**Dear Learner,**

**In this section of the course, you will be able to learn about how to develop database applications using  Python .**

**Here, you will learn about Python MySQL connector and how to connect to MySQL database and perform basic database operations such as SELECT,INSERT,UPDATE and DELETE.**

**Request you to go through the learning contents provided to have an understanding of the concepts.**

**At the end of the course, complete the practice assignments provided at the end of the module to have a better understanding of how to implement the concepts learnt.**

**Disclaimer: This content is proprietary to TCS, this may contain confidential or   privileged information. Any dissemination, use, review, distribution, printing or  copying of this content is strictly prohibited.**

**Happy Learning !!**

**Python MySQL Connector:**

Python programming language can be used to develop applications having database connectivity.

In Python, a database driver called "**MySQL Connector/Python"**is used to access the MySQL database server .

MySQL Connector/Python is an API implemented using pure Python.

We do not need to install any MySQL client library or any Python modules except the standard library for this driver.

How to Verify MySQL Connector/Python installation:

Before using , you need to verify whether the connector has been installed in your system or not.

Following are the steps you need to perform to verify the installation:

1. Open Python command line
2. Type the following command

>>> import mysql.connector

>>> mysql.connector.connect(host='localhost',database='mysql',user='root',password='your pass')

If the connector is installed in the system, you will get the following output after execution of the above commands.

**How to establish Connection:**

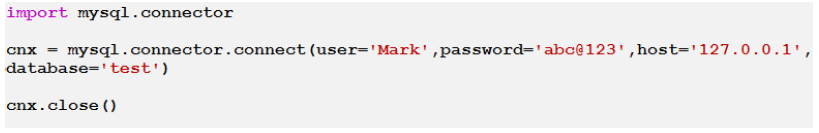
To establish the MySQL database connectivity following steps needs to be followed:

1. First, import the  mysql.connector and Error objects from the MySQL Connector/Python package.

2. Next, use the mysql.connector.connect() function of Connector/Python. The connect function creates a MySQL connection and returns

    a  connection object. The connect() function accepts four parameters: host, database, user and password as shown in the below

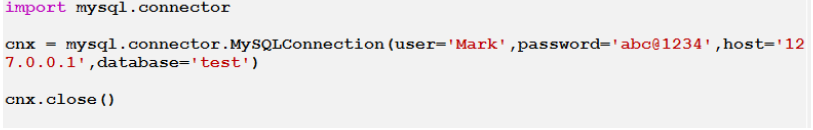
     code snippet.



If all the details are correctly provided, it returns a connection object. In the above example it is  cnx.

Connection can be established using the MySQLConnection Class also.

The MySQL Connection constructor initializes the attributes and tries to connect to the server. If the connection is successfully established, a connection object is returned as shown in the below code snippet.



The parameters in the MySQLConnection constructor can be passed as variable length arguments also.



 3. As the third step , we need to check if the connection to the MySQL database has been established successfully by  using the is\_connected() method. In case of an exception such as  MySQL server  not available, the database does not exist or invalid user name or password, Python will raise an exception. The  try and except block handles the exception and displays the error.

4. The fourth and the final step is to close the database connection using the  close() method of the  MySQLConnection object.

The following code snippet shows how to handle the connection errors and finally close the connection.



**Creating tables in the MySQL database using the connector:**

To create tables in the database, we need to use a handle structure called cursor, which is available in the connection class.

To create a new table in a MySQL database , we have to follow the below steps:

* First connect to the MySQL database server by creating a new MySQLConnection object.
* Next initiate a MySQLCursor object from the MySQLConnectionobject.
* Then execute the CREATE statement to create the table.
* Finally, Close the database connection.

The below code shows how to create tables using cursor.

**import mysql.connector as mysql**

**db = mysql.connect( host = "localhost", user = "root", passwd = "dbms", database = "test" )**

**cursor = db.cursor()**

**## creating a table called 'Persons' in the 'test' database**

**cursor.execute("CREATE TABLE PERSONS (name VARCHAR(25), address VARCHAR(35))")**

Make sure you define the name of the database when you create the connection. In the above example, **‘test’** is the name of the database.

**How to Insert Data into a Table:**

To insert new rows into a MySQL table, we have to follow the below steps:

* First connect to the MySQL database server by creating a new MySQLConnection object.
* Next initiate a MySQLCursor object from the MySQLConnectionobject.
* Then execute the INSERT statement to insert data into the table.
* Finally close the database connection.

MySQL Connector/Python provides API that allows insertion of one or multiple rows into a table at a time.

The below code is an example of how to insert data in a table called PERSONS.

