Multi DB Manager – Query Studio

Database Manager: A Complete Guide

Database Manager is the console inside SQL Planner for browsing, querying, and analysing your databases — across SQL Server, MySQL, PostgreSQL, and Oracle, all from one place. This guide walks through everything it does, step by step.

SQL Server MySQL PostgreSQL Oracle

01 Opening Database Manager
02 Connecting to a database
03 Browsing your databases
04 The workspace at a glance
05 Writing & running queries
06 Working with results
07 Browsing table data
08 Reading an object's script
09 Relationship diagrams
10 Object lineage
11 Execution plans
12 Query history
13 Keyboard shortcuts
14 Tips & troubleshooting

01 Opening Database Manager

Database Manager lives in the main SQL Planner navigation. Open it from the top menu — you'll land on a two-panel layout: your saved **Connections** on the left, and the workspace tabs (Query, Data, Script, and more) on the right. If you're already monitoring a server in SQL Planner, its databases are only a couple of clicks away here.

02 Connecting to a database

Everything starts with a connection. Each connection you save belongs to your user account, and its password is stored encrypted.

Add a connection

1 In the **Connections** panel, click the green **+** button to open **New Connection**.

2. Choose the **technology** — SQL Server, MySQL, PostgreSQL, or Oracle. The form adapts to that engine.
3. Fill in the **server / host** (and **port** where relevant), your **username** and **password**, and an optional **default database**.
4. Pick a **colour tag** if you like (see below), turn on **Encrypt connection** if your server requires an encrypted transport, and save.

Your new connection appears in the tree on the left, ready to expand.

Colour-code for safety

A colour tag is the simplest guard against running the right query on the wrong server. Tag environments however you like — a common convention is:

- 🔴 **Red** for production — the server where mistakes are expensive.
- 🟠 **Amber** for staging or shared test environments.
- 🟢 **Green** for local or development.

The colour shows as a stripe on the connection in the tree, and when that connection is active a matching **bar appears above the editor** naming the server and database — so you always know where a query is about to run.

> TIP
> Tag production first. The one second it takes pays for itself the first time it stops you.

Managing connections

Right-click or use the connection's controls to edit or remove it. Removing a connection only deletes the saved entry in SQL Planner — it never touches the database itself.

↑ back to top

---

03 Browsing your databases

The left panel is a live tree of your servers. Expand a server to see its **databases**; expand a database to see its **Tables, Views, Stored Procedures, Functions, Triggers, Synonyms,** and **Security**. The structure looks the same on every engine, so you never have to relearn where things live.

- **Select a database** — click (or double-click) it to make it active without expanding. The editor connects to it and is ready for a query.
- **Search the tree** — use the search box above the tree to filter to a table or object by name when the list is long.
- **Select an object** — click a table, view, or procedure to target it for the Data, Script, or Lineage tabs.

↑ back to top

04 The workspace at a glance

The right side is organised into tabs. Here's what each one is for — the rest of this guide covers them in detail.

TAB	WHAT IT'S FOR
Query	Write and run SQL, with autocomplete, snippets, formatting, and execution plans.
Data	Browse the rows of a selected table without writing a query.
Script	Read the definition (the CREATE script) of any object.
Diagram	See how tables relate, as a primary/foreign-key map.
Lineage	Trace what an object depends on, and what depends on it.
History	Find and re-open queries you've run before.

↑ back to top

05 Writing & running queries

The **Query** tab is a focused SQL editor with syntax highlighting and a few things that make everyday work faster.

Autocomplete that understands your query

As you type, suggestions appear automatically. Autocomplete reads the tables already in your statement, so it offers the right things in the right place:

- After `FROM` or `JOIN`, it suggests **tables and views**.
- In a `WHERE`, `SELECT`, `ON`, or `GROUP BY` clause, it suggests the **columns of the tables in**

your query — with each column's **data type** shown beside it.

- Type a **table alias** and a dot (for example `c.`) and it lists that table's columns.
- It also completes SQL **keywords and built-in functions**.

Press `Ctrl`+`Space` any time to bring suggestions up on demand.

Snippets

Open the **Snippets** menu for ready-made templates — a `SELECT` skeleton, a join, an insert, and more. Snippets adapt to the engine you're on (for example, a row-limited select uses `TOP` on SQL Server, `LIMIT` on MySQL and PostgreSQL, and `FETCH FIRST` on Oracle), so you don't have to remember each dialect.

Format

Click **Format** to tidy messy SQL — consistent indentation and line breaks — without changing what the query does.

