Python to MySQL Connection (Student Registration Project) - By Sunitha Mekala

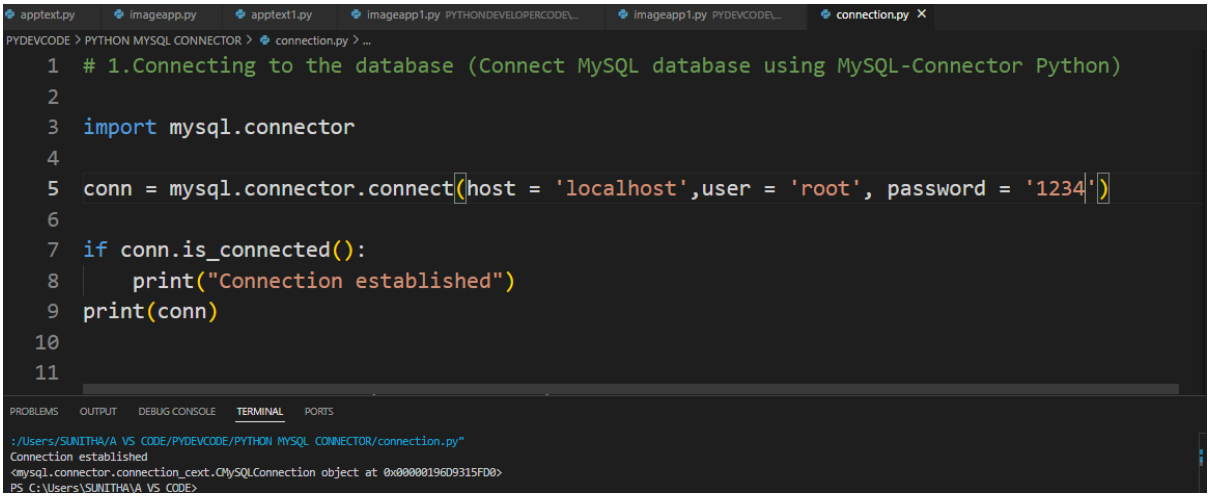
1) **How to install MySQL connector package in Python:**
mysql-connector method can be installed on Windows with the use of following command:

pip install mysql-connector-python

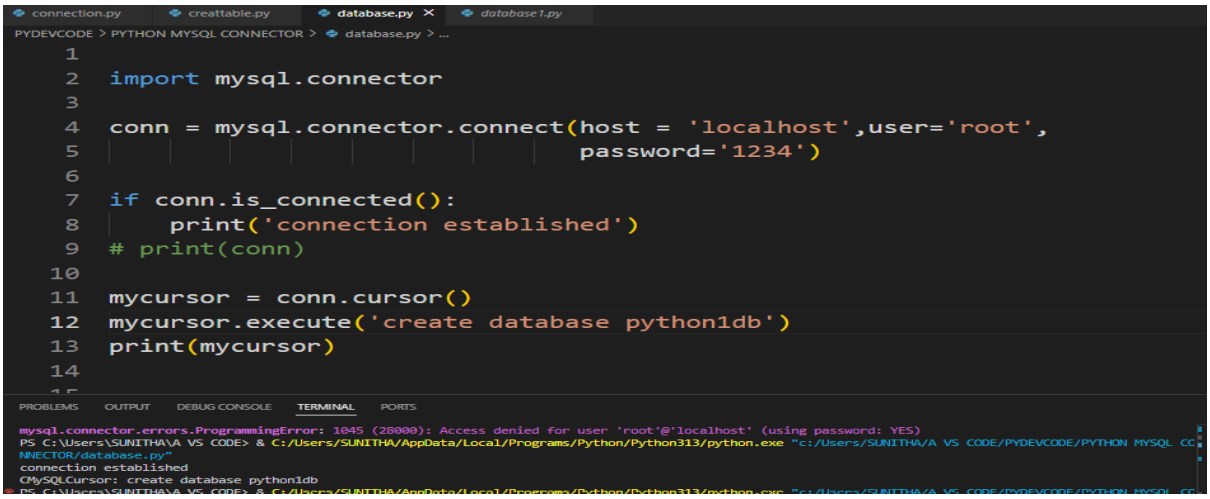
```
(base) C:\Users\A3MAX SOFTWARE TECH>pip install mysql-connector-python
Collecting mysql-connector-python
  Downloading mysql_connector_python-9.1.0-cp311-cp311-win_amd64.whl.metadata (6.2 kB)
Downloading mysql_connector_python-9.1.0-cp311-cp311-win_amd64.whl (16.1 MB)
   ---------------------------------------- 16.1/16.1 MB 6.4 MB/s eta 0:00:00
Installing collected packages: mysql-connector-python
Successfully installed mysql-connector-python-9.1.0

(base) C:\Users\A3MAX SOFTWARE TECH>pip install mysql-connector-python --upgrade
Requirement already satisfied: mysql-connector-python in c:\users\a3max software tech\anaconda3\lib\site-packages (9.1.0)
```

