBeaver.

You can also use brew (<http://macappstore.org/dbeaver-community/>):

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

Views

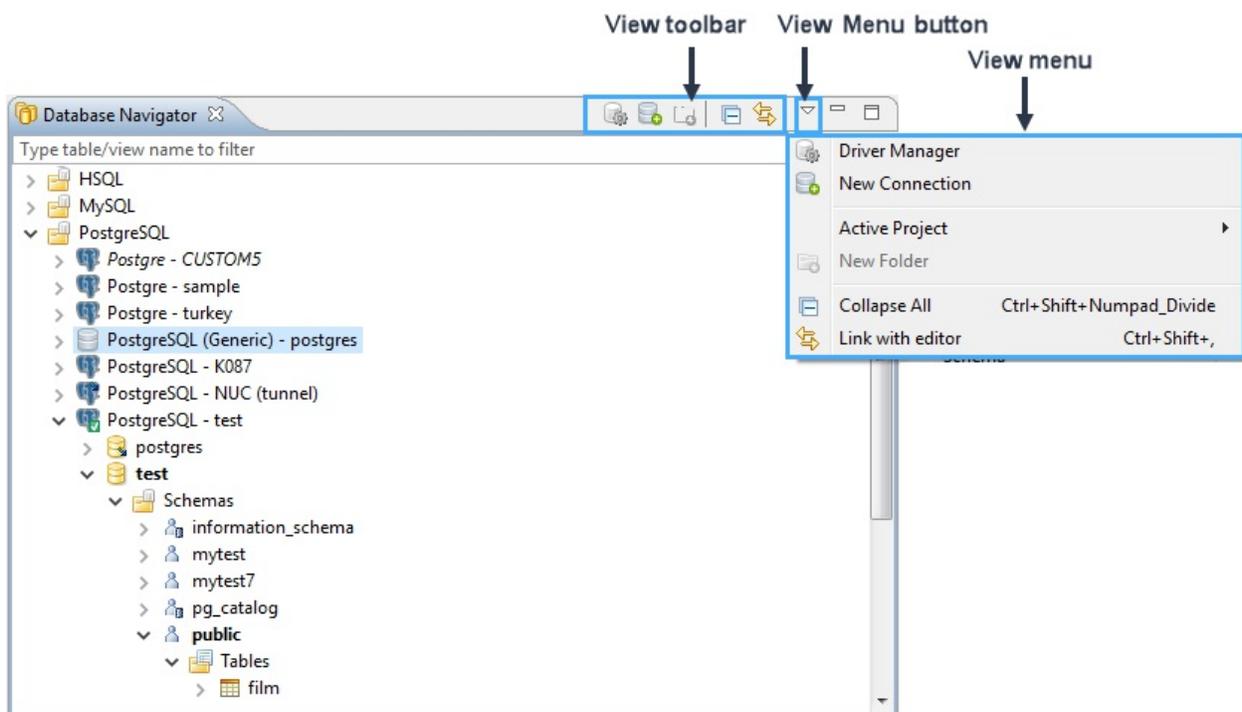
Views are windows within the workspace that provide presentations and ways to navigate the information. The main views in DBeaver are: Database Navigator (<https://github.com/serge-rider/dbeaver/wiki/Database-Navigator>), Projects (<https://github.com/serge-rider/dbeaver/wiki/Projects>) and Project Explorer (<https://github.com/serge-rider/dbeaver/wiki/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.

