Example:

```
# deleting student from student table

import mysql.connector as mysql
def main():
    cn=mysql.connect(database="db6pm",user="root",password="root")
    c=cn.cursor()
    rno=int(input("Enter Rollno"))
    c.execute("delete from student where rollno=%s",params=[rno])
    if c.rowcount==1:
        print("Student Deleted...")
        cn.commit()
        cn.close()
    else:
        print("Invalid Rollno")

main()
```

Output:

```
 ========= RESTART: F:/python6pmaug/dbtest6.py ========
Enter Rollno102
Student Deleted...
```

```
mysql> select * from student;
+--------+--------+---------+
| rollno | name   | fees    |
+--------+--------+---------+
|    101 | naresh | 3000.00 |
|    103 | ramesh | 8000.00 |
+--------+--------+---------+
2 rows in set (0.00 sec)
```

Reading data from database table

To read data from database table, python program should send “SELECT” command. “select” command read data from database table and store

inside cursor. From cursor python program fetch data using methods provided by cursor object.

1. fetchone()
2. fetchmany()
3. fetchall()

fetchone()
this method fetch one row from cursor object. This method returns tuple. This method returns None, if there are no more rows to fetch from cursor.

Example:
```
# write a program to read data from student table
import mysql.connector as mysql
def main():
    cn=mysql.connect(database="db6pm",user="root",password="root")
    c=cn.cursor()
    c.execute("select * from student")
    #r1=c.fetchone()
    #print(r1)
    #r2=c.fetchone()
    #print(r2)
    #r3=c.fetchone()
    #print(r3)

    while True:
        r=c.fetchone()
        if r==None:
            break
        print(r)

main()
```
Output:
```
========= RESTART: F:/python6pmaug/dbtest7.py =========
(101, 'naresh', 3000.0)
(103, 'ramesh', 8000.0)
```

Example:
```
# Login/Signin
```

```
import mysql.connector as mysql
def main():
    cn=mysql.connect(database="db6pm",user="root",password="root")
    c=cn.cursor()
    user=input("UserName :") # abc
    pwd=input("Password :")
    c.execute("select * from user_register where uname=%s and
pwd=%s",params=[user,pwd])
    row=c.fetchone()
    if row==None:
        print("invalid username or password")
    else:
        print(f'{user} welcome')

main()
```

Output:

```
========= RESTART: F:/python6pmaug/dbtest8.py =========
UserName :ramesh
Password :r123
invalid username or password

========= RESTART: F:/python6pmaug/dbtest8.py =========
UserName :ramesh
Password :ram123
ramesh welcome
```

fetchmany(n)

This method fetches n rows from cursor object. This method return list of tuples. This returns empty list if there are no rows to fetch from cursor object.

Example:

```
# Example of fetchmany

import mysql.connector as mysql
def main():
    cn=mysql.connect(database="db6pm",user="root",password="root")
    c=cn.cursor()
```

```
    c.execute("select * from student")
    rows=c.fetchmany(2)
    print(rows)
    rows=c.fetchmany(2)
    print(rows)
    rows=c.fetchmany(2)
    print(rows)


main()
```

Output:

```
========= RESTART: F:/python6pmaug/dbtest9.py =========
[(101, 'naresh', 3000.0), (103, 'ramesh', 8000.0)]
[(102, 'kishore', 1000.0), (104, 'rajesh', 5000.0)]
[]
```

fetchall()

This method fetches all rows from cursor object.

```
# Example of fetchall

import mysql.connector as mysql
def main():
    cn=mysql.connect(database="db6pm",user="root",password="root")
    c=cn.cursor()
    c.execute("select * from student")
    rows=c.fetchall()
    print(rows)
    tot=0
    for row in rows:
        print(f'{row[0]}\t{row[1]}\t{row[2]}')
        tot=tot+row[2]
    print("Total Fees ",tot)

main()
```

Output:

```
[(101, 'naresh', 3000.0), (103, 'ramesh', 8000.0), (102, 'kishore', 1000.0),
(104, 'rajesh', 5000.0)]
101   naresh      3000.0
103   ramesh      8000.0
102   kishore     1000.0
104   rajesh      5000.0
Total Fees  17000.0
```

MySQLCursor.executemany() Method

Syntax:

```
cursor.executemany(operation, seq_of_params)
```

This method prepares a database operation (query or command) and executes it against all parameter sequences or mappings found in the sequence seq_of_params.

