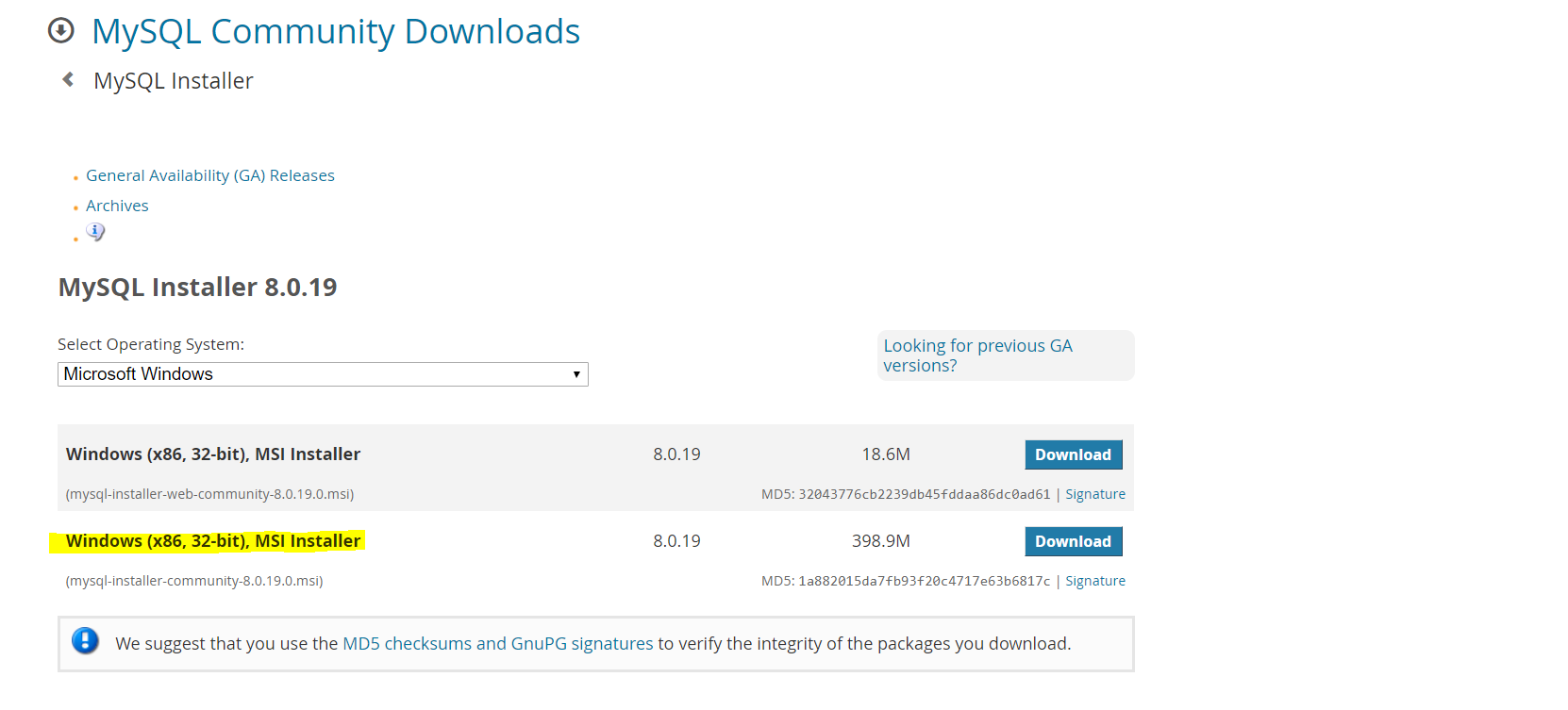
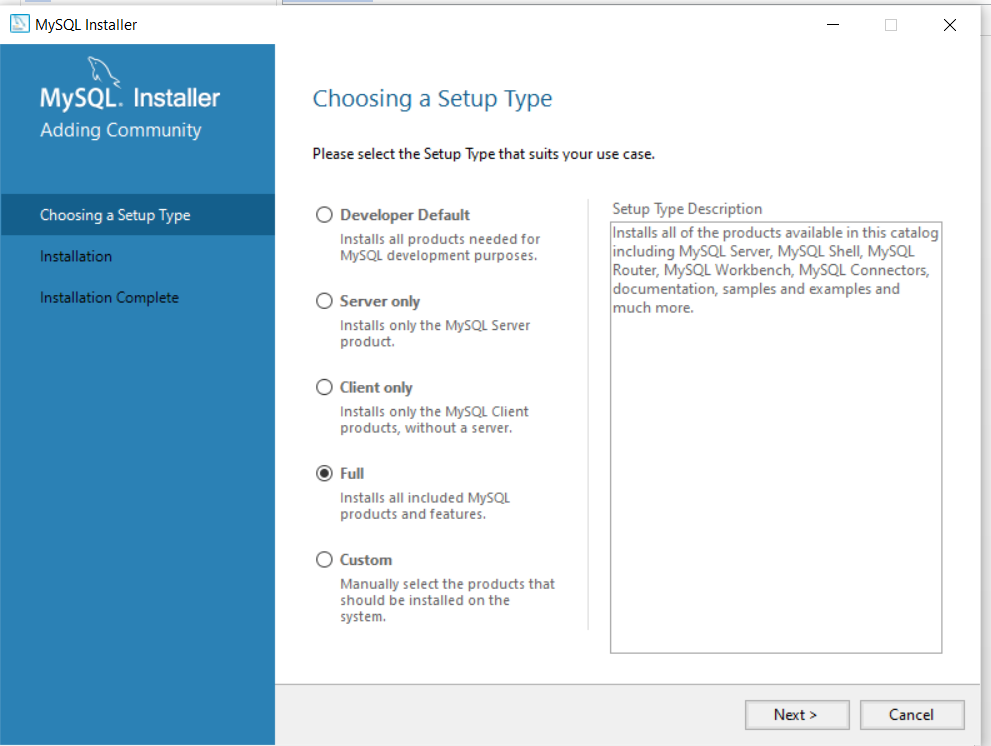
# **Working with MySQL and Python**

**Installing MySQL Workbench**

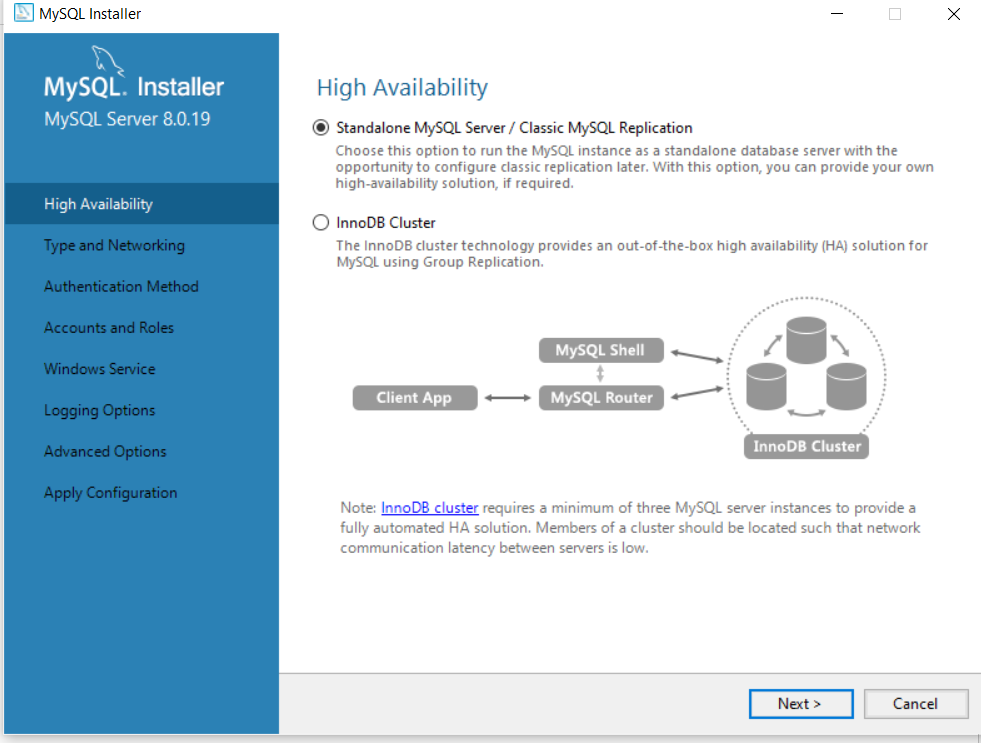
* Download MYSQL application. Go to page : <https://dev.mysql.com/downloads/installer/>
* Click on the link highlighted below.