2) **Connect MySQL database using MySQL-Connector Python:**
Create a new file in VSCode **– connection.py**



3) Save the below code in a new file – **database.py** for Creating a database from python code – **python1db**

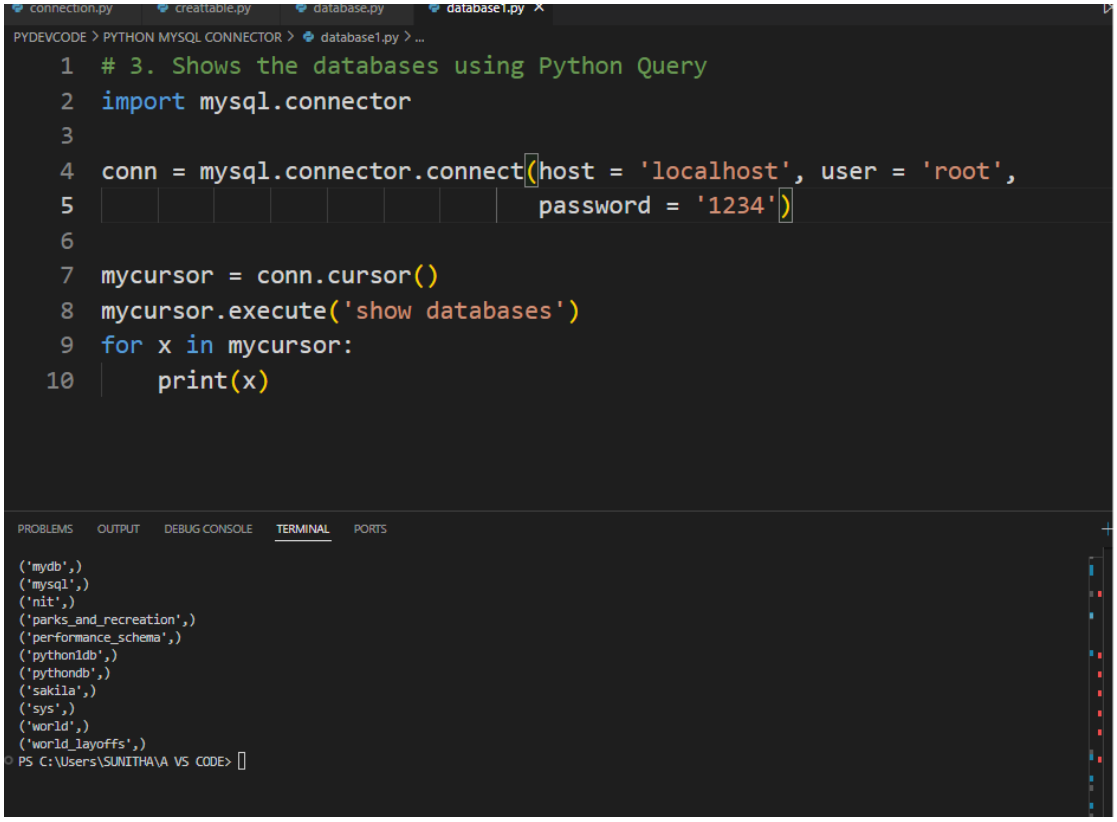


Manually check if our new database - **python1db** is created or not (Here its created)

```
mysql> show databases;
+----------------------+
| Database             |
+----------------------+
| information_schema   |
| mydb                 |
| mysql                |
| nit                  |
| parks_and_recreation |
| performance_schema   |
| python1db            |
| pythondb             |
| sakila               |
| sys                  |
| world                |
| world_layoffs        |
+----------------------+
12 rows in set (0.00 sec)

mysql>
```

