**Python MySQL Connectivity**

# Database Connection

In this section of the tutorial, we will discuss the steps to connect the python application to the database.

There are the following steps to connect a python application to our database.

1. Import mysql.connector module
2. Create the connection object.
3. Create the cursor object
4. Execute the query

## Creating the connection

* To create a connection between the MySQL database and the python application, the connect() method of mysql.connector module is used.
* Pass the database details like HostName, username, and the database password in the method call.
* The method returns the connection object.

**Consider the following example.**

### **Example**

**import mysql.connector**

**#Create the connection object**

**myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "", database = "mydb")**

**#printing the connection object**

**print(myconn)**

**Output:**

**<mysql.connector.connection.MySQLConnection object at 0x7fb142edd780>**

## Creating a cursor object

* The cursor object can be defined as an abstraction specified in the Python DB-API 2.0.
* It facilitates us to have multiple separate working environments through the same connection to the database.
* We can create the cursor object by calling the 'cursor' function of the connection object.
* The cursor object is an important aspect of executing queries to the databases.

### **Example**

**import** mysql.connector

#Create the connection object

myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "", database = "mydb")

#printing the connection object

**print**(myconn)

#creating the cursor object

cur = myconn.cursor()

**print**(cur)

**Output:**

<mysql.connector.connection.MySQLConnection object at 0x7faa17a15748>

MySQLCursor: (Nothing executed yet)

# Creating new databases

In this section of the tutorial, we will create the new database PythonDB.

### **Example:**

1. **import mysql.connector**
3. **#Create the connection object**
4. **myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "")**
6. **#creating the cursor object**
7. **cur = myconn.cursor()**
9. **try:**
10. **#creating a new database**
11. **cur.execute("create database PythonDB")**
13. **#getting the list of all the databases which will now include the new database PythonDB**
14. **dbs = cur.execute("show databases")**
16. **except:**
17. **myconn.rollback()**
19. **for x in cur:**
20. **print(x)**
22. **myconn.close()**

**Output:**

**('EmployeeDB',)**

**('PythonDB',)**

**('Test',)**

**('TestDB',)**

**('anshika',)**

# Creating the table

* In this section of the tutorial, we will create the new table Employee.
* We have to mention the database name while establishing the connection object.
* We can create the new table by using the CREATE TABLE statement of SQL.
* In our database PythonDB, the table Employee will have the four columns, i.e., name, id, salary, and department\_id

**Example**

1. **import** mysql.connector
3. #Create the connection object
4. myconn = mysql.connector.connect(host = "localhost", user = "root",passwd = "",database = "PythonDB")
6. #creating the cursor object
7. cur = myconn.cursor()
9. **try**:
10. #Creating a table with name Employee having four columns i.e., name, id, salary, and department id
11. dbs = cur.execute("create table Employee(name varchar(20) not null, id int(20) not null primary key, salary float not null, Dept\_id int not null)")
12. **except**:
13. myconn.rollback()
15. myconn.close()