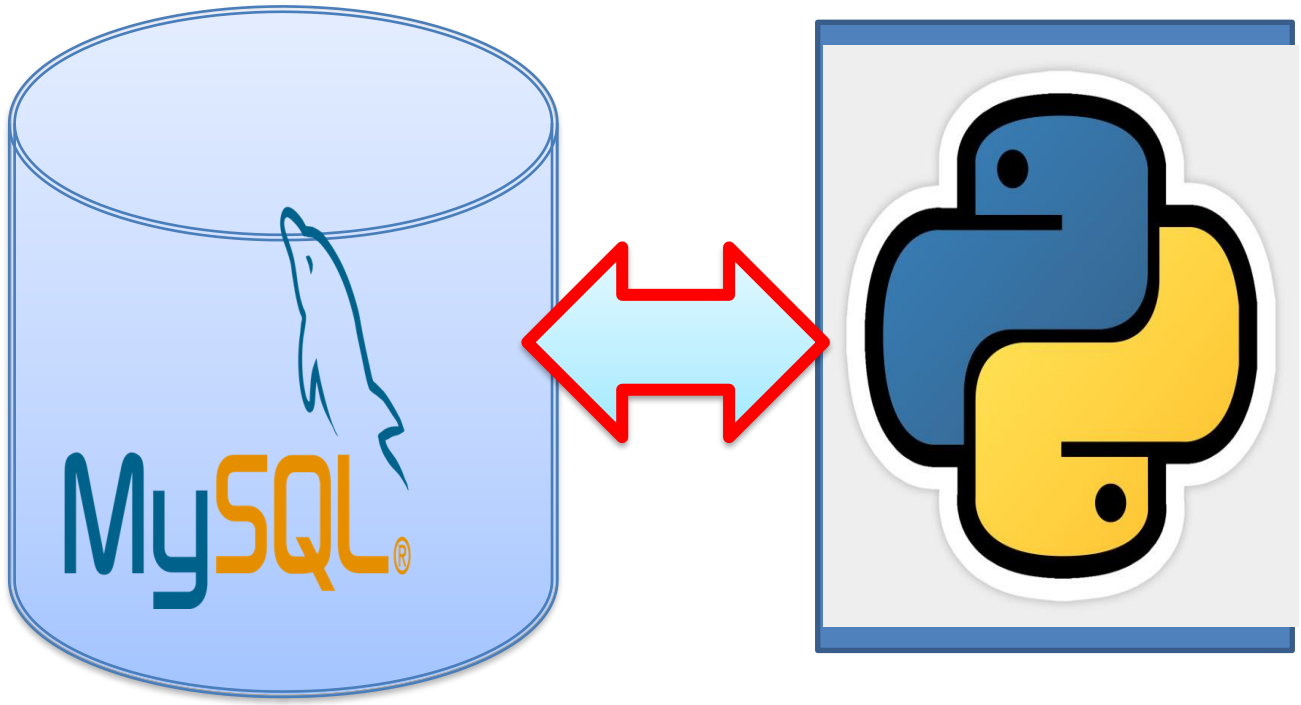
Python MySQL Connectivity



Introduction

MySQL and MariaDB are the most popular Opensource Relational Database Management Systems:

- MySQL and MariaDB are almost similar; they are written in C and C++
- Faster than most commercial databases so it can work well even with large datasets.
- Supported most operating systems and languages including Python, PHP, PERL, C, C++, and JAVA.
- Uses a standard form of well-known SQL language.
- MySQL is used for many small and big businesses.
- MariaDB is purely open-source database but MySQL has enterprise version managed by Oracle

Terminologies

Database schema - structure or format of a database, model described in a formal language supported by database management system

Database contains one or more tables manipulated using database management system.

