MYSQL Using Python

In [1]:

```
# pymysql

# mysql-connector-python

! pip install mysql-connector-python
```

```
Requirement already satisfied: mysql-connector-python in c:\users\pvashishtha\anaconda3\lib\site-packages (8.0.31)
Requirement already satisfied: protobuf<=3.20.1,>=3.11.0 in c:\users\pvashishtha\anaconda3\lib\site-packages (from
mysql-connector-python) (3.19.1)
```

In [2]:

```
! pip install pymysql
```

```
Requirement already satisfied: pymysql in c:\users\pvashishtha\anaconda3\lib\site-packages (1.0.2)
```

In [3]:

```

```

In [4]:

```
# 1. connect to database

# 2. create database cursor

# 3. CRUD- Create, Read, Update, Delete/Drop

# 4. Close connection
```

In [10]:

```
# connect to database

import mysql.connector

host='localhost'
user='root'
password=''
database='training'
port=3306

try:
    conn=mysql.connector.connect(host=host, user=user, password=password, database=database)
    print(conn)
except:
    print('Database Not Connected')
```

```
<mysql.connector.connection_cext.CMySQLConnection object at 0x0000026233537760>
```

In [11]:

```
# create cursor
try:
    cur=conn.cursor()
    print(cur)
except:
    print('Cursor not created')
```

```
CMySQLCursor: (Nothing executed yet)
```

In [38]:

```
# CRUD- INSERT

sql='insert into student values(101, "Prateek", "Delhi")'

cur.execute(sql)

conn.commit()
```

In [17]:

```
print(cur)
```

```
CMySQLCursor: insert into student values(103, "Deepika..
```

In [39]:

```
# CRUD- INSERT MANY

sql='insert into student values(%s, %s, %s)'
val=[
    (104, "Keerthana", "Hyderabad"),
    (105, "Ashritha", "Vizag"),
    (106, "Deepali", "Bhopal"),
    (107, "Ram", "Mumbai")
]

cur.executemany(sql, val)

conn.commit()
```

In [40]:

```
# CRUD- SELECT

sql='select * from student'
cur.execute(sql)
result=cur.fetchall() # fetchone()

for row in result:
    print('------------------------')
    print('Student ID: {0}\nName: {1}\nCity: {2}'.format(row[0],row[1],row[2]))
    print('------------------------')
```

```
------------------------
Student ID: 101
Name: Prateek
City: Delhi
------------------------
------------------------
Student ID: 102
Name: Vishal
City: Delhi
------------------------
------------------------
Student ID: 103
Name: Deepika
City: Chennai
------------------------
------------------------
Student ID: 104
Name: Keerthana
City: Hyderabad
------------------------
------------------------
Student ID: 105
Name: Ashritha
City: Vizag
------------------------
------------------------
Student ID: 106
Name: Deepali
City: Bhopal
------------------------
------------------------
Student ID: 107
Name: Ram
City: Mumbai
------------------------
```

In [41]:

```
import pandas as pd
sql='select * from student'
data=pd.read_sql(sql, conn)
data
```

```
C:\Users\pvashishtha\Anaconda3\lib\site-packages\pandas\io\sql.py:761: UserWarning: pandas only support SQLAlchemy
connectable(engine/connection) ordatabase string URI or sqlite3 DBAPI2 connectionother DBAPI2 objects are not test
ed, please consider using SQLAlchemy
  warnings.warn(
```

Out[41]:

	id	name	city
0	101	Prateek	Delhi
1	102	Vishal	Delhi
2	103	Deepika	Chennai
3	104	Keerthana	Hyderabad
4	105	Ashritha	Vizag
5	106	Deepali	Bhopal
6	107	Ram	Mumbai

In [42]:

```
# CRUD- UPDATE
sql='update student set city="Mumbai" where id=102'
cur.execute(sql)
conn.commit()
print('{0} rows updates in database'.format(cur.rowcount))
```

```
1 rows updates in database
```

In [43]:

```
# CRUD- DELETE
sql='delete from student where city="Delhi"'
cur.execute(sql)
print('{0} rows deleted from database'.format(cur.rowcount))
```

```
1 rows deleted from database
```

In [44]:

```
conn.commit()
```

In [45]:

```
# CRUD- DROP
sql='drop table student'
try:
    cur.execute(sql)
    print('Table Dropped!')
except:
    print('Drop Failed!')
```

```
Table Dropped!
```

In [46]:

```
# close connection

cur.close()

conn.close()
```

In [47]:

```
conn.disconnect()
```

Example

In [ ]:

```
# create a class that manages database.
# It has functionalities like:
# insert data, update, delete, select
# use company database and employee table
#
```

In [ ]:

```
# emp(eid int primary key, ename varchar(20), dept int, sal int);
```

In [83]:

```
import mysql.connector
class Employee:
    def __init__(self, host, user, password, database):
        try:
            self.conn=mysql.connector.connect(host=host, user=user, password=password,
                                              database=database)
            self.cur=self.conn.cursor()
            print('Database Connected, Start Using now!')
        except:
            print('Error Connecting DB')
        return None
    def addEmployee(self, eid, ename, dept, sal):
        sql='insert into emp values(%s,%s,%s,%s)'
        val=(eid, ename, dept, sal)
        self.cur.execute(sql, val)
        self.conn.commit()
        return self.cur.rowcount
    def updateEmployee(self, sql):
        self.cur.execute(sql)
        self.conn.commit()
        return self.cur.rowcount
    def displayAll(self):
        sql='select * from emp'
        self.cur.execute(sql)
        self.result=self.cur.fetchall()
        for self.row in self.result:
            print(self.row)
    def deleteEmployee(self, id):
        sql='delete from emp where eid={0}'.format(id)
        self.cur.execute(sql)
        self.conn.commit()
        return self.cur.rowcount
    def __del__(self):
        self.cur.close()
        self.conn.close()
        print('Database Closed')
```

In [87]:

```
emp=Employee('localhost','root','','company')
```

```
Database Connected, Start Using now!
```

In [89]:

```
emp.addEmployee(94, 'Kiran',10, 90000)
```

Out[89]:

```
1
```

In [90]:

```
emp.updateEmployee('update emp set sal=220000 where eid=90')
```

Out[90]:

```
0
```

In [91]:

```
emp.displayAll()
```

```
(90, 'Prateek', 30, 220000)
(91, 'Vishal', 20, 120000)
(92, 'Deepali', 10, 130000)
(93, 'Kiran', 10, 90000)
(94, 'Kiran', 10, 90000)
```

In [92]:

```
emp.deleteEmployee(91)
```

Out[92]:

```
1
```

In [94]:

```
del emp
```

...

In [ ]:

```

```