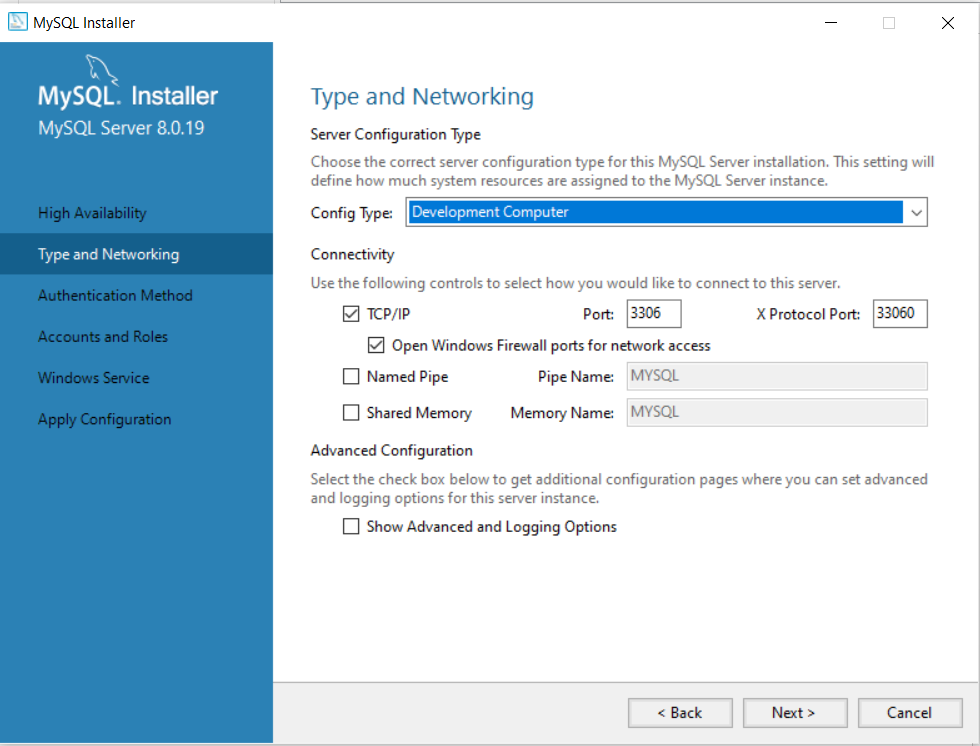
* Once downloaded, install the application. Follow the instructions and continue with installation.