Relation (or table) - contains tuples and attributes

Tuple (or row) - a set of fields that generally represents an "object" like a person or a music track

Attribute (also column or field) - one of possibly many elements of data corresponding to the object represented by the row

Database Schema

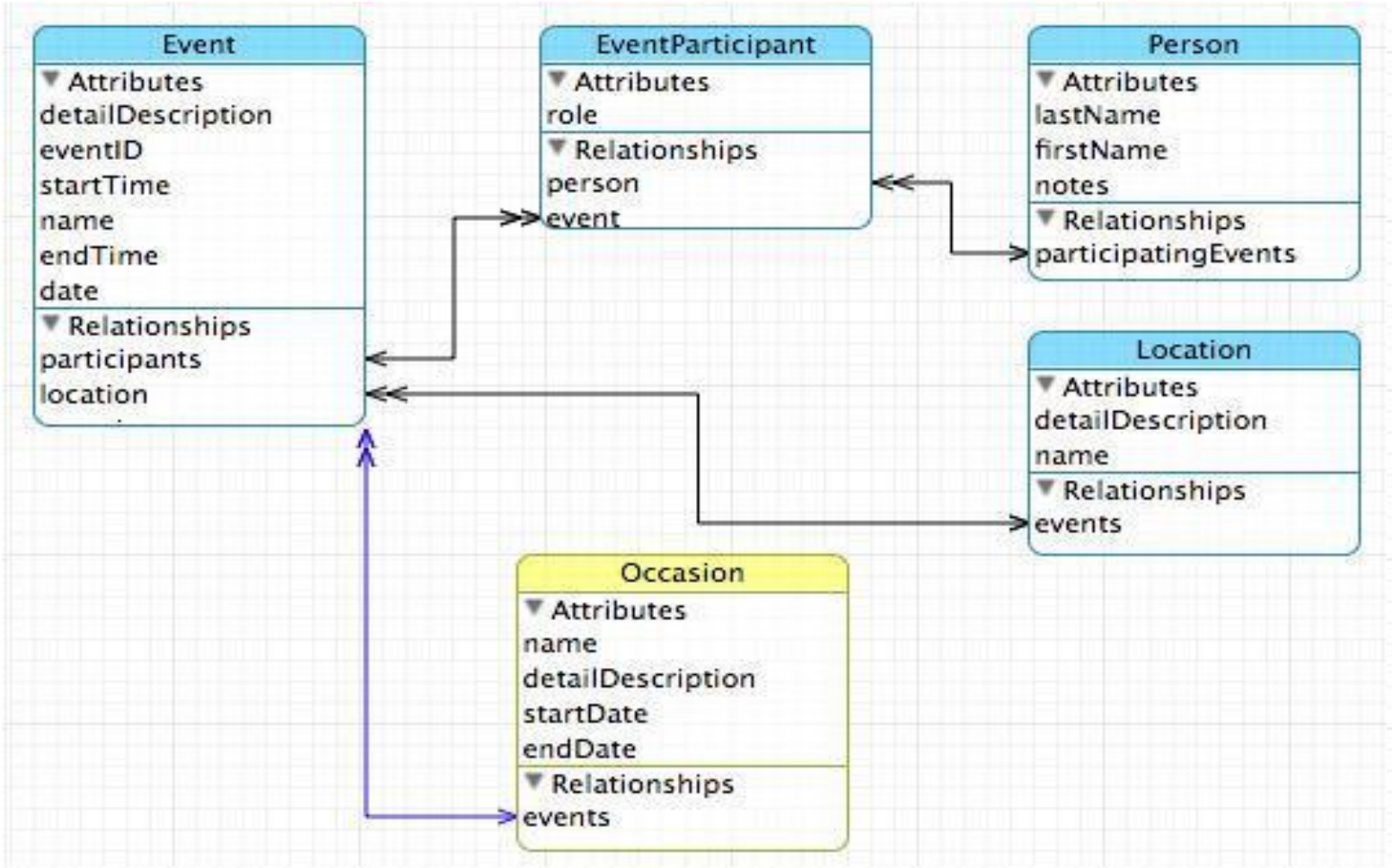
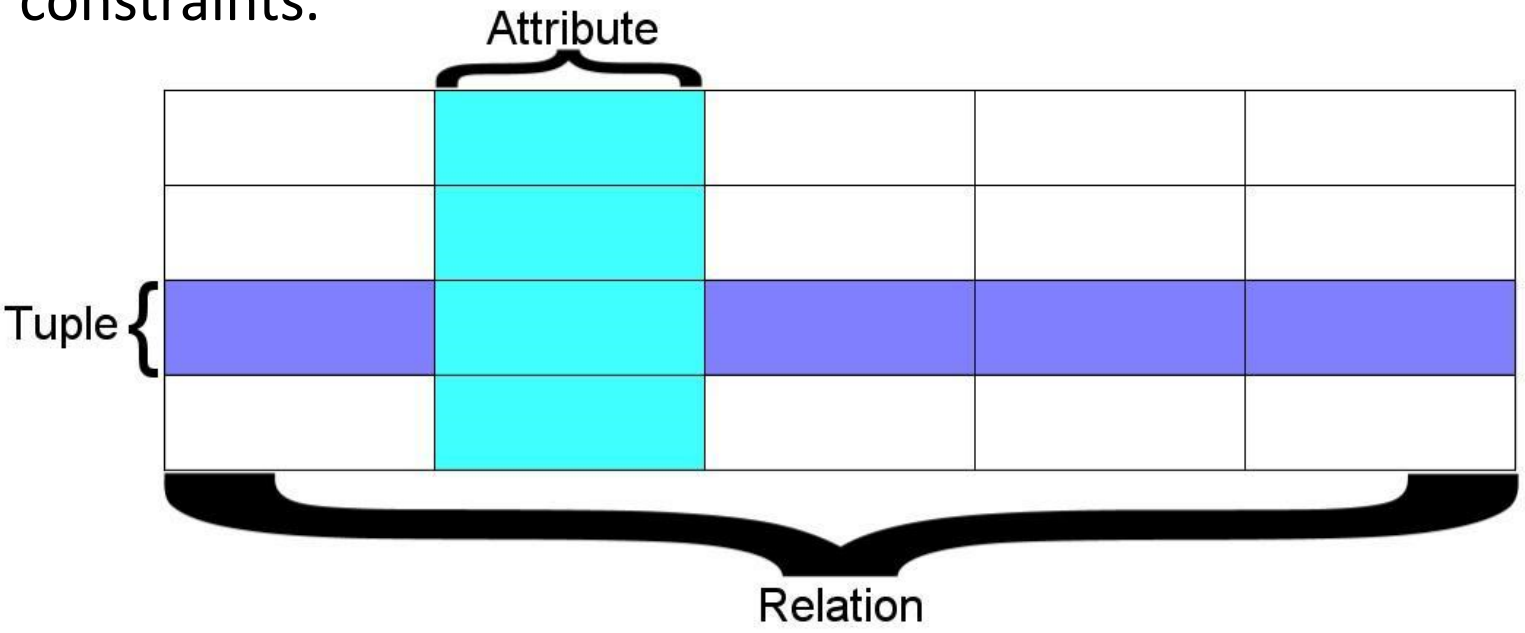
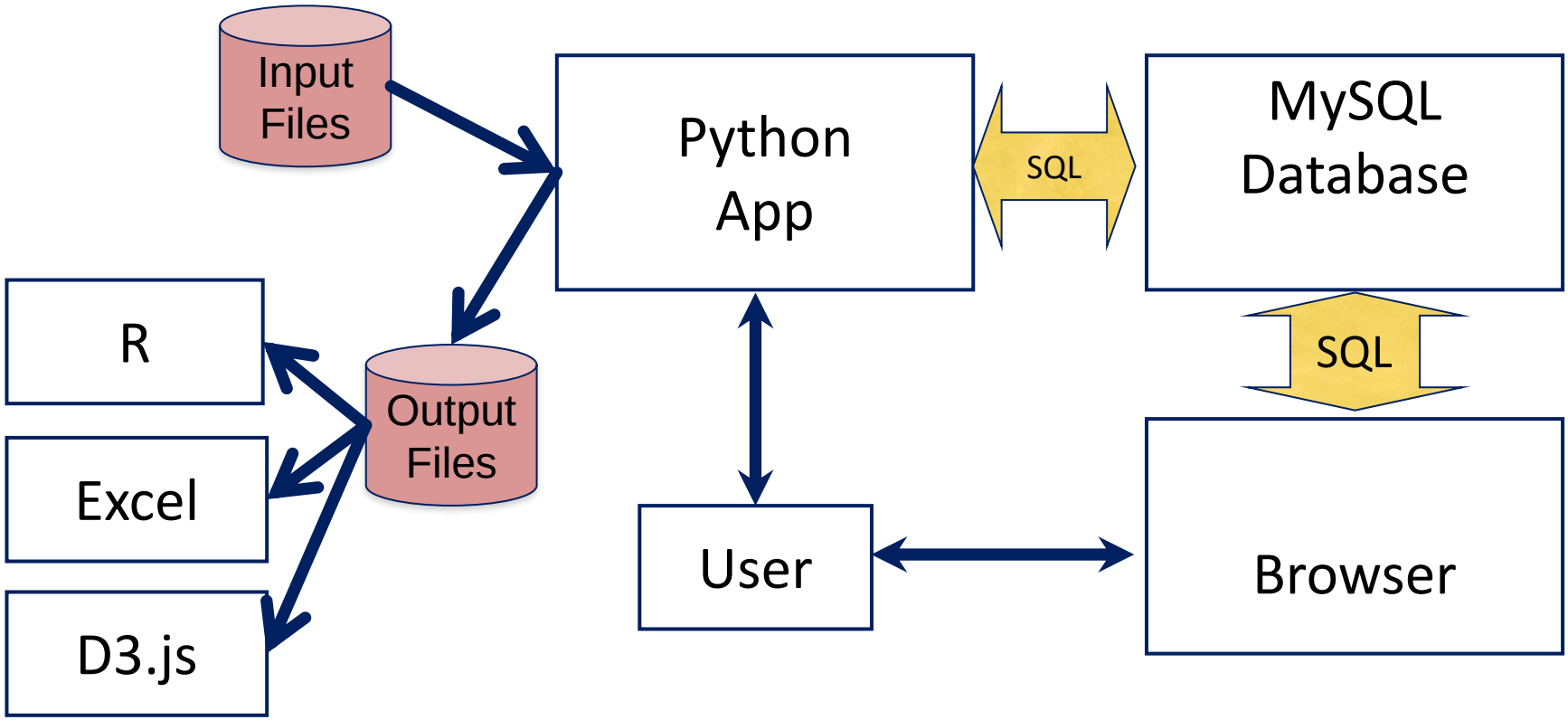