4) Shows the list of databases present – creating a new file name – **database1.py**



5) **Creating tables and showing tables –** for this create a file named - **creattable.py**

a) **Creating the table student**

```
PYDEVCODE > PYTHON MYSQL CONNECTOR > creattable.py > ...
1  # 4. Creating tables/show tables
2  import mysql.connector
3
4  conn = mysql.connector.connect(host = 'localhost',user='root',
5                                 password='1234',database='python1db')
6
7  mycursor = conn.cursor()
8  mycursor.execute('create table student(name varchar(50),branch varchar(10), id int)')
9
```

b) **Once the table is created we can see the tables in the sql**

```
connection.py  database.py  insertvalue.py  creattable.py  database1.py
PYDEVCODE > PYTHON MYSQL CONNECTOR > creattable.py > ...
2  import mysql.connector
3
4  conn = mysql.connector.connect(host = 'localhost',user='root',
5                                 password='1234',database='python1db')
6
7  mycursor = conn.cursor()
8  #mycursor.execute('create table student(name varchar(50),
9  #                                       branch varchar(10), id int)')
10 mycursor.execute('Show tables')
11
12 for x in mycursor:
13     print(x)

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
L CONNECTOR/creattable.py"
('student',)
PS C:\Users\SUNITHA\A VS CODE>
```

6) Inserting the records into student table – file name – **insertvalue.py**

```
PYDEVCODE > PYTHON MYSQL CONNECTOR > insertvalue.py > ...
1  # 5. Inserting the records
2  import mysql.connector
3
4  conn = mysql.connector.connect(host = 'localhost', user = 'root',
5                                 password = '1234', database = 'python1db')
6
7  mycursor = conn.cursor()
8
9  sql = 'insert into student (name,branch,id) values(%s,%s,%s)'
10
11 val = [('john','cse','56'),('mike','IT','78'),('tyson','me','80')]
12
13 mycursor.executemany(sql,val)
14 conn.commit()
15 print(mycursor.rowcount,'record inserted')

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
alue.py"
3 record inserted
PS C:\Users\SUNITHA\A VS CODE>
```

7.) Now we are working on creating a student registration form
Save with **studregistration.py file**

C:\Users\SUNITHA\A VS CODE\PYDEVCODE\PYTHON MYSQL CONNECTOR\empregistration.py

```
        e4.insert(0, fee)

# Create the main window
root = tk.Tk()
root.geometry('600x500')
root.title("Student Registration System")

# Labels and Entry Fields
tk.Label(root, text="Student ID").grid(row=0, column=0, padx=10, pady=10)
tk.Label(root, text="Name").grid(row=1, column=0, padx=10, pady=10)
tk.Label(root, text="Course").grid(row=2, column=0, padx=10, pady=10)
tk.Label(root, text="Fee").grid(row=3, column=0, padx=10, pady=10)

e1 = tk.Entry(root)
e1.grid(row=0, column=1, padx=10, pady=10)
e1.config(state="disabled")  # Initially disabled, editable when selecting a student

e2 = tk.Entry(root)
```