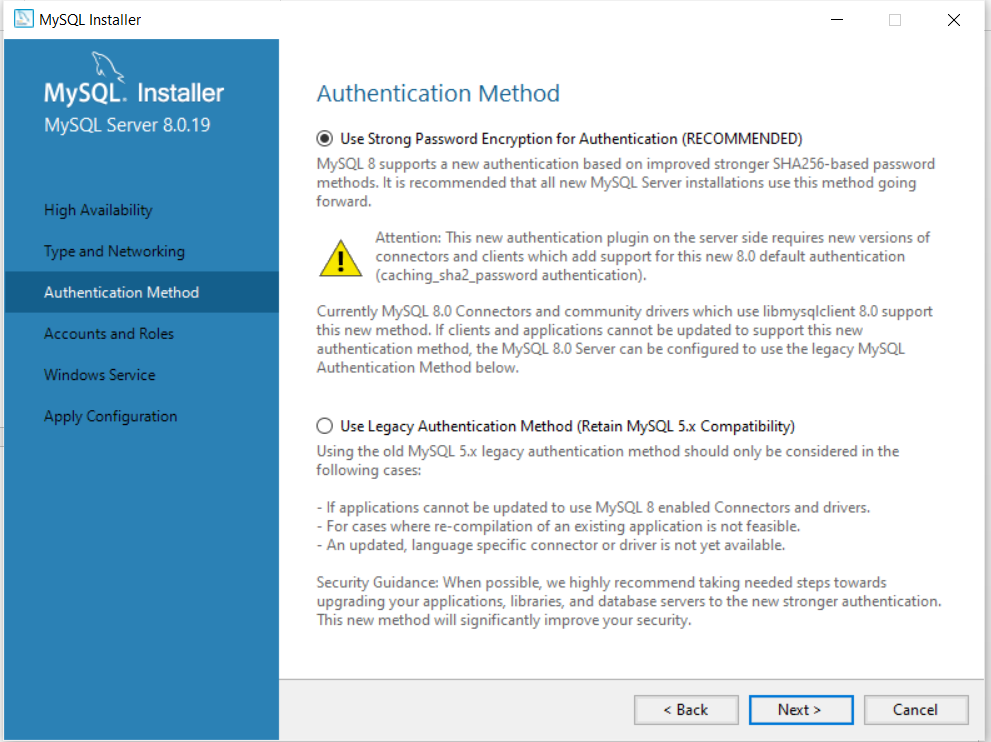


* Select Standalone MySQL server option

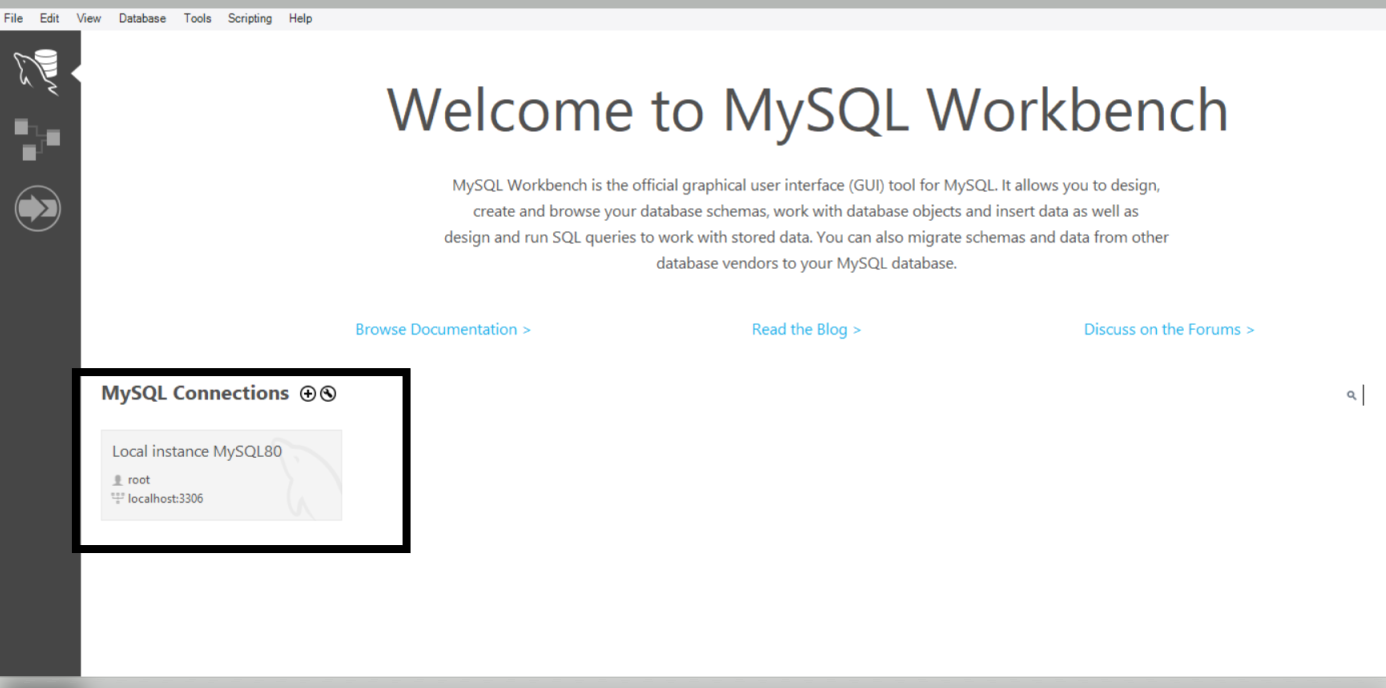


* Keep the default values for “Type and Networking” section

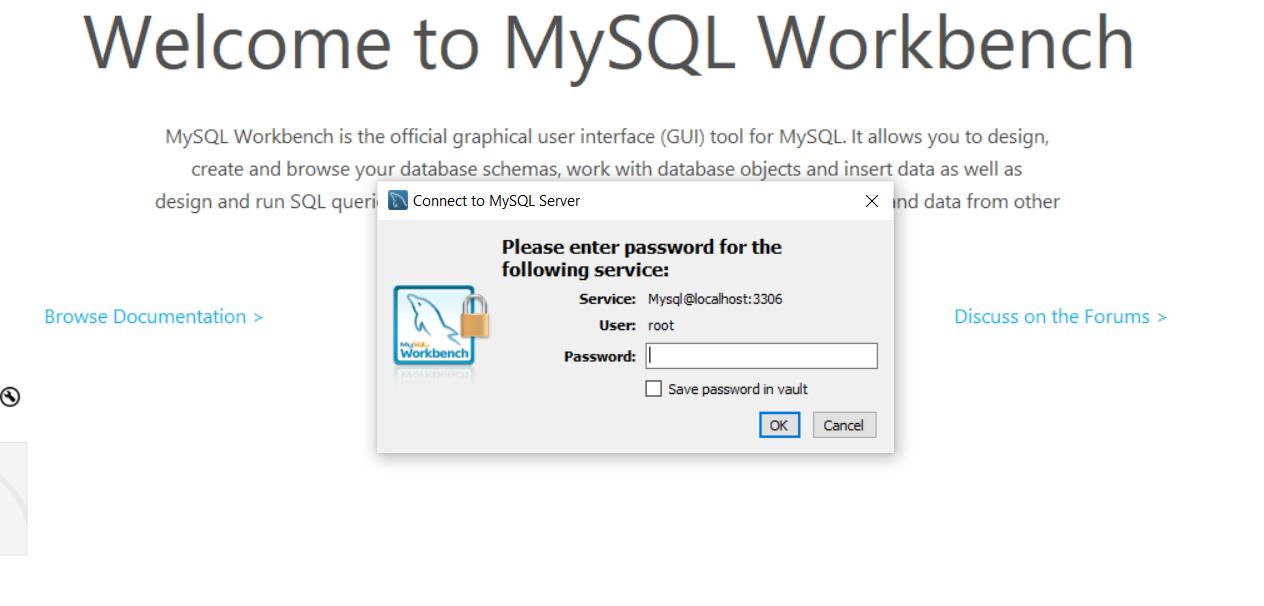




* Finish the installation by pressing next.
* Once the installation is complete, open the MYSQL workbench and click on the local instance connection. The window will look something like this:

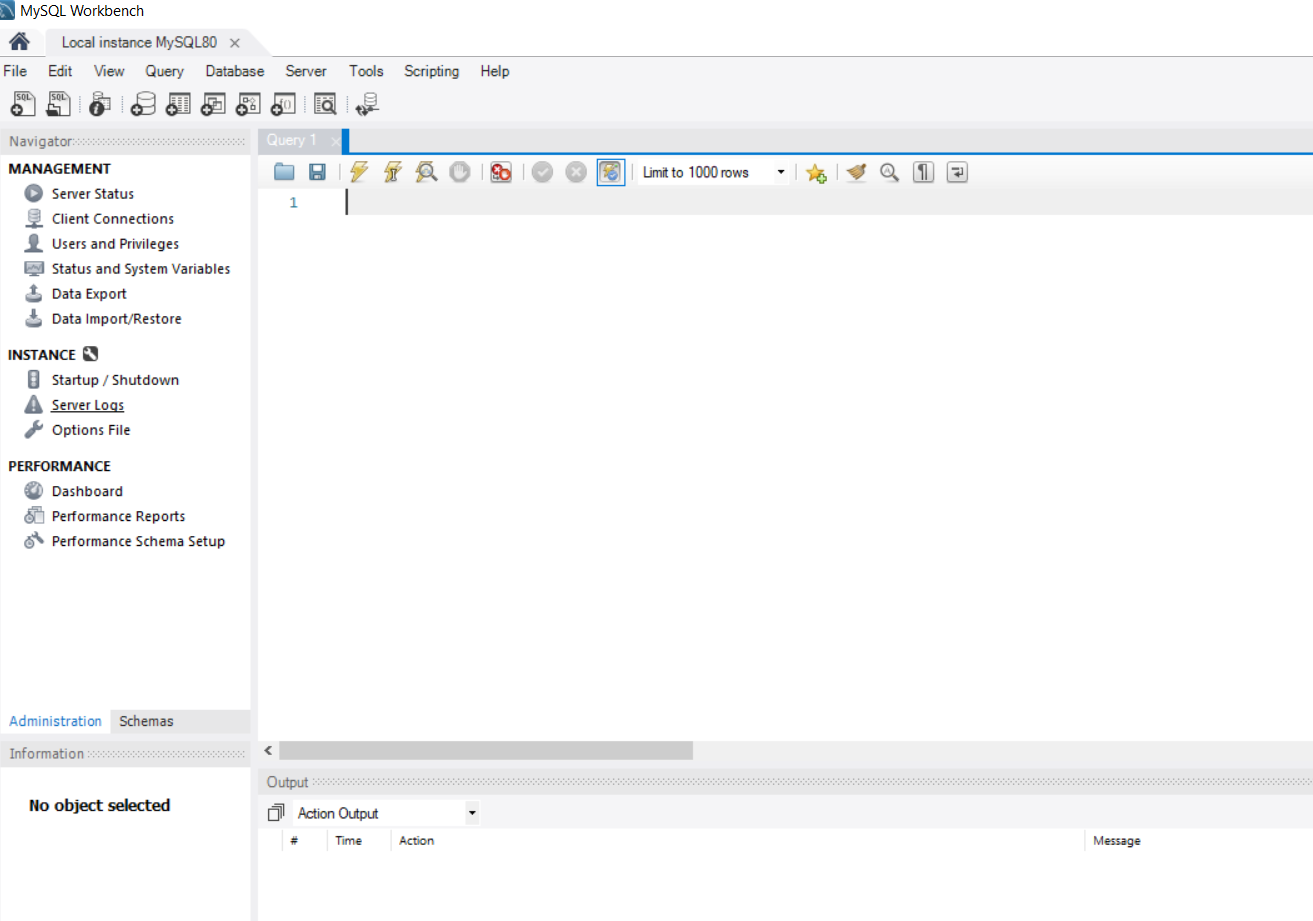


* Provide the password that you setup during installation:



**Remember the user name mentioned above as we will need it to connect python to Mysql.**

* Once the connection is established, you will see a window like this:

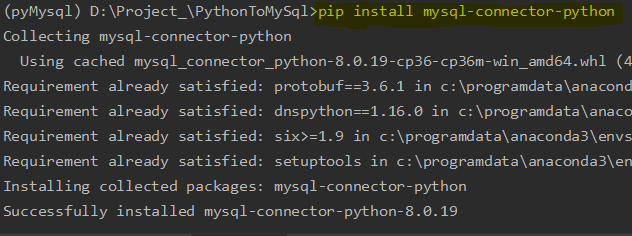


**Connecting MySql With Python**

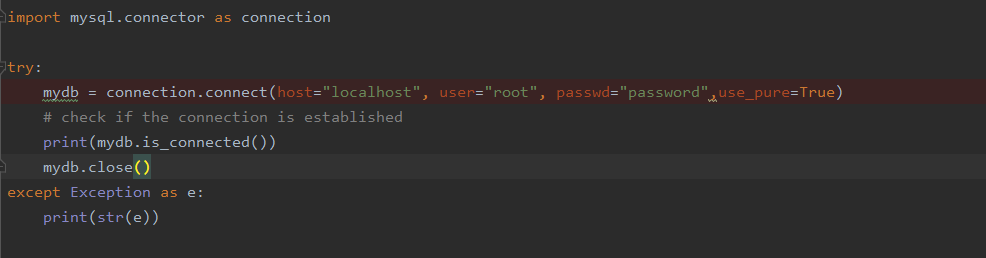
Let’s use python to connect with this database.

* First we need to install “mysql-connector-python” package to establish a connection with Mysql.

**“pip install mysql-conector-python”**



* Once the package is installed, we can go ahead with establishing the connection.

