**Experiment No. 6**

# Title: MySQL Database connectivity using Python

Batch: B1 Roll No: 1914049 Experiment No.:6

## Aim: CRUD Operations in Python with MySQL Database

Resources needed: Python IDE , MySQL Server

**Theory:**

MySQL Connector/python, a self-contained Python driver for communicating with MySQL servers. A connection with the MySQL server can be established using either the mysql.connector.connect () or the mysql.connector.MySQLConnection () class .

cnx = mysql.connector.connect(user='joe', database='test') cnx = mysql.connector.MySQLConnection(user='joe', database='test')

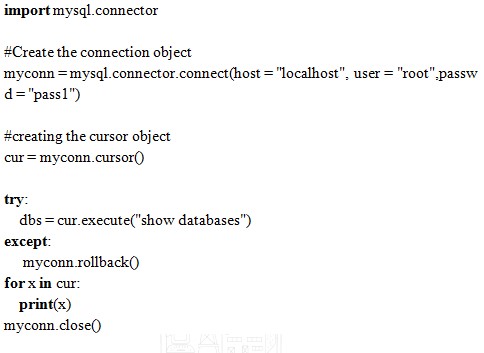
**Connection Arguments for Connector:**

* user (username\*) -The user name used to authenticate with the MySQL server.
* password (passwd\*)- The password to authenticate the user with the MySQL server.
* host - The host name or IP address of the MySQL server and default is 127.0.0.1
* database (db\*) -The database name to use when connecting with the MySQL server.
* port -The TCP/IP port of the MySQL server. Must be an integer and default is 3306

There are the following steps to connect a python application to the database.

1. Import mysql.connector module
2. Create the connection object.
3. Create the cursor object
4. Execute the query

# Example



**Activities:**

* Create a Employee Database to store all the Employee details.
* Read all the records from the Employee database.
* Give 15% hike in salary to all the employees who are having years of experience greater than 5.

Delete the records for the employees who are having age equals to 60 yrs.

Result: (script and output)

**CODE:**

#Importing Connector package

import mysql.connector

#established connection

database=mysql.connector.connect(host="localhost",user="root",password="1234")

#cursor() method create a cursor object

mycursor=database.cursor()

#Execute SQL Query to create a database

mycursor.execute("create database dbpython")

#Execute SQL Query to create a table into your database

mycursor.execute("create table employee(employee\_id INT,name VARCHAR(255), salary INT, years\_of\_experience INT, age INT)")

###INSERT RECORD

try:

   #Execute SQL Query to insert record

   mycursor.execute("insert into employee values(100,'Amitabh Bacchan',20000,20,65),(101,'Kareena Kapoor',50000,10,44),(102,'Varun Dhawan',80000,5,32),(104,'Alia Bhatt',100000,5,27),(105,'Vidya Balan',60000,14,50),(106,'Ananya Panday',10000,1,20)")

   # Commit is used for your changes in the database

   database.commit()

   print('Records inserted successfully...')

except:

   # rollback used for if any error

   database.rollback()

###DISPLAY RECORD

try:

   #Execute SQL Query to select all record

   mycursor.execute("select \* from employee")

   #fetches all the rows in a result set

   result=mycursor.fetchall()

   print('Employee Record:')

   for i in result:

      employee\_id=i[0]

      name=i[1]

      salary=i[2]

      years\_of\_experience = i[3]

      age = i[4]

      print(employee\_id,name,salary,years\_of\_experience,age)

except:

   print('Error:Unable to fetch data.')

###UPDATE RECORD

try:

   #Execute SQL Query to update record

   mycursor.execute("UPDATE employee SET salary=salary+0.15\*salary WHERE years\_of\_experience > 5")

   # Commit is used for your changes in the database

   database.commit()

   print('Record(s) updated successfully...')

except:

   # rollback used for if any error

   database.rollback()

###DELETE RECORD

try:

   #Execute SQL Query to detete a record

   mycursor.execute("DELETE FROM employee WHERE age>60")

    # Commit is used for your changes in the database

   database.commit()

   print('Record(s) deteted successfully...')

except:

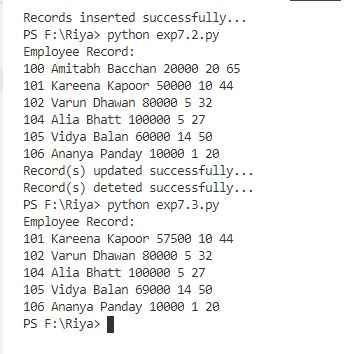
   # rollback used for if any error

   database.rollback()

