



CloudBeaver User Guide v.24.1.ea

User Guide

Table of contents

About CloudBeaver

CloudBeaver is a lightweight web application designed for comprehensive data management. It allows you to work with various data sources, including SQL, NoSQL, and cloud databases, all through a single secure cloud solution accessible via a browser.



Multiple Database Support and Drivers Management

Databases Support: Interact with different databases through a unified interface.

[Learn more](#)

Drivers Management: Create custom drivers and edit existing ones. [Learn more](#)

Secure Connectivity and Authentication

Centralized Access: Facilitate a centralized access management system.

[Learn more](#)

Connection Access Management: Harness administrator privileges to manage user access for shared connections. [Learn more](#)

Data and SQL Management

Data Management: Use flexible data filters and switch between views for interactive data handling.

[Learn more](#)

SQL Development Tools: Enhance your SQL development with autocomplete and syntax highlighting.

[Learn more](#)

Script Execution: Execute and manage scripts efficiently.

[Learn more](#)

Data Export: Easily export your data to various formats.

[Learn more](#)

Data Import: Import files into your database with ease.

[Learn more](#)

Data Types: Manage and display diverse data types.

Entity Relationship Diagrams and Visual Query Builder

ER Diagrams: Visualize database schemas and relationships.

[Learn more](#)

Visual Query Builder: Construct complex SQL queries effortlessly. [Learn more](#)

Artificial Intelligence Support

AI Smart Assistance: Enhance your SQL development with AI-powered smart completion. [Learn more](#)

Integrated Cloud Tools

Cloud Explorer: Access and manage resources from major cloud providers.

[Learn more](#)

Cloud File Explorer: Integrate smoothly with cloud-based file systems. [Learn more](#)

User Interface Configuration

Server Configuration: Modify CloudBeaver's server settings.

[Learn more](#)

Product Configuration: Customize various aspects of CloudBeaver.

[Learn more](#)

Learn more

Table of contents

[Drivers Management](#)

[Creating a Custom Driver](#)

[Uploading Binary Files](#)

[Editing an Existing Driver](#)

[Filtering Drivers](#)

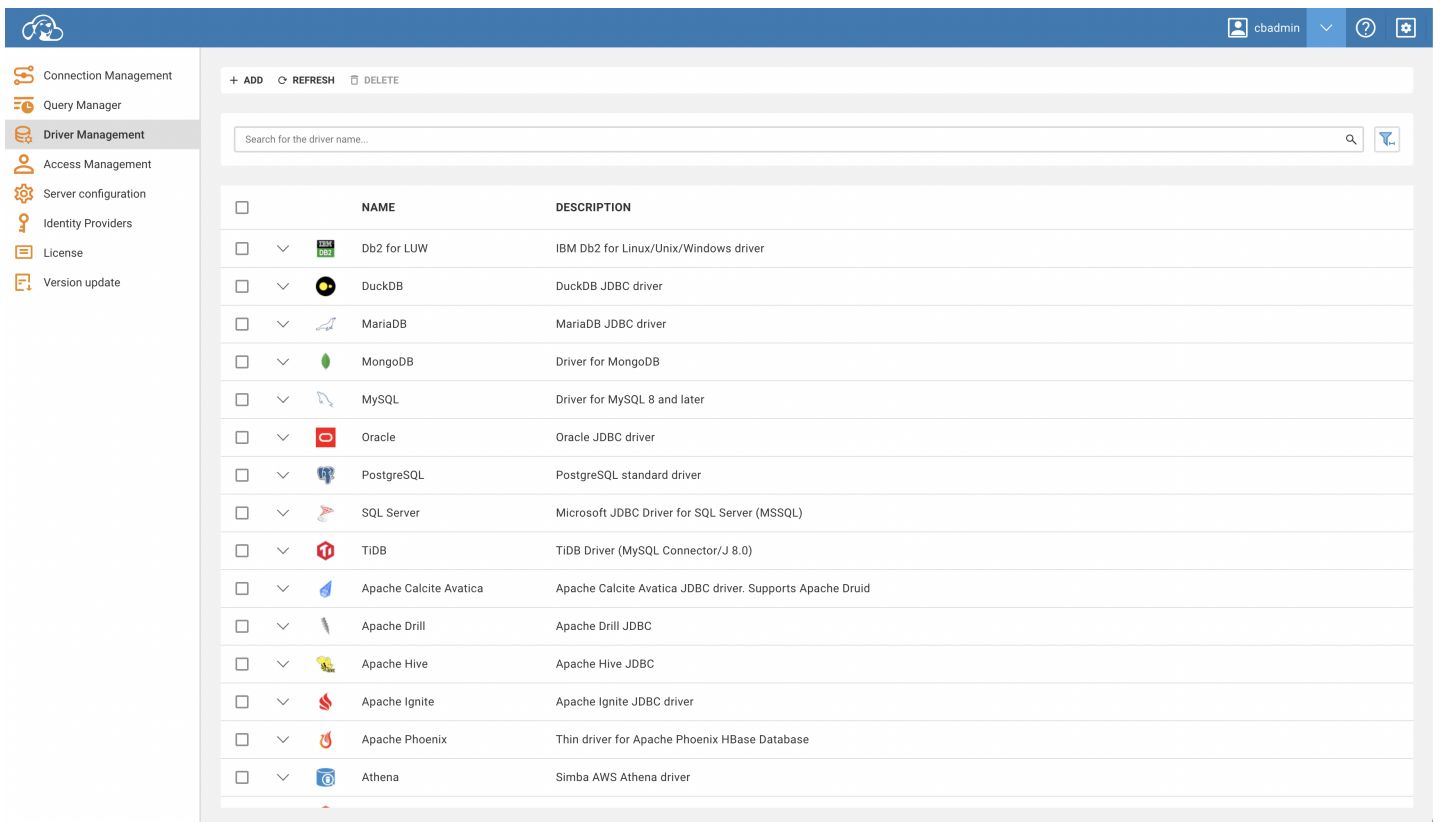
Note: This feature is available in [Enterprise](#) and [Team Edition](#) only.

Drivers Management

This wiki guide provides a step-by-step instruction for managing drivers using the product interface, including creating a new custom driver and editing an existing one.

Before you start managing your drivers, it is important to note that administrative privileges are required. Only users with administrator rights can perform actions such as creating, editing, or deleting drivers.

Creating a Custom Driver

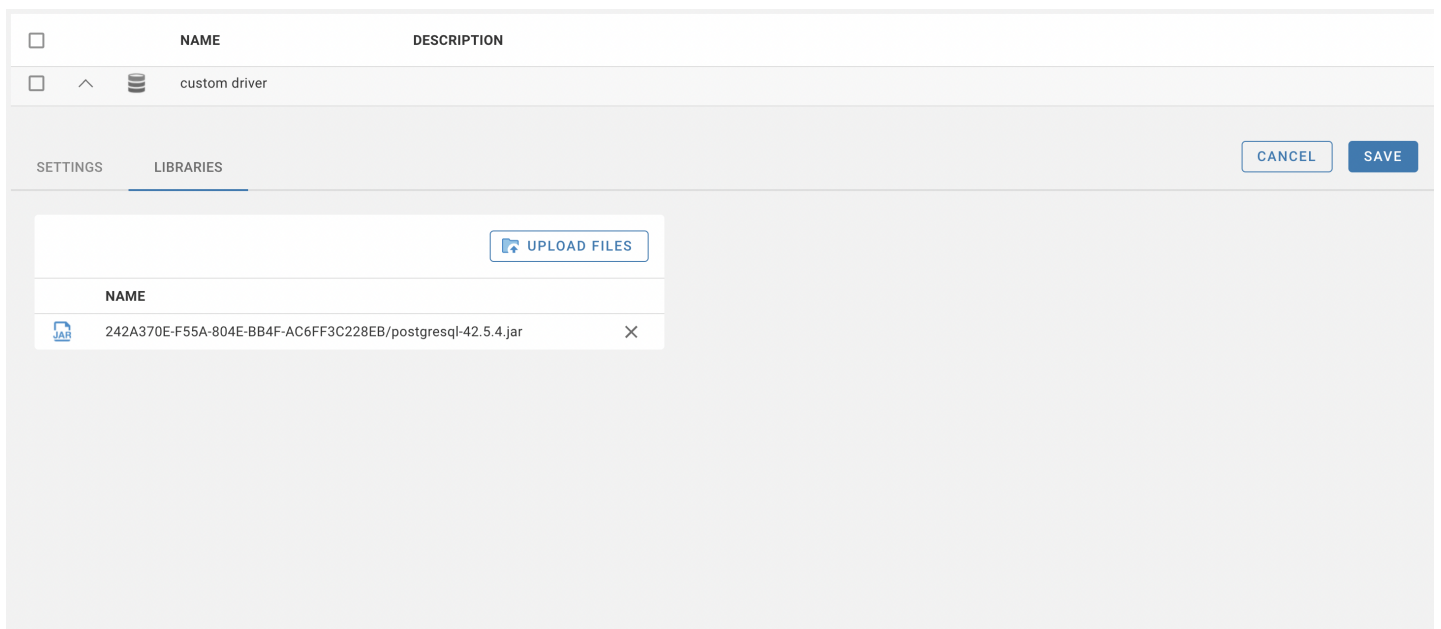


Follow the steps below to create a new custom driver:

1. Navigate to the **Administration** page of the interface.
2. Select the **Driver Management** tab.
3. Click on the **Add** button.
4. A form will appear in which you will have to fill in the necessary fields with the appropriate information.
5. Once completed, click **Create**.

Your new custom driver should now be successfully created.

Uploading Binary Files



In order to make your new driver functional, you will need to upload the `.jar` binary files. This is how you can do it:

1. Locate your newly created driver on the `Driver Management` table.
2. You can filter for your custom drivers by checking the relevant checkbox in the table filters.
3. Click on your custom driver to open it.
4. Navigate to the `Libraries` tab.
5. Click `Upload Files`, then select your `.jar` files.
6. Confirm that your files have appeared in the table.
7. Click `Save`.

Once the binary files have been uploaded, your custom driver will be ready to use. Whenever you need to access it, it can be selected from the driver list.

Editing an Existing Driver

	NAME	DESCRIPTION
<input type="checkbox"/>	Oracle	Oracle JDBC driver

SETTINGS LIBRARIES
CANCEL SAVE

Driver Name *

Driver Type *

Class Name *

URL Template

Description

DEFAULTS

Default Port

Default database

You can also edit any existing driver. By following these the steps:

1. Open the desired driver from the **Driver Management** table.
2. Modify the fields as required.
3. Click **Save** to preserve your changes.

Your driver is now updated and can be continued to be used with the modified settings.

Please note: Ensure you have the appropriate permissions and are aware of the potential impact before proceeding with these modifications.






Filtering Drivers

Drivers have 3 available filters:

- Search
- Custom/default driver
- Driver state (enabled/disabled/all)

In order to see all filters, you must click a filter button:

CloudBeaver interface showing the Driver Management section. The left sidebar contains navigation items: Query Manager, Driver Management (selected), Connection Templates, Access Management, Server configuration, AWS Settings, Identity Providers, License, and Version update. The main content area has a top bar with '+ ADD', 'REFRESH', and 'DELETE' buttons. Below is a search bar 'Search for the driver name...' with a search icon and a filter icon. A dropdown menu is set to 'Enabled' and a checkbox for 'Custom drivers' is present. The main area displays a table of drivers:

<input type="checkbox"/>	NAME	DESCRIPTION	ENABLED
<input type="checkbox"/>	 Db2 for LUW	IBM Db2 for Linux/Unix/Windows driver	<input checked="" type="checkbox"/>
<input type="checkbox"/>	 MariaDB	MariaDB JDBC driver	<input checked="" type="checkbox"/>
<input type="checkbox"/>	 MongoDB	Driver for MongoDB	<input checked="" type="checkbox"/>
<input type="checkbox"/>	 MySQL	Driver for MySQL 8 and later	<input checked="" type="checkbox"/>
<input type="checkbox"/>	 ODBC	Modern JDBC-ODBC bridge	<input checked="" type="checkbox"/>

Drivers can be disabled in the **Server configuration** tab

Learn more

Table of contents

[Provisioning](#)

Note: This feature is available in [Enterprise](#) and [Team Edition](#) editions only.

Users can be provisioned from the third-party system. This is useful when you want to pre-configure access for users.

Supported providers:

1. Microsoft AzureAD
2. Okta OpenID

Provisioning

You can find the **+ Import** button on the Users management page in administration. (only in Enterprise products)

The screenshot shows the CloudBeaver administration interface. The top navigation bar is blue and contains the user profile 'cbadmin', a help icon, and a settings icon. The sidebar on the left lists various management options: Query Manager, Driver Management, Connection Templates, Access Management (highlighted), Server configuration, AWS Settings, Identity Providers, License, and Version update. The main content area is titled 'USERS TEAMS' and features a '+ CREATE + IMPORT REFRESH' toolbar. Below the toolbar is a search bar with the placeholder text 'Search for the user name...'. The main area contains a table with the following data:

USER NAME	USER TEAM	ENABLED	
admin1	admin, user	<input checked="" type="checkbox"/>	
	user	<input checked="" type="checkbox"/>	
cbadmin	admin, dum, team_analytics, user	<input type="checkbox"/>	
empty	simple	<input checked="" type="checkbox"/>	
new	user	<input checked="" type="checkbox"/>	
wroud	simple, user	<input checked="" type="checkbox"/>	

Click on this button will open users importing form.

The screenshot shows the 'USER PROVISIONING' dialog box in the CloudBeaver interface. The dialog is titled 'USER PROVISIONING' and has a subtitle 'Import users from external service into the application'. It features a dropdown menu set to 'Azure' and a 'GET USERS LIST' button. Below this is a search bar for user names or emails. The main area contains two tables of user data. The left table has columns for 'EMAIL' and 'USER NAME', with checkboxes for selection. The right table has columns for 'EMAIL' and 'USER NAME', with checkboxes for selection. At the bottom right of the dialog is an 'IMPORT' button. Below the dialog, a table shows the list of imported users with columns for 'USER NAME', 'USER TEAM', and 'ENABLED'.

EMAIL	USER NAME
<input type="checkbox"/> aieksei@example.com	
<input type="checkbox"/> nikita@example.com	
<input type="checkbox"/> nikolai@example.com	
<input type="checkbox"/> anton@example.com	
<input checked="" type="checkbox"/> daria@example.com	
<input checked="" type="checkbox"/> denis@example.com	Denis Sinelnikov
<input checked="" type="checkbox"/> support@example.com	
<input type="checkbox"/> evgenia@example.com	

EMAIL	USER NAME
<input checked="" type="checkbox"/> daria@example.com	
<input checked="" type="checkbox"/> denis@example.com	Denis Sinelnikov
<input checked="" type="checkbox"/> support@example.com	

USER NAME	USER TEAM	ENABLED
11	user	<input checked="" type="checkbox"/>

To import users, you need to select users provisioning provider and click **Get Users List**. So that you know, you need to configure at least one supported provider before.

You will be asked to log in with the selected provider (in case you haven't logged in before) to load users.

After loading users, You can select users to import and verify them in the table on the right side. In the **Team Edition**, you must also select the role assigned to users.

To confirm importing, click on the **Import** button. Users will appear in the table.

