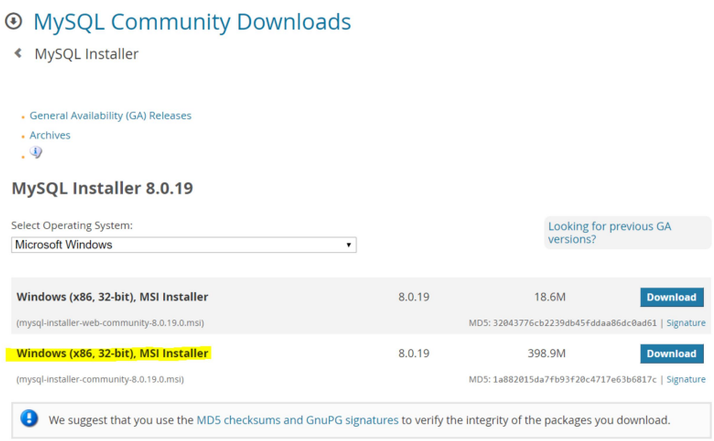
Author: Oluwaseun Tope Oyero
Title: Working with MySQL and Python

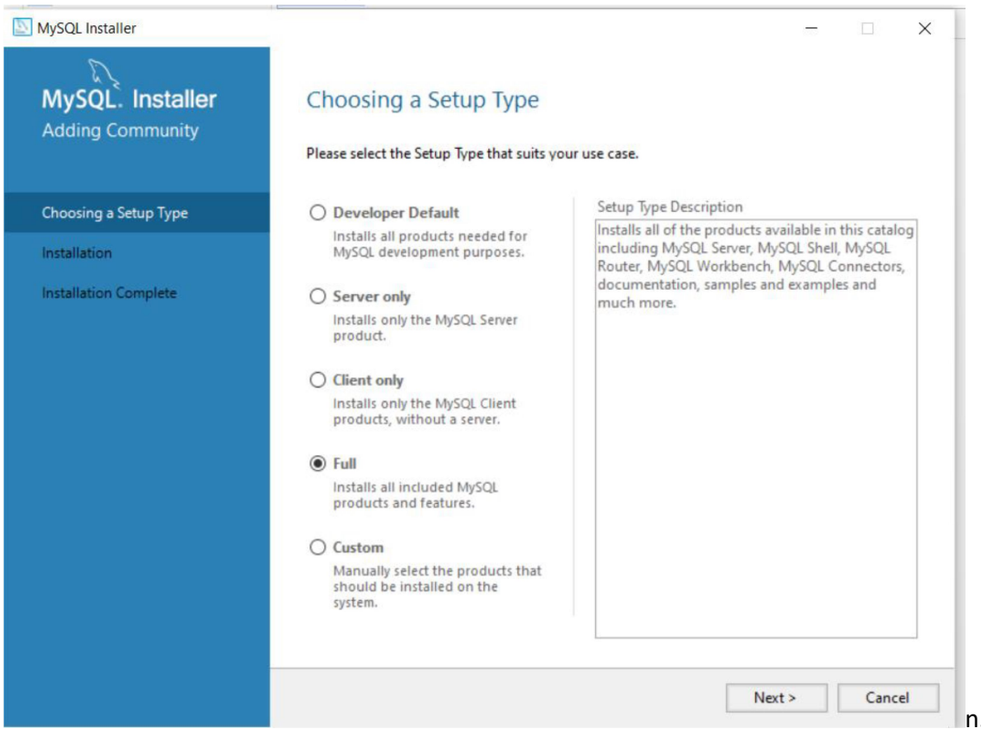
Installing MySQL Workbench

- Download MYSQL application. Go to page: https://dev.mysql.com/downloads/installer/

Click on the link highlighted below:

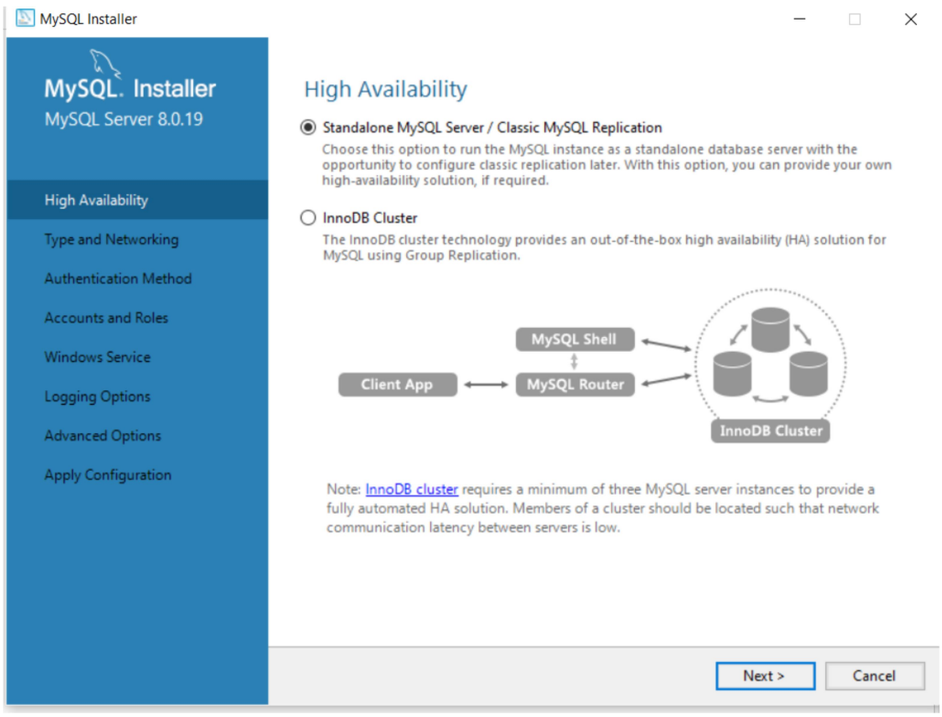


- Once downloaded, install the application. Follow the instructions and continue with installation.