Example:

```
# Example of executemany
import mysql.connector as mysql
def main():
    cn=mysql.connect(database="db6pm",user="root",password="root")
    c=cn.cursor()
    values=[(105,"kiran",2000),
            (106,"raman",6000)]
    c.executemany("insert into student values(%s,%s,%s)",values)
    print(c.rowcount)
    cn.commit()
    cn.close()

main()
```

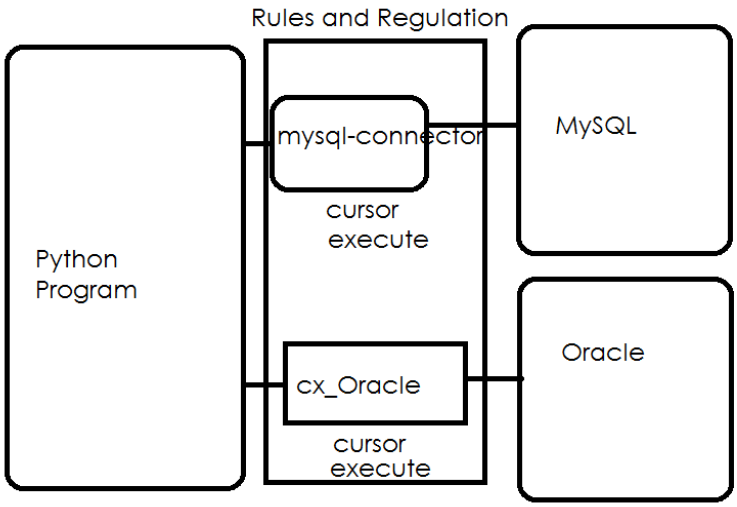
Output:

```
========= RESTART: F:/python6pmaug/dbtest10.py ========
2
```

Oracle Database

Install Oracle Database Software

www.oracle.com



In order to communicate with oracle database, install the following package library.

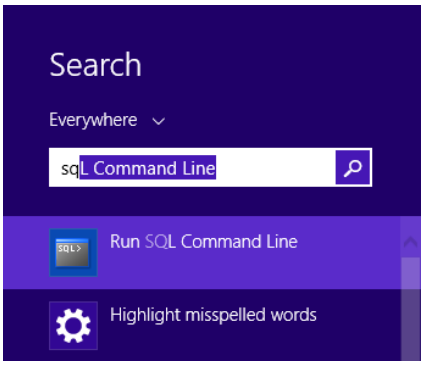
cx_Oracle

```
C:\Users\nit>pip install cx_Oracle
Collecting cx_Oracle
  Using cached cx_Oracle-8.3.0-cp310-cp310-win_amd64.whl (213 kB)
Installing collected packages: cx_Oracle
Successfully installed cx_Oracle-8.3.0

[notice] A new release of pip available: 22.2.2 -> 22.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\Users\nit>
```

```
>>> import cx_Oracle
>>>
```



Run SQL Command Line is tool provided by oracle to work with oracle database.

Establishing connection to oracle database
cx_Oracle.connect("username/password@hostname(OR)ipaddress")

```
>>> import cx_Oracle
>>> cn=cx_Oracle.connect("system/manager@localhost")
>>> print(cn)
<cx_Oracle.Connection to system@localhost>
>>> c=cn.cursor()
# write a program to create table

import cx_Oracle
def main():
    cn=cx_Oracle.connect("system/manager@localhost")
    c=cn.cursor()
    c.execute('''create table student(rollno number(5),
name varchar2(20),fee number(10,2))''')
    print("table created")

main()
```

Output:

```
======== RESTART: F:/python6pmaug/dbtest11.py ========
table created
```

```
SQL> connect
Enter user-name: system
Enter password:
Connected.
SQL> describe student
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------

 ROLLNO                                             NUMBER(5)
 NAME                                               VARCHAR2(20)
 FEE                                                NUMBER(10,2)
```

Example:

```
# program to insert data into student table

import cx_Oracle
def main():
    cn=cx_Oracle.connect("system/manager@localhost")
    c=cn.cursor()
    while True:
        rno=int(input("Rollno:"))
        n=input("Name :")
        f=float(input("Fee :"))
        c.execute("insert into student values(:1,:2,:3)",(rno,n,f))
        a=c.rowcount
        print(f'{a} rows inserted')
        ans=input("Add another student?")
        if ans=="no":
            break

    cn.commit()
    cn.close()
main()
```

Output:

```
======== RESTART: F:/python6pmaug/dbtest12.py ========
Rollno:1
Name :naresh
Fee :4000
1 rows inserted
```

Add another student?yes
Rollno:2
Name :suresh
Fee :3000
1 rows inserted
Add another student?no

```
SQL> select * from student;

    ROLLNO NAME                      FEE
---------- -------------------- ----------
         1 naresh                   4000
         2 suresh                   3000
```

Example:

```
# program to read data from student table of oracle database

import cx_Oracle
def main():
    cn=cx_Oracle.connect("system/manager@localhost")
    c=cn.cursor()
    c.execute("select * from student")
    rows=c.fetchall()
    for row in rows:
        print(row)
    cn.close()

main()
```

Output:

```
========= RESTART: F:/python6pmaug/dbtest13.py ========
(1, 'naresh', 4000.0)
(2, 'suresh', 3000.0)
```