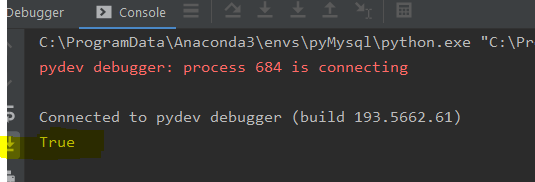


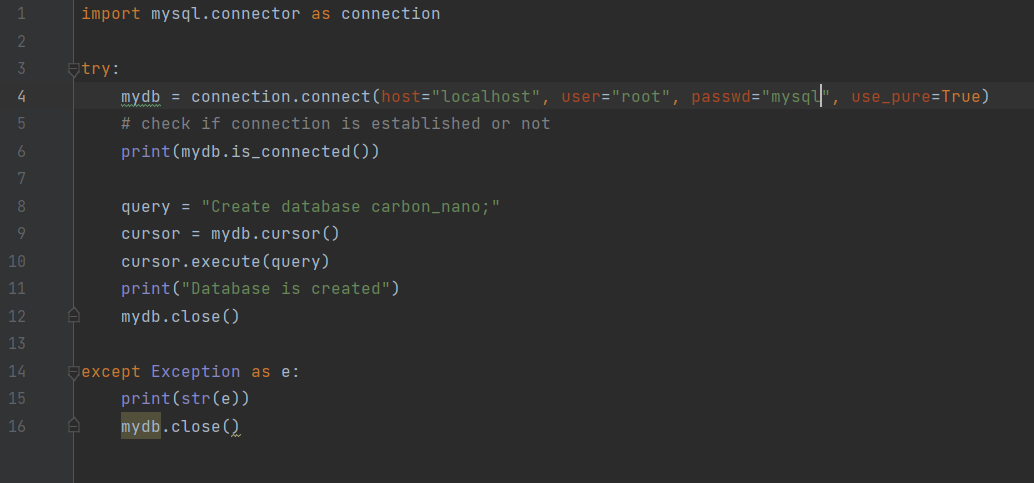
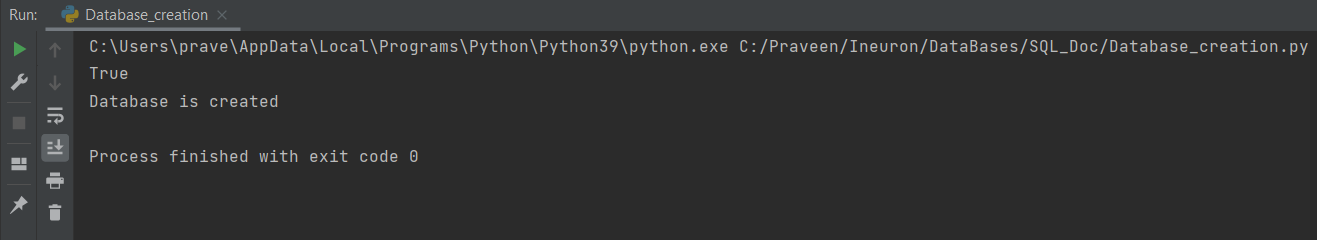
* Enter the details as shown above, your mysql server is running locally so **host** is “localhost”,

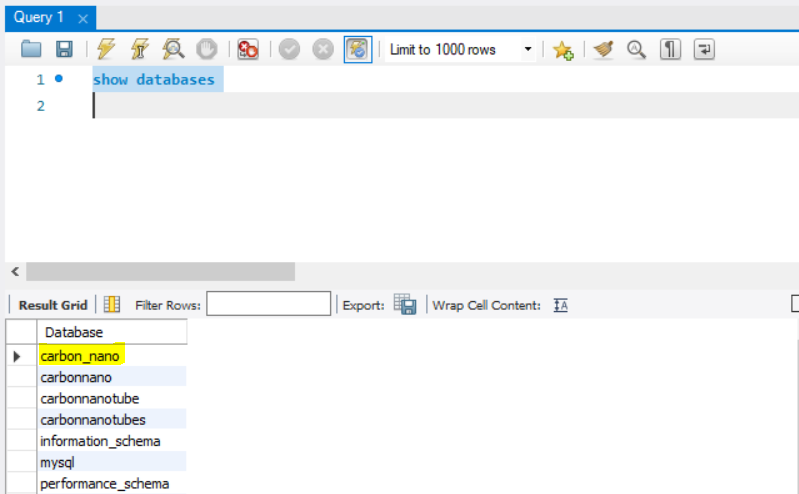
enter the **username** and **password** as was setup during installation.

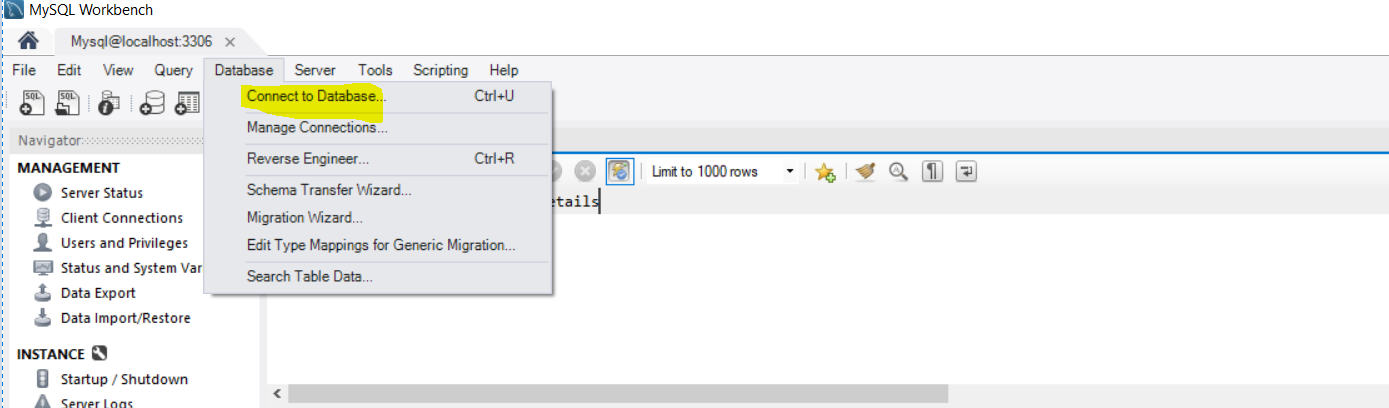
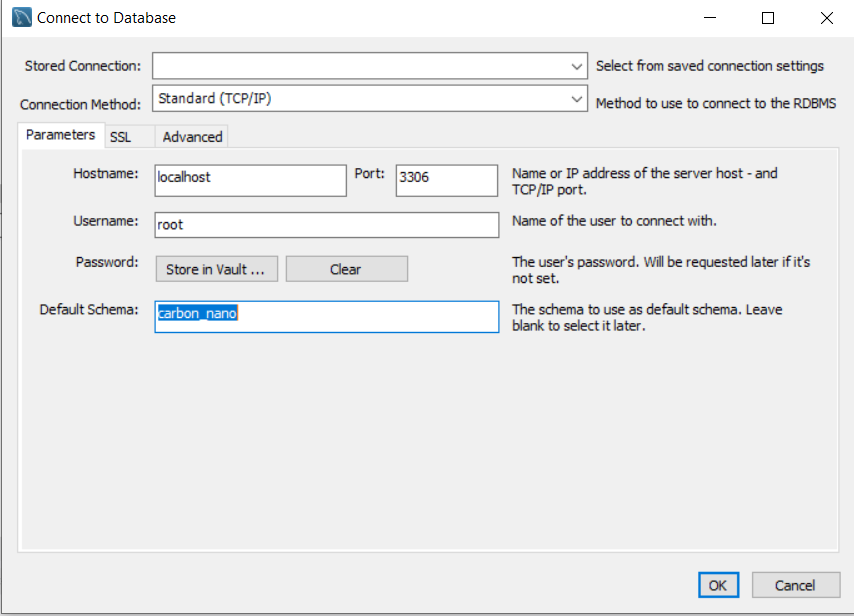
**Note: “use\_pure” argument forces mysqlConnector to user pure python connection instead of C extensions which leads to SSL error.**