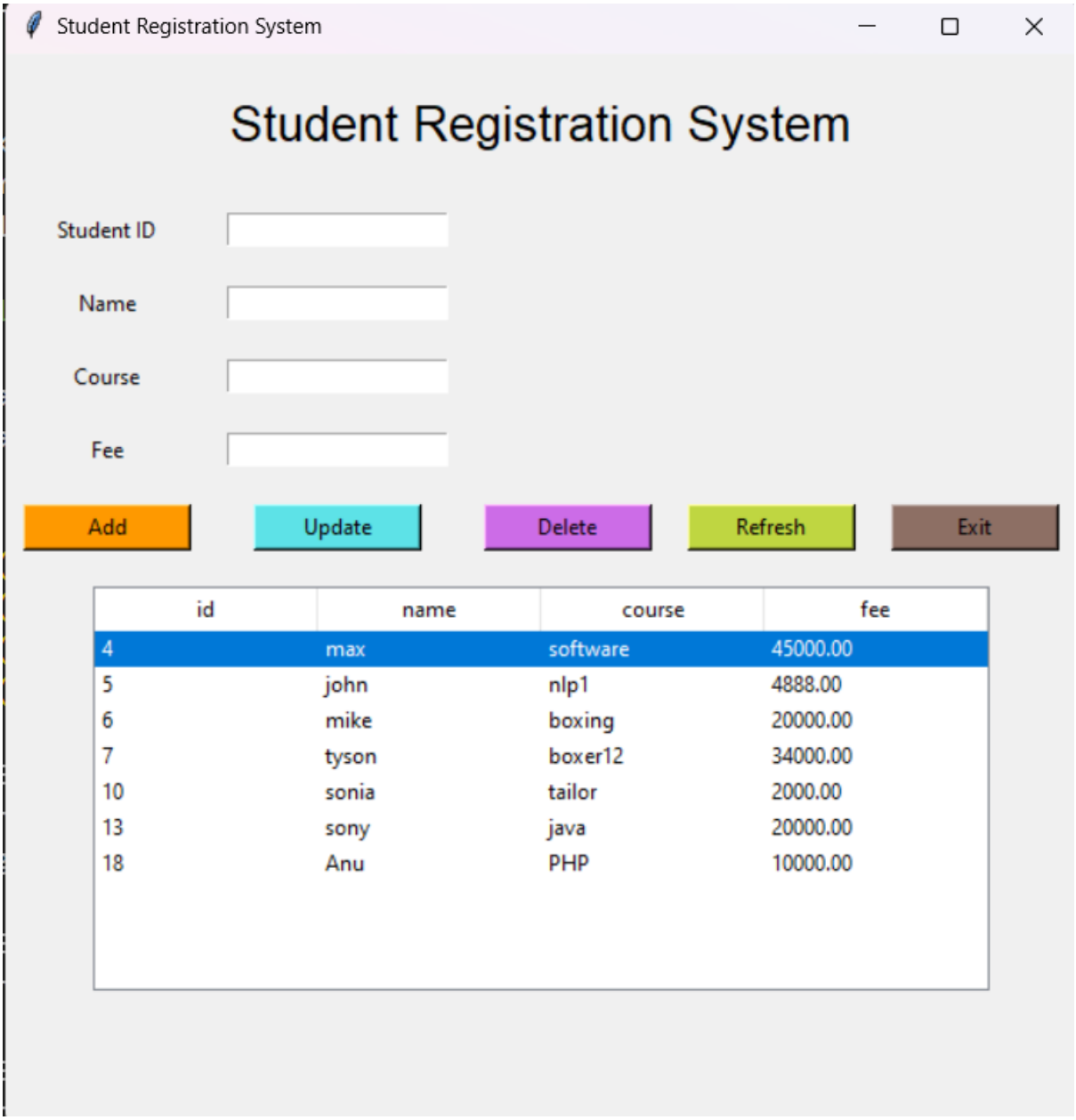
After that create the table and insert values,

```
mysql> create database webgui;
Query OK, 1 row affected (0.01 sec)
```

```
mysql> use webgui;
Database changed
mysql> CREATE TABLE registration (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     name VARCHAR(100) NOT NULL,
    ->     course VARCHAR(100) NOT NULL,
    ->     fee DECIMAL(10, 2) NOT NULL
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> desc registration;
+--------+---------------+------+-----+---------+----------------+
| Field  | Type          | Null | Key | Default | Extra          |
+--------+---------------+------+-----+---------+----------------+
| id     | int           | NO   | PRI | NULL    | auto_increment |
| name   | varchar(100)  | NO   |     | NULL    |                |
| course | varchar(100)  | NO   |     | NULL    |                |
| fee    | decimal(10,2) | NO   |     | NULL    |                |
+--------+---------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Once you write the backend code then below query will execute using **tkinter frontend**



Then, we can check in the webgui database to see if the records are inserted into the registration table.