Table schema

Relation is defined as a set of tuples that have the same attributes. A relation is usually described as a table, which is organized into rows and columns. Data referenced by an attribute are in the same domain and conform to the same constraints.



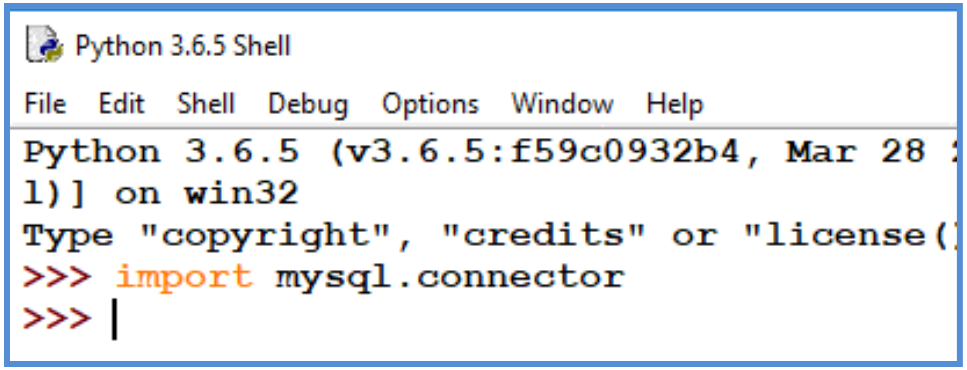
Python App


Input Files
Python App
SQL
MySQL Database
R
Output Files
SQL
Excel
User
Browser
D3.js

MySQL Python Connector

MySQL Python Connector is used to access the MySQL database from Python, you need a database driver. MySQL Connector/Python is a standardized database driver provided by MySQL.

1. Open Command window
2. type pip install mysql-connector
3. Open Python IDLE
4. Type import mysql



Connect Python App to MySQL

You have to install mysql.connector for Python that connects Python Scripts to the MySQL database. To use mysql.connector follow these steps:

1. Download mysql.connector.
2. Import mysql.connector module into python code
3. Create the database connection object.
4. Create the cursor object
5. Execute the query

Connect Python App to MySQL

Create connection using connect() method and pass connection details like HostName, username, and password. The method returns the connection object.