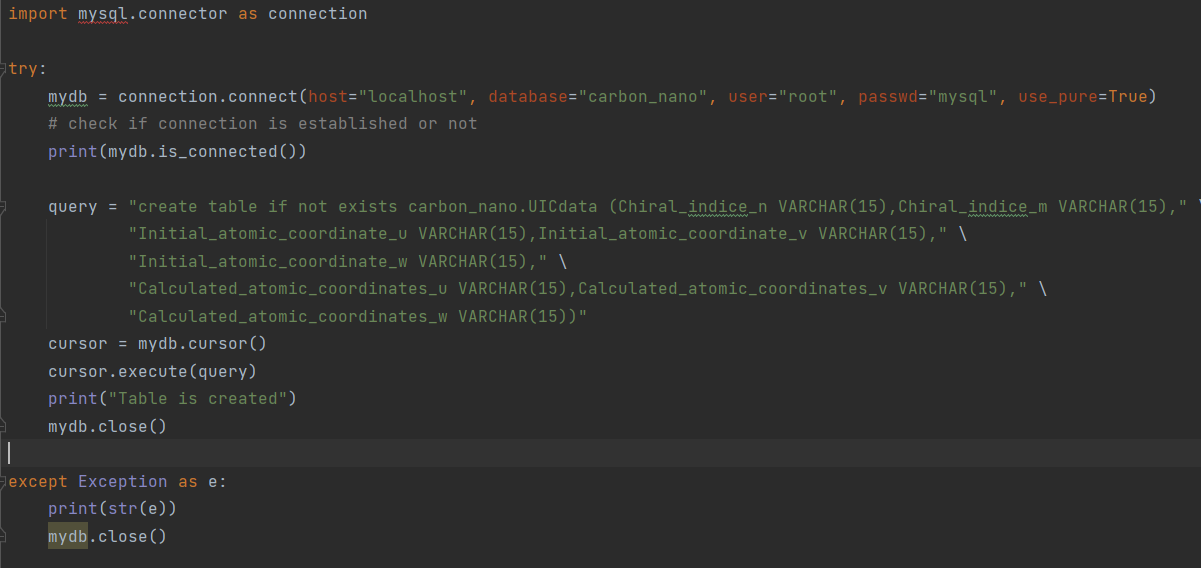
* To check if the connection is established, we can use **print (mydb.is\_connected)**. It will return **TRUE** if the connection is established else **FALSE**.

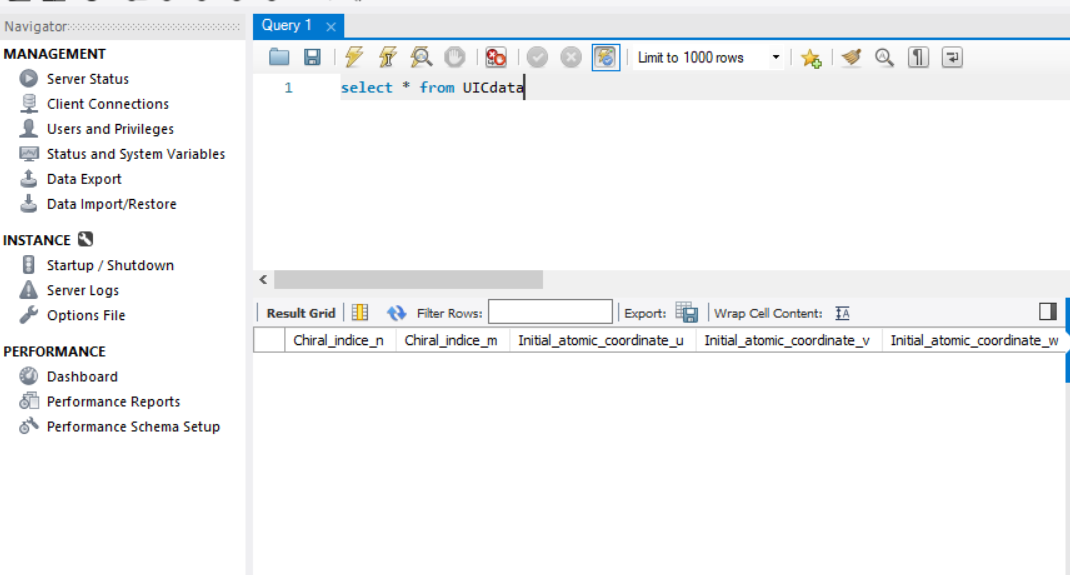


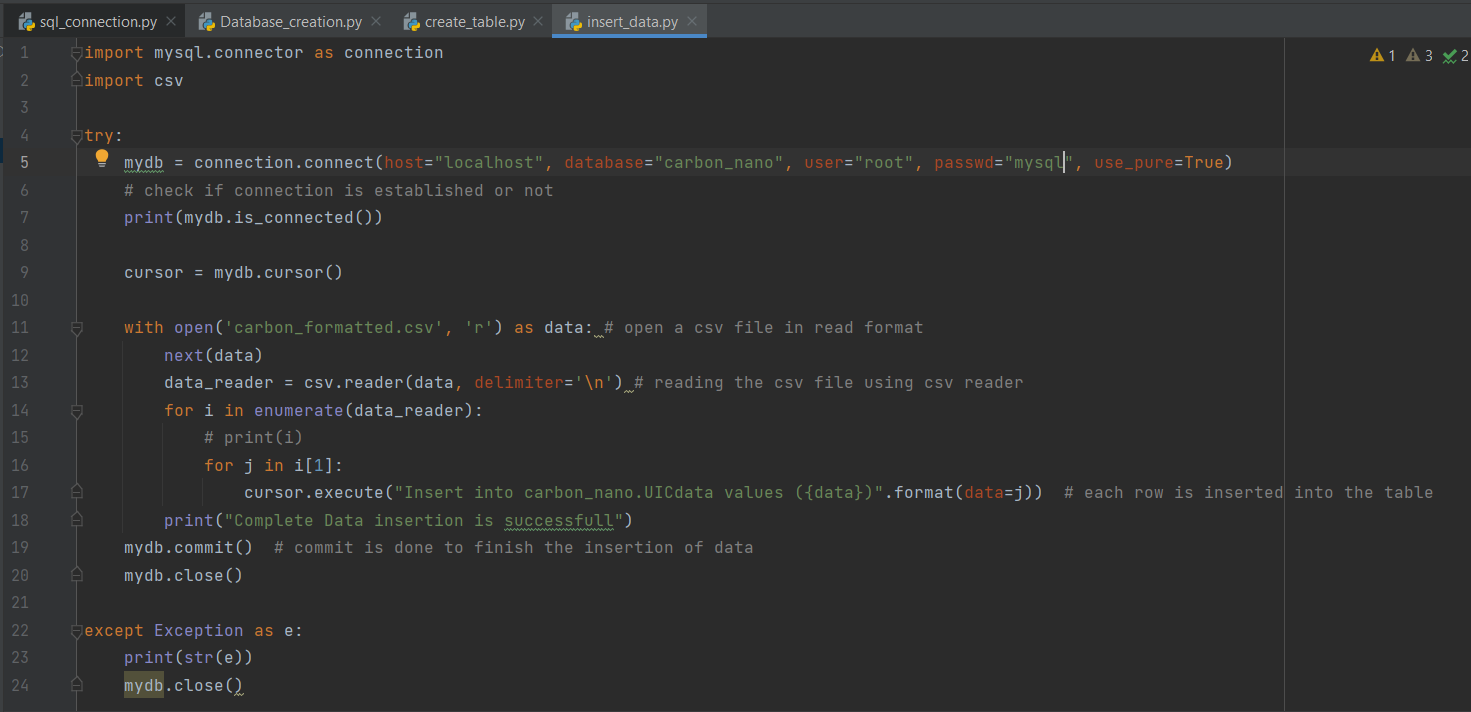
* Our connection is established now.
* Let’s start with creating a database with name carbon\_nano.
* Let’s check if database is created in our workbench

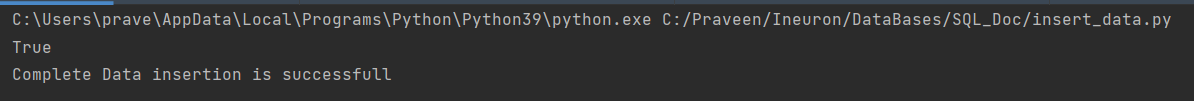


* Great! Database is created.
* Let’s start with creating tables. Let’s first connect to the created database in our workbench so that we can view the tables, once we create them from python.
* Give the database name:
* Now we are connected to the created database. Let’s start with creating tables.