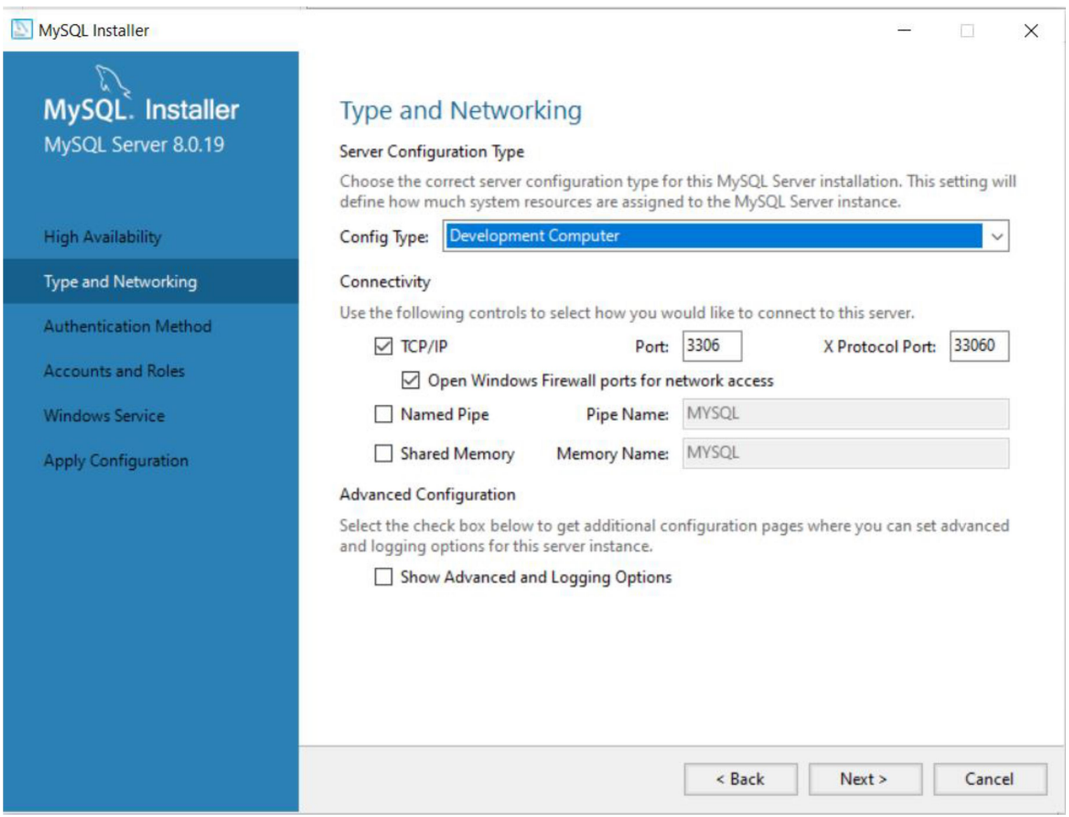


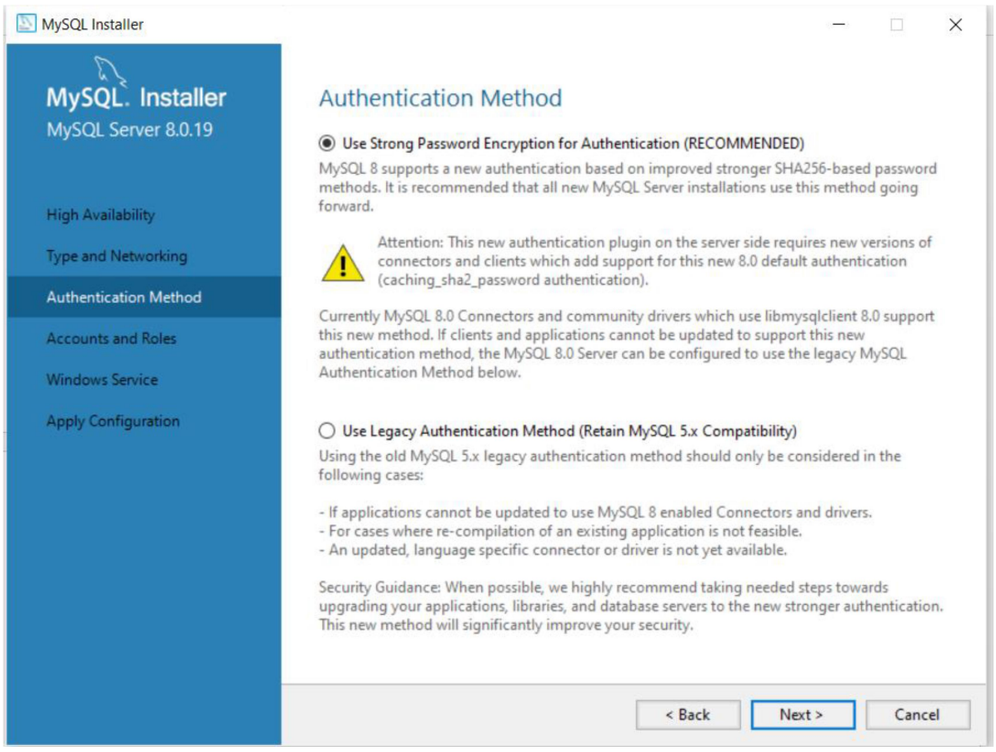
n.

- Select Standalone MySQL server option.

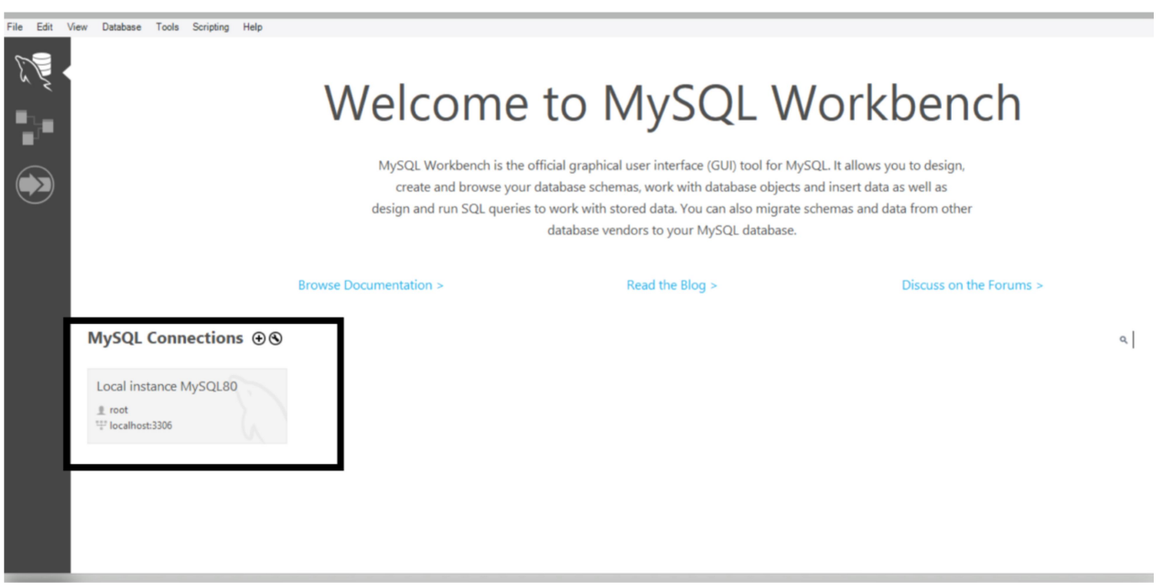


- Keep the default values for “Type and Networking” section.

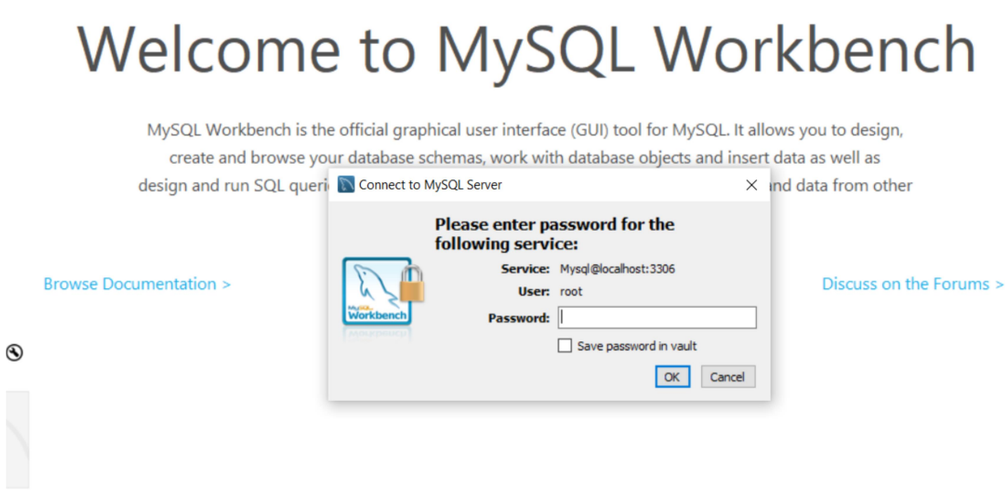




- Finish the installation by pressing next.
- Once the installation is complete, open the MYSQL workbench and click on the local instance connection. The window will look something like this:

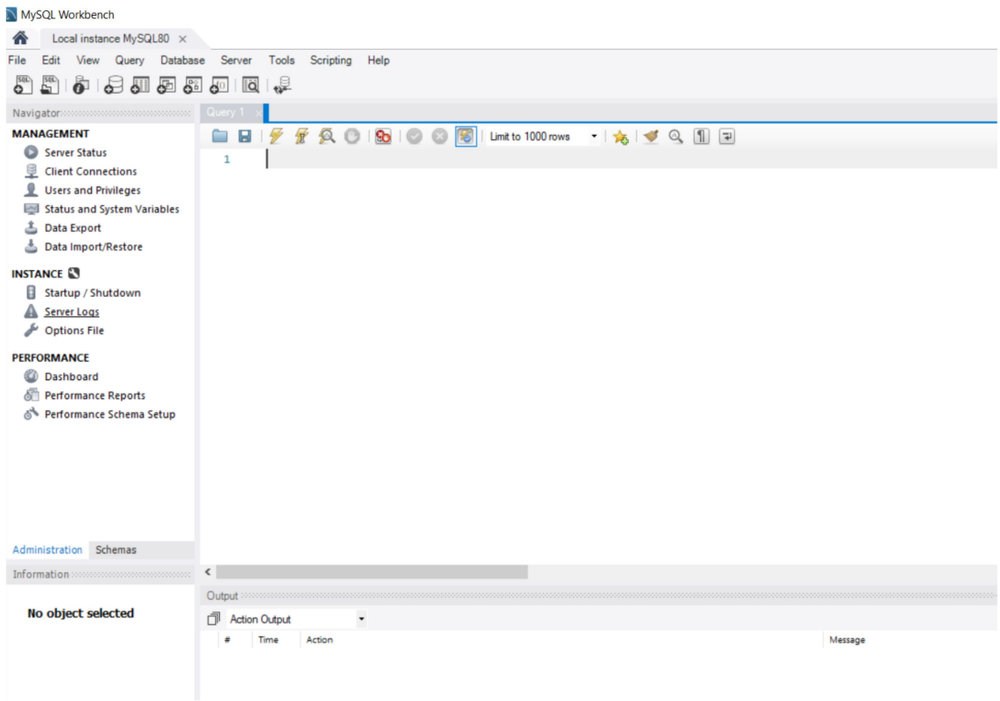


- Provide the password that you setup during installation:



<u>Remember the user name mentioned above as we will need it to connect python to Mysql.</u>

- 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"

```
(pyMysql) D:\Project_\PythonToMySql>pip install mysql-connector-python
Collecting mysql-connector-python
  Using cached mysql_connector_python-8.0.19-cp36-cp36m-win_amd64.whl (4
Requirement already satisfied: protobuf==3.6.1 in c:\programdata\anacond
Requirement already satisfied: dnspython==1.16.0 in c:\programdata\anaco
Requirement already satisfied: six>=1.9 in c:\programdata\anaconda3\envs
Requirement already satisfied: setuptools in c:\programdata\anaconda3\en
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-8.0.19
```

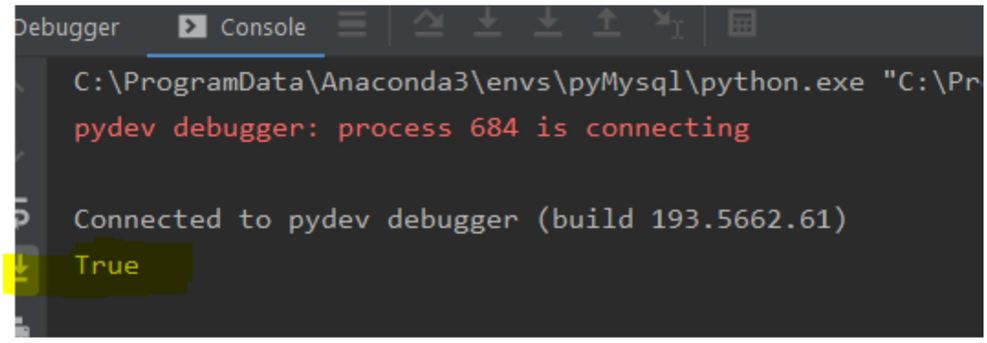
- Once the package is installed, we can go ahead with establishing the connection.

```
import mysql.connector as connection

try:
    mydb = connection.connect(host="localhost", user="root", passwd="password",use_pure=True)
    # check if the connection is established
    print(mydb.is_connected())
    mydb.close()
except Exception as e:
    print(str(e))
```

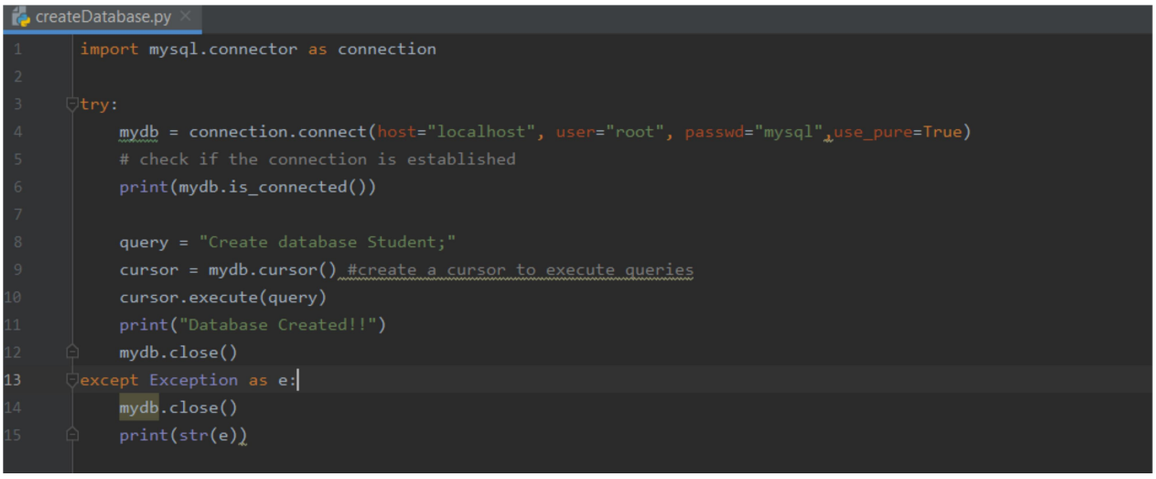
- Enter the details as shown above, your mysql server is running locally so host is "localhost", enter the **username** and **password** as was setup during installation.

Note: "use_pure" argument forces mysqlConnector to user pure python connection instead of C extensions which leads to SSL error.

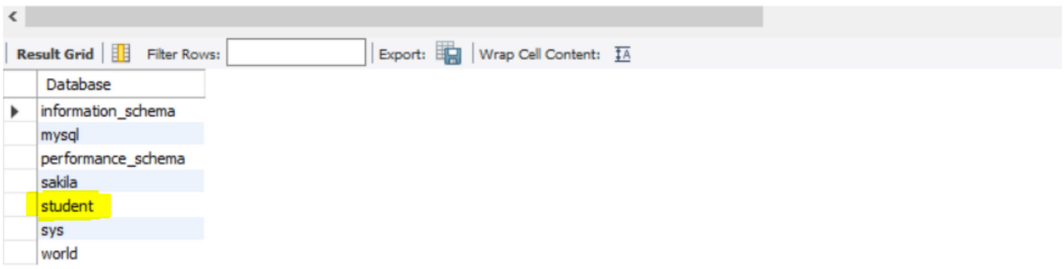
• To check if the connection is established, we can **use print (mydb.is_connected).** It will return **TRUE** if the connection is established else **FALSE**.