Create Connection

Table of contents

[How to Create a New Connection](#)

[Connection Access Management](#)

[Creating a Template in the Administration Section](#)

[Prerequisites:](#)

[Steps to Create a Template:](#)

[Shared vs Private Projects: Differences](#)

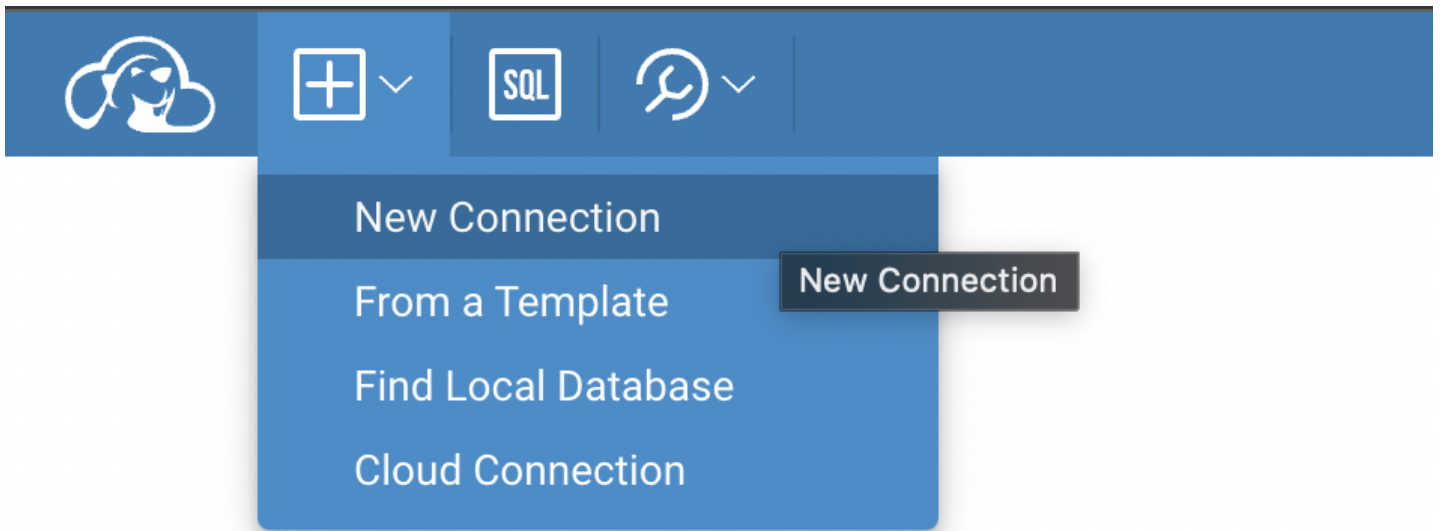
[Create shared/private project](#)

How to Create a New Connection

Follow the steps below to establish a new connection from the public part of the app:

1. Initiate New Connection:

- Click on the **+** button located in the top menu.
- Select **New Connection** from the dropdown options.



2. Choose a Driver:

- From the available options, select the appropriate driver.

3. Fill in Connection Details:

- A connection form will be presented.
- Input all necessary details into the form fields.

A screenshot of the CloudBeaver connection configuration form. The interface has a blue header bar with icons and a user profile 'cbadmin'. Below the header, there are tabs for 'OPTIONS', 'DRIVER PROPERTIES', 'SSH TUNNEL', and 'SSL'. The 'DRIVER PROPERTIES' tab is active. The form is divided into several sections: 'Driver' (PostgreSQL), 'Configuration' (Manual selected), 'Host *' (localhost), 'Port' (5432), 'Database' (postgres), 'AUTHENTICATION' (Database Native), 'User name' and 'User password' fields, 'Save credentials' checkbox, 'Connection name *' (PostgreSQL@localhost), 'Project' (Private), 'Description' text area, 'SETTINGS' (Show all databases, Show template databases, Show unavailable databases checkboxes), and 'User role' field. At the top right of the form area are 'CANCEL', 'TEST', and 'CREATE' buttons. A vertical sidebar on the right contains a back arrow button.

4. Set Up Advanced Settings (Optional):

- Additional settings such as SSH, SSL, and others can be adjusted.
- Navigate to the relevant tabs located at the top of the connection form to access these settings.

5. Test the Connection (Optional):

- Before finalizing the connection, you have the option to test it.
- Click the **Test** button to ensure the connection parameters are correct.

6. Finalize and Create the Connection:

- Once all details are in place, click the **Create** button.

7. Access and Modify Connection:

- The newly created connection will be visible in the connection navigator menu.
- To edit the connection details:
 - Click on the context button next to the connection name.
 - Select **Edit Connection** from the dropdown menu.

By following these steps, users can successfully create and modify connections in the app.

Connection Access Management

If you have **admin permissions**, you'll have the capability to manage connection's permissions. When working with a shared connection, you will notice an **Access** tab. This is where you can manage who has access to the connection.

Creating a Template in the Administration Section

Before proceeding with the creation of a template, ensure that you have the necessary administrative permissions.

Prerequisites:

- You must have **admin permissions** to access this feature.

Steps to Create a Template:

The steps are similar to connection creation; the only difference is that it must be done in the administration part, under the "Connection Templates" tab. You can learn more about creating connection templates [here](#).

Shared vs Private Projects: Differences

In our product you can create Private and Shared project

- Private projects are exclusive spaces owned by an individual user. They operate as personal projects, safeguarding sensitive information and allowing for undisturbed individual work.
- Shared projects are collaborative hubs where multiple users have access to contribute and edit content. These projects foster teamwork.

The choice between shared and private projects hinges on the project's nature and objectives. Whether opting for the autonomy of a private project or embracing the teamwork in a shared project, understanding these differences empowers individuals and teams to align their project management strategies with their specific needs.

Create shared/private project

If you want to be able to create private projects, your administrator must enable this feature in the **server configuration** panel:

CONFIGURATION

Enable private connections

Allows users to create private connections

Navigator simple view

By default, all user's new connections will contain only basic information in navigation tree

RESOURCE MANAGER

Enable Resource Manager

Enable Resource Manager functionality

When you create new connection, you can choose **Private** or **Shared** project, depending on the purpose for which you need to create:

The screenshot shows the 'DRIVER PROPERTIES' tab of the CloudBeaver connection configuration window. At the top right, there are three buttons: 'CANCEL', 'TEST', and 'CREATE'. The interface is divided into several sections:

- Driver:** A dropdown menu is set to 'PostgreSQL'.
- Configuration:** Two radio buttons are present: 'Manual' (selected) and 'URL'.
- Host *:** A text input field contains 'localhost'.
- Port:** A text input field contains '5432'.
- Database:** A text input field contains 'postgres'.
- Connection name *:** A text input field contains 'PostgreSQL@localhost'.
- Project:** A dropdown menu is set to 'Private', with a list of options below it: 'Private' and 'Shared'.
- AUTHENTICATION:** A dropdown menu is set to 'Database Native'.
- User name:** An empty text input field.
- User password:** An empty text input field with a visibility toggle icon.
- Save credentials
- MISC:** A section with a dropdown arrow.
- User role:** An empty text input field.
- DATABASE LIST:** A section with a right-pointing arrow.
- SQL:** A section with a right-pointing arrow.

Learn more

Table of contents

[Features](#)

[Shortcuts](#)

Features

- [Data Filters](#)
- [Data Ordering](#)
- [Value Panel](#)
- [Grouping panel](#)
- [JSON and Document View](#)
- [Working with spatial/GIS data](#)

Shortcuts

Shortcut	Description
Enter Backspace a-Z and 0-9 keys	Start inline editing
Alt+Insert	Add a new row
Ctrl+Alt+Insert	Duplicate row
Delete	Delete row
Escape	Revert changes
Ctrl+V	Past value to cell from clipboard
Ctrl+C	Copy data to clipboard

Learn more

Table of contents

[Overview](#)

[Shortcuts](#)

[Statement Execution](#)

[Script Execution](#)

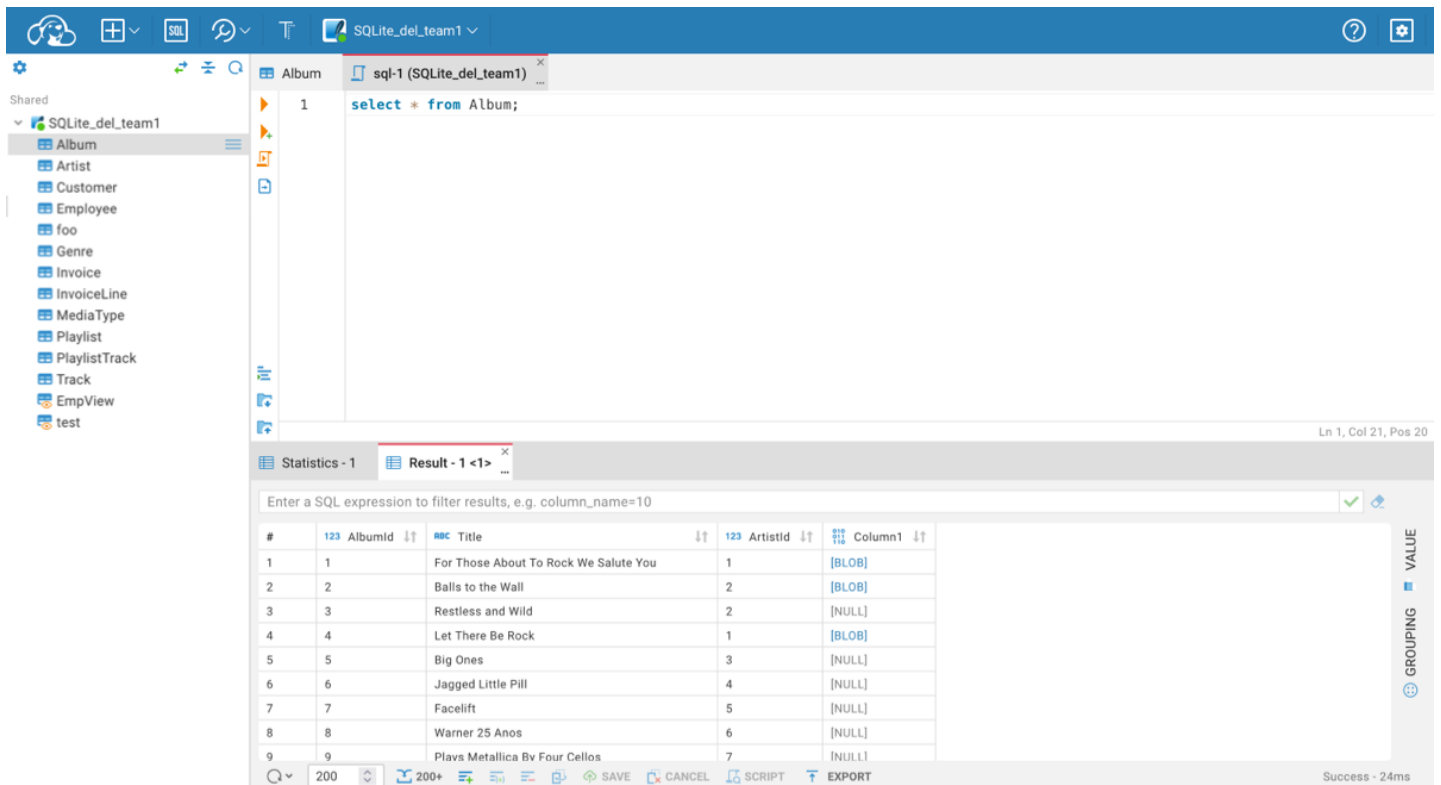
[Auto and Manual commit](#)

[Switching to Manual commit](#)

[Limitations of Rollback capabilities](#)

Overview

SQL Editor supports autocomplete, syntax highlight, statement execution, script execution, and execution plan for some databases.



The screenshot displays the SQL Editor interface. On the left, a tree view shows the database schema for 'SQLite_del_team1', including tables like Album, Artist, Customer, Employee, foo, Genre, Invoice, InvoiceLine, MediaType, Playlist, PlaylistTrack, Track, EmpView, and test. The main editor area shows a SQL query: `select * from Album;`. Below the editor, the 'Result - 1 <1>' tab is active, displaying a table of results. The table has columns for row number, AlbumId, Title, ArtistId, and Column1. The results show 9 rows of album data.

#	AlbumId	Title	ArtistId	Column1
1	1	For Those About To Rock We Salute You	1	[BLOB]
2	2	Balls to the Wall	2	[BLOB]
3	3	Restless and Wild	2	[NULL]
4	4	Let There Be Rock	1	[BLOB]
5	5	Big Ones	3	[NULL]
6	6	Jagged Little Pill	4	[NULL]
7	7	Facelift	5	[NULL]
8	8	Warner 25 Anos	6	[NULL]
9	9	Plays Metallica Bv Four Cellos	7	[NULL]

Shortcuts

Shortcut	Description
Ctrl+Enter	Execute SQL statement
Ctrl+ or Ctrl+Shift+Enter	Execute SQL statement in new tab
Alt+X	Execute script
Shift+Ctrl+E	Show execution plan
Alt+T	Open SQL Editor in separate browser tab
Shift+Ctrl+F	Format script
Ctrl+Z or CMD+Z	Undo
Ctrl+Y or Ctrl+Shift+Z or Shift+CMD+Z or CMD+Y	Redo

Statement Execution

Place the cursor on the line with the statement or select part of the script to execute the statement. Click on the **Run** icon in the left toolbar or use the `Ctrl+Enter` shortcut. The result of the statement execution will be shown under the script editor area. Results will be grouped (`Result - 1 <1>` , `Result - 1 <2>`) if statement execution is finished with more than one result.