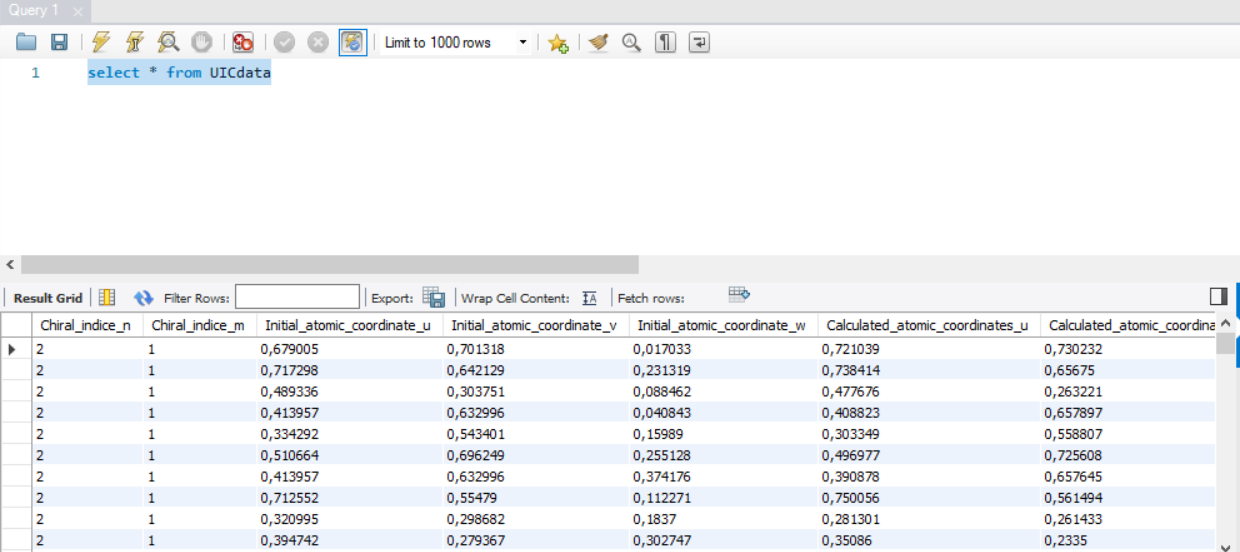


* We need to pass an **additional parameter, database name, while connecting to server**. We have passed “Carbon\_nano” database in which we are going to create the table.
* Let’s see if the table is connected in our Mysql.
* Our table is now created in the mentioned database.
* Let’s start with inserting values in our table from a csv file:



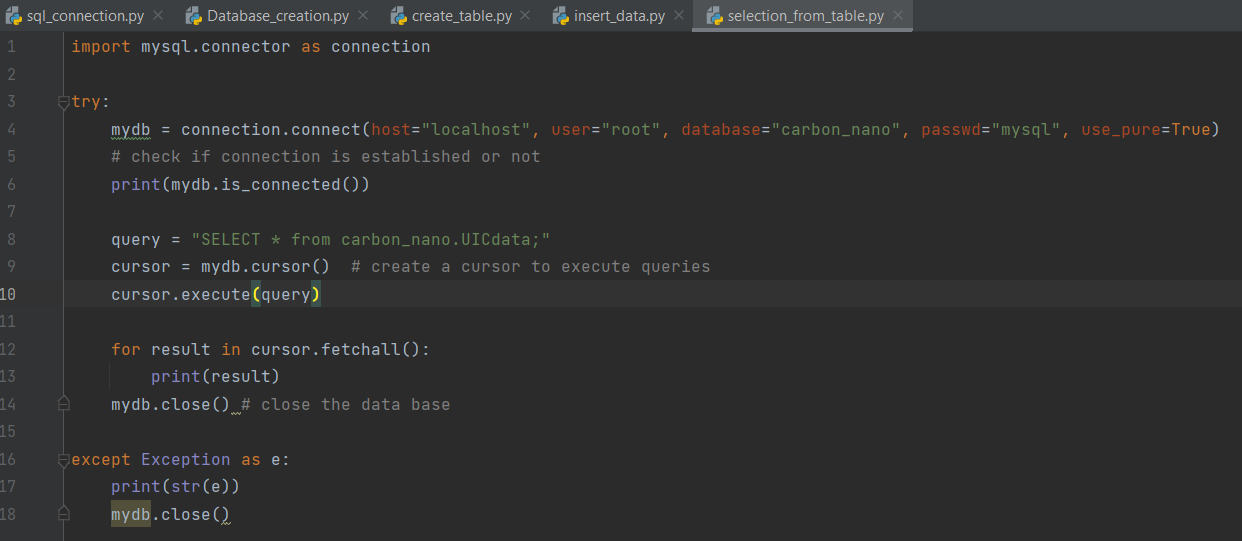


* Let’s now **insert values into a new table in a new database from a file**.
* We are loading all the values in the file “carbon\_formatted.csv” into our table.
* We created a new database named “carbon\_nano” and a new table “UICdata” in it.
* We are reading each row from the file and inserting into the table.
* Let’s check the values are inserted.

