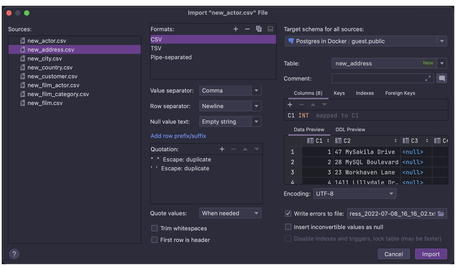
What's New in DataGrip 2022.2

DataGrip 2022.2 is here! This is the second major update of 2022, and it's packed with various enhancements. Let's take a look at what's inside!

Option to import multiple CSV files

We've enhanced the file importing process by implementing the ability to select several CSV files and import them at once!



Playground and *Script* resolve modes

In order to resolve database objects in SQL scripts, DataGrip needs to understand what the objects mean. This is necessary for the IDE to provide you with correct code completion and treat the usages of objects as links.

In general, the resolve context should match the running context, but that is not a trivial task for DataGrip to handle. This is why there were, until recently, several resolve-related problems. Here's what would sometimes happen:

- All of the objects were resolved, but the script could not be run in the database.
- When the *Auto-qualify* option was turned on, the objects would sometimes stay unqualified if they were located in the default database or schema.
- If there were multiple objects with the same name in different databases or schemas, existing columns would fail to be resolved and even the *Expand column list* action would sometimes generate the wrong list. At the same time, there was also the risk of resolving non-existing columns.

The cause of these issues was that, for each console or local file, DataGrip resolved their objects to the context selected in the top right-hand drop-down menu and the default database or schema (or the entry from the *SQL Resolution scopes* setting).

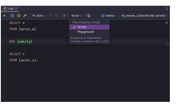
This was fine for working with scripts containing `USE` or `SET SEARCH PATH` statements, which switch the context. However, the IDE needs to know where to resolve to at the beginning of the file, too. So, it treated the default database or schema (or the entry from the *SQL Resolution scopes* setting) as the most appropriate context for the beginning of the file.

In other words, if there are no `USE` or `SET SEARCH PATH` statements in your file, then DataGrip shouldn't try to resolve the beginning of the file to somewhere out of context. To address this situation, we created a way for DataGrip to know which approach to follow by introducing two resolve modes: *Playground* and *Script*.

In *Playground* mode, DataGrip resolves all objects to the context, which is the value in the schema chooser, resolution scope, or default database. It works best if your file is just a set of unconnected queries, independent of each other and in no particular sequence. **Playground mode is now the default for query consoles**.

In *Script* mode, the beginning of the file is resolved to the context, but any `USE` statements in the script change the context for the resolve because they are part of the script's sequential logic. This is a good choice for when your queries have sequential logic and should be run as a single script. **Script mode is now the default for local files**.

Switching between the modes is easy. Just use the drop-down menu on the toolbar, as in the screenshot.

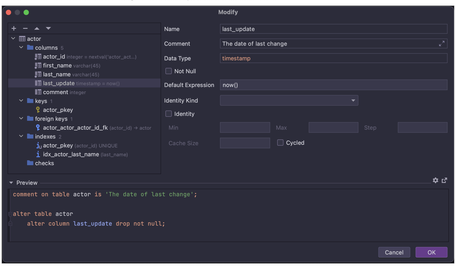


Code generation

New *Modify* UI

The new *Modify* UI, a basic version of which was introduced in the 2022.1 release, has become the default option. Starting with DataGrip 2022.2, all of a table's child objects can be added and edited using this new UI.

The old UI will remain available through this release cycle via the context menu.



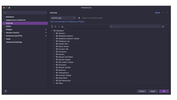
Support for collations and charsets SQL Server

Collations and charsets are now generated with the table's DDL.

Categories in the keymap

Previously, all of the actions related to database functionality were stored in four different places in the keymap. This structure, which has very little logic, was a historical consequence of the IDE's development.

To help you see all of the available actions, we've re-arranged the structure and placed all of the actions in groups under the Database parent group.



Scopes in DDL data sources

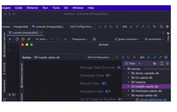
You can now set file scopes as sources for DDL data sources. This will let you easily filter folders for the DDL data sources. For example, you can exclude subfolders.

Miscellaneous

Merge All Project Windows action on macOS

For macOS users, we've introduced the ability to merge all open project windows into one, turning them into tabs. This action is available from the *Window* menu.

GIF animation.^



Password expiration warning Oracle, Vertica

Oracle and Vertica databases can provide warnings if the password is going to expire. *DataGrip* now retrieves this information and displays it after *Test Connection* is activated.

Basic support for *DuckDB*, *Mimer SQL*, and *Apache Ignite*

We've added 3 new databases to the basic support list: *DuckDB*, *Mimer SQL*, and *Apache Ignite*.

Data editor

Quality improvements

- DBE-10971: The table order is now saved in the data editor.
- DBE-7888: February is back in the date picker!
- DBE-15454: Binary columns with mixed content types are now editable.
- DBE-7804: CSV import no longer fails on the last line if the last value is empty and there is no new line.
- DBE-15335: The *SQL-Insert-Multirow* extractor generates the correct query.
- DBE-14980: When opening the *Quick Documentation* for cells, the related data is loaded automatically.
- DBE-15639: The incorrect placement of *Origin* and *Target* objects in the *Migration* window of the *DDL diff* preview has been fixed.
- DBE-15694: Operator family matching is now supported.
- DBE-15644: PostgreSQL *Fire* mode is now respected when dumping to a DDL datasource.
- DBE-8557: Azure SQL Database `SET ROWCOUNT` is executed before Azure queries to limit page size.

Query console

Quality improvements

- DBE-996: The search bar no longer hides the toolbar.
- DBE-11616: ClickHouse `MODIFY ORDER BY` is supported for `ALTER TABLE`.
- DBE-14643: SQL Server The *condition is always false* warning no longer appears during variable assignment.
- DBE-12232: SQL Server `CHANGETABLE` is now supported.
- DBE-13312: SQL Server `END CONVERSATION` is now supported.
- DBE-12435: SQLite The `PRAGMA_TABLE_INFO` and `PRAGMA_TABLE_XINFO` tables are resolved correctly.
- DBE-14997: PostgreSQL Columns are now correctly resolved when using `CTE` with `INSERT INTO`.

Download

Free for non-commercial use