The screenshot shows the SQL editor interface with a script editor and a results pane. The script editor contains the following SQL statements:

```

1 select * from Album;
2 select * from Customer;

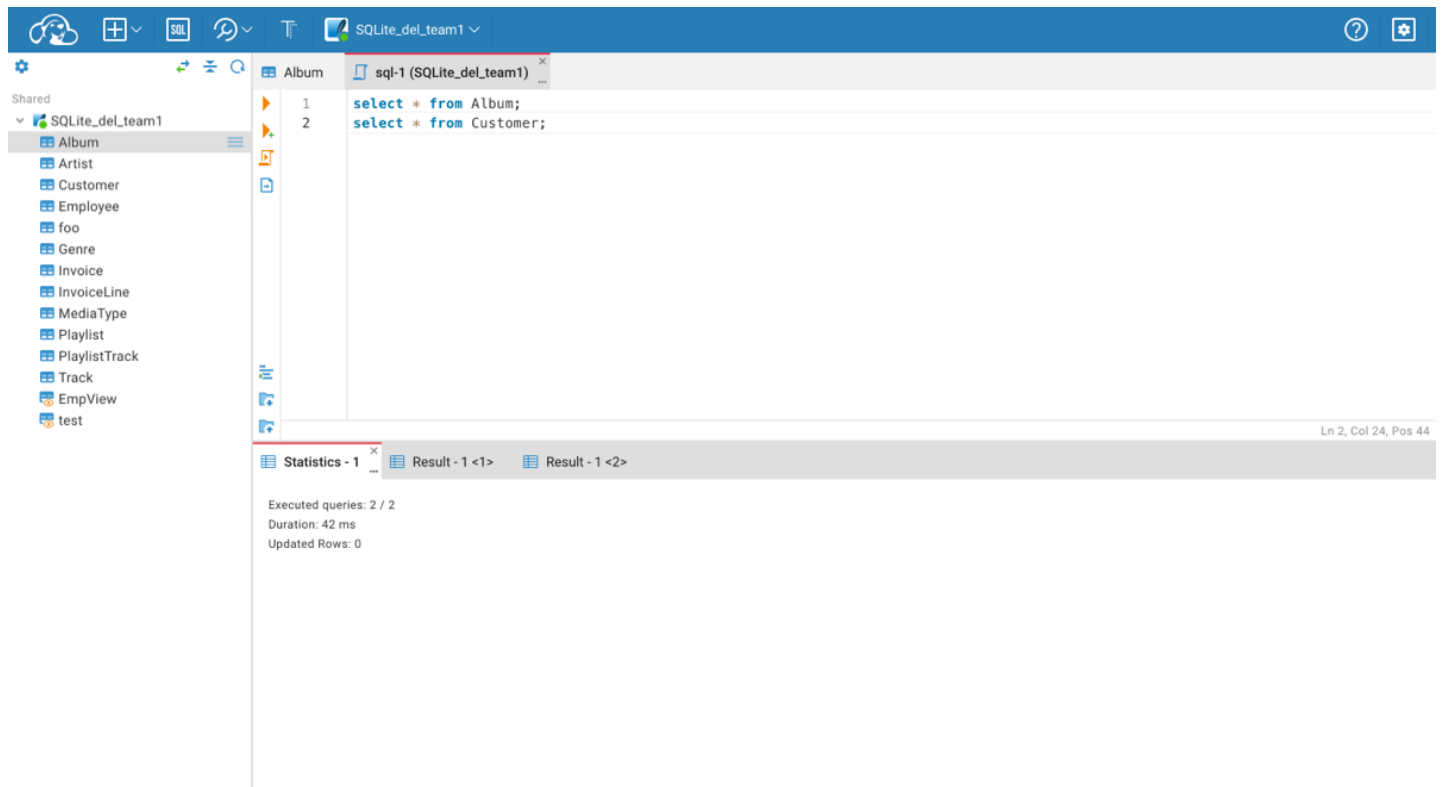
```

The results pane shows the execution results for the second statement, grouped as "Result - 1 <2>". The results are displayed in a table format with columns for Customer ID, First Name, Last Name, Company, Address, and City.

#	Customerid	FirstName	LastName	Company	Address	City
1	1	Luis	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José dos Campos
2	2	Leonie	Köhler	[NULL]	Theodor-Heuss-Straße 34	Stuttgart
3	3	François	Tremblay	[NULL]	1498 rue Bélanger	Montréal
4	4	Björn	Hansen	[NULL]	Ullevålsveien 14	Oslo
5	5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague
6	6	Helena	Holý	[NULL]	Rilská 3174/6	Prague
7	7	Astrid	Gruber	[NULL]	Rotenturmstraße 4, 1010 Innere Stadt	Vienne
8	8	Daan	Peeters	[NULL]	Grétrystraat 63	Brussels
9	9	Kara	Nielsen	[NULL]	Sander Boulevard 51	Copenhagen

Script Execution

Click on the **Script** icon in the left toolbar or use the `Alt+X` shortcut to execute the script. The summary result will be shown in the **Statistics** tab, and results will be shown in separate **Result** tabs.



Auto and Manual commit





By default, all database connections in the SQL Editor operate in **Auto-commit** mode, meaning that changes are automatically committed after each SQL statement is executed. To gain precise control over your transactions, you may switch to **Manual commit** mode.

Switching to Manual commit

In **Manual commit** mode, you manually determine when to commit or rollback transactions. This mode is essential when batch operations need to be treated as a single unit or when you need to inspect changes before making them permanent.

Here is a table outlining the actions and their corresponding icons in **Manual commit** mode:

Action	Icon	Description
--------	------	-------------

Switch to manual commit		Click to open a menu for manual transaction control. You will need to Commit or Rollback changes explicitly.
Commit		Click after executing SQL statements to save the changes to the database.
Rollback		Click to revert changes made by your SQL statements, undoing current transaction changes.
Switch to auto-commit		Click to return to Auto-commit mode, where changes are automatically committed.

Tip: After committing in **Manual commit** mode, performing a **Refresh** is necessary to see the newly added data in the Result tab.

Limitations of Rollback capabilities

In database management, not all commands support rollback operations. It is crucial to understand that Data Definition Language (DDL) commands—such as **CREATE**, **DROP**, or **ALTER** cannot be reversed with transactions for some databases. This means that once these commands are executed, they cannot be reversed even in manual commit mode.

Important: Always check the transaction support for the specific database you are working with to avoid irreversible operations.

Learn more

Table of contents

[Overview](#)

[Shortcuts](#)

[Statement Execution](#)

[Script Execution](#)

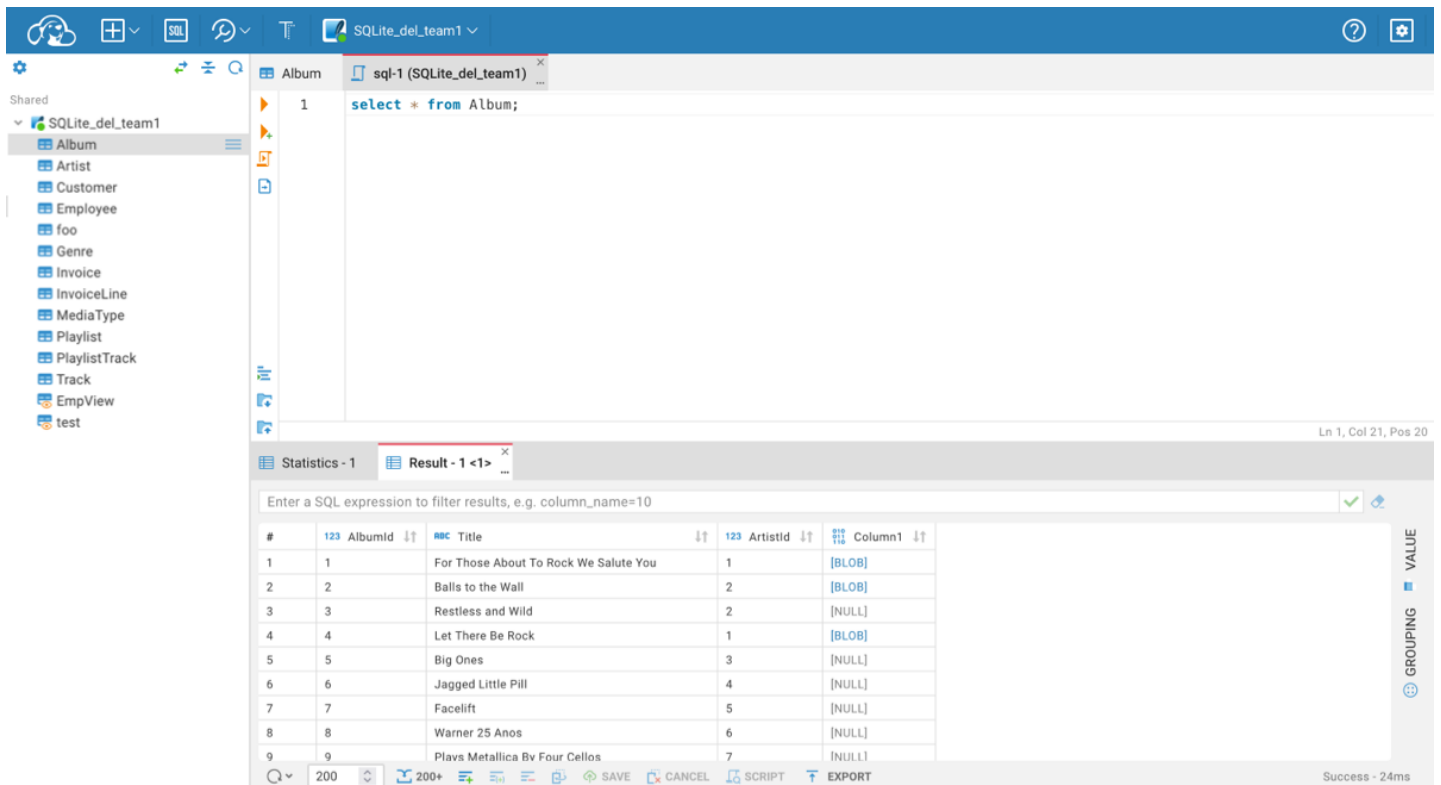
[Auto and Manual commit](#)

[Switching to Manual commit](#)

[Limitations of Rollback capabilities](#)

Overview

SQL Editor supports autocomplete, syntax highlight, statement execution, script execution, and execution plan for some databases.



The screenshot displays the SQL Editor interface. On the left, a tree view shows the database schema for 'SQLite_del_team1', including tables like Album, Artist, Customer, Employee, foo, Genre, Invoice, InvoiceLine, MediaType, Playlist, PlaylistTrack, Track, EmpView, and test. The main editor area contains the SQL query: `select * from Album;`. Below the editor, the 'Result - 1 <1>' tab shows a table of results. The table has columns for row number, AlbumId, Title, ArtistId, and Column1. The results are as follows:

#	AlbumId	Title	ArtistId	Column1
1	1	For Those About To Rock We Salute You	1	[BLOB]
2	2	Balls to the Wall	2	[BLOB]
3	3	Restless and Wild	2	[NULL]
4	4	Let There Be Rock	1	[BLOB]
5	5	Big Ones	3	[NULL]
6	6	Jagged Little Pill	4	[NULL]
7	7	Facelift	5	[NULL]
8	8	Warner 25 Anos	6	[NULL]
9	9	Plays Metallica Bv Four Cellos	7	[NULL]

The interface also shows a status bar at the bottom with 'Success - 24ms' and various action buttons like SAVE, CANCEL, SCRIPT, and EXPORT.

Shortcuts

Shortcut	Description
Ctrl+Enter	Execute SQL statement
Ctrl+ or Ctrl+Shift+Enter	Execute SQL statement in new tab
Alt+X	Execute script
Shift+Ctrl+E	Show execution plan
Alt+T	Open SQL Editor in separate browser tab
Shift+Ctrl+F	Format script
Ctrl+Z or CMD+Z	Undo
Ctrl+Y or Ctrl+Shift+Z or Shift+CMD+Z or CMD+Y	Redo

Statement Execution

Place the cursor on the line with the statement or select part of the script to execute the statement. Click on the **Run** icon in the left toolbar or use the `Ctrl+Enter` shortcut. The result of the statement execution will be shown under the script editor area. Results will be grouped (`Result - 1 <1>` , `Result - 1 <2>`) if statement execution is finished with more than one result.

The screenshot shows the SQL editor interface. The script editor contains the following SQL statements:

```

1 select * from Album;
2 select * from Customer;

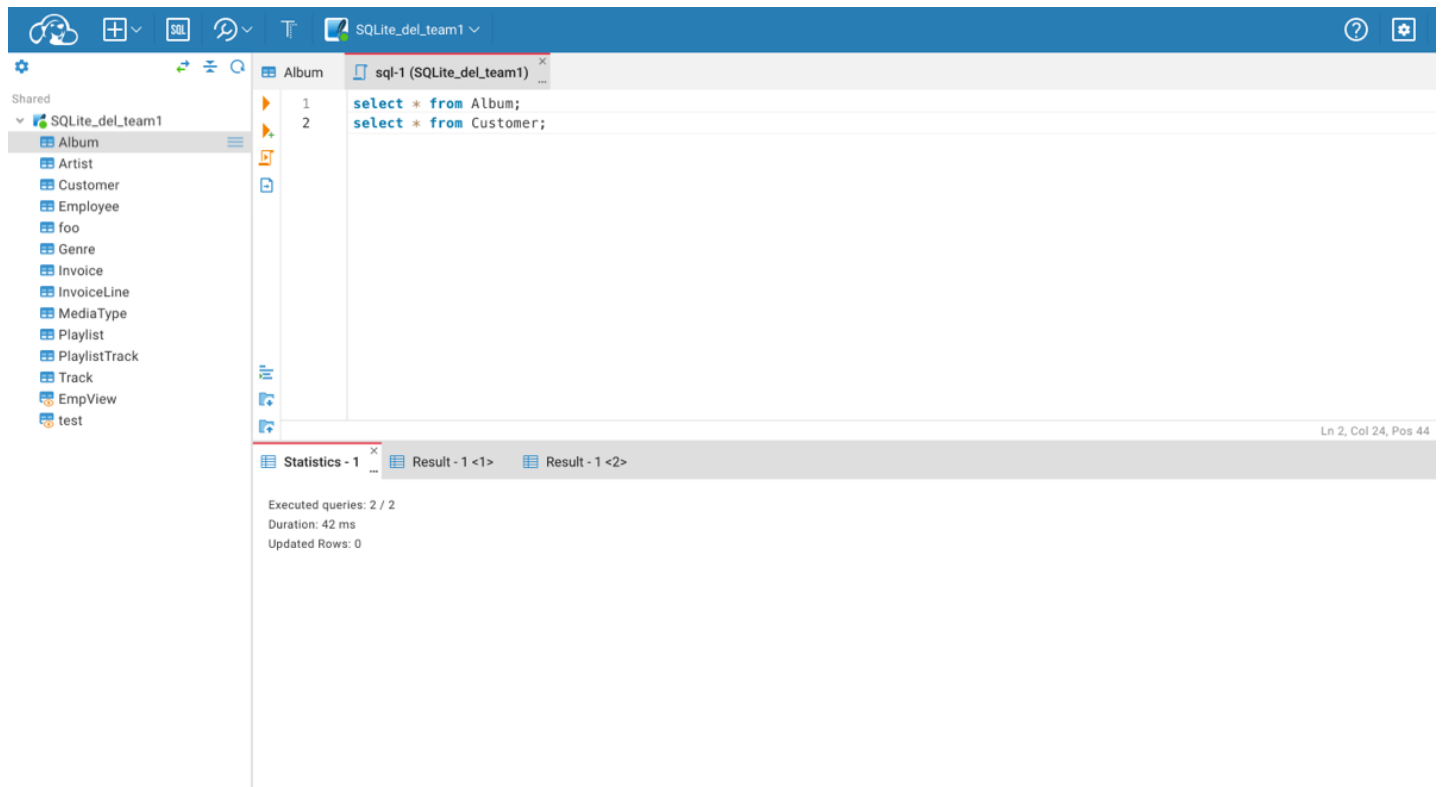
```

The results are displayed in a table with the following columns: #, CustomerID, FirstName, LastName, Company, Address, and City. The data is as follows:

#	CustomerID	FirstName	LastName	Company	Address	City
1	1	Luis	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José dos Campos
2	2	Leonie	Köhler	[NULL]	Theodor-Heuss-Straße 34	Stuttgart
3	3	François	Tremblay	[NULL]	1498 rue Bélanger	Montréal
4	4	Björn	Hansen	[NULL]	Ullevålsveien 14	Oslo
5	5	František	Wichterlová	JetBrains s.r.o.	Klanova 9/506	Prague
6	6	Helena	Holý	[NULL]	Rilská 3174/6	Prague
7	7	Astrid	Gruber	[NULL]	Rotenturmstraße 4, 1010 Innere Stadt	Vienne
8	8	Daan	Peeters	[NULL]	Grétrystraat 63	Brussels
9	9	Kara	Nielsen	[NULL]	Sander Boulevard 51	Copenhagen

Script Execution

Click on the **Script** icon in the left toolbar or use the `Alt+X` shortcut to execute the script. The summary result will be shown in the **Statistics** tab, and results will be shown in separate **Result** tabs.