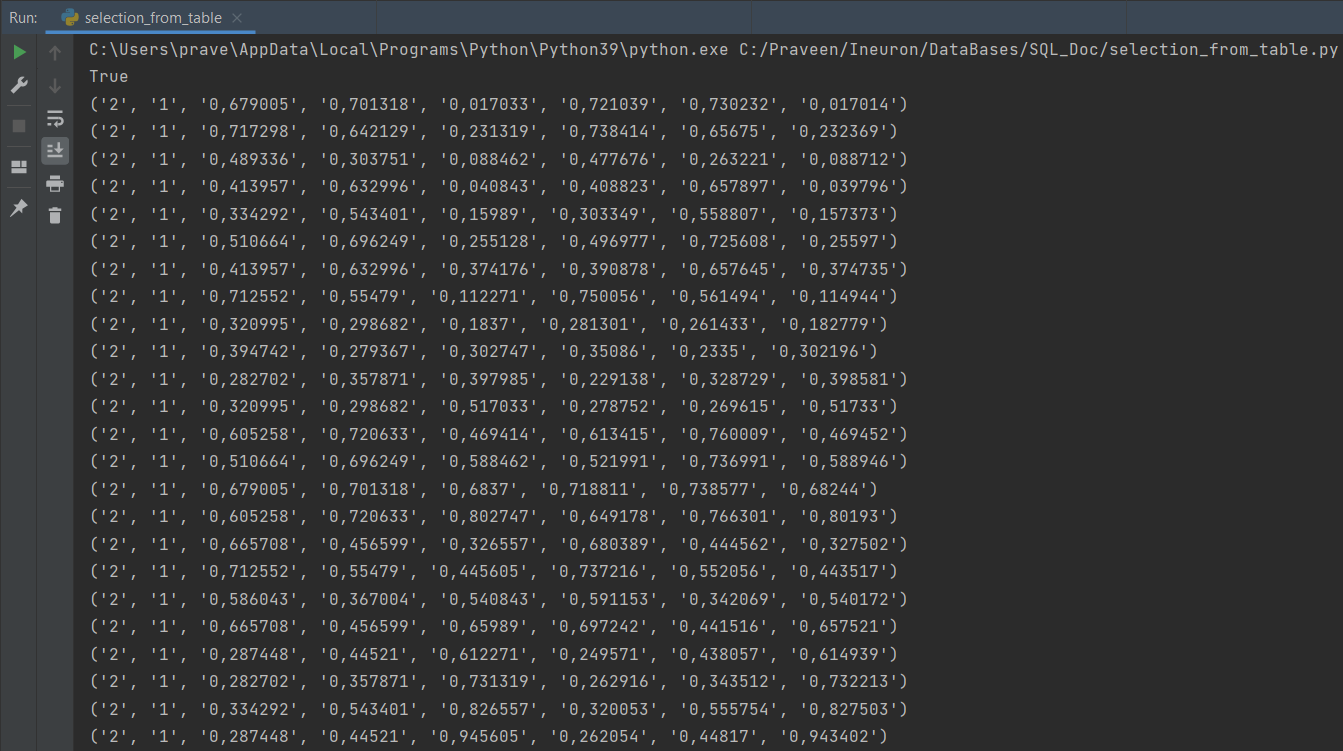


**Let’s see some other commands:**

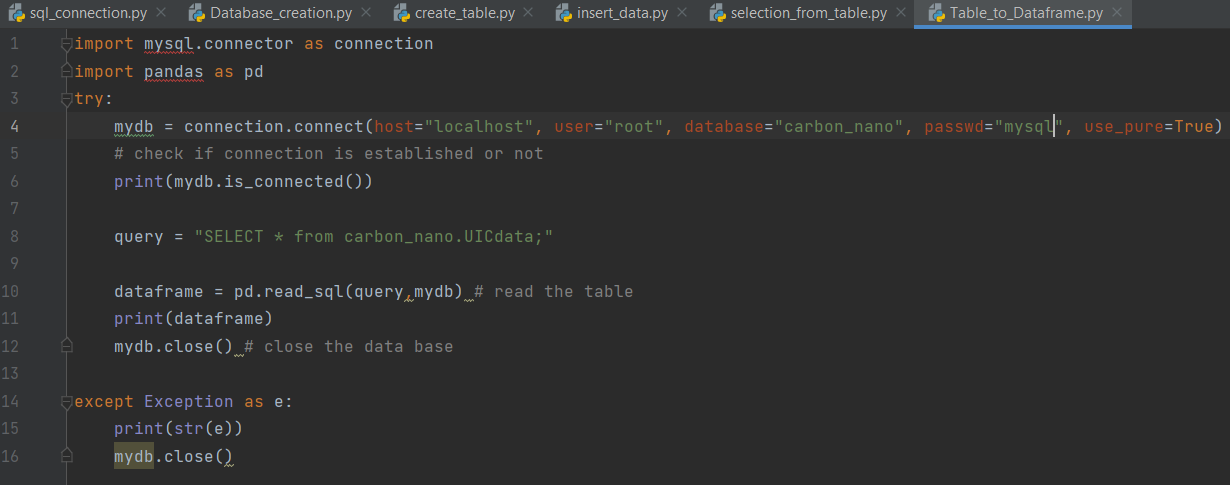
1. **Selecting from table**



Result:

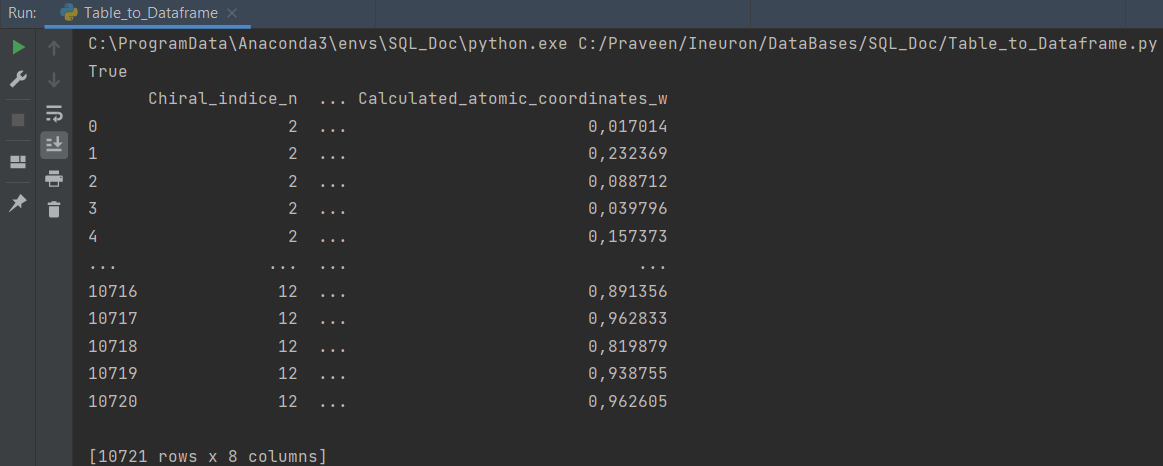


* Let’s try to store all the select values into a Dataframe.

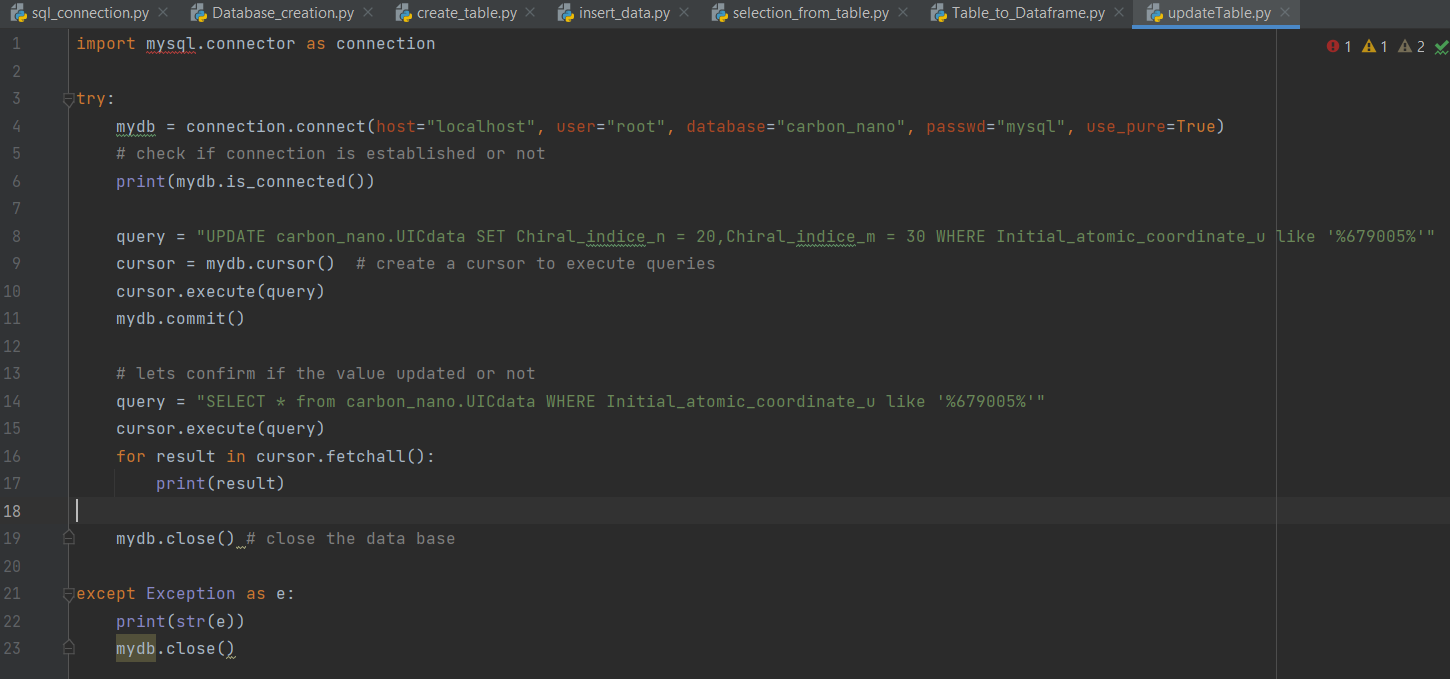


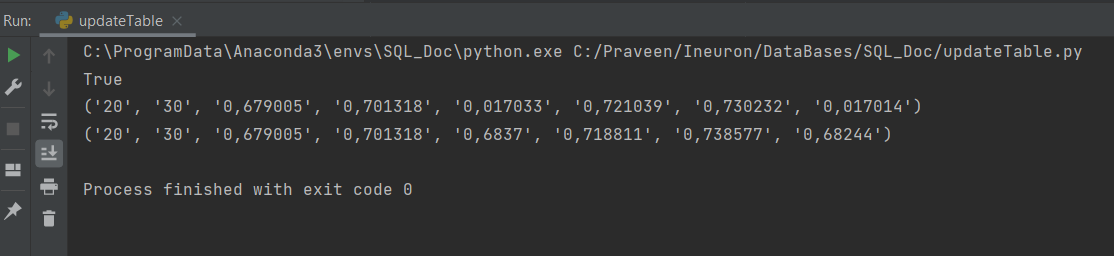
We can use **pandas.read\_sql** to store the values in a dataframe.