ConnObject = mysql.connector.connect
(host = <hostname>, user = <username> , passwd = <password>)

mysqlconn1.py - C:/Users/Sangeeta Chauhan/AppData/Local/Programs/Python/Python36-32/mysqlconn1.py (3.6.5)

File Edit Format Run Options Window Help

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
          (host = "localhost", user = "root",passwd = "mysql")

print(conn) #printing the connection object
```

```
 RESTART: C:/Users/Sangeeta Chauhan/AppData/Local/Programs/Python/Python36-32/m
sqlconn1.py
<mysql.connector.connection.MySQLConnection object at 0x03C66D50>
>>>
```

Connect Python App to MySQL

- Python Database Application Programming Interface specifies Connection objects and methods for manipulating any database
- The Cursor object, created by Connection object, manipulates and retrieves data, We need cursor to execute our MySQL Queries
- **Cursors** are created by the **connection.cursor()** method : they are bound to the connection for the entire lifetime of the program and all the commands are executed in the context of the database session wrapped by the connection.
- Three methods for fetching rows of a query result set – fetchone, fetchmany and fetchall

Display MySQL databases

File Edit Format Run Options Window Help

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
         (host = "localhost", user = "root",passwd = "mysql")

#creating the cursor object
MyCur = conn.cursor()
MyCur.execute("show databases")  # With the help of CUrsor executing Query

for x in MyCur: #Fetching database from MyCur one by one
    print(x)
conn.close()
```

Python Create database

File Edit Format Run Options Window Help

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
         (host = "localhost", user = "root",passwd = "mysql")

#creating the cursor object
MyCur = conn.cursor()
MyCur.execute("create database TelephoneDir")# Creating Database TelephoneDir
MyCur.execute("show databases")  # With the help of CUrsor executing Query

for x in MyCur: #Fetching database from MyCur one by one
    print(x)
conn.close()
```

Python create tables

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
         (host = "localhost", user = "root",passwd = "mysql")

#creating the cursor object
MyCur = conn.cursor()
MyCur.execute("use telephonedir")
MyCur.execute("create table customer (cust_id int(5),cust_name varchar(40)\
            ,tele_no varchar(15),date_of_reg date)")# Creating Table Customer
MyCur.execute("show tables")  # With the help of CUrsor executing Query

for x in MyCur: #Fetching database from MyCur one by one
    print(x)
conn.close()
```

Python view table structure

File Edit Format Run Options Window Help

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
          (host = "localhost", user = "root",\
           passwd = "mysql",database="telephonedir")

#creating the cursor object
MyCur = conn.cursor()
MyCur.execute("desc customer")  # With the help of CUrsor executing Query

for x in MyCur: #Fetching result from MyCur one by one
    print(x)
conn.close()
```

Python insert data

File Edit Format Run Options Window Help

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
         (host = "localhost", user = "root",passwd = "mysql")

#creating the cursor object
MyCur = conn.cursor()
MyCur.execute("use telephonedir")


MyCur.execute('insert into customer values("T101","Mr. C.P. Gupta",9812343134,"2018/01/21")')

MyCur.execute("Select * from customer")  # With the help of CUrsor executing Query

for x in MyCur: #Fetching database from MyCur one by one
    print(x)
conn.close()
```

Python insert data

mysqlconn_insert.py - C:/Users/Sangeeta Chauhan/AppData/Local/Programs/Python/Python36-32/mysqlconn_insert.py (3.6.5)

File Edit Format Run Options Window Help

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
          (host = "localhost",\
           user = "root",\
           passwd = "mysql",\
           database="telephonedir")

#creating the cursor object
MyCur = conn.cursor()


Cid=int(input("Enter Customer Id"))
name=input("Enter Name of Customer")
t_num=input("Enter telephone number")
dt=input("Enter date of registration (year/mm/dd)")

qry='insert into customer (cust_id,cust_name,tele_no,date_of_reg)\
                values(%s,%s,%s,%s)'
val=(Cid,name,t_num,dt)

MyCur.execute(qry,val)
conn.commit()
MyCur.execute("Select * from customer")  # With the help of CUrsor executing Query

for x in MyCur: #Fetching result from MyCur one by one
    print(x)
conn.close()
```

With **INSERT INTO** statement, we can mention the format specifier (%s) in place of values.

Python insert multiple records

File Edit Format Run Options Window Help

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
          (host = "localhost",\
           user = "root",\
           passwd = "mysql",\
           database="telephonedir")

#creating the cursor object
MyCur = conn.cursor()

#inserting the values into the table
qry = "insert into customer(cust_id,cust_name,tele_no,date_of_reg) values (%s, %s, %s, %s)"
val = [(1006, 'Jayant', '0751567887', '2018/04/01'),\
       (1007, 'Mayank Shinde', '07512341997', '2018/04/27'),\
       (1008, 'Namrata', '0751556787', '2017/03/05')]

MyCur.executemany(qry,val)

conn.commit()
MyCur.execute("Select * from customer")  # With the help of CUrsor executing Query

for x in MyCur: #Fetching result from MyCur one by one
    print(x)
conn.close()
```

