

Heidi

A MySQL Client

Install

Heidi is a Windows-only application.

1. Installing Heidi is a simple process. Download the installer from here:
<https://www.heidisql.com/download.php>
2. Follow the installation instructions. You can keep all the defaults in the main options.

Guide - Connect to Localhost

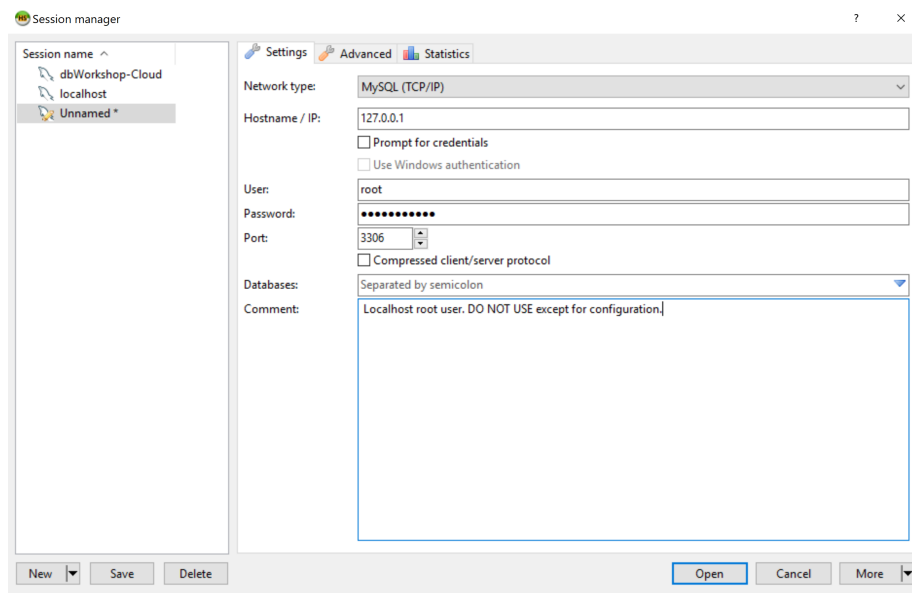
When you first run Heidi, you are presented with a Session Manager window. This is where you configure connections to different SQL Servers. You need to configure at least one connection to start using Heidi. You can add others later to connect to different servers, or the same server as different users.

Part 1: Connect to localhost as root

We recommend not using the root user after the initial setup, but it is useful to use first to get your other user accounts setup.

1. Connect to Localhost

To connect to your localhost MySQL Server, open Heidi, and use the following settings to configure your session:



- Network type: determines the type of server you're connecting to, and the method of connection. Use TCP/IP for localhost.
- Hostname: the location of the host. 127.0.0.1 is a specific IP that refers to your local machine.
- User + Password: The user account credentials for the MySQL user account.
- Port: Most MySQL servers run on port 3306. You only need to change this if your server admin tells you.
- Databases: Leave this blank to see all the databases your user account can connect to.
- Comment: Add a comment here to remind yourself what this connection is for.

Save and open the connection.

2. Create a database

When you connect, you will see the main view in Heidi. The server and available objects are listed in the left panel. You will see some databases already exist:

- information_schema
- mysql
- performance_schema
- sys

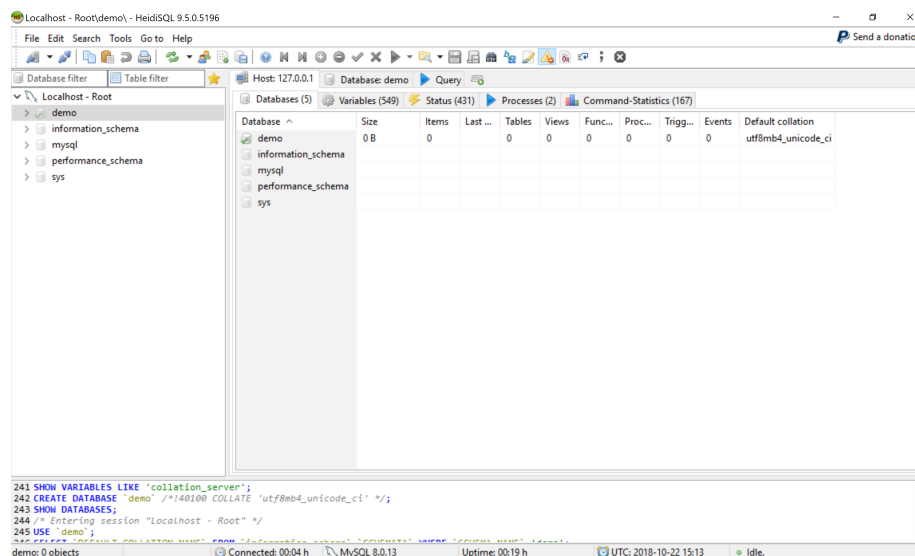
These are config databases, and should never be manually edited.