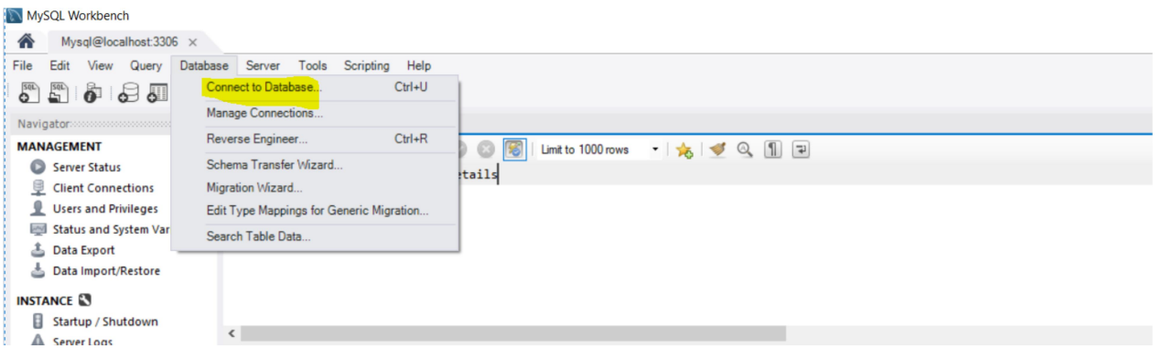
- Our connection is established now.
- Let's start with creating a database with name Student.



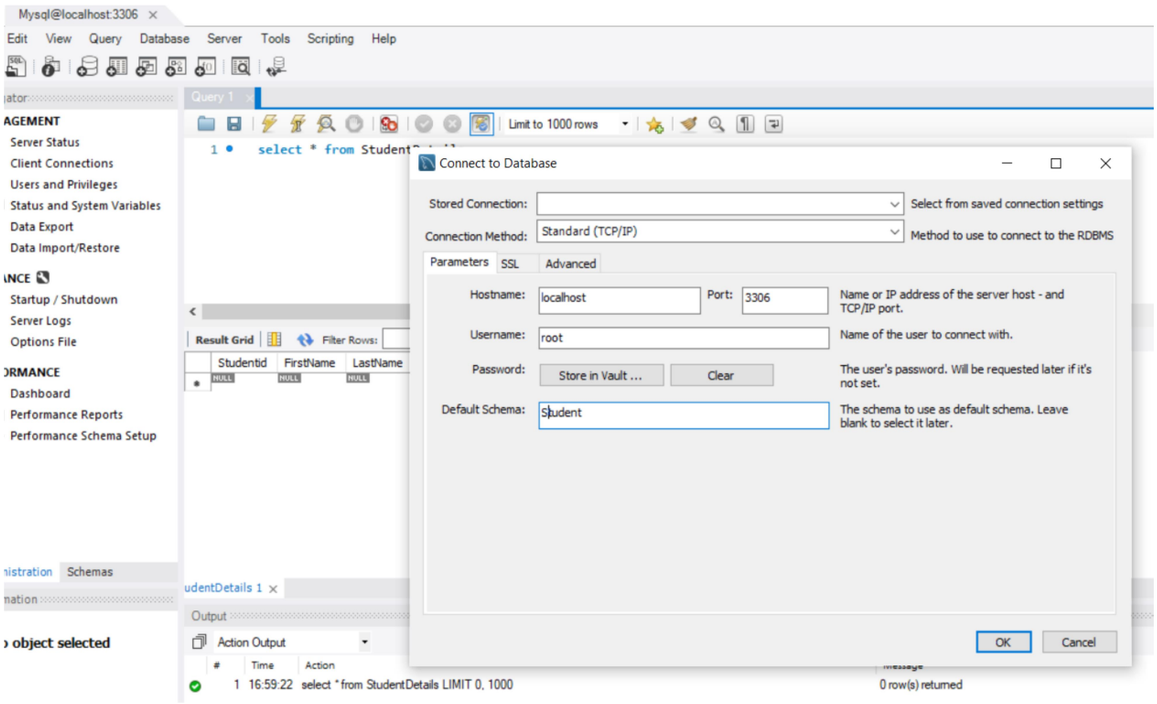
- Let's check if database is created in our MySQL Workbench.



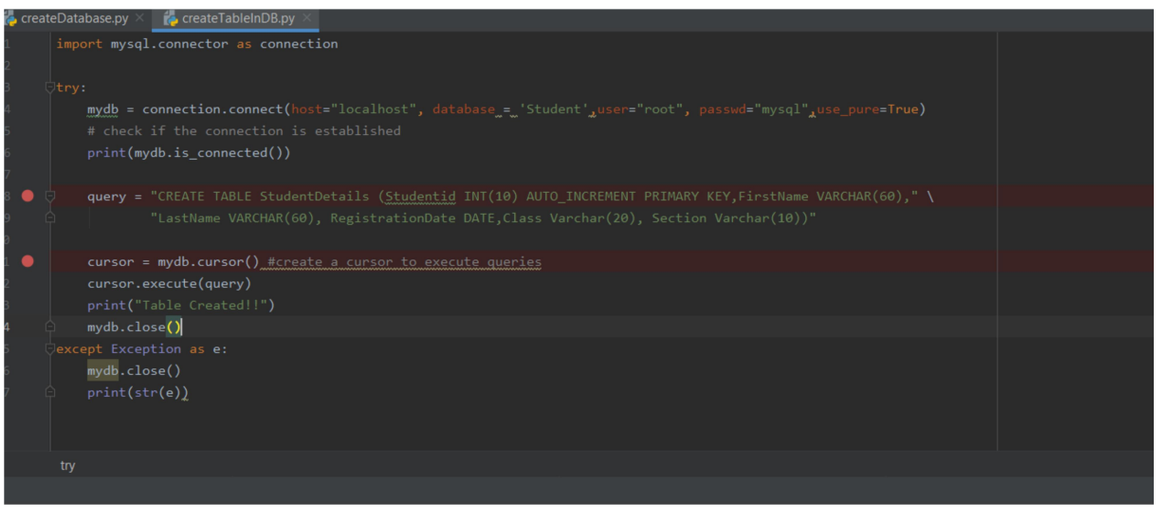
- Great! Database is created.
- Let's start with creating tables. Let's first connect to the created database in our workbench so that we can view the tables, once we create them from python.



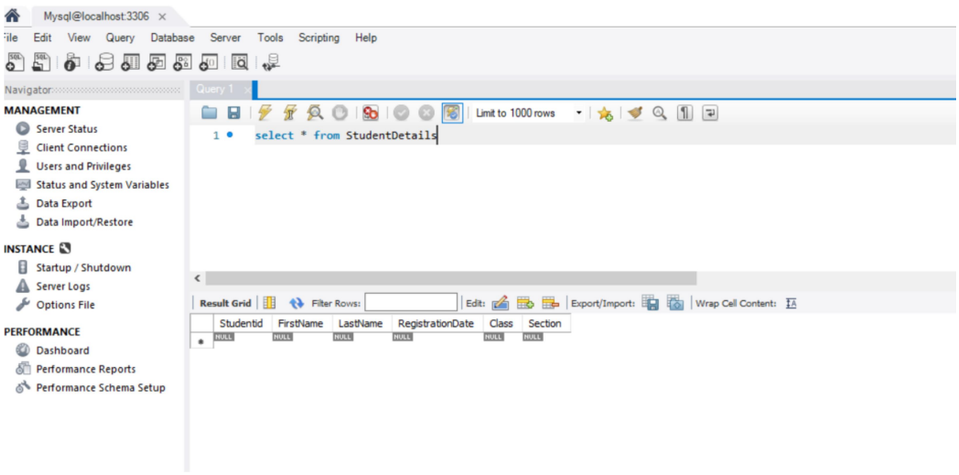
- Give the database name:



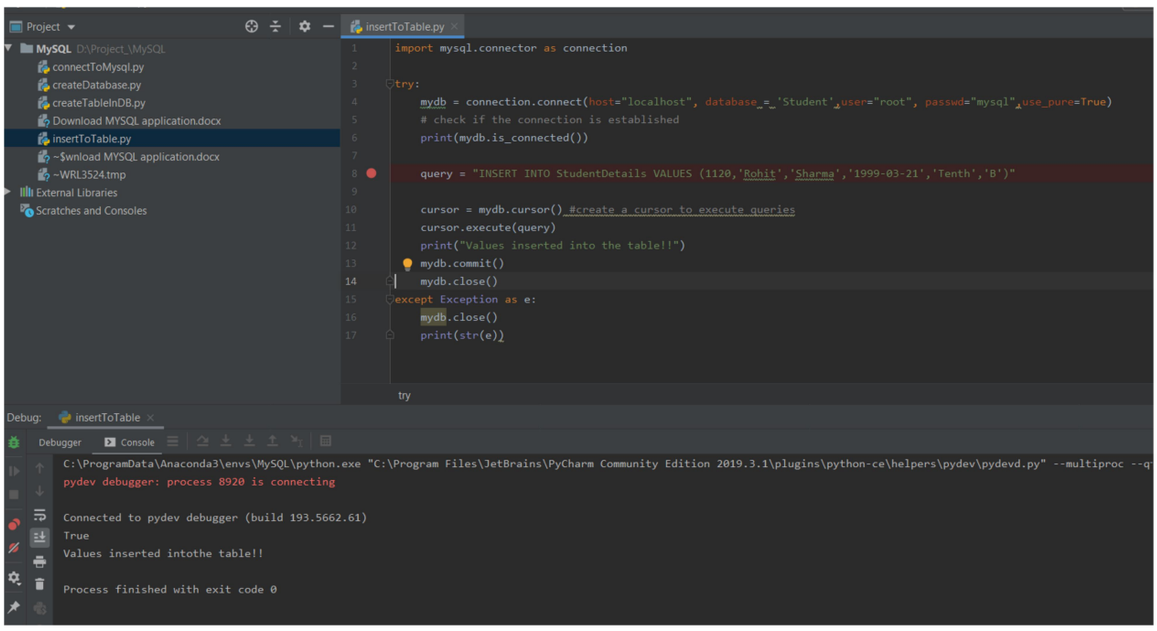
- Now we are connected to the created database. Let's start with creating tables.



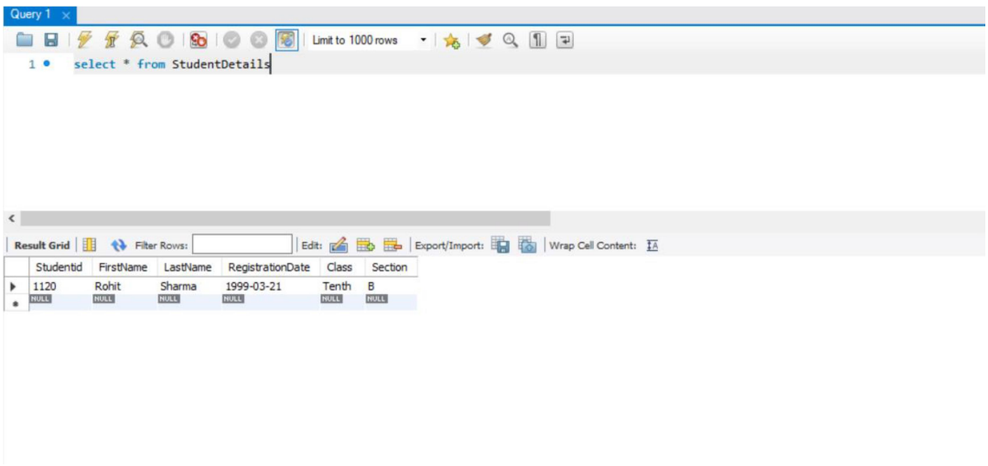
- We need to pass an additional parameter, database name, while connecting to server. We have passed "Student" database in which we are going to create the table.
- Let's see if the table is connected in our Mysql.



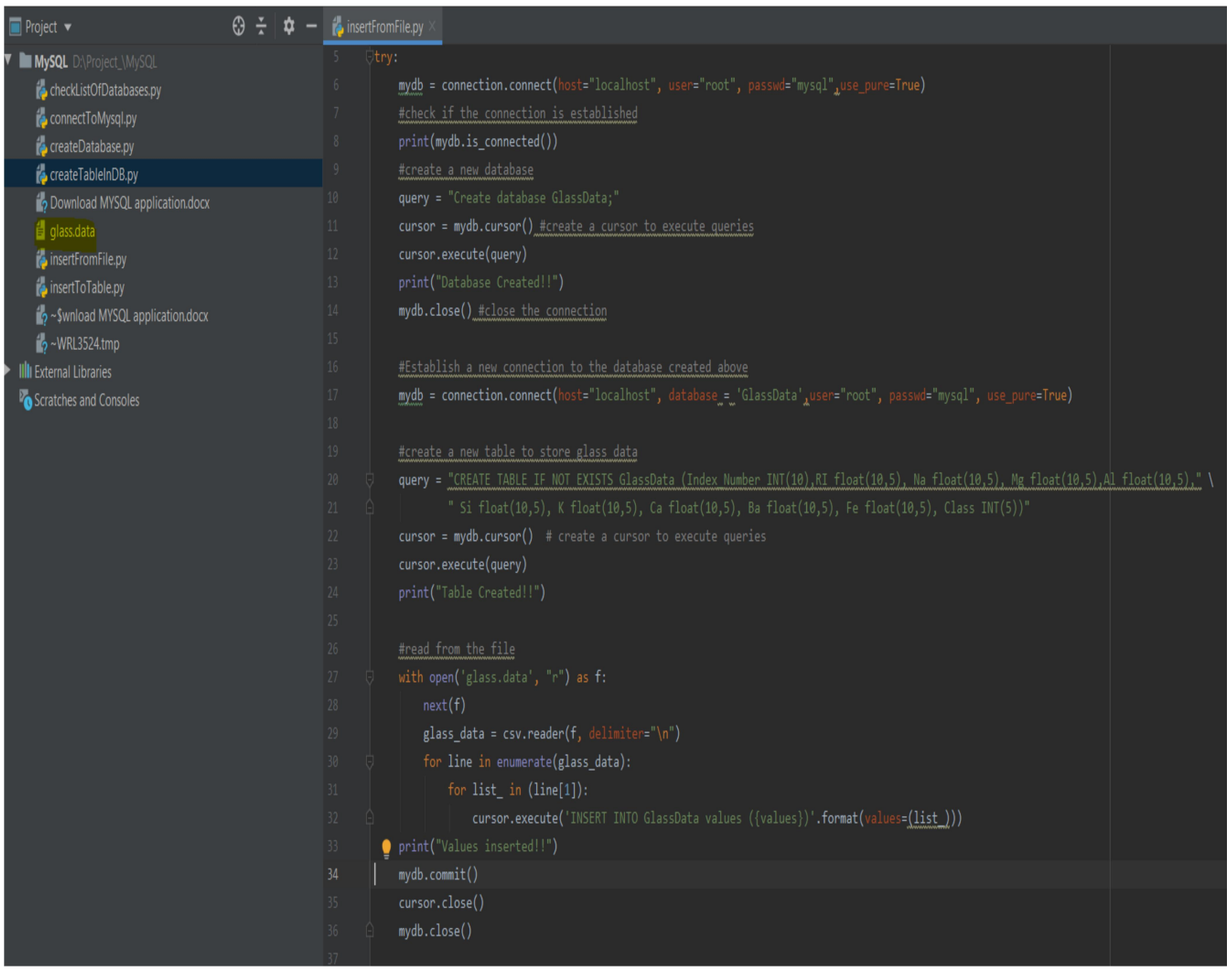
- Our table is now created in the mentioned database.