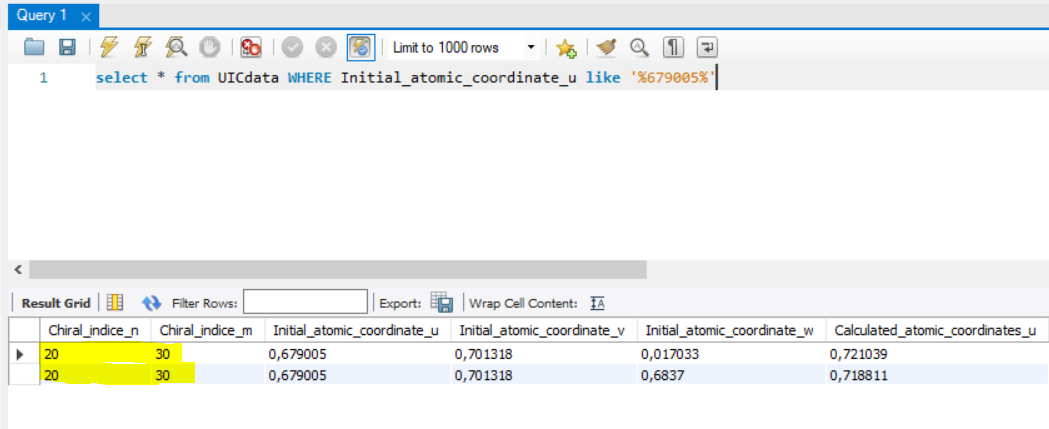
Let’s see the result:



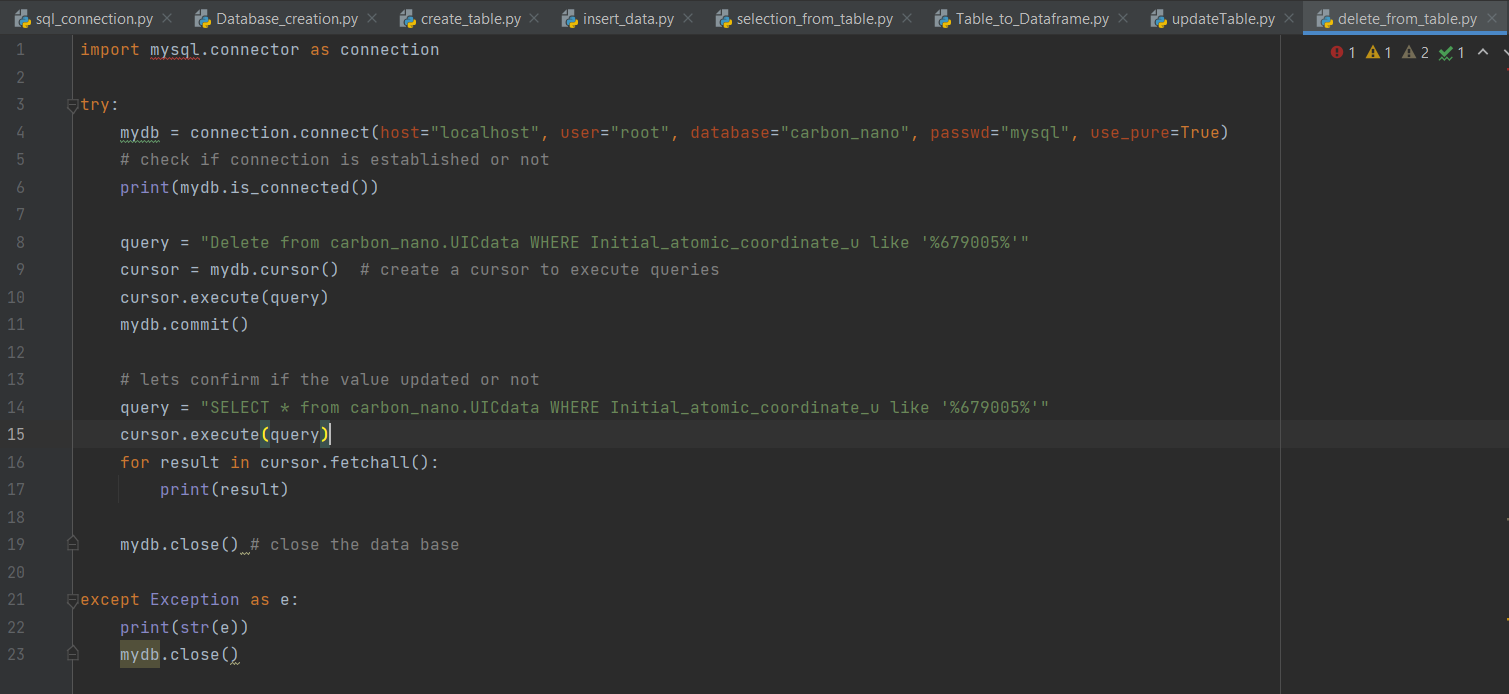
1. **Update statement**



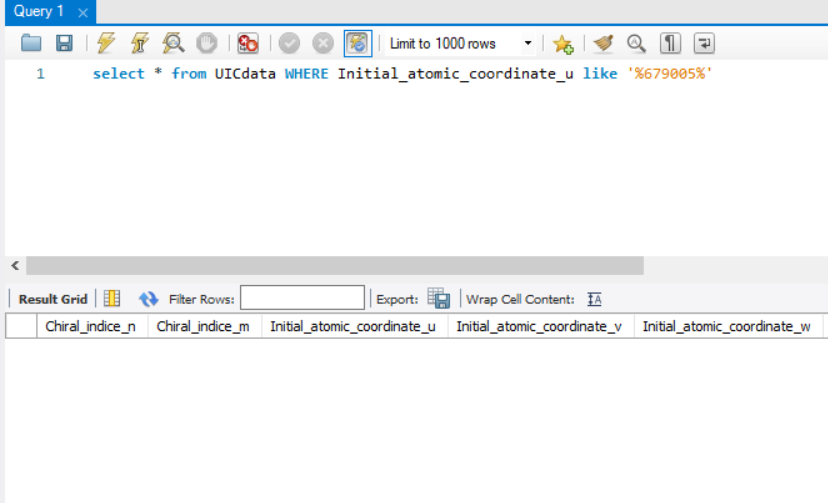


* Let’s check in MySql workbench:

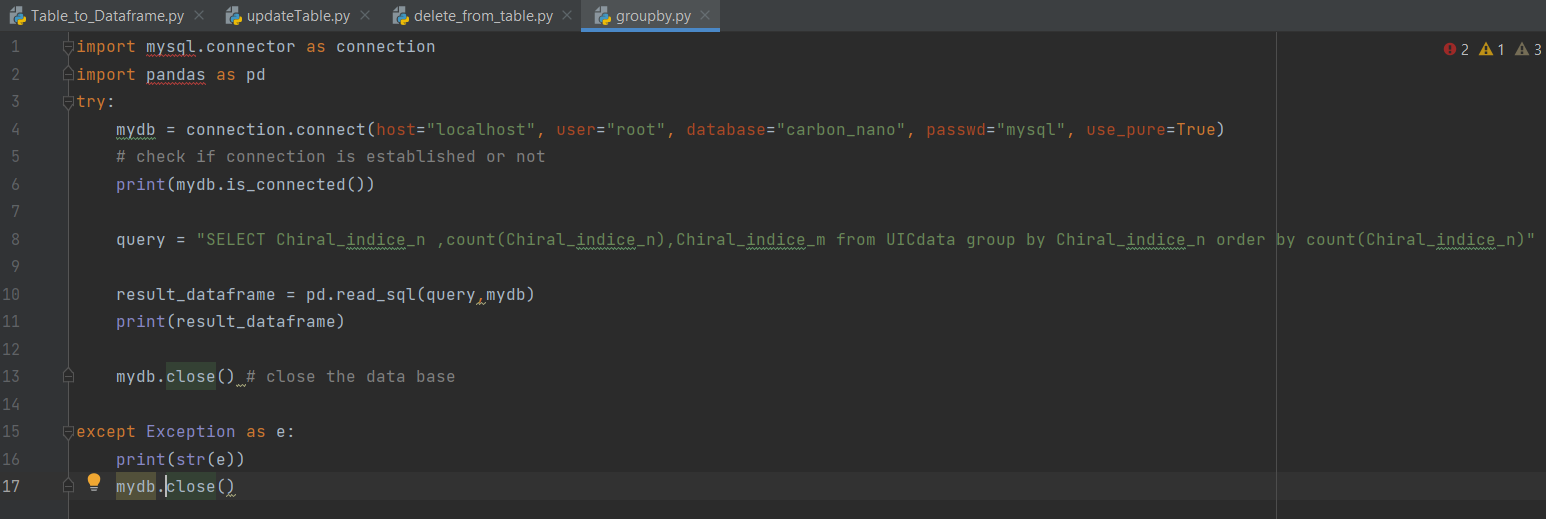
1. **Delete statement**



* Let’s check in MySql workbench:



Result above shows empty in workbench since we have deleted the values

1. **Group by, Order by**

**Group by, Order by result :**