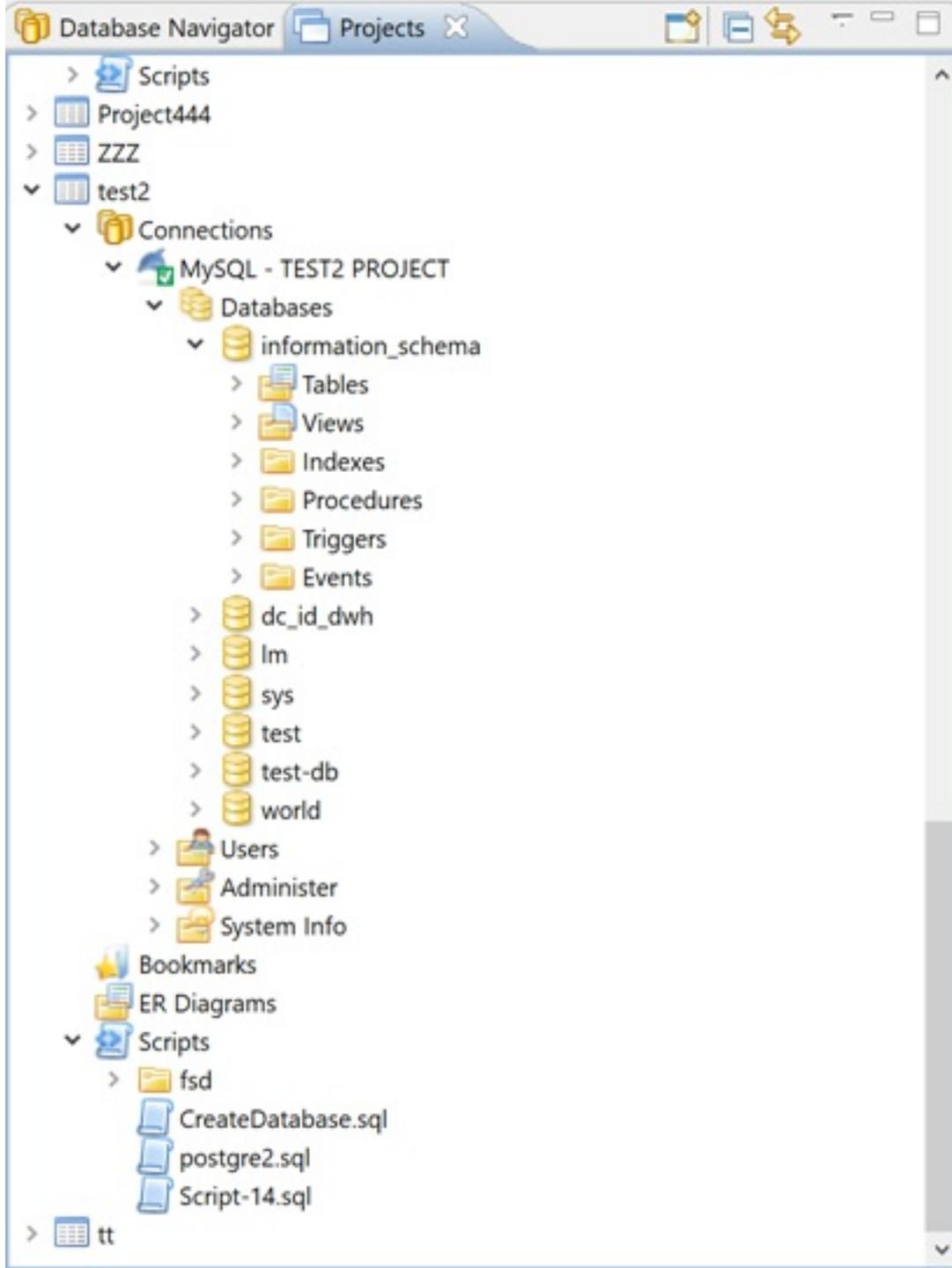
Views also provide context menus for objects they display. To open a context menu for an object, right-click the object.

Projects

Projects

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 (<https://github.com/serge-rider/dbeaver/wiki/Application-Window-Overview>) article.

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

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

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

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

Icon	Item	Description
	Create Project	Opens the Create Project wizard
	Refresh Projects	Refreshes the projects tree to display changes caused by creating modifying or deleting projects
	Collapse All	Collapses the tree to the root level
	Link with editor	<ul style="list-style-type: none"> - Enabled when at least one editor is open, otherwise disabled - Highlights the object in the tree that

		has its editor open
--	--	---------------------

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 (<https://github.com/serge-rider/dbeaver/wiki/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.