**import mysql.connector as mysql**

**#database connection**

**connection = mysql.connect(host="localhost", user="root", passwd="", database="test")**

**cursor = connection.cursor()**

**# queries for inserting values**

**insert1 = "INSERT INTO PERSONS(NAME, ADDRESS) VALUES('Ravi', '3rd Block NEDFi House' );"**

**insert2 = "INSERT INTO PERSONS(NAME, ADDRESS) VALUES('Sadduz', 'Geetanjali Park,Kolkata' );"**

**#executing the quires**

**cursor.execute(insert1)**

**cursor.execute(insert2)**

**#commiting the connection then closing it.**

**connection.commit()**

**connection.close()**

**Insert Data into a Table Using Python variables in a MySQL INSERT query**:

Sometimes we need to insert a value stored in  python variable into a table’s column.

* We can insert Python variable into the table using the prepared statement and parameterized query.
* Using a parameterized query we can pass python variables as a query parameter in which placeholders (%s) is used for parameters.

Following  code is an example for  how to insert data into a table PERSONS using Python variables.

**import mysql.connector as mysql**

**db = mysql.connect( host = "localhost", user = "root", passwd = "dbms", database = "test" )**

**cursor = db.cursor()**

**## defining the Query**

**query = "INSERT INTO persons (name, address) VALUES (%s, %s)"**

**## storing values in a variable**

**values = ("Hafeez", "Subhash Marg,New Delhi")**

**## executing the query with values**

**cursor.execute(query, values)**

**## to make final output we have to run the 'commit()' method of the database object**

**db.commit()**

**print(cursor.rowcount, "record inserted")**

**Inserting Multiple Rows**

Sometimes we need to **insert multiple rows into a table.**

**To insert multiple rows into the table, we have to use the executemany() method.**

**This method takes a query as the first argument and a list of tuples containing the data as a second argument.**

**The below code is an example of how to insert multiple rows.**

**import mysql.connector as mysql**

**db = mysql.connect( host = "localhost", user = "root", passwd = "dbms", database = "test" )**

**cursor = db.cursor()**

**## defining the Query**

**query = "INSERT INTO Persons (name, address) VALUES (%s, %s)"**

**## storing values in a variable**

**values = [ ("Peter","M G Marg,Delhi"), ("Amy", "New Street,Kolkata"), ("Michael", "Lane 6,Bangaluru"), ("Hennah", "NEDFi House,Guwahati") ]**

**## executing the query with values**

**cursor.executemany(query, values)**

**## to make final output we have to run the 'commit()' method of the database object**

**db.commit()**

**print(cursor.rowcount, "records inserted")**

**How To Retrieve data from MySQL database using the connector:**

To retrieve data in a MySQL database from Python, we need to follow the below steps:

1. First, we need to connect to the database and get a MySQLConnection object.
2. Next , we need to instantiate a  MySQLCursor object from the the MySQLConnection object and use the cursor to execute a query by calling its  execute() method. We can use fetchone() ,  fetchmany() or  fetchall() method to fetch data from the result set.
3. Finally, we need to close the cursor as well as the database connection by calling the  close() method of the corresponding object.

The following code demonstrates how data can be fetched from database using cursor.

The following code demonstrates how data can be fetched from database using cursor, and how the exceptions are to be taken care of.

In the code below, the "table does not exist error" is being taken care of.

If there are no errors, the data that  being fetched is to be displayed one by one from the cursor.

**import mysql.connector**

**from mysql.connector import Error**

**try:**

**connection = mysql.connector.connect(host='localhost',database='test',user='root',password='dbms')**

**sql\_select\_Query = "select \* from Persons"**

**cursor = connection.cursor()**

**cursor.execute(sql\_select\_Query)**

**records = cursor.fetchall()**

**print("Total number of rows in Persons is: ", cursor.rowcount)**

**print("\nPrinting each person record")**

**for row in records:**

**print("Name = ", row[0], )**

**print("Address = ", row[1])**

**except Error as e:**

**print("Error reading data from MySQL table", e)**

**finally:**

**if (connection.is\_connected()):**

**connection.close()**

**cursor.close()**

**print("MySQL connection is closed")**

**How to delete data in the MySQL database using the connector:**

To delete data from a table from a Python program, we need to follow the below steps:

1. First, we have to connect to the database by creating a new MySQLConnection object.
2. Next, we need to instantiate a new cursor object and call its  execute() method with the SQL statement to be used to delete record as per requirement.
3. Next , to make the changes permanent , we have to ensure that the commit() method of the MySQLConnection object is called after the  execute() method.
4. Finally , we need to close the cursor and database connection by calling  close() method of the corresponding objects.

The following code gives a demo on how to remove unwanted data from my database by using the unquoted %s markers and data in tuples.

**Import mysql.connector**  
  
**mydb = mysql.connector.connect(**  
**host="localhost",**  
**user="root",**  
**passwd="passwd1",**  
**database="test"**  
**)**  
  
**mycursor =mydb.cursor()**  
  
**sql ="DELETE FROM personss WHERE address = 'Mountain 21'"**  
  
**mycursor.execute(sql)**  
  
**mydb.commit()**  
  
**print(mycursor.rowcount, "record(s) deleted")**