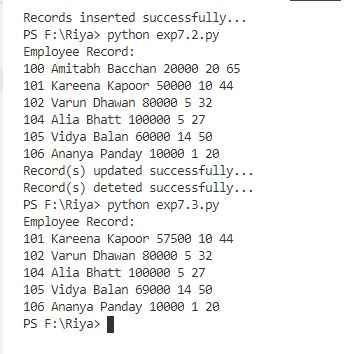
database.close()#Connection Close

**OUTPUT:**

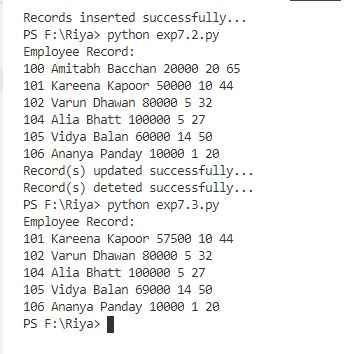
After inserting record:

****

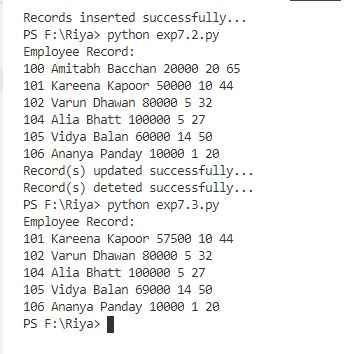
Reading (and displaying) the inserted records:

****

After updating the records(15% hike in salary to all the employees who are having years of experience greater than 5).

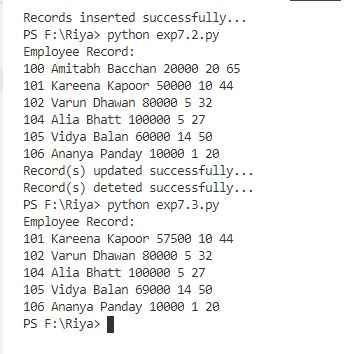
****

After deleting records for the employees who are having age equals to 60 yrs.

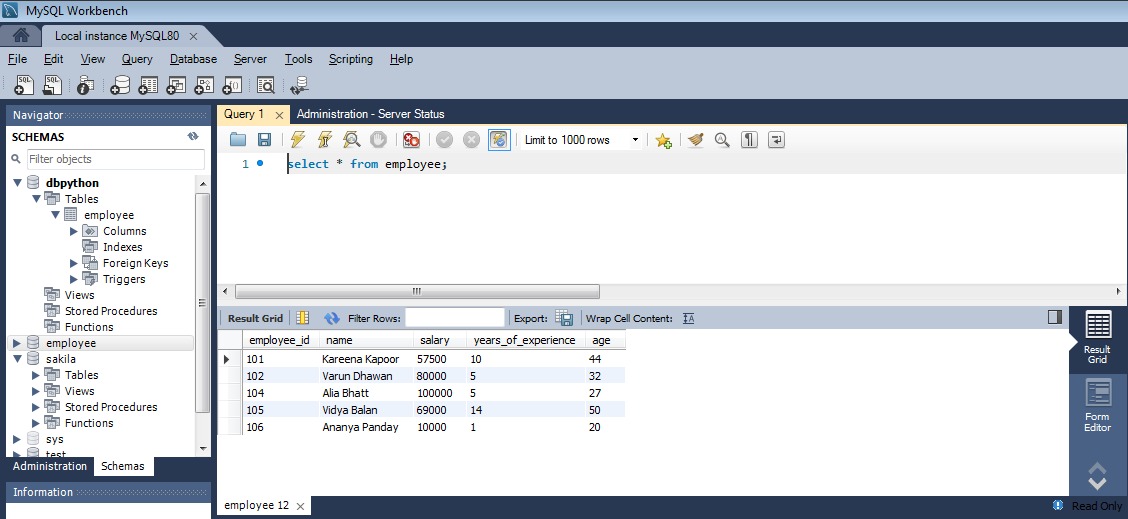
****

Updated records:

As you can see record ‘Amitabh Bacchan’ who was above 60 yrs has been deleted and all the employees having years of experience greater than 5 have been given a raised salary(raised by 15%).

****

Screenshot of table that has been created in the database:



Outcomes:

CO3: Demonstrate handling database with python.

Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

In this experiment, we learnt all the CRUD operations in Python with MySQL Database and implemented the same by creating a MySQL database and a table in that database. we further explored these operations by adding, deleting, updating and reading the records in the database using python.

References:

1. Yuxi Liu ; Python Machine Learning By Example: The easiest way to get into machine learning 1st Edition, Kindle Edition , Packt publishing Ltd , 1st edition 2017.
2. Martin C. Brown, The Complete Reference Paperback, Osborne, 2nd edition 2001
3. Frank Millstein, Data Analytics with Python: Data Analytics In Python Using Pandas, Copyright at Frank Millstein, 1st edition 2018
4. <https://www.python.org/dev/peps/pep-0249/>