Auto and Manual commit





By default, all database connections in the SQL Editor operate in **Auto-commit** mode, meaning that changes are automatically committed after each SQL statement is executed. To gain precise control over your transactions, you may switch to **Manual commit** mode.

Switching to Manual commit

In **Manual commit** mode, you manually determine when to commit or rollback transactions. This mode is essential when batch operations need to be treated as a single unit or when you need to inspect changes before making them permanent.

Here is a table outlining the actions and their corresponding icons in **Manual commit** mode:

Action	Icon	Description
--------	------	-------------

Switch to manual commit		Click to open a menu for manual transaction control. You will need to Commit or Rollback changes explicitly.
Commit		Click after executing SQL statements to save the changes to the database.
Rollback		Click to revert changes made by your SQL statements, undoing current transaction changes.
Switch to auto-commit		Click to return to Auto-commit mode, where changes are automatically committed.

Tip: After committing in **Manual commit** mode, performing a **Refresh** is necessary to see the newly added data in the Result tab.

Limitations of Rollback capabilities

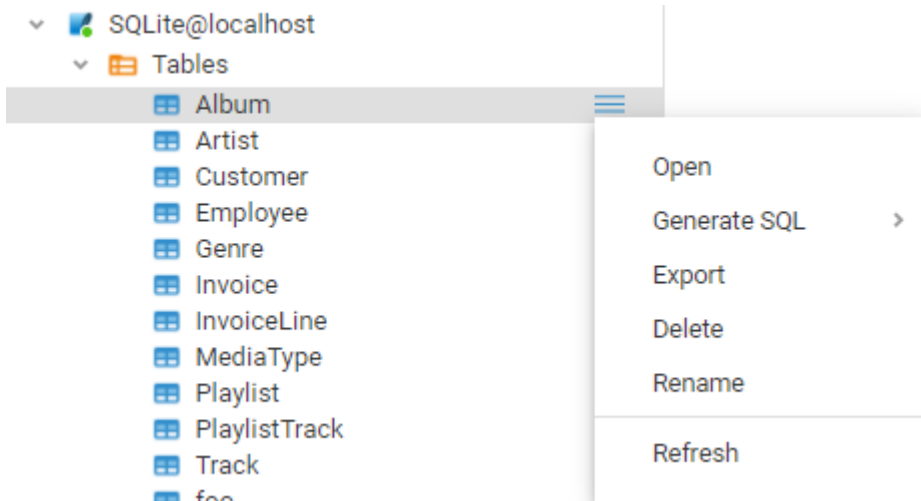
In database management, not all commands support rollback operations. It is crucial to understand that Data Definition Language (DDL) commands—such as **CREATE**, **DROP**, or **ALTER** cannot be reversed with transactions for some databases. This means that once these commands are executed, they cannot be reversed even in manual commit mode.

Important: Always check the transaction support for the specific database you are working with to avoid irreversible operations.

Learn more

You can perform data export for database tables:

1. Select a table you want to export in the Database Navigator or the Metadata Editor. In the context menu, choose **Export**. You can also export data from the Data Editor and the ResultSet in the SQL Editor.



2. Choose your export format. CloudBeaver supports different output formats:

- CSV
- DBUnit
- JSON
- Markdown
- Source code
- SQL
- TXT
- XML
- XLSX
- HTML

3. Set the export configuration options. They are specific to the data format you chose.

Export configuration (CSV)



NAME	VALUE	
Characters escape	quotes	▼
Delimiter	,	
File extension	csv	
Format numbers	false	
Header	top	▼
Header format	label	▼
NULL string		
Quote always	disabled	▼
Quote character	"	

BACK

CANCEL

EXPORT

4. Press **Export**. \ Note: avoid changing data in the tables you have selected to be exported while the exporting is in progress.
5. Press **Download** in the pop up dialog.

Learn more

Table of contents

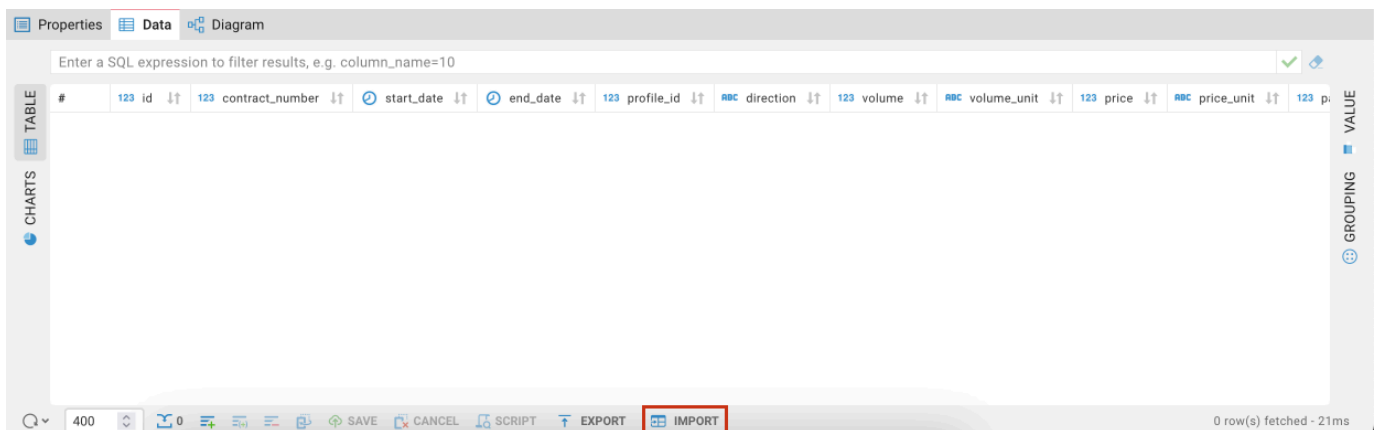
[Import process steps](#)

[Import configuration details](#)

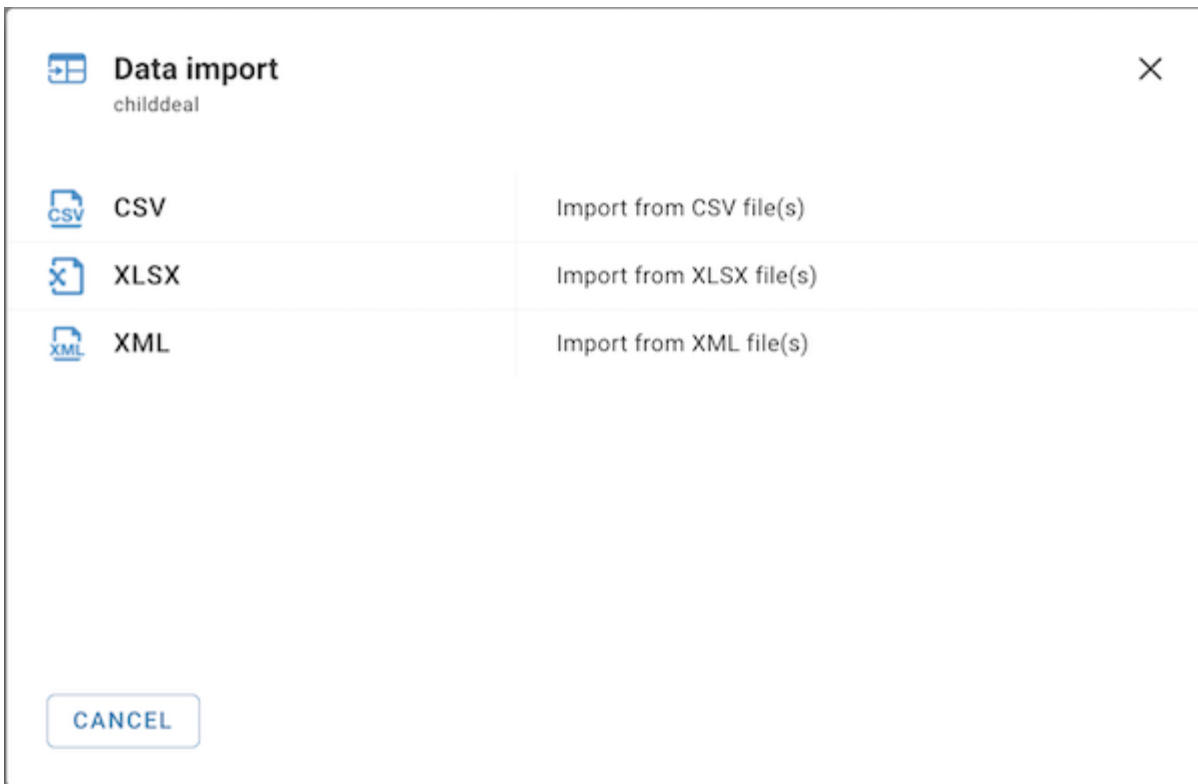
In CloudBeaver, data import is accessible through the [Data Editor](#) when a table is open.

Import process steps

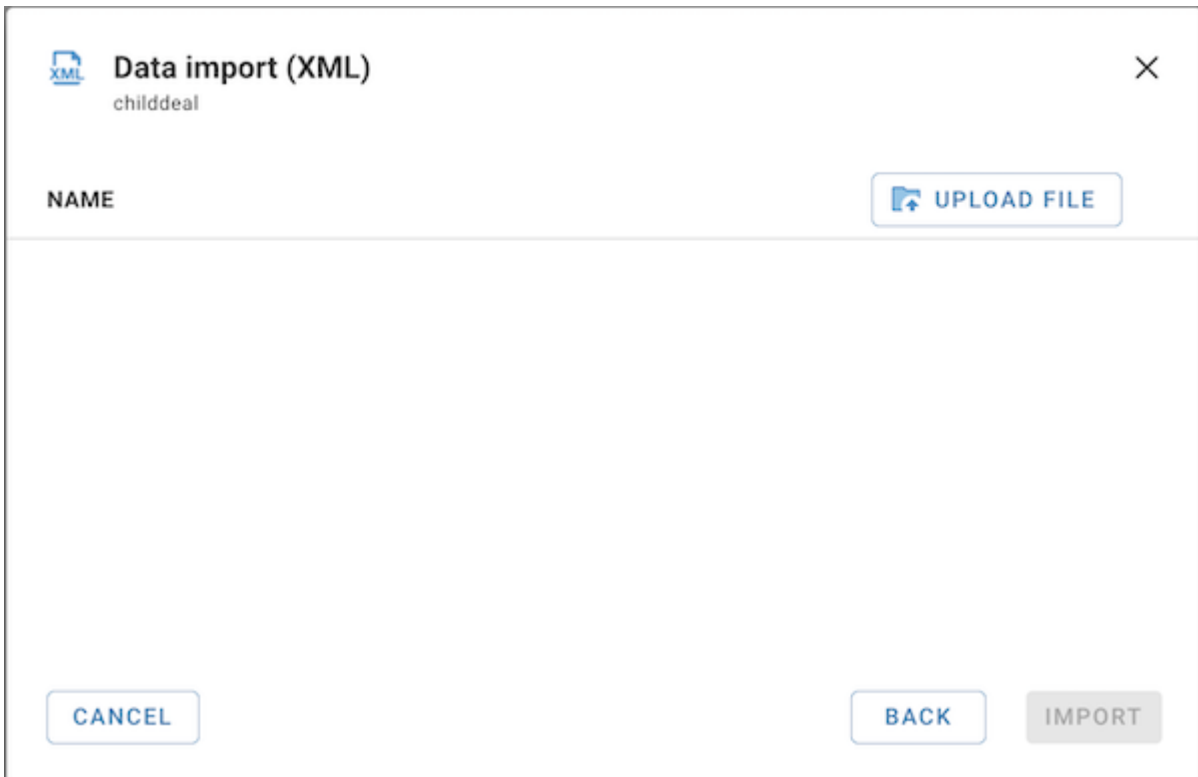
1. Click the **Import** button to open a pop-up window for the import process.



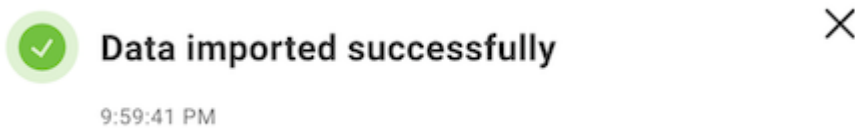
2. Choose the file type for import.



3. Choose the file from local storage and press **Import**.



4. Upon successful import, a confirmation message will appear.



Import configuration details

The import process in CloudBeaver includes several key features and limitations:

- The Community Edition (CE) only allows for `CSV` files. Pro versions include `CSV`, `XLSX`, and `XML` files.
- In CloudBeaver Team Edition, data import is available to users with the roles of Editor, Manager, Developer, and Administrator.
- `CSV` files should be comma-delimited.
- The structure (DDL) of the uploaded table must match the existing table, specifically in terms of columns.
- Only unique primary key values are accepted to ensure data integrity and avoid duplicates.
- The import operation does not block the interface, allowing for continued work while the import is processed.
- The system currently supports uploading one file at a time.

