Database Schema Compare Guide

SQL PLANNER · USER GUIDE

Using Schema Compare

A step-by-step guide to comparing two databases, reading the differences, and deploying a synchronization script in your database's native SQL.

ON THIS PAGE

Before you start · Open Schema Compare · Run a comparison · Connecting each database · Read the results · Generate & deploy · Troubleshooting · FAQ

Before you start

You'll need a few things ready on both the source and the target:

- A login for each server with permission to read the schema (catalog views). To deploy changes, the login also needs permission to create or alter objects.
- Network access from the SQL Planner machine to each database server and its port.
- Both sides on the **same platform** — SQL Server with SQL Server, MySQL with MySQL, and so on.

Good habit: compare into a copy or a lower environment first, and always keep a recent backup of the target before you deploy anything.

Open Schema Compare

From the top menu, open **Menu** → **SQL Tools** → **Compare Database Schema**. The tool opens on a short wizard that walks you from picking a platform to reviewing a script.

Run a comparison

1 **Choose your database technology.** Pick SQL Server, MySQL, PostgreSQL, or Oracle. This sets the dialect for the whole session, so both sides use it.

2 **Connect the source.** Enter the server, sign in, and click **Test Connection**. When it succeeds, choose the database to read from.

3 **Connect the target.** Repeat for the database you want to bring in line with the source.

4 **Compare.** The tool reads both catalogs and lists every object that is added, missing, or different between the two.

5 **Review & deploy.** Open the side-by-side diff, generate the native script, and run it when you're satisfied.

Connecting each database

The connection screen is the same idea everywhere — a server, a login, and a database list — but each platform has a couple of specifics worth knowing.

> **Server and port.** If your server uses a non-default port, you can put the port in its own box or type it directly in the server box as `host:port`. Both are accepted.

SQL Server

Server: `HOST\INSTANCE` or `host,1433`

Auth: SQL login (user + password), or Windows where available.

Identifiers are wrapped in `[brackets]` in generated scripts.

MySQL

Server: `host` with **Port** `3306` (or `host:3306`).

Cloud / managed: use the endpoint host and open the firewall / security group to your machine.

The tool negotiates a secure connection automatically and falls back for local servers. Identifiers use `` `back-ticks` ``.

PostgreSQL

Server: `host` with **Port** `5432`.

Auth: the server's host-based rules (`pg_hba.conf`) must allow password sign-in from your address. See troubleshooting if you get an authentication error.

Identifiers use `"double-quotes"`; sequences are scripted natively.

Oracle

Server: host, port `1521`, and a **service name**.

Schema: choose the schema to read; object names are typically uppercase.

Identifiers use `"double-quotes"`; statements end with `/` and synonyms are included.

Read the results

After the comparison runs, the differences appear as a list of objects. Selecting an object opens the **side-by-side diff** docked at the bottom of the screen:

- **Source on the left, target on the right.** Each side shows the object as a real `CREATE` statement in the platform's own SQL, lined up so you can spot exactly what differs.
- **Added, missing, or changed.** Objects that exist only on one side, or whose definitions differ, are called out so you can decide what to do with each.
- The panel slides up from the bottom and can be resized, so you can widen it while you read and collapse it when you're done.

> **Reading tip:** differences in columns include the things that usually bite — data type, length, nullability, and identity / auto-increment — so check those first when a table looks "almost the same."

Generate & deploy the script

When you're happy with what the comparison found, generate the synchronization script. It is written entirely in the target platform's native SQL, ready to run.

1. **Generate** the script to bring the target in line with the source.
2. **Read it.** The script is yours to inspect — nothing has changed on the target yet.
3. **Run it** when you're ready, either from the tool or by copying it into your own SQL client.

> **Before you deploy:** back up the target, and prefer running against a test copy first. Generated changes can drop or alter objects to match the source, so review the script in full — especially any statement that removes something — before you run it on a production database.

Troubleshooting

Most problems are connection-related. Here are the messages you're most likely to meet and what to do about them.

If you see	What it means	Do this
Unable to connect to any of the specified MySQL hosts	The host or port couldn't be reached.	Check the host spelling and that the port is `3306`. Put the port in the Port box, or type `host:3306` in the server box. Confirm the firewall / security group allows your machine.

If you see	What it means	Do this
MySQL connection rejected over TLS (often on cloud / managed MySQL)	The server requires a secure connection your client must negotiate.	Use the server's endpoint host and standard port, and make sure inbound access is open. The tool selects a secure mode automatically; for a local server it falls back on its own.
PostgreSQL: password authentication failed *(SQLSTATE 28000)*	The server's host-based rules didn't apply password sign-in to your connection.	In `pg_hba.conf`, make sure a `scram-sha-256` (or `md5`) rule matches your client address and isn't shadowed by an earlier rule. Set the role's password, then reload the server config. (Other clients fail the same way until this is fixed.)
Login failed / insufficient permission	The login can sign in but can't read the catalog or change objects.	Grant the login read access to the schema for comparing, and create/alter rights if you intend to deploy.
Oracle: table or schema looks empty	Reading the wrong schema, or a name-case mismatch.	Pick the correct **schema** on the connection screen. Remember Oracle names are usually uppercase.
Comparison runs but a table shows no columns	The chosen database or schema doesn't contain that object.	Confirm both sides point at the database/schema you expect, then run the comparison again.

FAQ

Can I compare a SQL Server database to a MySQL database?

No — a comparison runs within one platform on both sides. That's deliberate: it's what keeps the generated script perfectly native to your database instead of a generic approximation.

Does running the comparison change anything?

No. Comparing only reads the two schemas. Changes happen only when you choose to run the synchronization script.

Which direction does the script go?

The script brings the **target** in line with the **source**. Decide which database is which before you generate it.

What's actually compared?

Tables and columns (type, length, nullability, identity/auto-increment, defaults), primary and foreign keys, indexes, check constraints, views, stored procedures, functions, sequences, and — on Oracle — synonyms.

SQL Planner — Schema Compare user guide