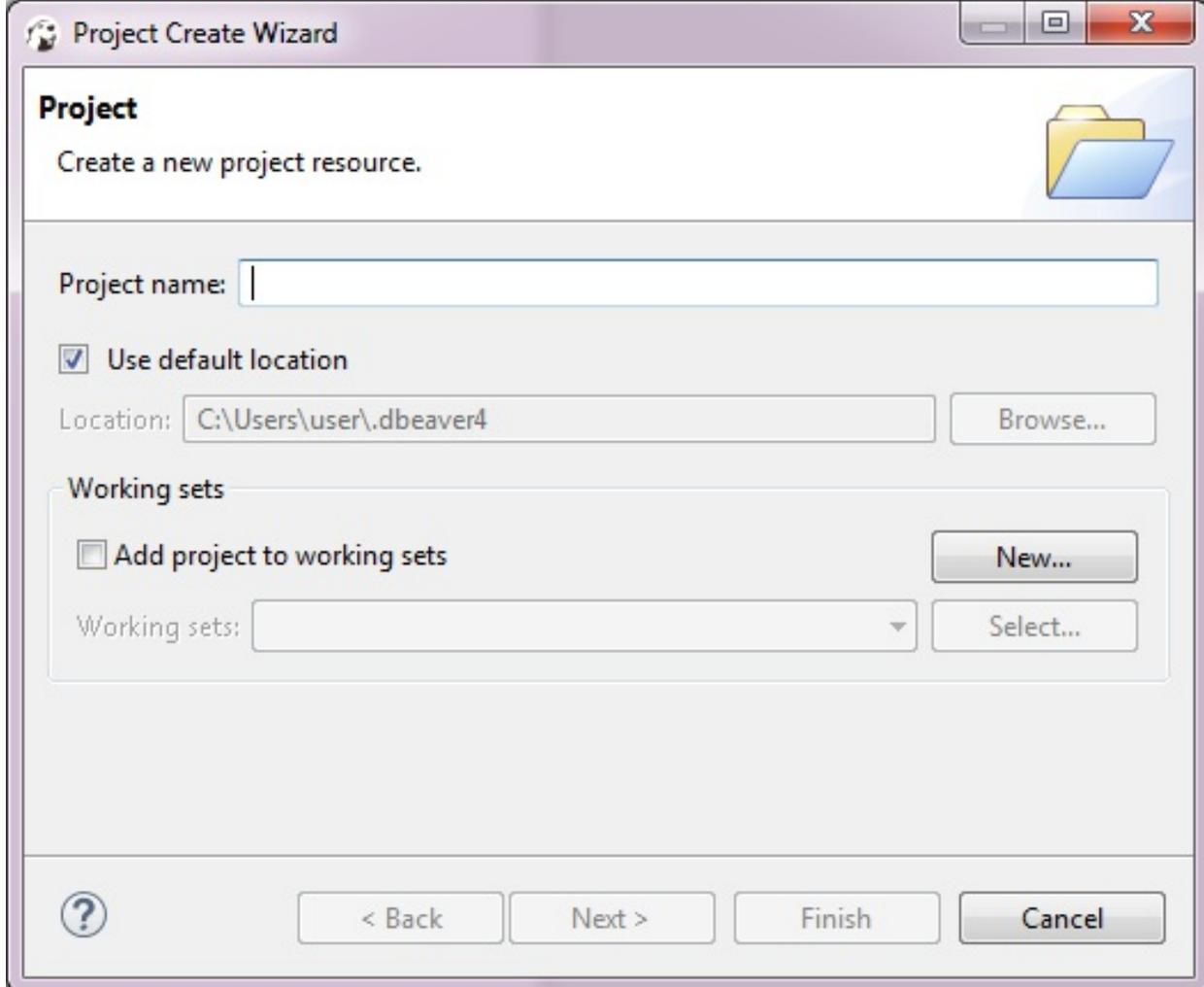
- The **Set Active Project** menu item (for a project root node) makes the project active, that is visible in the Database Navigator.
- The **Link File (SQL Script)** and **Link Folder** menu items allow creating links to files and folders in the file system.

Managing Projects

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

Creating Project

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



TBA

Deleting Project

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

1. Delete object dialog box is to confirm the deletion of the project itself. Click **Yes** if you are sure you want to delete it. Otherwise, click **No**.
2. Delete project dialog box is to confirm the deletion of the project`s contents: these are the data stored in the file system, database connections are not affected. Click **Yes** if

you want the contents to be deleted as well. To keep the contents, click **No**.

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

Managing Bookmarks

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

To create a bookmark:

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

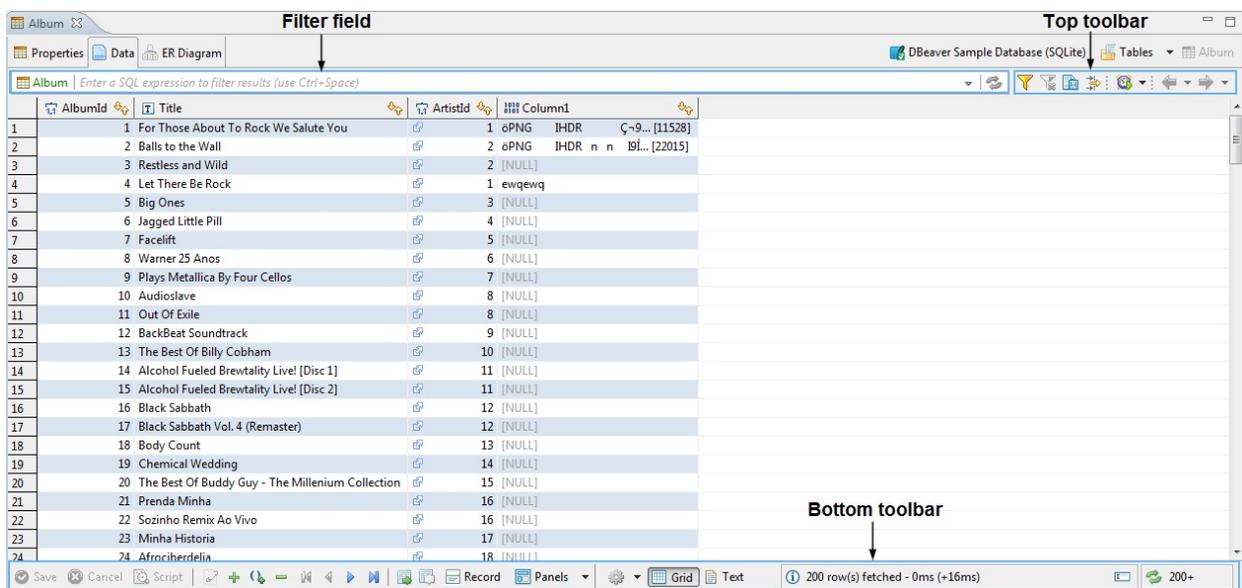
To open an object using its bookmark, double-click the bookmark or right-click it and click **Open Bookmark** on the context menu. You can rename and delete bookmarks using the context menu as well.

Data editor

The Data editor appears:

- As the **Data** tab of the Database Object Editor (<https://github.com/serge-rider/dbeaver/wiki/Database-Object-Editor>) available only for tables and views.
- As the **Results** tab when you run a custom SQL query in SQL Editor (<https://github.com/serge-rider/dbeaver/wiki/SQL-Editor>)

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



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
	Cuts the content of the current cell or

Cut	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
Paste	Pastes the copied content to the cells in focus
Advanced Paste	Pastes several values delimited with a tabulation or line break
Delete	Deletes the row that has the cell in focus NOTE: In fact, when users click Delete , the system only highlights the row red while the actual deletion happens when users click Save .
Edit cell	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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

<p>Save to file...</p>	<ul style="list-style-type: none"> - 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
<p>Load from file...</p>	<ul style="list-style-type: none"> - Appears only for columns with BLOB/CLOB data - Opens a standard window for opening files
<p>Order/Filter</p>	<p>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 ...</p> <p>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.</p>
<p>View/Format</p>	<p>Opens a submenu that provides tools for formatting and modifying the view of data, see ...</p>
<p>Navigate</p>	<p>Opens a submenu that helps users navigate throughout the data table, see ...</p>
<p>Layout</p>	<p>Changes the layout of data, see ...</p>
<p>Export</p>	<p>Opens the Data Transfer wizard that guides you through the steps to select a format and export data</p> <p>NOTE: The system exports the whole result</p>

Resultset...	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

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:

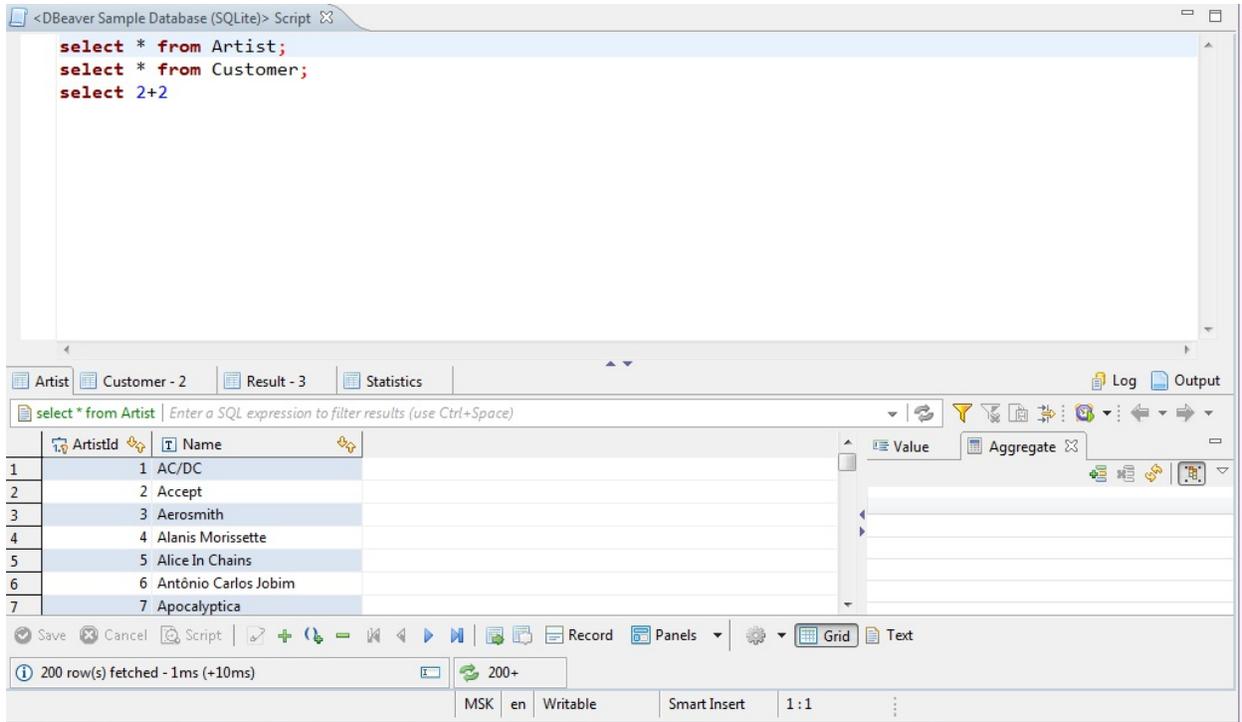
- Click this connection in the Database Navigator (<https://github.com/serge-rider/dbeaver/wiki/Database-Navigator>) view and press `F3` or click **SQL Editor -> SQL Editor** on the main menu. Alternatively, you click **SQL Editor** on the context menu of this connection. DBeaver opens the Choose SQL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

