

# **MySQL**

Now that you know what a database, table, and column are, the next thing that you would need to do is install a database service where you would be running your SQL queries on.

We will be using MySQL as it is free, open-source, and very widely used.

## **Installing MySQL**

Depending on your operating system, to install MySQL run the following commands.

### **Install MySQL on Ubuntu**

To install MySQL on a Linux or Ubuntu machine, run the following commands:

- First update your apt repository:

```
sudo apt update -y
```

- Then install MySQL:

```
sudo apt install mysql-server mysql-client
```

We are installing two packages, one is the actual MySQL server, and the other is the MySQL client, which would allow us to connect to the MySQL server and run our queries.

To check if MySQL is running, run the following command:

```
sudo systemctl status mysql.service
```

To secure your MySQL server, you could run the following command:

```
sudo mysql_secure_installation
```

Then follow the prompt and choose a secure password and save it in a secure place like a password manager.

With that, you would have MySQL installed on your Ubuntu server. The above should also work just fine on Debian.

### **Install MySQL on Mac**

I would recommend installing MySQL using [Homebrew](#) ():

```
brew install mysql
```

After that, start MySQL:

```
brew services start mysql
```

And finally, secure it:

```
mysql_secure_installation
```

In case that you ever need to stop the MySQL service, you could do so with the following command:

```
brew services stop mysql
```

## Install MySQL on Windows

To install MySQL on Windows, I would recommend following the steps from the official documentation here:

<https://dev.mysql.com/doc/refman/8.0/en/windows-installation.html> (<https://dev.mysql.com/doc/refman/8.0/en/windows-installation.html>)

## Accessing MySQL via CLI

To access MySQL run the `mysql` command followed by your user:

```
mysql -u root -p
```

## Creating a database

After that, switch to the `demo` database that we created in the previous chapter:

```
USE demo;
```

To exit the just type the following:

```
exit;
```

## Configuring .my.cnf

By configuring the `~/.my.cnf` file in your user's home directory, MySQL would allow you to log in without prompting you for a password.

To make that change, what you need to do is first create a `.my.cnf` file in your user's home directory:

```
touch ~/.my.cnf
```

After that, set secure permissions so that other regular users could not read the file:

```
chmod 600 ~/.my.cnf
```

Then using your favourite text editor, open the file:

```
nano ~/.my.cnf
```

And add the following configuration:

```
[client]
user=YOUR_MYSQL_USERNAME
password=YOUR_MYSQL_PASSWORD
```

Make sure to update your MySQL credentials accordingly, then save the file and exit.

After that, if you run just `mysql`, you will be authenticated directly with the credentials that you've specified in the `~/.my.cnf` file without being prompted for a password.

## The `mysqladmin` command

As a quick test, you could check all of your open SQL connections by running the following command:

```
mysqladmin proc
```

The `mysqladmin` tool would also use the client details from the `~/.my.cnf` file, and it would list your current MySQL process list.

Another cool thing that you could try doing is combining this with the `watch` command and kind of monitor your MySQL connections in almost real-time:

```
watch -n1 mysqladmin proc
```

To stop the `watch` command, just hit `CTRL+C`

## GUI clients

If you prefer using GUI clients, you could take a look at the following ones and install them locally on your laptop:

- [MySQL Workbench](https://www.mysql.com/products/workbench/) (<https://www.mysql.com/products/workbench/>)
- [Sequel Pro](https://www.sequelpro.com/) (<https://www.sequelpro.com/>)
- [TablePlus](https://tableplus.com/) (<https://tableplus.com/>)

This will allow you to connect to your database via a graphical interface rather than the `mysql` command-line tool.

If you want to have a production-ready MySQL database, I would recommend giving DigitalOcean a try:

[Worry-free managed database hosting](https://www.digitalocean.com/products/managed-databases/) (<https://www.digitalocean.com/products/managed-databases/>)