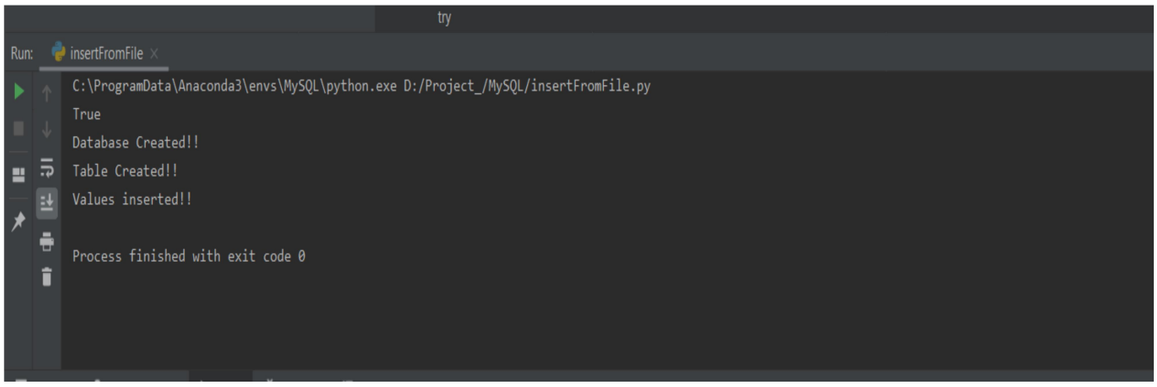


- Let's check if the values are inserted:

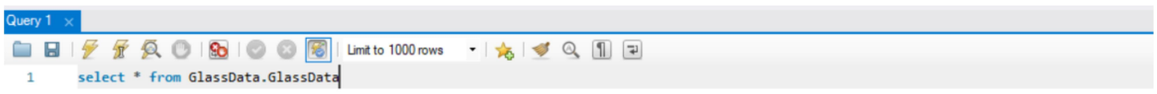


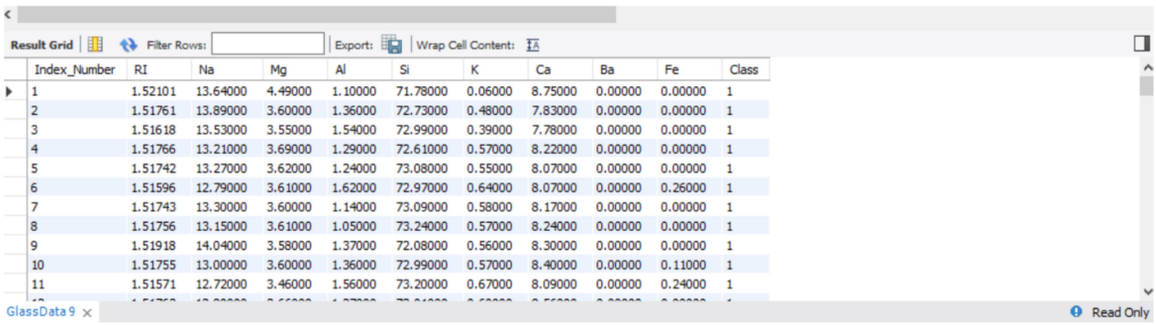
- Let's now insert values into a new table in a new database from a file.
- We are loading all the values in the file "glass. Data" into our table.
- We created a new database named "Glass Data" and a new table "Glass Data" in it.
- We are reading each row from the file and inserting into the table.





- Let's check if the values are inserted in the table.





Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Index_Number	RI	Na	Mg	Al	Si	K	Ca	Ba	Fe	Class
1	1.52101	13.64000	4.49000	1.10000	71.78000	0.06000	8.75000	0.00000	0.00000	1
2	1.51761	13.89000	3.60000	1.36000	72.73000	0.48000	7.83000	0.00000	0.00000	1
3	1.51618	13.53000	3.55000	1.54000	72.99000	0.39000	7.78000	0.00000	0.00000	1
4	1.51766	13.21000	3.69000	1.29000	72.61000	0.57000	8.22000	0.00000	0.00000	1
5	1.51742	13.27000	3.62000	1.24000	73.08000	0.55000	8.07000	0.00000	0.00000	1
6	1.51596	12.79000	3.61000	1.62000	72.97000	0.64000	8.07000	0.00000	0.26000	1
7	1.51743	13.30000	3.60000	1.14000	73.09000	0.58000	8.17000	0.00000	0.00000	1
8	1.51756	13.15000	3.61000	1.05000	73.24000	0.57000	8.24000	0.00000	0.00000	1
9	1.51918	14.04000	3.58000	1.37000	72.08000	0.56000	8.30000	0.00000	0.00000	1
10	1.51755	13.00000	3.60000	1.36000	72.99000	0.57000	8.40000	0.00000	0.11000	1
11	1.51571	12.72000	3.46000	1.56000	73.20000	0.67000	8.09000	0.00000	0.24000	1

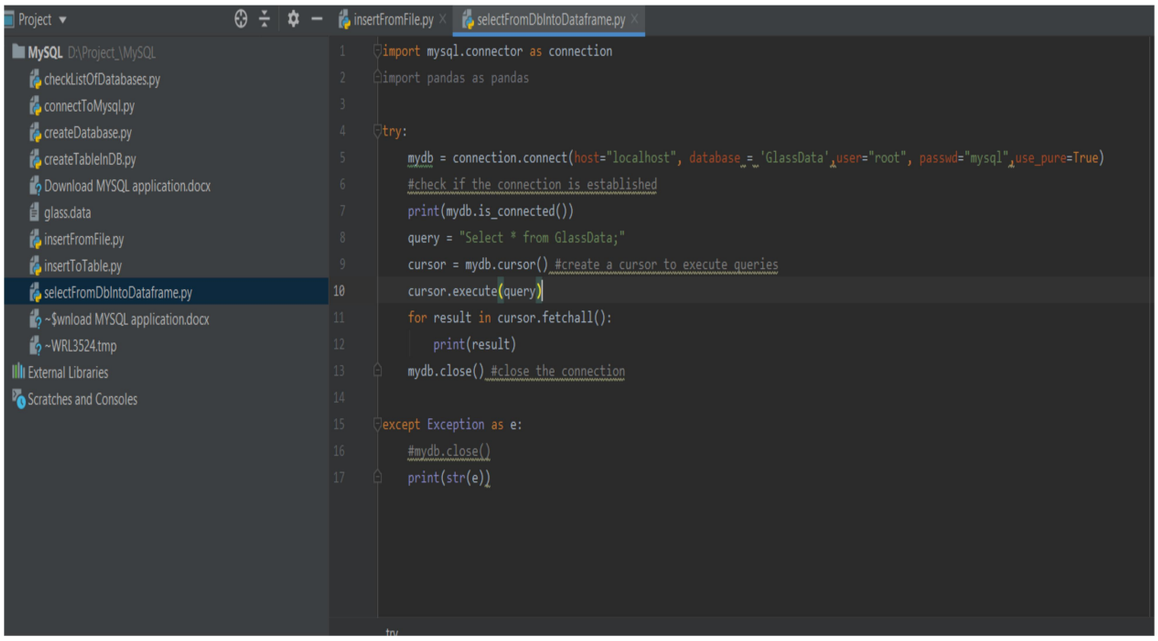
GlassData 9 | Read Only

- The values are inserted.