Learn more

Table of contents

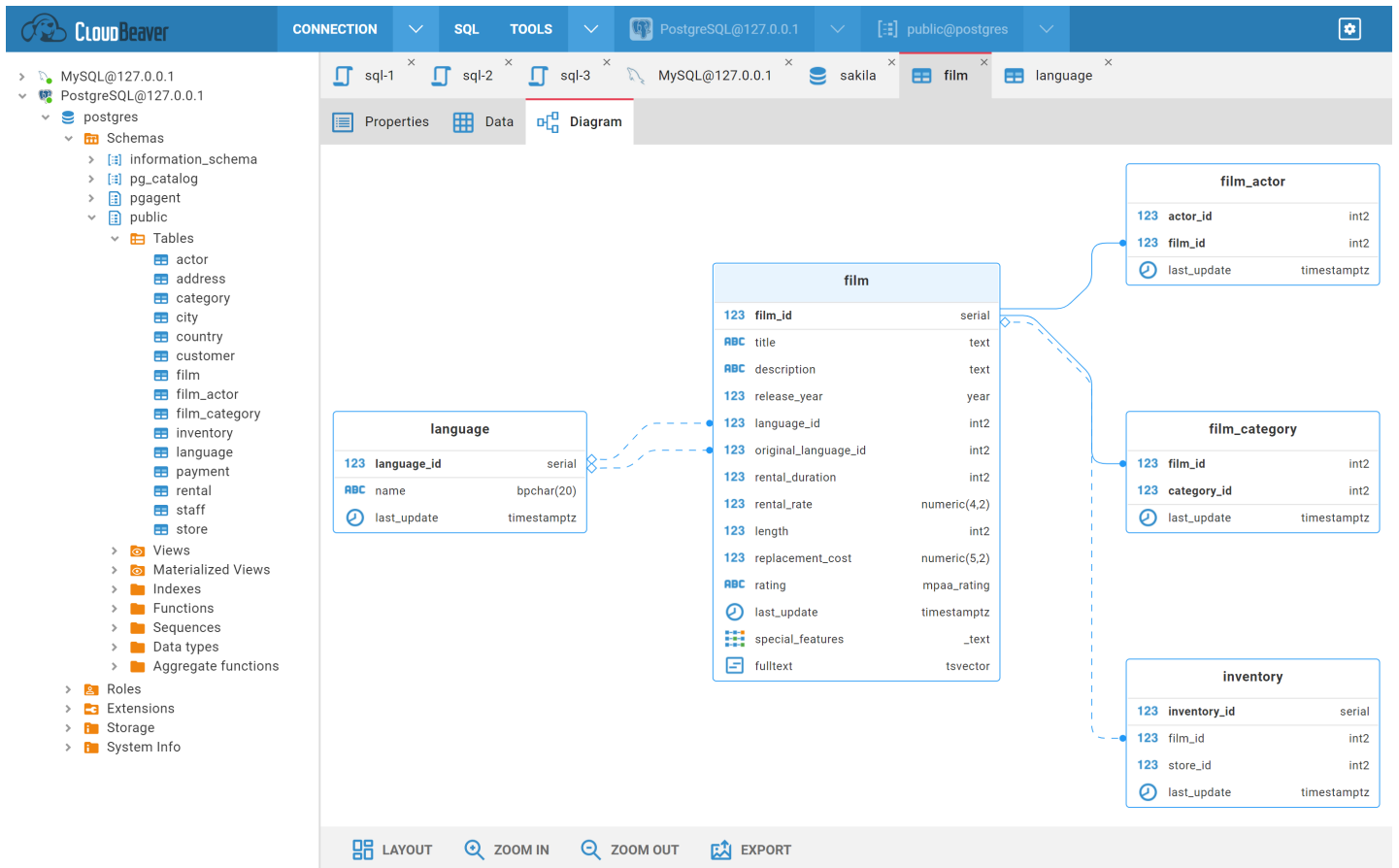
[Overview](#)

[Display entities with attributes](#)

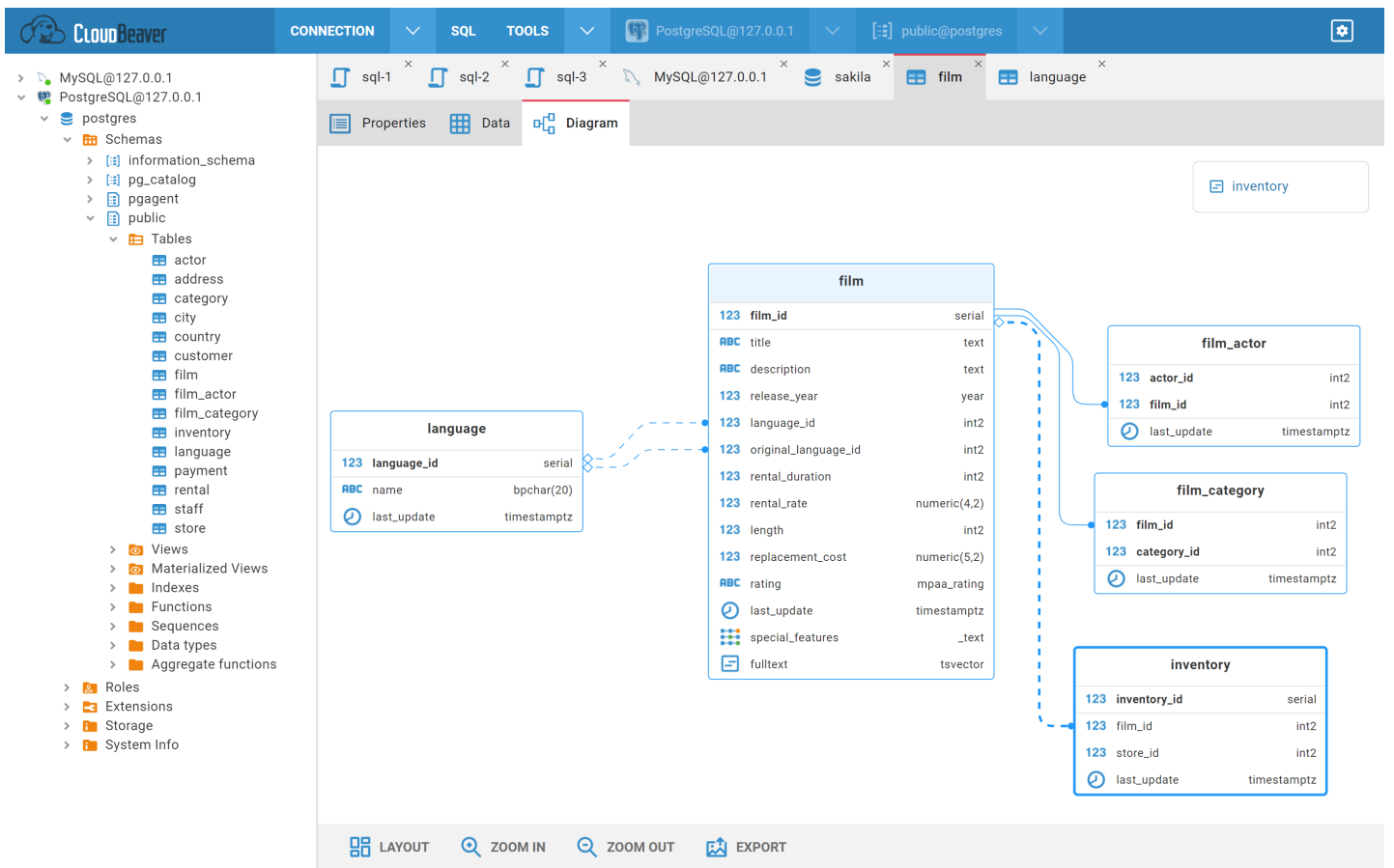
Overview

Display entities with attributes

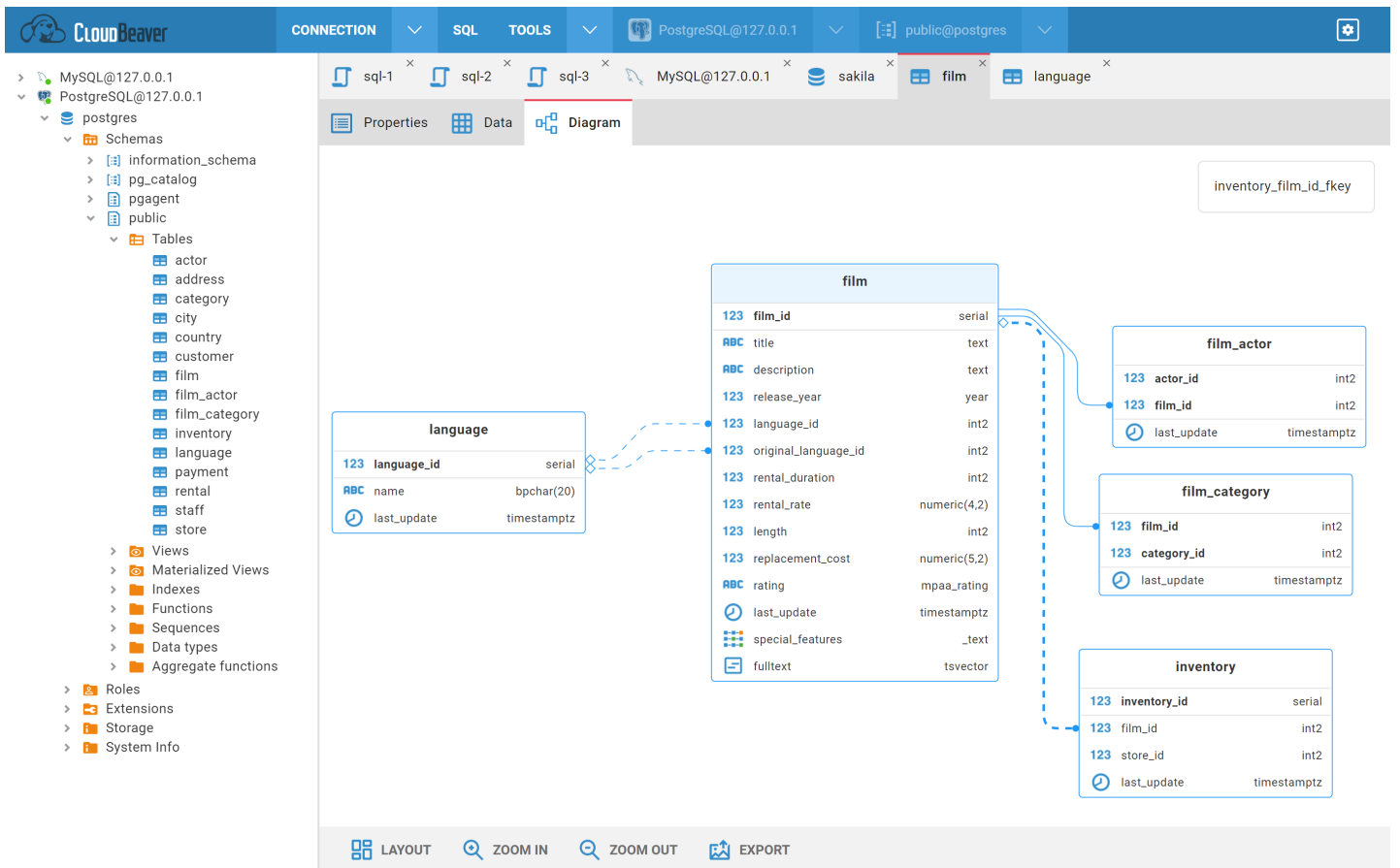
1. Navigate to your connection and open a table or schema
2. Select the "Diagram" tab (if the tab is not presented then the object does not support the diagram presentation)



You can click on an entity to highlight it:



You can click on a relation to get highlight it:

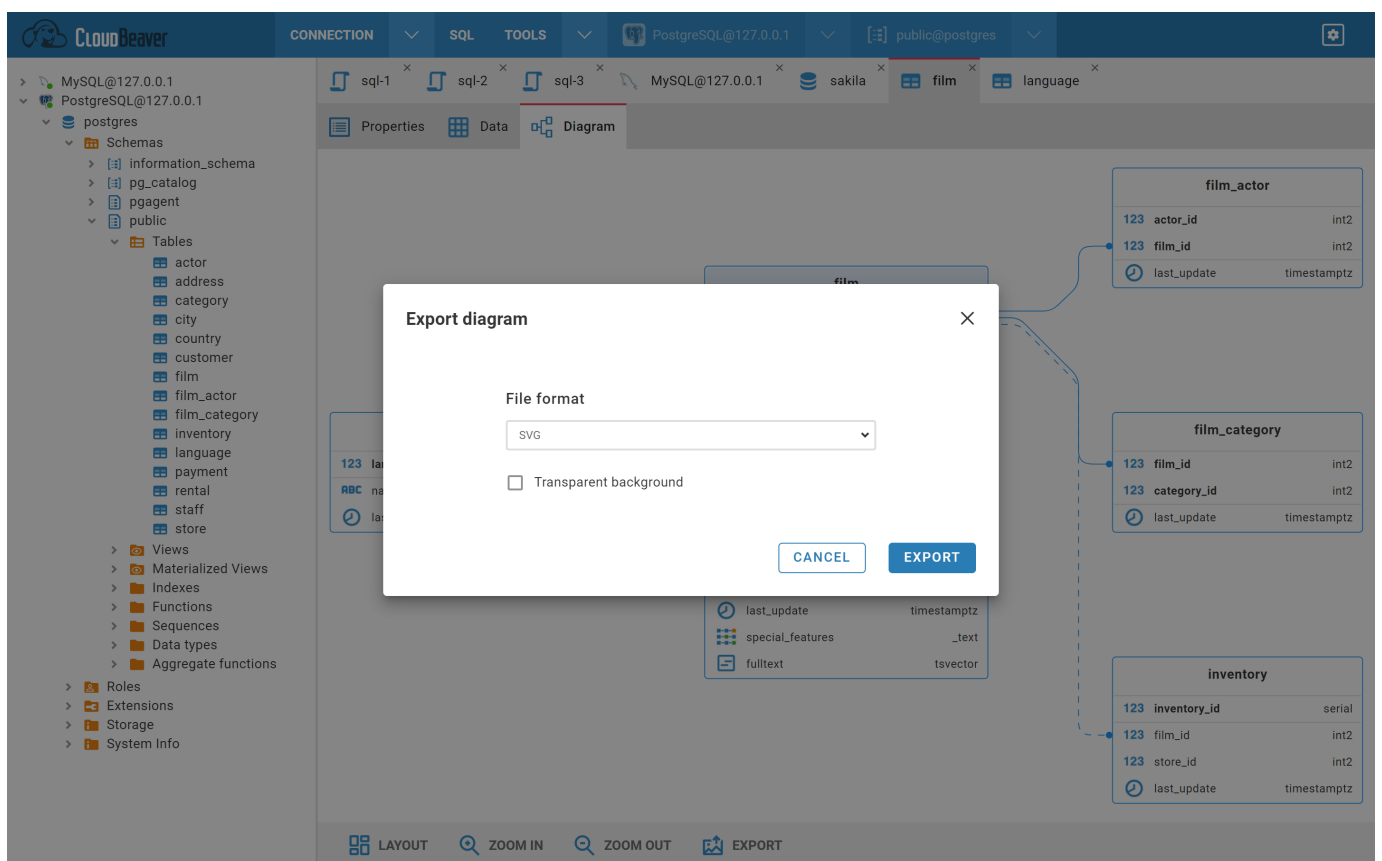


You can open the entity in the metadata editor by:

- double-clicking on the entity
- double-clicking on the entity attribute
- clicking on the link in the entity tooltip

On the bottom toolbar you can find different buttons:

1. Layout - diagram auto layout
2. Zoom in/out
3. Export - export diagram in a `png` or `svg` format



Learn more

Table of contents

[Overview](#)

[Creating a Visual Query](#)

[Filtering](#)

[Sorting](#)

[Executing a Visual Query](#)

[Shortcuts](#)

[The Visual Query Builder symbols](#)

[Table symbols](#)

[Join symbols](#)

[Settings](#)