Run, limit, and cancel

- **Run** your query with the **Run (F5)** button, or press `F5` / `Ctrl`+`Enter`.
- **Set a row limit** before running so a broad `SELECT *` never floods the grid. You can raise it when you genuinely need more rows.
- **Cancel** a long-running query with the **Cancel** button if you change your mind.

Query statistics

After each run, the status line reports how long the query took, how many rows came back, and how many result sets it produced — a quick read on cost without any extra work.

↑ back to top

---

06 Working with results

Every statement that returns rows gets its own result grid. If a query returns several result sets, each one lands in its own grid.

- **Sort** — click a column header to sort by it.
- **Filter** — type in the **Filter rows...** box to search across the whole grid, or use the **per-column filters** to narrow specific columns. Column filters combine together, then with the

global filter.

- **Chart** — turn any result into a **bar, line, or pie chart** to read the shape of the data at a glance.
- **Insights** — open the built-in insights for quick highlights about a result set without writing more queries.
- **Export & copy** — export to **CSV, JSON,** or **Markdown**, or copy the grid (including as Markdown) straight to your clipboard.

> GOOD TO KNOW
> Filtering and sorting act on the rows already returned to the grid (up to your row limit). To work over more data, raise the row limit or narrow the query itself.

↑ back to top

07 Browsing table data

To look at a table's contents without writing SQL, select the table in the tree and open the **Data** tab. It loads the rows into a grid with the same sorting, filtering, charting, and export tools as query results — handy for a quick look at what's in a table.

↑ back to top

08 Reading an object's script

Select any object — a table, view, stored procedure, or function — and open the **Script** tab to read its definition with syntax highlighting. From there you can:

- **Download** the script as a `.sql` file.
- **Copy** it to your clipboard.
- Toggle **word wrap** for long lines.

It's the fastest way to see exactly how something is built before you change or depend on it.

↑ back to top

09 Relationship diagrams

The **Diagram** tab draws your tables as a map of primary and foreign keys, so you can see how a database fits together.

- **Drag** any table box to arrange the layout the way you think about it.
- **Auto-arrange** to let Database Manager group connected tables for you; unrelated tables sit below a divider.
- **Hover a relationship line** to see exactly which columns link the two tables.
- **Show or hide tables** from the panel to focus on the part of the schema you care about.
- **Toggle data types** to show each column's type and length on the boxes.
- **Export PNG** to save the diagram as an image for docs or a review.

↑ back to top

10 Object lineage

Before you change an object, it helps to know what touches it. Select an object and open the **Lineage** tab to trace its dependencies — what it relies on, and what relies on it. It's a quick way to gauge the blast radius of a change.

↑ back to top

11 Execution plans

Want to know how the database will run a query before you commit to it? Write the query in the **Query** tab and click **Explain**. Database Manager asks the engine for its **estimated execution plan** — without running the statement — and shows it below the editor.

This works on all four engines. You'll see the plan in the form each engine produces:

ENGINE	WHAT YOU GET
SQL Server	The estimated showplan, as a grid of operations and costs.
MySQL	The `EXPLAIN` output, as a grid.
PostgreSQL	The `EXPLAIN` plan, as readable text.
Oracle	The explain-plan output, as readable text.

Use it to catch an expensive scan or a missing index early — while it's still cheap to fix.

↑ back to top

12 Query history

Every query you run is saved to the **History** tab, tied to your account. Open it to find that statement you ran earlier, copy it back into the editor, and run it again. You can clear your history whenever you want a clean slate.

↑ back to top

13 Keyboard shortcuts

SHORTCUT	ACTION
`F5`	Run the query
`Ctrl` + `Enter` / `⌘` + `Enter`	Run the query
`Ctrl` + `Space`	Show autocomplete suggestions

↑ back to top

14 Tips & troubleshooting

Autocomplete isn't suggesting columns

Autocomplete needs to know the database's schema first. Select the database in the tree (a single click is enough) and give it a moment to load — then column suggestions will appear inside your query. On a database with very many tables, the first load can take a little longer.

A connection won't open

Double-check the host, port, username, and password, and confirm the database server allows connections from where SQL Planner runs. If your server requires an encrypted transport, turn on **Encrypt connection** in the connection settings.

My result grid looks short

Results are capped by the row limit so a broad query can't overwhelm the page. Raise the limit before running, or add a `WHERE` clause to focus the query.

Where did my query go?

Open the **History** tab — every run is saved there and can be re-opened in the editor.

> HABIT WORTH KEEPING
>
> Colour-code production red and watch for the connection bar above the editor before you run anything that changes data. It’s the cheapest insurance in the tool.

↑ back to top