

Database in Python -Week 8.1

Database (MySQL) with Python

1. What if you want to insert multiple rows into a table in a single insert query from the Python application. Use the cursor's executemany() function to insert multiple records into a table.

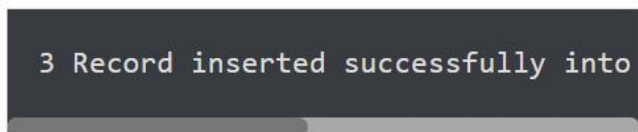
Syntax of the `executemany()` method: `cursor.executemany(operation, seq_of_params)`

Note: Each tuple is enclosed within parentheses and separated by commas. For example, to insert multiple rows in a laptop table, we can use the following SQL Query:

INSERT INTO Laptop (Id, Name, Price, Purchase_date) VALUES (%s, %s, %s, %s)

`records_to_insert = [(4, 'HP Pavilion Power', 1999, '2019-01-11'), (5, 'MSI WS75 9TL-496', 5799, '2019-02-27'), (6, 'Microsoft Surface', 2330, '2019-07-23')]`

Output:



```
3 Record inserted successfully into
```

Id	Name	Price	Purchase_date
1	Lenovo ThinkPad P71	6459	2019-08-14
2	Area 51M	6999	2019-04-14
3	MacBook Pro	2499	2019-06-20
4	HP Pavilion Power	1999	2019-01-11
5	MSI WS75 9TL-496	5799	2019-02-27
6	Microsoft Surface	2330	2019-07-23

MySQL Laptop table after inserting
multiple rows

Refer to fetch data from the MySQL table to verify your result.

Note:

- Using `cursor.executemany(sql_insert_query, records_to_insert)` we are inserting multiple rows (from a List) into the table.
- Using the `cursor.rowcount` we can find the number of records inserted.

2. Insert timestamp and `DateTime` into a MySQL table using Python

3. Fetch single row from MySQL table using cursor's fetchone

Output:

Printing first record (1, 'Lenovo ThinkPad P71', 6459.0, datetime.date(2019, 8, 14))
MySQL connection is closed

4. Python Fetch MySQL row using the column names

To select records from my MySQL table using a column name, we only need to change the cursor creation. Replace the standard cursor creation with the following code, and you are ready to fetch records from my MySQL table using a column name.

cursor = connection.cursor(dictionary=True)

Output:

Fetching each row using column name

1 Lenovo ThinkPad P71 6459.0 2019-08-14
2 Area 51M 6999.0 2019-04-14
3 MacBook Pro 2499.0 2019-06-20
4 HP Pavilion Power 1999.0 2019-01-11
5 MSI WS75 9TL-496 5799.0 2019-02-27
6 Microsoft Surface 2330.0 2019-07-23
7 Acer Predator Triton 2435.0 2019-08-17
10 Lenovo ThinkPad P71 6459.0 2019-08-14
11 Lenovo ThinkPad P71 6459.0 2019-08-14

MySQL connection is closed

5. Update Multiple Rows of MySQL Table using Python

It is possible to update multiple rows in a single SQL Query. You can also call it a bulk update. Use the **cursor.executemany()** method of cursor object to update multiple rows of a table.

The syntax of the executemany() method : cursor.executemany(operation, seq_of_params)

Output:

2 Records of a laptop table updated successfully
connection is closed

6. Update Datetime and timestamp column of a MySQL table from Python

Suppose you have a date column in a MySQL table and you want to update a **datetime.datetime()** object into this column. Let's see how to prepare an update query to update the datetime column of a table

Output:

Purchased Date Updated successfully
connection is closed

7. Delete All rows from a table in Python

It is possible to delete all rows from a MySQL database table using a truncate SQL query. Truncate SQL queries remove all data from a table, typically bypassing the number of integrity enforcing mechanisms.

Output:

All Record Deleted successfully
MySQL connection is closed

8. Delete MySQL Table and Database from Python

You can delete old, unused tables and temporary databases and tables using a DROP TABLE and DROP DATABASE statement.

Output:

Table and Database Deleted successfully
MySQL connection is closed

9. Delete MySQL Table column from Python

Use alter table drop column command to delete a column from a MySQL table.

Output:

Column Deleted successfully
MySQL connection is closed