**MIT xPRO Data Engineering Certificate**

**MySQL Connector**

**MySQL Python Connector**

pip install mysql-connector-python

To check package has been loaded, run a python file ‘check.py’ with the following contents:

import mysql.connector

And check for errors in the terminal.

**MySQL Shell for Windows**

Download MySQL Shell

\? for help

\sq to switch to SQL mode

\c root@localhost:3306 to connect to MySQL running on local machine

Now, enter your sql script

There is also a pre-configured command-line client called “MySQL 8.0 Command Line Client”

**Pandas as an in-memory database**

See app.py This connects to the MySQL Server using a python connector and then reads an SQL query to a pandas dataframe.

**What Is a *Server*? What Is a *Client*?**

Within a network, a computer machine that shares resources, such as memory, hard drives, and printers, with other computers is called a *server*. *Servers* also provide access to data or particular services, such as large-scale computations. Within the same networks, any other computer that is not a *server* is called a *client*. *Clients* are generally capable of receiving information or using a particular service from the *server*.

Therefore, *servers* are usually more powerful and expensive machines because they share their resources with other computers in the network. *Clients* are less powerful machines, and you can think of them as personal computers that users operate to connect to a *server*. The *clients*’ machines don’t need to be extremely powerful because, generally, they don’t need to share resources with other computers.

**What Is a *Client*-*Server* Interface?**

As you know, a user can perform data analysis on a database using SQL locally. This means that the database is stored on the user machine and all the operations happen on the user’s computer. However, SQL works particularly well on a *client-server* system. In this case, users on different *clients’* machines can access a database that is stored on a *server* machine via wired communications (i.e., LAN) or online channels.

In this specific case, the computer you are using to manipulate a database and where you have installed MySQL Workbench is the *client* and, so far, you have only performed operations locally.

In the last video, you learned how to extend the applications of SQL by using a different type of *client* by using a Python *driver* that acts on the database.

**What Is a *Driver*?**

Let’s try to understand what a *driver* is with an example.

You have probably heard of the word *driver* when installing a printer on your machine. Imagine you are working in an office where multiple printers are connected to a network and you want to print a certain document from a specific printer.

It would be convenient to simply say "print this document" but, of course, because each printer is unique, the signal cannot be exactly the same for all the printers in your office. In fact, as you probably know, if you don’t specify which printer you want to connect to, you won’t be able to print anything.

Here’s where *drivers* become useful. Rather than having to send a different signal to each type of printer (and hoping that one works for you), it is more convenient to locally install a *driver* for a particular printer that translates the "print this document" command into something more specific that the desired printer hardware needs to understand.

In other words, a *driver* is a piece of software related to hardware that contains a mapping/translation so that an operating system can communicate with hardware. It's a piece of code that sits between a program and hardware.

In conclusion, both *servers* and *clients* can use *drivers* for communication whenever it is necessary.