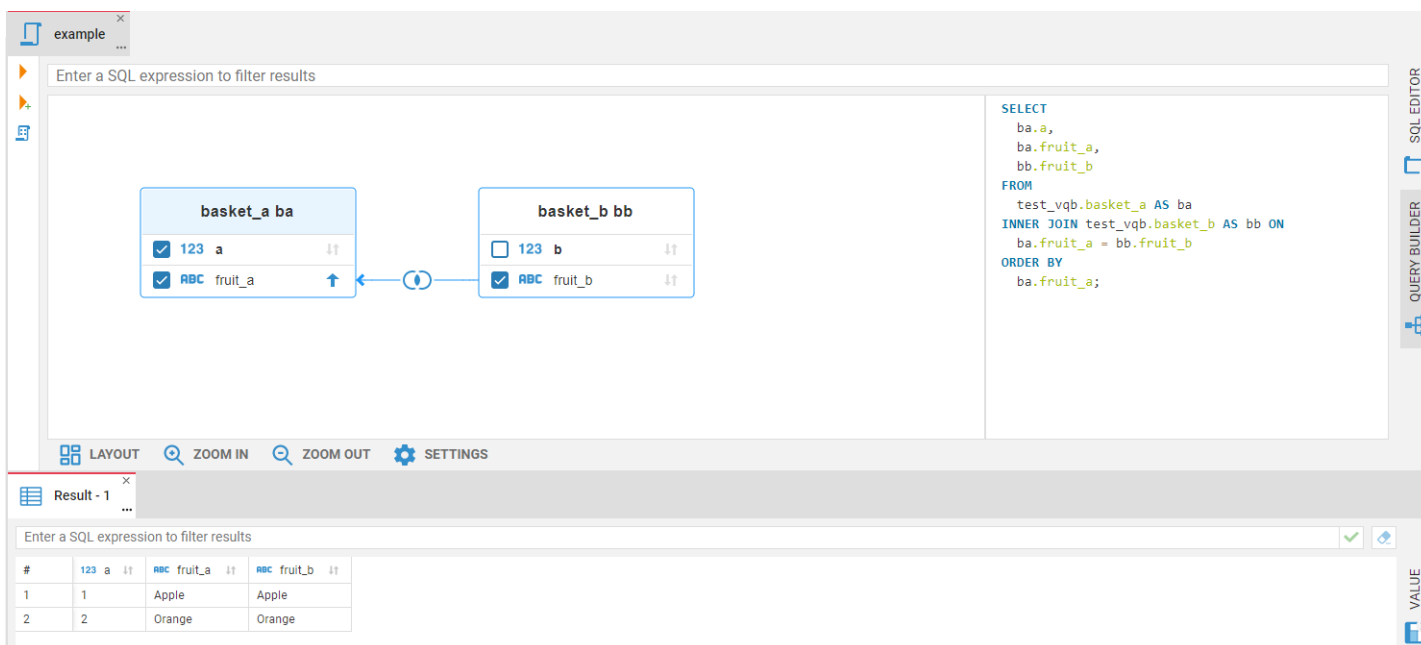
[Visualization of an existing SQL query](#)

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

Overview

The Visual Query Builder is a user-friendly visualization tool that can help you to create queries to the database and see results. You do not need to know SQL language to work in it. The Visual Query Builder may be useful for:

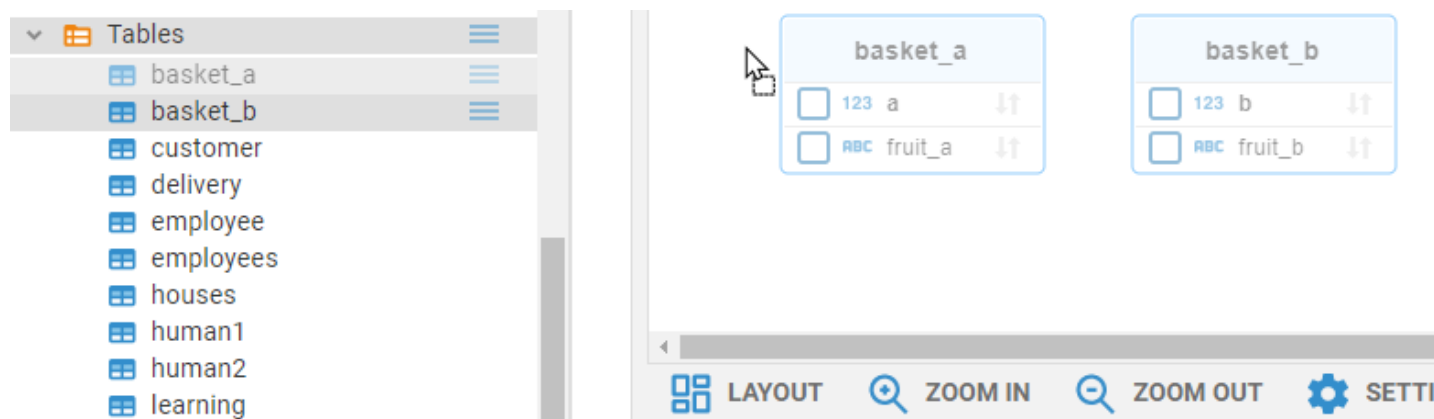
- building queries;
- complex queries analysis;
- easy query editing.



To open the Visual Query Builder, click the Query Builder tab in the SQL Editor right toolbar.

Creating a Visual Query

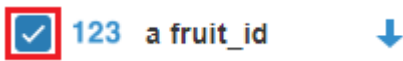
1. Select tables in the Navigator tree and drag-and-drop them into the Visual Query Builder area. The existing connections between the tables will automatically be displayed. The tables will also be added to the SQL expression which can be found in the field to the right of the diagram.



2. To create a new join between tables, connect their columns holding the left mouse button. The connection between the selected columns of the tables will appear in the diagram and the Inner Join will be added to the SQL script.



3. You can change a join type clicking the join label on the connection line.
4. To remove a join between tables, click on the line, then press the Delete button. The connection will be removed from the diagram and the join will disappear from the SQL script.
5. By default all tables' columns are included in the query. If you only want to see certain columns in your query result, select the checkbox near the column name.



Filtering

6. WHERE condition with the filter value is used for filtering. To add a filter, write it in the top filter field.

Column name	Operation Sign	Value
A table column name. You have to write a table alias before if another column has the same name	The most common signs: =, >, <, <>, LIKE, ILIKE, BETWEEN	A column value, used as a parameter. Text and time values must be rounded by single quotes, numeric values do not need any quotes

Filter example:

e.employeeid>2

```

SELECT
*
FROM
test_vqb.customer AS c
INNER JOIN test_vqb.payment AS p ON
c.customerid = p.customerid
INNER JOIN test_vqb.employees AS e ON
p.employeeid = e.employeeid
WHERE
e.employeeid>2;

```

LAYOUT ZOOM IN ZOOM OUT SETTINGS

Result - 1

Enter a SQL expression to filter results



#	123 customerid	abc name	123 paymentid	123 customerid	123 employeeid	123 amount	123 employeeid	abc name
1	4	Abid	3	4	3	3000	3	Hassan
2	5	Sia	4	5	4	4000	4	Anna
3	3	Hussain	5	3	5	4000	5	Sau
4	6	Kait	6	6	6	5000	6	Kelsie
5	7	Tony	7	7	7	5000	7	Tory
6	8	Sam	8	8	8	5000	8	Salley

Sorting

7. To apply a sorting condition to a column, press the sorting icon next to a column name on the diagram. The column will be sorted in ascending order and the conditional expression ORDER BY will be added to the SQL script. To sort the column in descending order, press the sorting icon again to select the down arrow. If you want to remove a condition, continue to click the sorting icon to deactivate it. Sorting can be applied to multiple columns in different tables. First, apply sorting on the first column you wish to sort, and then on the second, third and so on. You can sort numbers, texts, dates, time and other values.



Executing a Visual Query

Use the Execute SQL statement button  on the left pane to execute a query and get the results in the same tab. If you want to see the result in a new tab, press the Execute SQL statement in a new tab button .

Shortcuts






You can use the same shortcuts as in the SQL Editor to execute the Visual Query.

Key	Description
Ctrl+Enter	Execute the SQL statement
Ctrl+\ or Ctrl+Shift+Enter	Execute the SQL statement in a new tab

The Visual Query Builder symbols




The Visual Query Builder uses the following visual tools to display queries on the diagram:

Table symbols

Symbol	Description
	Table Primary Key is bold and displayed at the top of the table.
	Table Alias is used to shorten your Join Statement.
	Colored table header marks the first table in your Join Statement.
	Colorless header marks a joined table in your Join Statement.
	Line goes from the joined table to the first table.

Join symbols

Available Join types are described in the table below. The Visual Query Builder can show results only for those types of Joins that are supported by your database.

Symbol	Description
	Inner Join
	Left Join
	Left Outer Join

	Right Join
	Right Outer Join
	Full Join
	Full Outer Join
	Cross Join

Settings

You can customize the diagram view using the bottom toolbar to make the work with the diagram easier.



- **Layout** updates the diagram view to display all of its objects in the most optimal way.
- **Zoom in** and **Zoom out** enlarges or shrinks the diagram view.
- **Settings** menu contains additional settings of the Visual Query Builder. Press the Settings button at the bottom toolbar to open it.
 - **Layout on update** enables Auto-layout feature. As soon as you add a new object to the diagram, the diagram view will automatically be updated to display all of its objects in the most optimal way.
 - **Show join type on entities** moves Join labels from lines into headers of joined tables.
 - **Show Type** adds information about column types into entities.
 - **Show Icons** adds icons of column types into entities.
 - **Notation** changes the representation of connection lines. Simple notation is set by default. You can change it to the IDEF1X language type.

Visualization of an existing SQL query

If you write a JOIN statement by yourself and then want to convert it to the diagram view, just switch the SQL Editor with your statement to the Visual Query Builder.

Note: the Visual Query Builder can transform the syntax of your query, but it does not affect the query result in the Result set.

Learn more

Table of contents

[Understanding the AI integration in CloudBeaver](#)

[Initial setup](#)

[Data privacy](#)

[AI settings and customization](#)

[Credentials for OpenAI](#)

[Credentials for Azure AI](#)

[Preferences](#)

[AI smart completion usage](#)

[Accessing prompts history](#)

[Disabling AI features](#)

[Best practices for question formulation](#)

CloudBeaver offers the ability to construct SQL queries using natural language through **AI smart completion** feature. This capability is achieved through integrations with both OpenAI's [GPT-3 language model](#) and [Azure OpenAI](#).

Note: CloudBeaver is not affiliated with OpenAI. Integration is achieved through the public API.

- To utilize this feature, register with OpenAI and obtain a secret key.

Understanding the AI integration in CloudBeaver

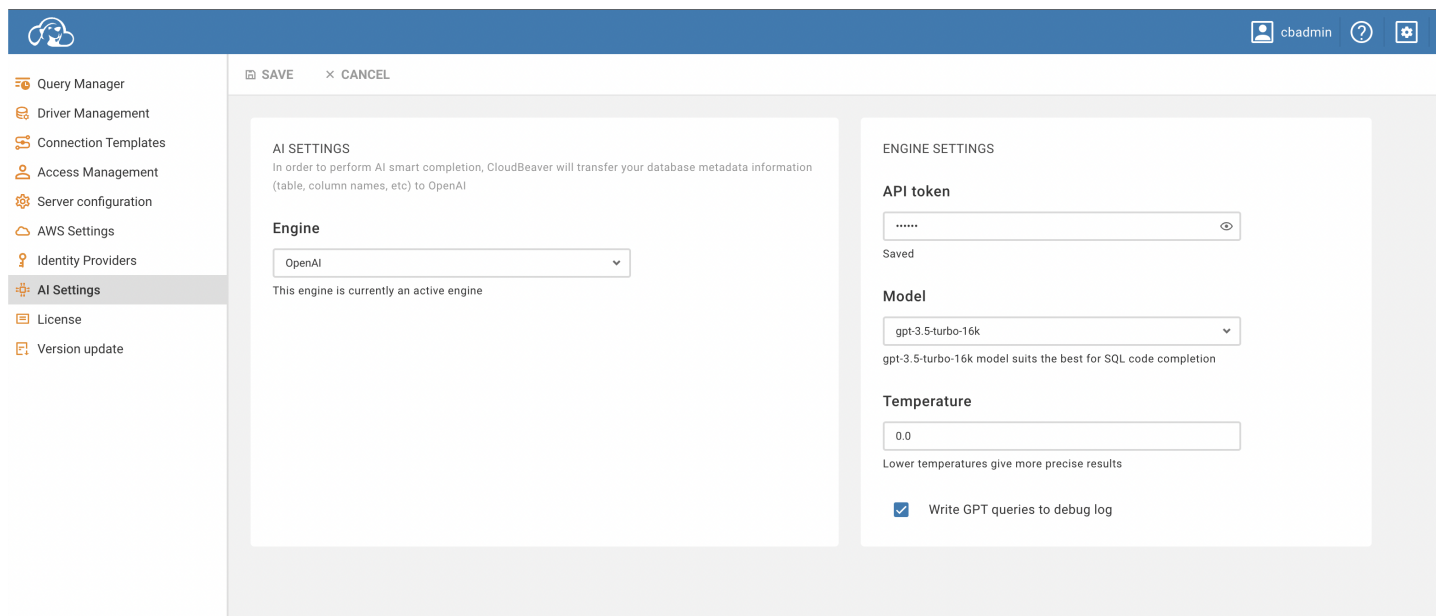
With the **AI smart completion** feature, you can type queries in natural language and CloudBeaver will convert them into SQL statements. This tool simplifies writing complex queries by interpreting your input and automatically generating the correct SQL code.

Initial setup

To activate the AI features in CloudBeaver, configure the API token:

1. Navigate to **Administration page** -> **Server Configuration tab**
2. Ensure the **AI Service** option is activated.
3. Navigate to **AI Settings tab** -> **Choose an engine**
4. In the **API token** field, input your AI secret key.
5. Save the changes.

For instructions on utilizing the AI features, visit the .



Data privacy

We prioritize data safety and user privacy. In this section, we outline how data is managed and the measures taken to protect user privacy when using the AI features.

To enable the AI features capabilities, table and column names from the current database schema are transmitted to OpenAI. This step is crucial for accurately translating user requests into SQL queries. Key considerations regarding data privacy are as follows:

- **No Table Data:** Only metadata like table and column names are shared with OpenAI. Actual table data is not transmitted.
- **Log Transparency:** The entire request can be logged for your review. To enable this, navigate to **AI Settings tab** and check the **Write GPT queries to debug log** option.

- **Azure OpenAI privacy:** If you use Azure OpenAI, be aware that it operates under its own [privacy policy](#). It's recommended to review their terms before using.

AI settings and customization

To utilize the AI-enhanced functionalities within CloudBeaver, certain configurations and setup processes are required. This section offers a comprehensive guide on initial setup and customization options to tailor the AI integration according to specific preferences.

Credentials for OpenAI

1. Sign up on the [OpenAI platform](#).
2. Navigate to the [API Keys section](#) and generate a new secret key.
3. Insert this key into CloudBeaver's **Engine Settings**.

Credentials for Azure AI

1. Sign up on the [Azure platform](#).
2. Navigate to the [Azure Portal](#) and create a new AI service under the AI + Machine Learning section.
3. Generate and copy the credentials for the newly created service.
4. Insert these credentials into CloudBeaver's **Engine Settings**.

Preferences

For specific requirements or troubleshooting, you might want to adjust some of the following settings:

- Navigate to **Administration page** -> **AI Settings** -> **Engine settings** to access these settings.

Setting	Description
API token	Input your secret key from the OpenAI platform.

Model	Choose the AI model (recommended: gpt-3.5-turbo for SQL).
Temperature	Control AI's creativity from 0.0 (more precise) to 0.9 (more diverse). Note that higher temperature can lead to less predictable results.
Write GPT queries to debug log	Logs your AI requests.