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

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

You can see all your saved SQL scripts in the Project Explorer (<https://github.com/serge-rider/dbeaver/wiki/Project-Explorer>) view in the **Scripts** folder.

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



Results Pane

The results pane displays tabs with results in various formats. The tabs resulting from script execution represent instances of the Data Editor (<https://github.com/serge-rider/dbeaver/wiki/Data-Editor>). You can create, edit and execute SQL scripts in the script pane and then see the results in the result tabs.

- To toggle (hide/show) the results pane, press **F6** .
- To maximize the results pane, press **Shift+F6** or double-click the results tab name.

- To switch between the script pane and the results pane, press `Alt+F6`.

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

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



Time	Type	Text	Duration	Rows	Result
21:41:31	SQL / USER	select count(*) from customer	0 ms		An I/O error occur...
16:52:09	Transaction	Commit	0 min 0 sec		
16:52:09	Transaction	Rollback	0 min 0 sec		
16:43:32	Transaction	Commit	8 min 36 sec		
16:43:32	Transaction	Commit	0 min 0 sec		
16:43:32	Transaction	Rollback	0 min 0 sec		
16:43:26	Transaction	Commit	0 min 6 sec		
16:43:26	Transaction	Commit	0 min 0 sec		
16:43:25	Transaction	Rollback	0 min 0 sec		

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

Hyperlinks

You can press `Ctrl` and move mouse over SQL text. If DBeaver recognizes some identifier as a table/view name then it presents it as a hyperlink which will navigate to this object's editor.

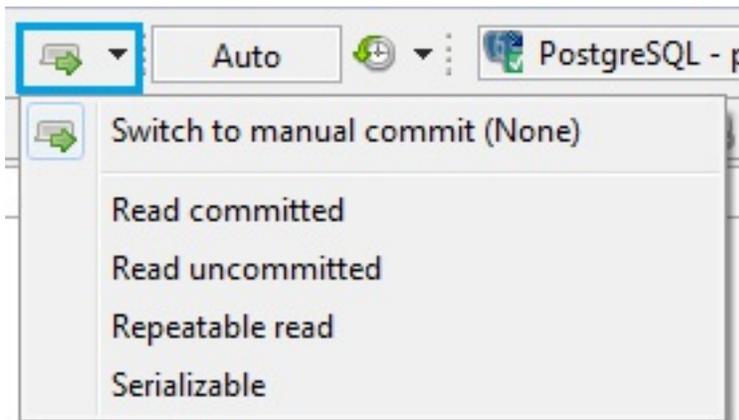
Active Database/Schema Selection

To change the connection associated with the current SQL editor or change the active database/schema, use the combo boxes in the main toolbar:



Transactions Management

You can toggle auto-commit mode and transactions isolation level on the **Database** menu or in the main toolbar:



MongoDB

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

Browsing Mongo collections

You can view/edit Mongo collections content as standard relational tables (grid/plain text presentations) or as JSON documents.

Presentation can be switched in the Results Viewer toolbar.

In grid DBeaver will try to unify all documents in some particular collection (as they have the same structure/the same set of properties).