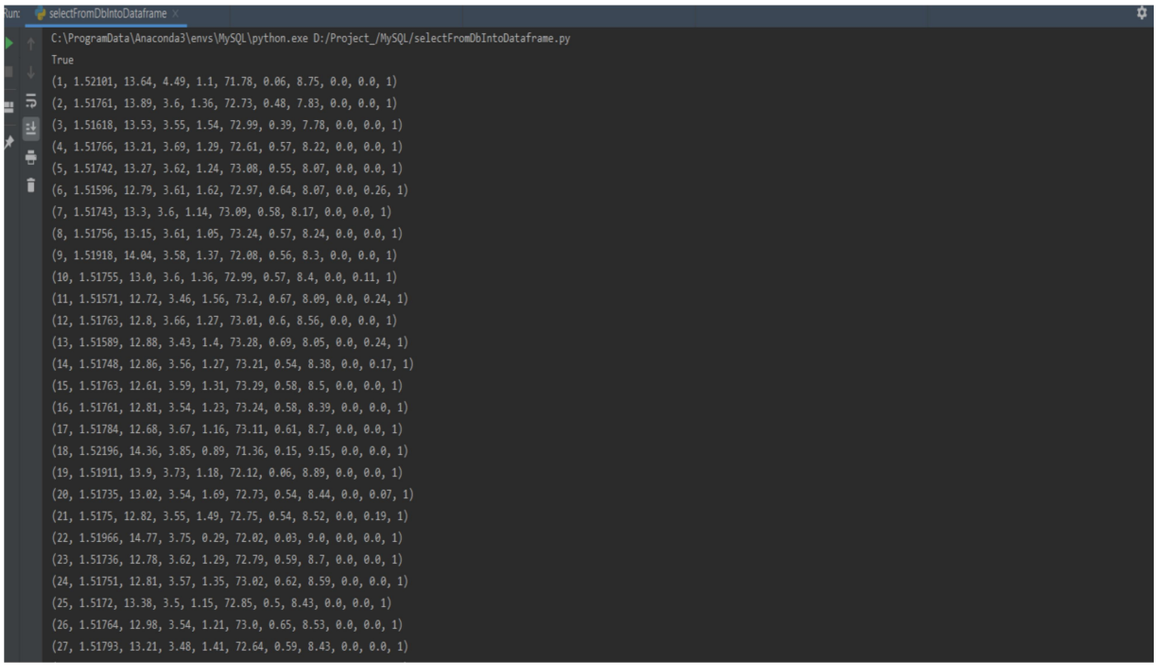
Let's see some other commands:

1) Selecting from table

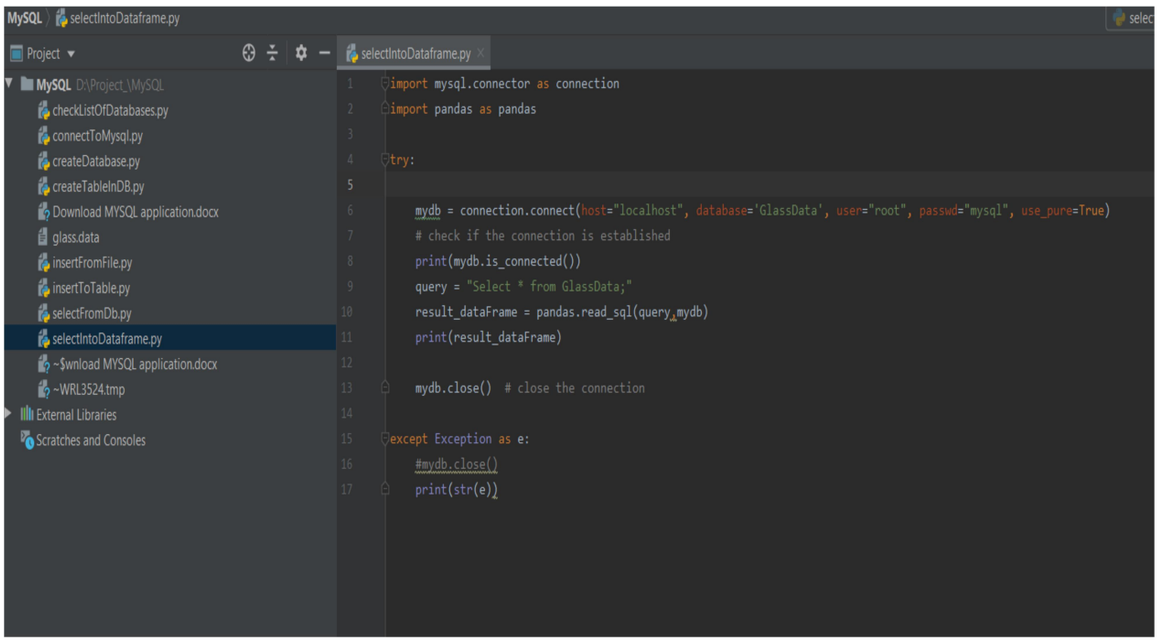
Oluwaseun Tope Oyero



Result:



- Let's try to store all the select values into a Dataframe.

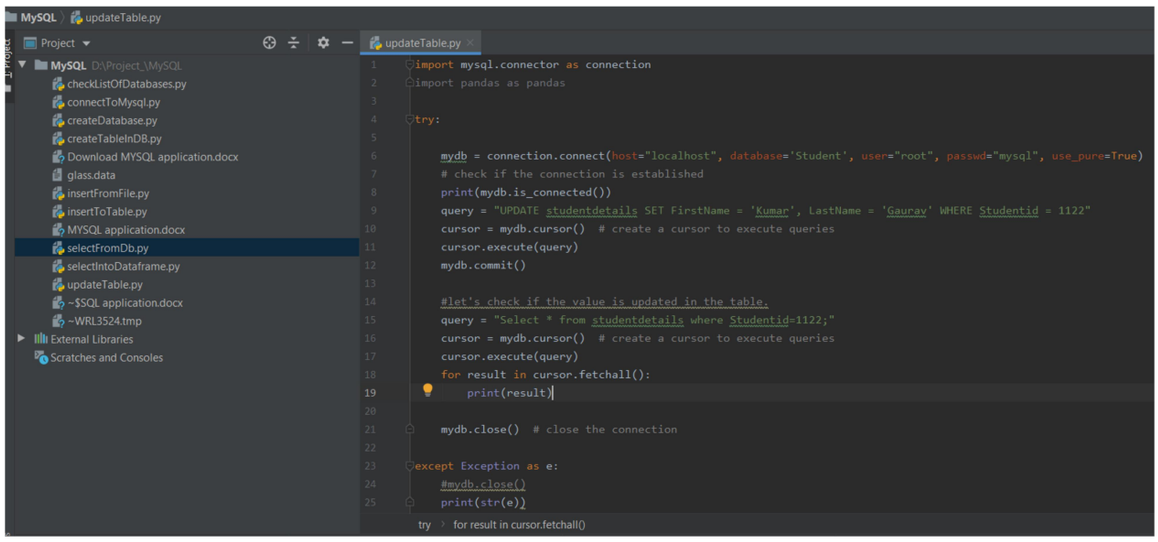


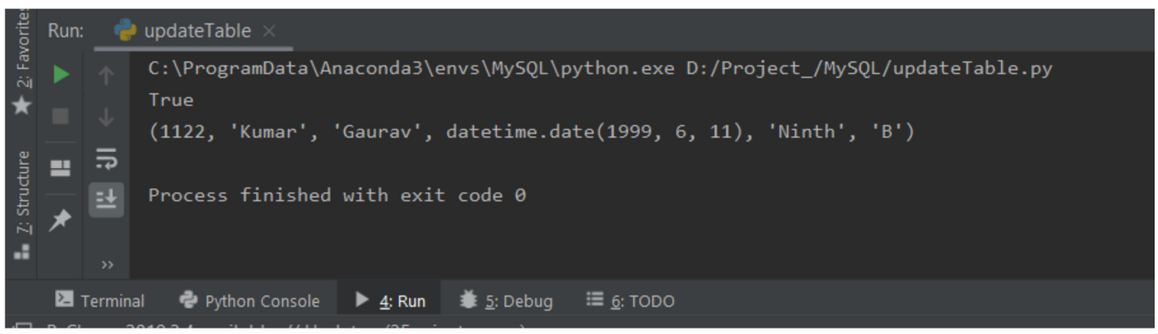
We can **use pandas.read_sql** to store the values in a **dataframe.**

Let's see the result:

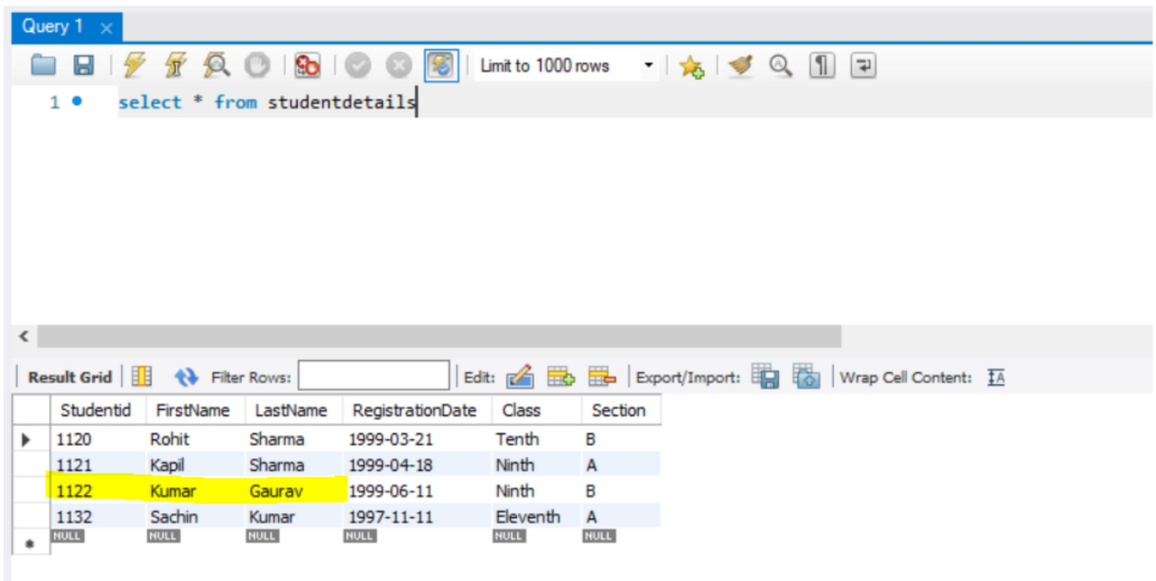
```
True
     Index_Number       RI     Na    Mg    Al  ...     K    Ca    Ba   Fe  Class
0               1  1.52101  13.64  4.49  1.10  ...  0.06  8.75  0.00  0.0      1
1               2  1.51761  13.89  3.60  1.36  ...  0.48  7.83  0.00  0.0      1
2               3  1.51618  13.53  3.55  1.54  ...  0.39  7.78  0.00  0.0      1
3               4  1.51766  13.21  3.69  1.29  ...  0.57  8.22  0.00  0.0      1
4               5  1.51742  13.27  3.62  1.24  ...  0.55  8.07  0.00  0.0      1
..            ...      ...    ...   ...   ...  ...   ...   ...   ...  ...    ...
209           210  1.51623  14.14  0.00  2.88  ...  0.08  9.18  1.06  0.0      7
210           211  1.51685  14.92  0.00  1.99  ...  0.00  8.40  1.59  0.0      7
211           212  1.52065  14.36  0.00  2.02  ...  0.00  8.44  1.64  0.0      7
212           213  1.51651  14.38  0.00  1.94  ...  0.00  8.48  1.57  0.0      7
213           214  1.51711  14.23  0.00  2.08  ...  0.00  8.62  1.67  0.0      7

[214 rows x 11 columns]

Process finished with exit code 0
```

2) Update

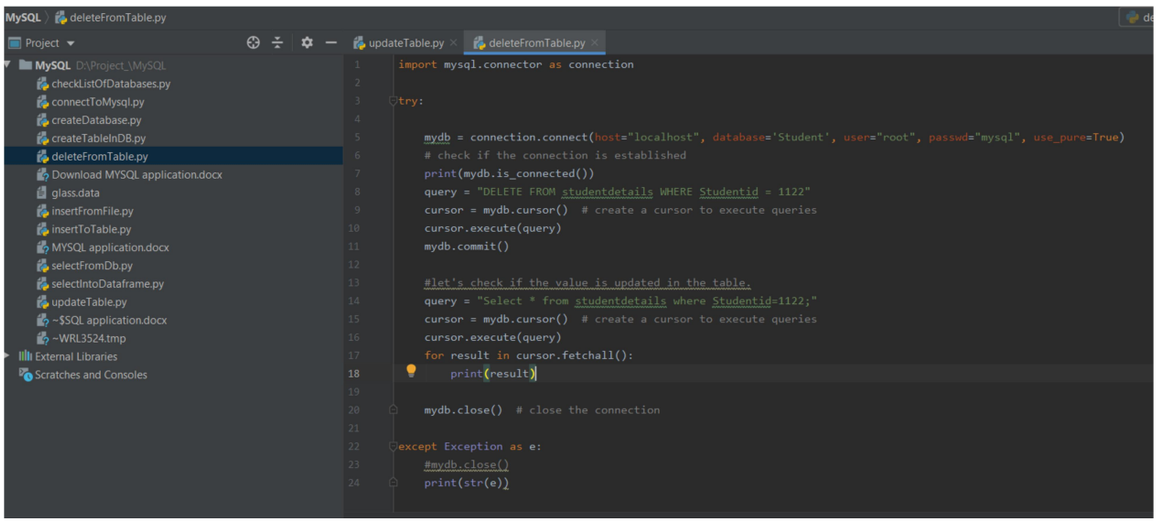




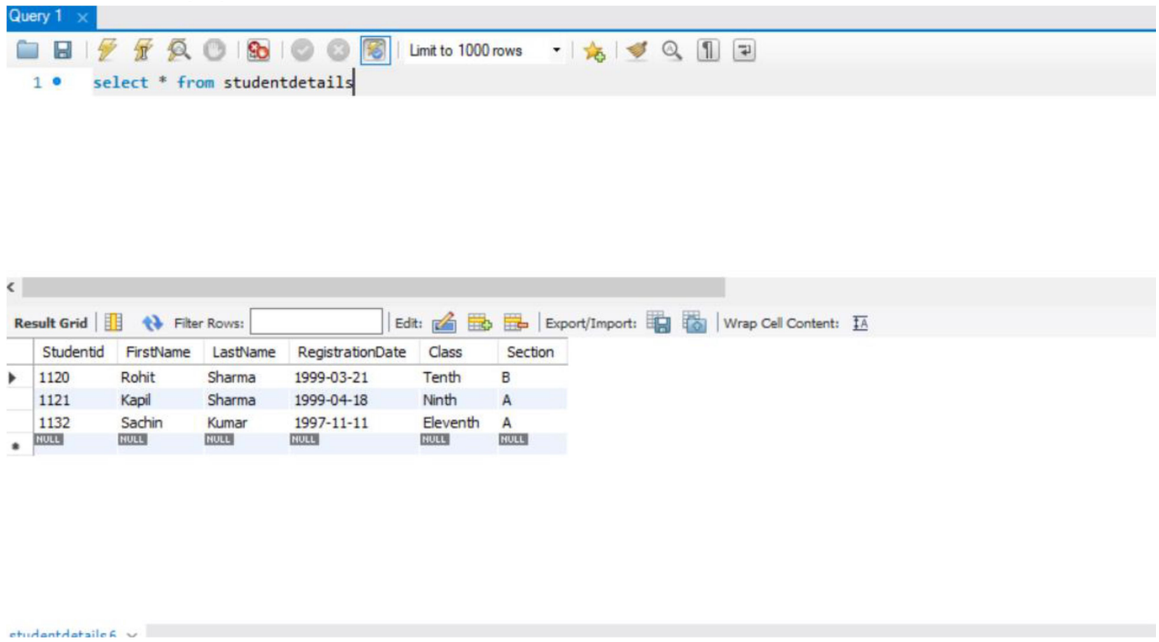
➢ **Let's check in MySql workbench:**



3) Delete statement



- **Let's check in MySql workbench:**



4) Group by, Order by