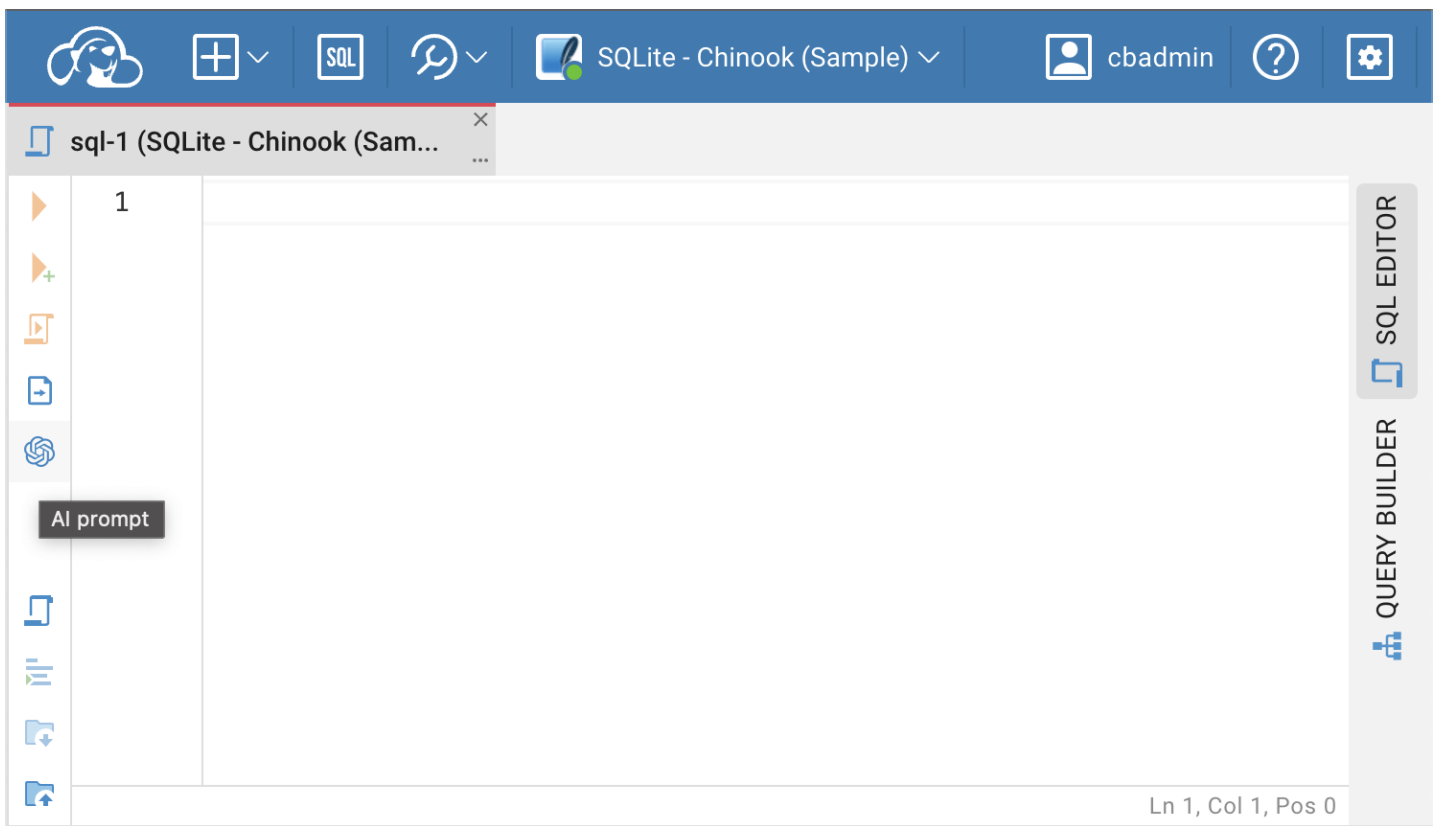
There is also an option to switch the **Engine** from OpenAI to Azure OpenAI. Azure provides a set of distinct settings:

Setting	Description
Endpoint	Configure a custom endpoint URL for Azure OpenAPI interactions.
API version	Select the version of the API you wish to use.
Deployment	Specify the deployment name chosen during model deployment.
Context size	Choose the context size between 2048 and 32768 . A larger number allows the AI to use more data for better answers but may slow down response time. Choose based on your balance of accuracy and speed.

AI smart completion usage

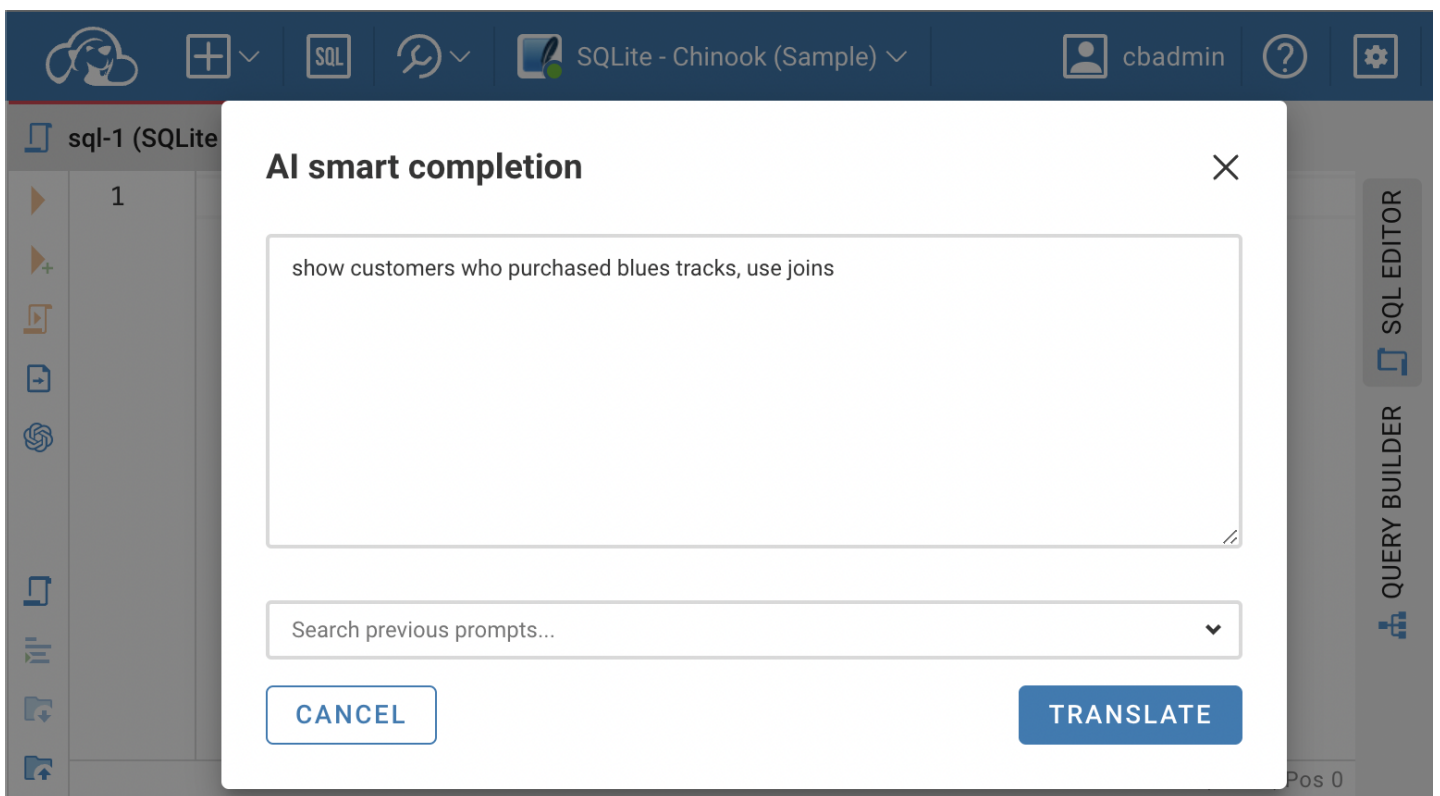
To interact with databases using the **AI Smart completion** feature:

1. Launch the **SQL Editor**.
2. Click on the **AI smart completion** icon  located in the left toolbar of the **SQL Editor**.



3. Input your natural language request in the **AI smart completion** window.

4. Click **Translate** to obtain the SQL query.



Accessing prompts history

Prompts history allows you to review previous prompts in the scope of the session for the chosen tab.

Disabling AI features

To hide the **AI smart completion** icon in the SQL Editor:

- Navigate to **Administration page** -> **Server Configuration tab** -> **Services section**.
- Deselect **AI option**.

Best practices for question formulation

When using AI to generate SQL queries, it's essential to provide clear and specific input. Here's how to optimize your questions:

- **Language:** While AI supports multiple languages, it's recommended to use English for best results.
- **Database knowledge:** Familiarity with your database structure enhances the accuracy of generated queries.
- **Explicit details:** If you know certain tables or columns that should be part of the query, include them in your request for better accuracy.

For instance, if you're using the CloudBeaver sample SQLite database, you might phrase your request as:

Example 1: "List all customers from Italy"

Resulting SQL:

```
SELECT *  
FROM customers  
WHERE country = 'Italy';
```

Example 2: "montre les clients de France"

```
SELECT *  
FROM customer  
WHERE country = 'France';
```

Example 3: "show customers who purchased blues tracks, use joins"

```
SELECT c.FirstName, c.LastName, t.Name, g.Name
FROM Customer c
      JOIN Invoice i ON c.CustomerId = i.CustomerId
      JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
      JOIN Track t ON il.TrackId = t.TrackId
      JOIN Genre g ON t.GenreId = g.GenreId

WHERE g.Name = 'Blues'
ORDER BY c.LastName, c.FirstName;
```

Example 4: "get names of customers who purchased blues tracks, use joins"

```
SELECT DISTINCT c.FirstName, c.LastName
FROM Customer c
      JOIN Invoice i ON c.CustomerId = i.CustomerId
      JOIN InvoiceLine il ON i.InvoiceId = il.InvoiceId
      JOIN Track t ON il.TrackId = t.TrackId
      JOIN Genre g ON t.GenreId = g.GenreId
WHERE g.Name = 'Blues';
```


Learn more

Please note: This article has been updated and its content is now available in a revised form at a new location.

Please refer to the updated article [Cloud Explorer](#) for the most current information.

Learn more

Table of contents

[Overview](#)

[Configure the Cloud storage](#)

[Work with SQL scripts](#)

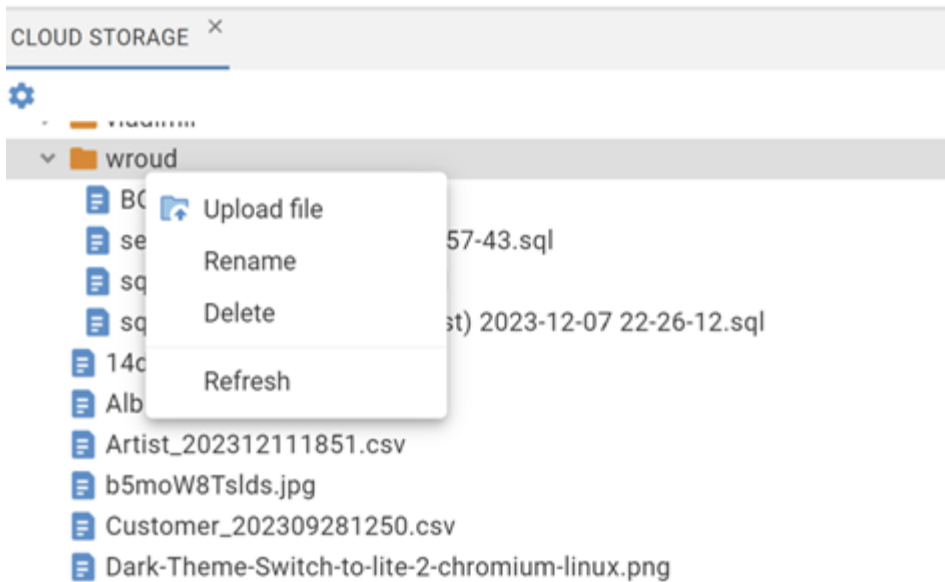
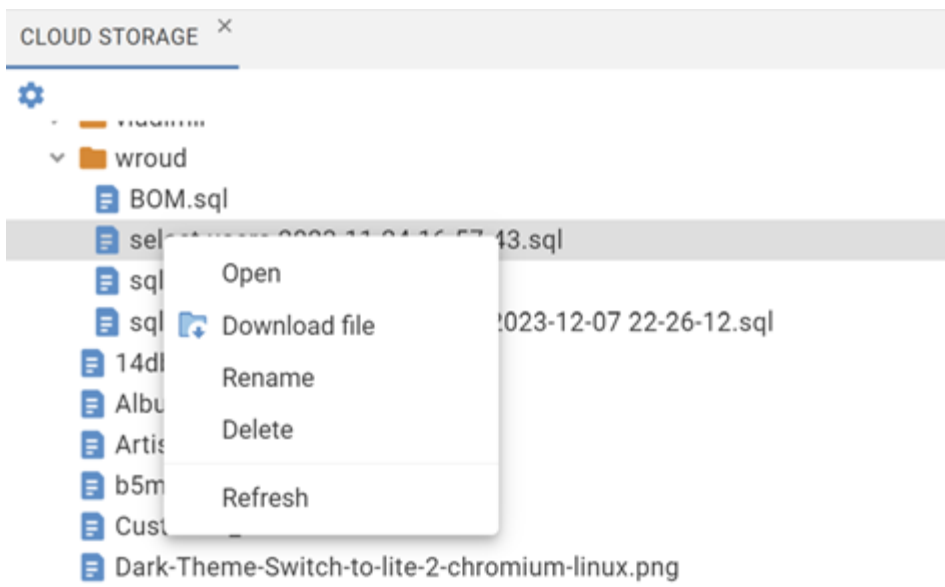
Note: This feature is available in [Enterprise](#) and [Team Edition](#) editions only.

Overview

CloudBeaver provides the ability to use your cloud storage services through a web interface. Currently, we support Amazon Simple Storage Service and Google Cloud Storage.