Executing JS

JS statements can be executed in SQL editor as usual.

Mongo scripting reference

(<https://docs.mongodb.com/v3.0/administration/scripting/>)

Following example creates a user in the current database.

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

This example returns all documents in collection 'test_col':

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

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

Executing SQL

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

```
SELECT * FROM test_col
WHERE propName.subProp='value';
```

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

Nested JSON properties can be divided by dot.

Office

This extension supports data export in Office formats (XLS).

It is included in EE version.

Installation in CE version:

1. Main menu `Help -> Install New Software`
2. Install Office extension. You can do this from online Eclipse Marketplace
(<https://marketplace.eclipse.org/content/dbeaver-office-integration> (<https://marketplace.eclipse.org/content/dbeaver-office-integration>)). Or you can do this manually:
 - **Paste** `https://dbeaver.jkiss.org/update/office/latest/` **into**
`Work with field` and press `Enter`
 - **Check** `DBeaver Extension->DBeaver Office Extension` **item**
3. Click Next->Finish. Restart DBeaver.

Note: this works only if DBeaver version matches Office extension version (e.g. latest).

SVG format

This extension supports data export in Office formats (XLS).

It is included in EE version.

Installation in CE version:

1. Main menu `Help -> Install New Software`
2. Install Office extension. You can do this from online Eclipse Marketplace
(<https://marketplace.eclipse.org/content/dbeaver-svg-support>
(<https://marketplace.eclipse.org/content/dbeaver-svg-support>)). Or you can do this manually:
 - **Paste** `https://dbeaver.jkiss.org/update/svg/latest/` **into** `Work`
`with` `field` and **press** `Enter`
 - **Check** `DBeaver Extension->DBeaver Office Extension` **item**
3. Click `Next->Finish`. Restart DBeaver.

Note: this works only if DBeaver version matches Office extension version (e.g. latest).

Diagrams (ERD)

Entity Relation Diagrams (ERD) are graphic presentations of database entities and relations between them. DBeaver allows viewing diagrams of existing tables and whole database schemas, see Database Structure Diagrams (<https://github.com/serge-rider/dbeaver/wiki/Database-Structure-Diagrams>), as well as create custom diagrams, see Custom Diagrams (<https://github.com/serge-rider/dbeaver/wiki/Custom-Diagrams>)

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

Structure Adjustment

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

You can do the following structure changes in diagrams.

- Add new tables to a diagram by drag-n-dropping them onto the diagram field from the Database Navigator (<https://github.com/serge-rider/dbeaver/wiki/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 (<https://github.com/serge-rider/dbeaver/wiki/Custom-Diagrams>) only) - connect tables with a connector: click the **Show Palette** button () in the upper-left corner of the diagram tab and then, in the Palette panel, click **Connection**. Now click the tables that you want to connect with each other in turn one by one, and then to stop the connection line double-clicking the last table.
- (Available for Custom Diagrams (<https://github.com/serge-rider/dbeaver/wiki/Custom-Diagrams>) only) - remove tables: right-click the table and click **Delete** on the context menu.

View Adjustment

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

- Enable/disable the diagram grid: Click **Toggle Grid** () in the toolbar.
- Modify attributes visibility: Right-click the diagram and, on the context menu, click **Show Attributes** and then one of the options:
 - **All** - all attributes
 - **Any keys** - primary and foreign keys
 - **Primary key** - only primary keys
 - **None** - no attributes
- Modify attributes presentation: Right-click the diagram and, on the context menu, click **Attribute Styles** and then one of the options:

- **Show Icons**
- **Show Data Types**
- **Show Nullability**
- **Show Comments**
- Change the color of entities/notes: Right-click the header of the entity or comment and then click **Set color** on the context menu. Then you can select the color and click **OK**.
- For elements located in front of/behind others, bring an element to front or send to back: Right-click the element and then click **Bring to front / Send to back** on the context menu.
- Zoom the diagram in/out: Click the **Zoom In/Zoom Out** buttons or choose the scaling value in the dropdown list in the toolbar: 

Refresh

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

Notes Creation

You can create notes only in Custom Diagrams (<https://github.com/serge-rider/dbeaver/wiki/Custom-Diagrams>). To create a note, click the **Show Palette** button () in the upper-

left corner of the diagram tab, then, in the Palette panel, click **Note** and then click anywhere in the diagram tab. Now you can double-click **Note** box to enter the note text.

Search in Diagram Entities

To search among entities of a diagram, click the **Search items** button () in the toolbar, then type in the search combination. The entities that contain the search combination are highlighted in the diagram.

To remove the filter, click the cross icon next to the search field.

