MySQL-PYTHON CONNECTIVITY
VIVA QUESTIONS

1. **What is MySQL Connector/Python?**
   MySQL Connector/Python is a standardized database driver provided by MySQL. It is used to access the MySQL database from Python.

2. **What are the five major steps for connecting MySQL and Python?**

   There are five major steps for connecting MySQL and Python.

   - Import MySQL connector
   - Open a connection to a database
   - Create a cursor object
   - Execute a query
   - Close the connection

3. **How do we create a connection object?**
   Connection object is created with the help of connect() function. The connect() function of mysql.connector package is used to establish a connection to MySQL database. For example:

   conn1 = mysql.connector.connect(host='localhost', database='test', user='root', password='tiger')

   Here, mysql.connector.connect is a MySQL-python library function that creates the connection and returns a connection object, ***conn1***. It connects to a specific MySQL database (***test*** in this case) using the given host (***localhost*** in this case), username (***root*** in this case) and password (***tiger*** in this case).

4. **How do we create a cursor object?**
   The connection object returned by the connect() method is used to create a cursor object. You can create a **cursor** object using the cursor() method of the connection object/class. The cursor object is used to execute statements to perform database operations. Foe example:

   cursor1 = conn1.cursor()

   Here, ***cursor1*** is a cursor object created using connection object ***conn1*** using cursor() method.

5. **How do we execute SQL query through Python?**
   To execute SQL queries through Python, cursor object is used along with execute() method. For example:

   cursor1.execute("select * from emp;")

   Here, ***cursor1*** is a cursor object which uses execute method to run the SQL query. The sql query is given in the form of a string. The output of the SQL query is stored in the cursor object in the form of a result set.

6. **What is the difference between fetchall() and fetchnone() methods?**
   The fetchall() method fetches all rows of a result set and returns a list of tuples. The fetchnone() method returns a single record as a list/ tuple and None if no more rows are available.

7. **What is the purpose of rowcount parameter?**
   It returns the number of rows affected by the last execute method for the same cur object.