Within the file browser, you can interact with it like a regular file system, allowing you to:

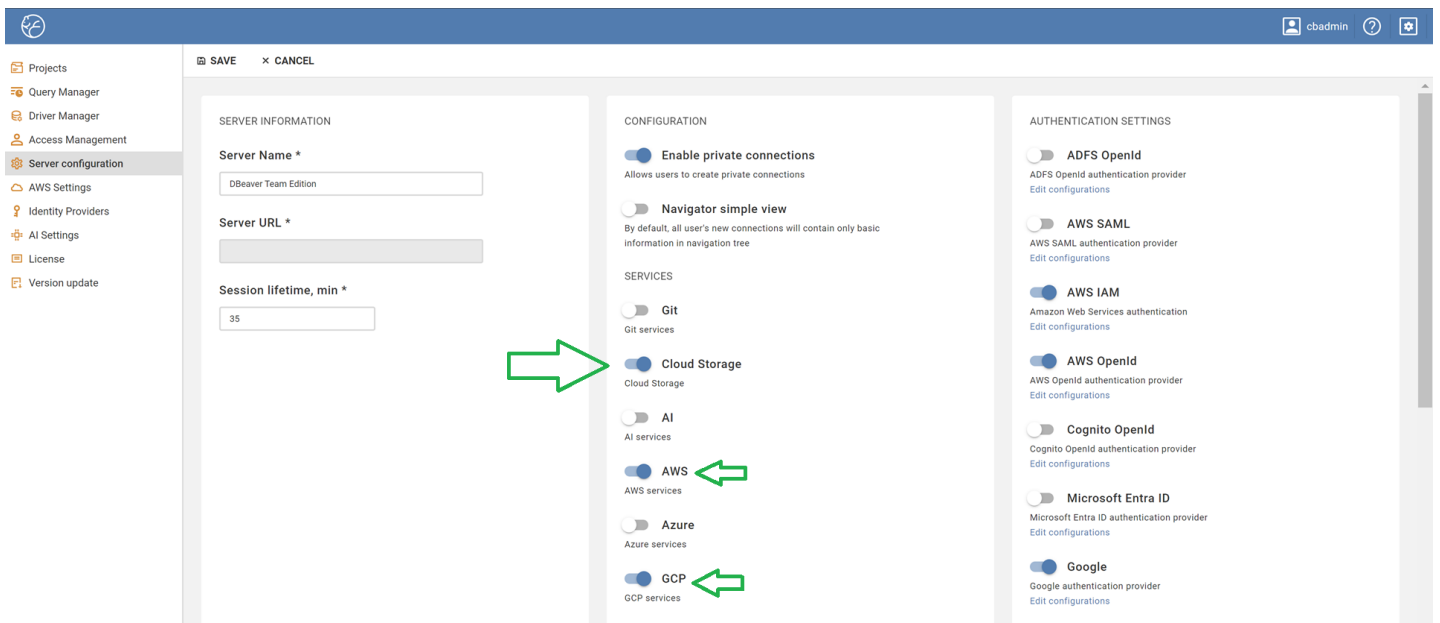
- upload new files to the bucket from your local storage;
- download files to your local storage;
- delete and rename files;
- drag-and-drop files between buckets and different file systems.



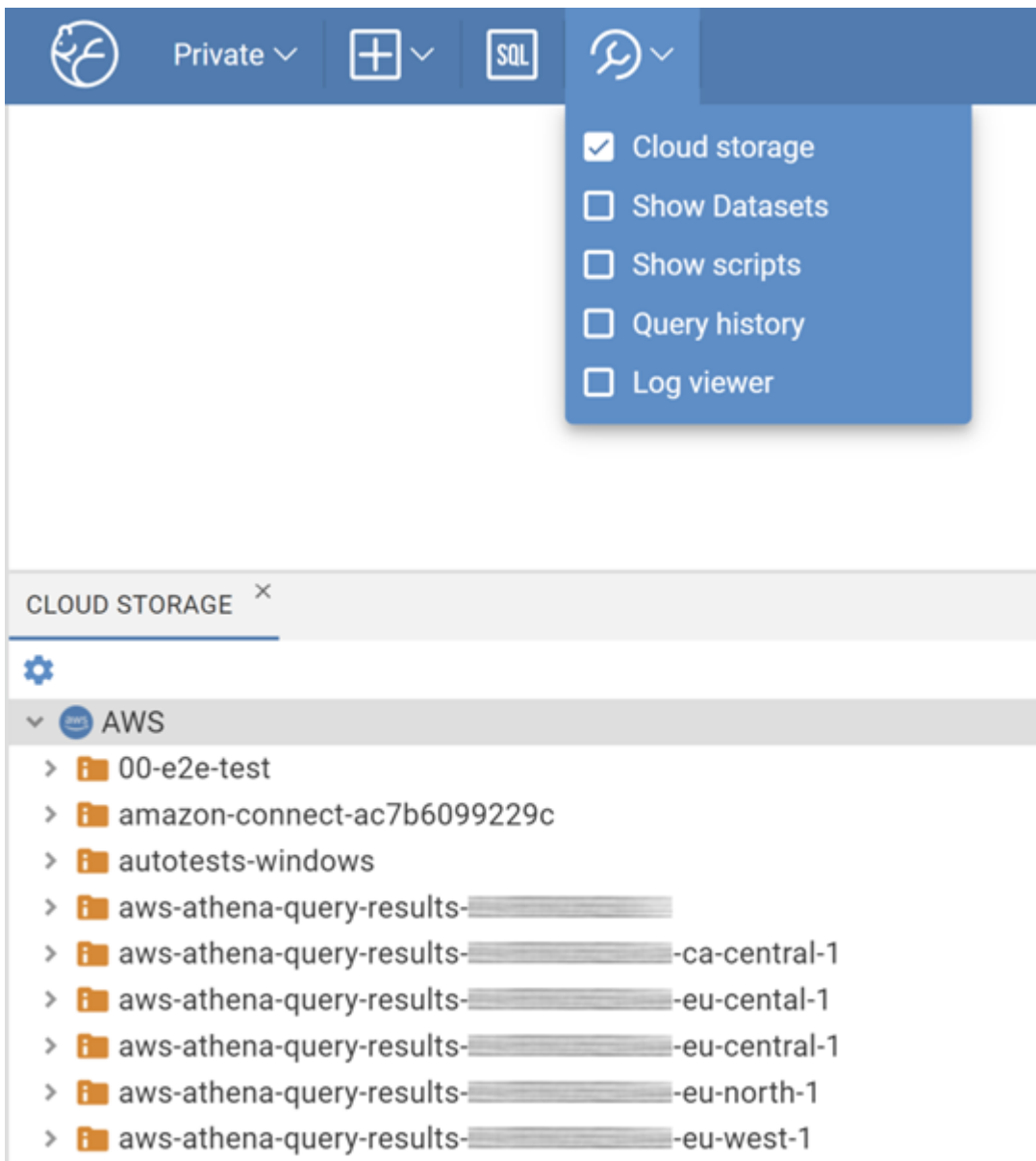
Configure the Cloud storage

To gain access to your storage, you should create a cloud configuration - [AWS](#) or [GCP](#).

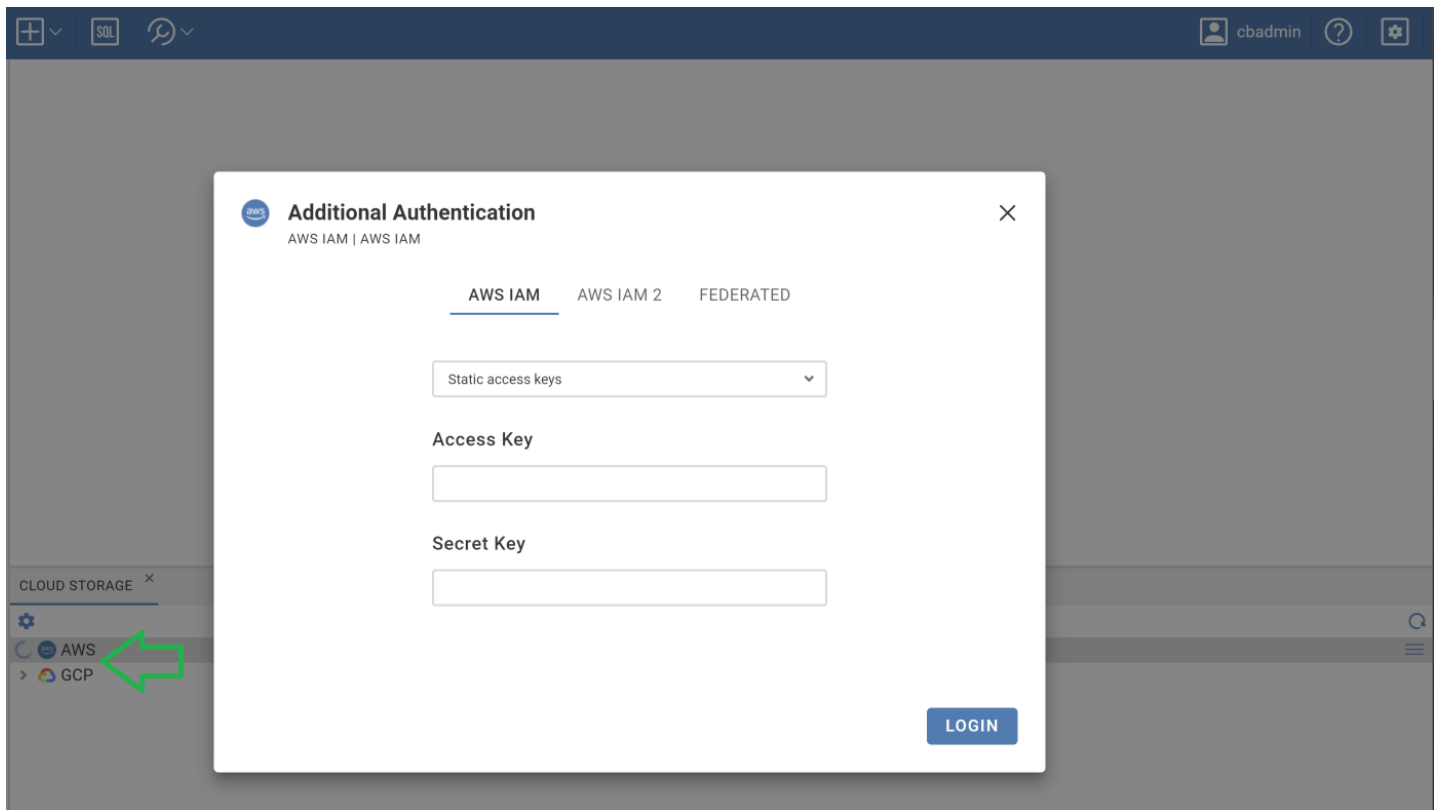
Then enable 'Cloud storage' in the Server configuration. Ensure that the relevant service (AWS, GCP) is also enabled.



Navigate to the Tools menu on the main page of the app. Check the 'Cloud storage' option.



And finally, to view your buckets, expand the file system tree. If you are not logged in under the required provider, a login window will appear. Use the preconfigured authentication.



Work with SQL scripts

The file type is recognized automatically. Only SQL scripts (.sql) could be opened in the app. Double-click on the file to open it in SQL editor. You can easily execute statements or make changes and save them directly to the cloud storage.

select users

1

```
select * from users u where age > 18;
```

CLOUD STORAGE



wroud

BOM.sql

select users 2023-11-24 16-57-43.sql

sql-1 (share) 2023-10-10.sql

sql-2 (PostgreSQL@localhost) 2023-12-07 22-26-12.sql

14dbeaver.pdf

Album_202312111259.csv

Artist_202312111851.csv

b5moW8Tsls.jpg

Customer_202309281250.csv

Learn more

Table of contents

[Server information](#)

[Configuration](#)

[Custom connections](#)

[Navigator simple view](#)

[Services](#)

[AWS](#)

[Authentication settings](#)

[Security](#)

[Save credentials](#)

[Save users credentials](#)

CloudBeaver offers different settings that allow configuring the server. The administrator can set the Server configuration settings when configuring the app for the first time, or it can be done later in the Administration Menu.

The screenshot displays the CloudBeaver Administration interface. The top navigation bar includes the CloudBeaver logo, a user profile icon for 'cbadmin', and a help icon. A sidebar on the left lists navigation options: Connection Management, Access Management, Server configuration (highlighted), Identity Providers, License, and Version update. The main content area is titled 'SAVE X CANCEL' and is divided into several sections:

- SERVER INFORMATION:** Contains three input fields: 'Server Name *' (value: Cloudbeaver EE Web Server), 'Server URL *' (value: http://localhost:3100), and 'Session lifetime *' (value: 30).
- CONFIGURATION:** Features three toggle switches: 'Enable custom connections' (checked), 'Navigator simple view' (unchecked), and 'AWS' (unchecked).
- SERVICES:** Includes a toggle for 'AWS' (unchecked).
- AUTHENTICATION SETTINGS:** Contains three sections: 'Allow anonymous access' (unchecked), 'Local' (checked), and 'AWS IAM' (unchecked). Below these are 'Openid' and 'SAML' sections, each with a toggle (unchecked) and a link to 'Edit configurations'.
- SECURITY:** Features two toggle switches: 'Save credentials' (checked) and 'Save users credentials' (checked).

Server information

Basic settings such as Server name and Session lifetime.

Configuration

Custom connections

Whether users can create connections by themselves or it can be done only from the Administration Menu.

Navigator simple view

Defines how the [Database navigator](#) structure will look like.

You can read more about Simple and Advanced mode [here](#).

Services

AWS

Enables AWS Services.

Authentication settings

Define different authentication methods.

You can read more about authentication methods [here](#).

Security

Save credentials

Allow saving credentials for the pre-configured database.

Save users credentials

Allow saving credentials for non-admin users.

Learn more

Table of contents

[Product config structure](#)

[Product config example](#)

[Shortcuts](#)

[Configuration file locations](#)

[Table of settings](#)

[Explanation](#)

Product config structure

example of packages:

`core-ui` (package name) -> `ui` (name in config)

`plugin-notifications` (package name) -> `notifications` (name in config)

```
{
  core: {
    [core package name]: {
      [property name]: [property value]
    },
    ...
  },
  plugin: {
    [plugin package name]: {
      [property name]: [property value]
    },
    ...
  }
}
```

Product config example

```

{
  // Global properties
  core: {
    authentication: {
      baseAuthProvider: 'local',
      primaryAuthProvider: 'local'
    },
    browser: {
      'cookies.disabled': false
    },
    theming: {
      defaultTheme: 'light'
    },
    localization: {
      defaultLanguage: 'en'
    },
    'navigation-tree': {
      childrenLimit: 500,
      editing: true,
      deleting: true
    }
  },
  plugin: {
    'sql-editor': {
      maxFileSize: 100
    },
    notifications: {
      notificationsPool: 5,
      maxPersistentAllow: 5
    },
    'data-spreadsheet': {
      hidden: false
    },
    'data-viewer': {
      disableEdit: false
    },
    'log-viewer': {
      refreshTimeout: 3000,
      maxLogRecords: 1000,
      logBatchSize: 2000,
      maxFailedRequests: 3
    },
    'data-export': {
      disabled: false
    },
    'erd-viewer': {
      maxColumnsToDisplay: 15000
    }
  }
}

```

Shortcuts

Config below is equivalent to example config for plugins: 'log-viewer', 'data-export', 'erd-viewer'

```

{
  ...
  'plugin.log-viewer.refreshTimeout': 3000,
  'plugin.log-viewer.maxLogRecords': 1000,
  'plugin.log-viewer.logBatchSize': 2000,
  'plugin.log-viewer.maxFailedRequests': 3,
  'plugin.data-export.disabled': false,
  'plugin.erd-viewer.maxColumnsToDisplay': false
}

```

Configuration file locations

`webapp/packages/product-default/src/config.json5` (webapp)

`conf/product.conf` (server)

`workspace/.data/.product.runtime.conf` (runtime, highest priority)

listed in priority order

Table of settings

Variable	Deprecated	Value	Description
<code>plugin.notifications.notificationsPool</code>	<code>core_events.notificationsPool</code>	5	Maximum notifications
<code>plugin.notifications.maxPersistentAllow</code>	<code>core_events.maxPersistentAllow</code>	5	Maximum persistent notifications
<code>core.browser.cookies.disabled</code>	<code>core.cookies.disabled</code>	false	Whether an app can use cookies or not

Explanation

For example, if you want to disable the data export functionality and increase refresh timeout for the [Log Viewer](#), you can do it this way.

1. Open or create `.product.runtime.conf`
2. Paste the following code

```
{
  plugin: {
    'log-viewer': {
      refreshTimeout: 7000
    },
    'data-export': {
      disabled: false
    }
  }
}
```