Diagram Export

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

Diagram Printing

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

Settings

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

Query Manager

In the Query Manager view you can see all SQL queries DBeaver executes.

DBeaver logs every query you execute. You can see all previously executed queries in the special view “Query Manager”. You can open it through main menu “Window->Show View”.

You can see query execution statistics (execution time, number of fetched/updated rows, errors, etc).

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

By default QM shows only queries explicitly executed by user (in SQL editor or in Data Editor). You can configure logs visibility, monitor metadata reading queries, utility SQL, etc. QM configuration available in global preferences.

Command-Line

Command line parameters

Command line parameters might be passed directly to `dbeaver[.exe]` executable.

On Windows you also can use `dbeaver-cli.exe` executable (it doesn't spawn new window so you can see output messages).

DBeaver control

Name	Value	Example
-help	Prints help message	
-stop	Quits DBeaver	
-dump	Prints DBeaver thread dump	
-f	Opens file in DBeaver UI	<code>-f c:\some-path\some-file.sql</code>
-con	Opens database connection in DBeaver UI	See connection parameters table
-closeTabs	Closes all open editor tabs	
-disconnectAll	Closes all open connections	
	Force reuse of single	

- reuseWorkspace	workspace by multiple DBeaver instances	
-newInstance	Force new DBeaver instance creation (do not try to reuse already running one)	

System parameters

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

VM arguments

You can pass any advanced Java parameters supported by your local JVM (Oracle, OpenJDK, IBM, etc).

Parameters supported by Oracle JVM (1.8):

<https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jav>
(<https://docs.oracle.com/javase/8/docs/technotes/tools/windows/jav>)

Parameters supported by all JVMs:

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

Connection parameters

All connection parameters must be supplied as a single command line argument, parameters are divided by pipe (|). Parameter name and value are divided by = .

Example: -con

driver=sqlite|database=C:\db\SQLite\Chinook.db|name=SQLiteChi

Name	Value	Example
name	Connection name	Test connection
driver	Driver name or ID	driver=sqlite , driver=mysql , €
	Connection URL. Optional	

url	(JDBC URL may be constructed by driver from other parameters)	url=jdbc:sqlite:C:\db\SQLite\C:
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
	Do not ask use	

savePassword	password on connect	savePassword=true
connect	Connect to this database	connect=false
openConsole	Open SQL console for this database (sets <code>connect</code> to true)	openConsole=true
folder	Put new connection in a folder	folder=FolderName
autoCommit	Sets connection auto commit flag (default value depends on driver)	autoCommit=true
prop.propName	Advanced connection parameters (depend on driver)	prop.connectTimeout=30

Managing connections

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

Provide predefined connections

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

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

Importing connections from CSV/XML

You can import connection from CSV or XML files.

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

Supported names:

	Name	Meaning	
	name	Connection name	

url	JDBC URL	
host	Database server host name	
port	Database server port	
database	Database/schema name	
user	User name	
password	User password	

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

Sample CSV

```
name,host,port,server,database,url,user,password,type
Postgre Import XML 1,localhost,5432,,postgres,jdbc:postgresql://localhost:5432/postgres,postgres,postgres,dev
Postgre Import XML 2,localhost,5432,,postgres2,jdbc:postgresql://localhost:5432/postgres2,postgres2,postgres2,prod
```

Sample XML

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

Secure connections from editing

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

- Generate MD5 hash of your password. You can do it from command line using Linux utility md5sum (`md5sum <<<"your password")` or you can do it online - just google "MD5 hash online".
- Add attribute `lockPassword` in connection descriptor (in `.dbeaver-data-sources.xml` file in `<dataSource>` tag. So it will look like this:

```
<data-source
id="postgresql-1294077257514--1782558860"
provider="postgresql"
driver="postgres-jdbc"
name="Postgre - sample"
create-date="1294077257524"
update-date="1468749632751"
login-date="1469021247956"
save-password="true"
show-system-objects="true"
read-only="false"
lockPassword="2ba81a47c5512d9e23c435c1f29373cb">
```

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

Managing drivers

Configure drivers with pre-installed jars

You can customize drivers configuration in

`${HOME}/.dbeaver4/.metadata/.plugins/org.jkiss.dbeaver.core/drivers.xml` file. If you have some pre-installed jar files you can reference them in `drivers.xml`. Example:

```
<library type="jar" path="absolute-jar-folder-path\driver-jar.jar"
custom="true"/>
```

Also in `drivers.xml` you can use following variables to specify relative paths:

	Variable	Meaning	
	<code>drivers_home</code>	Standard DBeaver drivers location (<code>\${HOME}/.dbeaver-drivers</code> by default)	
	<code>dbeaver_home</code>	DBeaver installation folder	
	<code>home</code>	User home folder	
	<code>workspace</code>	DBeaver workspace path (<code>\${HOME}/.dbeaver4</code> by default)	

For instance:

```
<library type="jar" path="${dbeaver_home}\drivers\my-driver.jar" cu
stom="true"/>
```

Develop in Eclipse

DBeaver is an Eclipse RCP

(https://wiki.eclipse.org/Rich_Client_Platform) application.