You can create any number of databases in your server. To create a new database, right-click on the connection name and choose create new → database.

- Give the database a name.
- Use the collation utf8mb4_unicode_ci. (This collation uses utf-8 to allow for accented and non-standard latin characters. utf8mb4 also extends it to allow storing of emoji characters - probably not useful in many situations, but it doesn't appear to do any harm compared to using utf8_unicode_ci)
- You will see the database added to the list of available databases in the left panel.

NOTE: When you perform any action in Heidi, it will show you the actual SQL command being run in the bottom panel. This can be useful, if you want to know how to do something via a script, you can do it in a menu and look at the SQL output.

For example, look at the code used to create the database:



2. User account management:

As mentioned, using root user for day-to-day work is not recommended. One reason for this is that root has access to the config databases (information_schema, etc), and this clutters the work environment. So, next we will create a new user account, and give them access only to the new database.

To see and edit user accounts on the current server, go to the “User manager” screen. (In the menu: Tools→User manager, or with the icon shown)

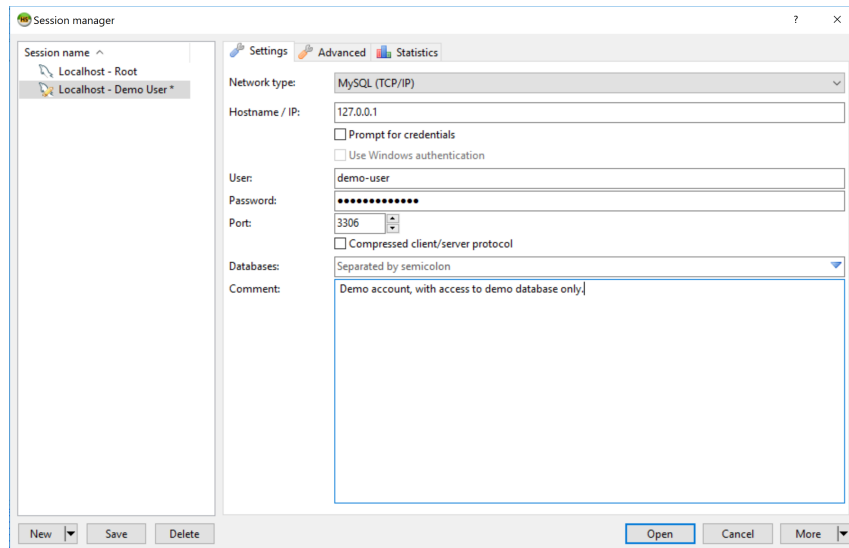
This screen shows all the user accounts on the current server. If you created a second account during server setup, it will appear here.

- To create a new user account, click Add. Give the user a user name, and assign a password. Keep the “From host” as localhost.
- Look at the Global privileges list. They’re split into 3 colours:
 - Green privileges are read actions, like SELECT, SHOW databases and EXECUTE pre-defined functions.
 - Red privileges are write actions. These are potentially destructive, so be careful when assigning these. We recommend that only the root user has these assigned at the global level.
 - Blue privileges are admin actions, like creating users, granting privileges to other users and performing actions on the server like Shutdown and setting up replication.
- We don’t want this user to have any global permissions, so leave them all blank.
- Click “Add object”, then select your new database from the list of server objects. This lets you assign privileges specific to this database. Click Ok.
- Give the user all permissions on this new database.
 - Notice that many actions in the global list are not available at the database list. This is because many actions (like create user) are only applicable at the server level.
- Click save
- If you are interested, you can also see the code used to create that user:

```
255 CREATE USER 'demo-user'@'localhost' IDENTIFIED BY 'alexonflavour';
256 GRANT USAGE ON *.* TO 'demo-user'@'localhost';
257 GRANT SELECT, EXECUTE, SHOW VIEW, ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, EVENT, INDEX, INSERT, REFE
258 FLUSH PRIVILEGES;
259 SHOW GRANTS FOR 'demo-user'@'localhost';
```

Connected: 00:11 h MySQL 8.0.13 Uptime: 00:26 h UTC: 2018-10-22 15:20 Idle

- Right-click on the connection name in the left panel, and choose “disconnect”.
- Now, we will setup a new connection using our new user. Create a new session, and use the username and password for your new user:



- Save and open this new connection. You will see your new database on the left. You will also see the `information_schema` database, which is visible to everyone, but read-only to anyone without root access.

You now have a new, empty database, and a user account with full control over this database, but without any access to edit the main server configuration.