**Experiment No. 6**

**Title: MySQL Database connectivity using Python**

Batch: B2 Roll No: 1914078 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')

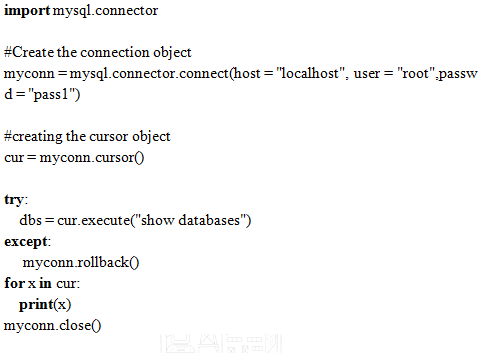
**Connection Arguments for Connector:**

* 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)

### Create new database

### import mysql.connector

### cnx = mysql.connector.connect(user='root', password='', host='127.0.0.1')

### #creating the cursor object

### cur = cnx.cursor()

### #to print all databases

### try:

### #creating a new database

### cur.execute("create database sampleDB")

### dbs = cur.execute("show databases")

### except:

### # Return to previous transaction state

### cnx.rollback()

### for x in cur:

### print(x)

### cnx.close()

### 

### 

### Create new table

### import mysql.connector

### cnx = mysql.connector.connect(user='root', password='', host='127.0.0.1', database='sampledb')

### #creating the cursor object

### cur = cnx.cursor()

### try:

### #creating a new table

### cur.execute("create table Devansh\_employees(name varchar(20) not null, id int(20) not null primary key, salary float not null, age int not null, experience int not null)")

### except:

### cnx.rollback()

### cnx.close()

### 

### 

### Insert multiple data into table

### import mysql.connector

### cnx = mysql.connector.connect(user='root', password='', host='127.0.0.1', database='sampledb')

### #creating the cursor object

### cur = cnx.cursor()

### sql = "insert into Devansh\_employees(name, id, salary, age, experience) values (%s, %s, %s, %s, %s)"

### #The row values are provided in the form of tuple

### val = []

### for i in range(10):

### innerVal = []

### innerVal.append(input("Enter name :"))

### innerVal.append(int(input("Enter id :")))

### innerVal.append(float(input("Enter salary :")))

### innerVal.append(int(input("Enter age :")))

### innerVal.append(int(input("Enter experience :")))

### val.append(tuple(innerVal))

### #to print all databases

### try:

### #Executing Insert query

### cur.executemany(sql,val)

### #commit the transaction

### cnx.commit()

### print(cur.rowcount,"records inserted!")

### except:

### cnx.rollback()

### cnx.close()

### 

### Selecting data

### import mysql.connector

### cnx = mysql.connector.connect(user='root', password='', host='127.0.0.1', database='sampledb')

### #creating the cursor object

### cur = cnx.cursor()

### try:

### cur.execute("select \* from Devansh\_employees")

### #fetching the rows from the cursor object

### result = cur.fetchall()

### #printing the result

### for x in result:

### print(x)

### except:

### cnx.rollback()

### cnx.close()

### 

### Updating data

### import mysql.connector

### cnx = mysql.connector.connect(user='root', password='', host='127.0.0.1', database='sampledb')

### #creating the cursor object

### cur = cnx.cursor()

### try:

### cur.execute("update Devansh\_employees set salary=salary\*1.15 where experience>5")

### cnx.commit()

### except:

### print("rollback")

### cnx.rollback()

### cnx.close()

### 

### Post update

### 

### Deleting data

### import mysql.connector

### cnx = mysql.connector.connect(user='root', password='', host='127.0.0.1', database='sampledb')

### #creating the cursor object

### cur = cnx.cursor()

### try:

### cur.execute("DELETE FROM Devansh\_employees where age>=60 ")

### cnx.commit()

### except:

### cnx.rollback()

### cnx.close()

### 

### Post delete

### 

### Outcomes: Demonstrate handling database with python

### Conclusion: We performed CRUD Operations in Python with MySQL Database

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/>