It consists from a set of Eclipse plugins, features and products.

Sources structure:

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

Opening workspace in Eclipse

You will need to setup "Eclipse for RCP and RAP developers (<https://www.eclipse.org/downloads/packages/eclipse-rcp-and-rap-developers/oxygen1a>)". Currently default Eclipse version is Oxygen (4.7). Build may fail on newer or older versions. Then you will need to install a few additional Eclipse plugins which DBeaver depends on (and which are not included in the standard Eclipse distribution):

- Open main menu Help -> Install New Software
- Copy <http://dbeaver.jkiss.org/eclipse-repo/> into "Work with" field and press enter images/eclipse-deps-install.png
- Check all plugins and finish the installation process

Now you need to checkout DBeaver repository with `git clone https://github.com/serge-rider/dbeaver.git` command.

In Eclipse create a new workspace and run Import wizard. Choose "General->Existing projects into Workspace".

Select DBeaver checkout directory as root a directory. Check all plugins and features Eclipse will find and finish the wizard.

Now you can build DBeaver (`Ctrl+B`). To run the standalone version find file `DBeaver.product` (it is located in `product/standalone` directory). Open it and click on "Launch an Eclipse application" or "Launch an Eclipse application in Debug mode".

If product won't start with `No application id has been found` error try this workaround:

<https://stackoverflow.com/questions/13575695/eclipse-rcp-no-application-id-has-been-found>

(<https://stackoverflow.com/questions/13575695/eclipse-rcp-no-application-id-has-been-found>)

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

That's it.

Notes:

Everything above covers Community Edition version.

Features structure

Features Structure

The DBeaver features are structured to support both "IDE" and "RCA" mode

(<https://github.com/serge-rider/dbeaver/wiki/images/development/DBeaver-CE-Features.png>)

Reset workspace

Sometimes (especially after multiple DBeaver versions upgrade) workspace become messy.

Some keyboard shortcuts may stop working, toolbars layout may be broken, etc, etc.

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

1. Shutdown DBeaver

2. Go to directory to

```
%HOMEPATH%\dbeaver\metadata\plugins\org.eclipse.e4.workbench\
```

- **Windows:** `Win+R` , `enter`

```
%HOMEPATH%\dbeaver\metadata\plugins\org.eclipse.e4.workben
```

- **Linux:** `cd`

```
$HOME/.dbeaver/.metadata/.plugins/org.eclipse.e4.workbench/
```

3. Delete file `workbench.xmi`

4. Start DBeaver

If that doesn't help then you can try to remove

`%HOMEPATH%\dbeaver\metadata\` folder. This will erase all your UI settings and SQL scripts configurations (but all connections and scripts will remain). Do it only if nothing else helps!

That's it.

Posting issues

A few tips.

- Check existing issues for your issue (including closed ones). Duplicating an issue is slower for both parties so search through open and closed issues to see if what you're running into has been addressed already.
- Be clear about what your problem is: what was the expected outcome, what happened instead? Detail how someone else can recreate the problem.
- If you posting a bug report check "Error Log" view. If there are any errors related to your bug then post a complete stacktrace. Sometimes there are no errors in Error Log - then try to find them in log files.
- If your issue is related to a database data or metadata management - check the Query Manager view. It contains information about all queries DBeaver executes (explicitly or implicitly). To see more detailed information you can configure Query Manager in Preferences.  images/qm-view.png
- Depending on the nature of your bug report provide information about:
 - Operating system
 - Window manager (for Linux)
 - Database (name and version)
 - Database driver (name and version)

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

Log files

DBeaver writes different log files. Most of them are Eclipse logs. Usually log files resides in the workspace.

Default workspace location is `${HOME}/.dbeaver` (`${HOME}/.dbeaver4` for DBeaver 4.x+).

Two standard log files:

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

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

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

If you are reporting about some error please attach logs (not complete file but valuable part of it).

Logs are very useful, big number of errors can't be reproduced/fixed without full error stacktrace.

Thread dump

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

Mac and Linux

Run the following on your terminal:

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



Windows

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

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

in Command Prompt.

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