Note that here we have to use executemany() instead of execute()

Python Update records

File Edit Format Run Options Window Help

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
          (host = "localhost",\
           user = "root",\
           passwd = "mysql",\
           database="telephonedir")

#creating the cursor object
MyCur = conn.cursor()


Cid=int(input("Enter Id of Customer of which name is to be modified "))
name=input("Enter Name of Customer")


qry=("update customer set cust_name=%s where Cust_id=%s")
data=(name,Cid)


MyCur.execute(qry,data)
conn.commit()
MyCur.execute("Select * from customer")  # With the help of CUrsor executing Que

for x in MyCur: #Fetching result from MyCur one by one
    print(x)
conn.close()
```

Python delete records

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
          (host = "localhost",\
           user = "root",\
           passwd = "mysql",\
           database="telephonedir")

#creating the cursor object
MyCur = conn.cursor()


Cid=int(input("Enter Id of Customer of which record is to be deleted "))


qry=("delete from customer where cust_id=%s ")
data=(Cid,)


MyCur.execute(qry,data)
print("Record Deleted")

conn.commit()
MyCur.execute("Select * from customer")  # With the help of CUrsor executing Que

for x in MyCur: #Fetching result from MyCur one by one
    print(x)
conn.close()
```

Python count records

Cursor.rowcount displays number of records fetched after running. You can Query.run this command after executing any query.

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
           (host = "localhost",\
            user = "root",\
            passwd = "mysql",\
            database="telephonedir")

#creating the cursor object
MyCur = conn.cursor()


qry='Select * from customer'
MyCur.execute(qry)

for x in MyCur: #Fetching result from MyCur one by one
    print(x)
print("Total ",MyCur.rowcount," records present ")
conn.commit()
conn.close()
```

Python display records

To display all records from a table, use fetchall() method as illustrated below:

```
import mysql.connector

#Create the connection object
conn = mysql.connector.connect\
          (host = "localhost",user = "root",\
           passwd = "mysql",database="telephonedir")

#creating the cursor object
MyCur = conn.cursor()


qry='Select * from customer'
MyCur.execute(qry)
for row in MyCur.fetchall():
         print ("row:", row)
print ("rowcount:", MyCur.rowcount)


conn.close()
```

Python display records

- cursor.fetchall() fetches all the rows of a query result. The method returns all the rows as a list of tuples. An empty list is returned if there is no record to fetch.
- cursor.fetchmany(size) returns the number of rows specified by size argument. When called repeatedly this method fetches the next set of rows of a query result and returns a list of tuples. If no more rows are available, it returns an empty list.
- cursor.fetchone() method returns a single record or None if no more rows are available.

Python error handling

For error handling in python, we need to import two more module:

```
>>>from mysql.connector import Error
>>>from mysql.connector import errorcode
```

MySQL connector Error object is used to show us an error when we failed to connect Databases or if any other database error occurred while working with the database.

Python error handling

```
import mysql.connector
from mysql.connector import Error
from mysql.connector import errorcode

def display():
      try:
            cnx = mysql.connector.connect(user='root', password='mysql',\
            host='localhost',database='Telephonedir')
            Cursor = cnx.cursor()
            query = ("SELECT * FROM Customer")
            Cursor.execute(query)
            for (cid,cname,tel,dt) in Cursor:
                  print("=================================================")
                  print("Customer Id   : ",cid)
                  print("Customer Name : ",cname)
                  print("Contact Number: ",tel)
                  print("Registered On : ",dt)
                  print("=================================================")
            Cursor.close()
            cnx.close()
            print("You have done it!!!!!!")
      except mysql.connector.Error as err:
            if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
                  print("Something is wrong with your user name or password")
            elif err.errno == errorcode.ER_BAD_DB_ERROR:
                  print("Database does not exist")
            else:
                  print(err)
      else:
            cnx.close()

display()
```