

SQL using Python | Set 3 (Handling large data)

It is recommended to go through [SQL using Python | Set 1](#) and [SQL using Python and SQLite | Set 2](#)

In the previous articles the records of the database were limited to small size and single tuple. This article will explain how to write & fetch large data from the database using module SQLite3 covering all exceptions.

A simple way is to execute the query and use fetchall(). This has been already discussed in SET 1.

■ `executescript()`

This is a convenience method for executing multiple SQL statements at once. It executes the SQL script it gets as a parameter.

Syntax:

```
sqlite3.connect.executescript(script)
```

```
import sqlite3
```

```
# Connection with the DataBase
```

```
# 'library.db'
```

```
connection = sqlite3.connect("library.db")
```

```
cursor = connection.cursor()
```

```
# SQL piece of code Executed
```

```
# SQL piece of code Executed
```

```
cursor.executescript("""
```

```
    CREATE TABLE people(
        firstname,
        lastname,
        age
    );
```

```
    CREATE TABLE book(
        title,
        author,
        published
    );
```

```
INSERT INTO
```

```

        book(title, author, published)
    VALUES (
        'Dan Clarke''s GFG Detective Agency',
        'Sean Simpsons',
        1987
    );
    """

sql = """
SELECT COUNT(*) FROM book;"""

cursor.execute(sql)

# The output is fetched and returned
# as a List by fetchall()
result = cursor.fetchall()
print(result)

sql = """
SELECT * FROM book;"""

cursor.execute(sql)

result = cursor.fetchall()
print(result)

# Changes saved into database
connection.commit()

# Connection closed(broken)
# with DataBase
connection.close()

```

[Run on IDE](#)

Output:

```

[(1,)]
[('Dan Clarke's GFG Detective Agency', 'Sean Simpsons', 1987)]

```

Note: This piece of code may not work on online interpreters, due to permission privileges to create/write database.

■ executemany()

It is often the case when, large amount of data has to be inserted into database from Data Files(for simpler case take Lists, arrays). It would be simple to iterate the code many a times than write every time, each line into database. But the use of loop would not be suitable in this case, the below example shows why. Syntax and use of executemany() is explained below and how it can be used like a loop.

```

import sqlite3

# Connection with the DataBase
# 'library.db'
connection = sqlite3.connect("library.db")
cursor = connection.cursor()

# SQL piece of code Executed
cursor.execute("""

```

```

        CREATE TABLE book(
            title,
            author,
            published);"""

List = [('A', 'B', 2008), ('C', 'D', 2008),
        ('E', 'F', 2010)]

connection. executemany("""

    INSERT INTO
    book(title, author, published)
    VALUES (?, ?, ?)""", List)

sql = """
    SELECT * FROM book;"""
cursor.execute(sql)
result = cursor.fetchall()
for x in result:
    print(x)

# Changes saved into database
connection.commit()

# Connection closed(broken)
# with DataBase
connection.close()

```

[Run on IDE](#)

Output:

```

Traceback (most recent call last):
  File "C:/Users/GFG/Desktop/SQLITE3.py", line 16, in
    List[2][3] = [['A', 'B', 2008], ['C', 'D', 2008], ['E', 'F', 2010]]
NameError: name 'List' is not defined

```

The use of executemany(), can make the piece of code functional.

```

import sqlite3

# Connection with the DataBase
# 'library.db'
connection = sqlite3.connect("library.db")
cursor = connection.cursor()

# SQL piece of code Executed
cursor.execute("""
    CREATE TABLE book(
        title,
        author,
        published);""")

List = [('A', 'B', 2008), ('C', 'D', 2008),
        ('E', 'F', 2010)]

connection. executemany("""
    INSERT INTO
    book(title, author, published)
    VALUES (?, ?, ?)""", List)

```

```

sql = """
SELECT * FROM book;"""
cursor.execute(sql)
result = cursor.fetchall()
for x in result:
    print(x)

# Changes saved into database
connection.commit()

# Connection closed(broken)
# with DataBase
connection.close()

```

[Run on IDE](#)

Output:

```

('A', 'B', 2008)
('C', 'D', 2008)
('E', 'F', 2010)

```

■ Fetch Large Data

```

import sqlite3

# Connection created with the
# database using sqlite3.connect()
connection = sqlite3.connect("company.db")
cursor = connection.cursor()

# Create Table command executed
sql = """
    CREATE TABLE employee (
        ID INTEGER PRIMARY KEY,
        fname VARCHAR(20),
        lname VARCHAR(30),
        gender CHAR(1),
        dob DATE);"""
cursor.execute(sql)

# Single Tuple inserted
sql = """
    INSERT INTO employee
    VALUES (1007, "Will", "Olsen", "M", "24-SEP-1865");"""
cursor.execute(sql)

# Multiple Rows inserted
List = [(1008, 'Rkb', 'Boss', 'M', "27-NOV-1864"),
        (1098, 'Sak', 'Rose', 'F', "27-DEC-1864"),
        (1908, 'Royal', 'Bassen', "F", "17-NOV-1894")]

connection.executemany(
    "INSERT INTO employee VALUES (?, ?, ?, ?, ?)", List)

print("Method-1\n")

# Multiple Rows fetched from
# the Database
for row in connection.execute('SELECT * FROM employee ORDER BY ID')
    print (row)

```

```
print("\nMethod-2\n")

# Method-2 to fetch multiple
# rows
sql = """
    SELECT * FROM employee ORDER BY ID;"""

cursor.execute(sql)
result = cursor.fetchall()

for x in result:
    print(x)

connection.commit()
connection.close()
```

[Run on IDE](#)

Output:

Method-1

```
(1007, 'Will', 'Olsen', 'M', '24-SEP-1865')
(1008, 'Rkb', 'Boss', 'M', '27-NOV-1864')
(1098, 'Sak', 'Rose', 'F', '27-DEC-1864')
(1908, 'Royal', 'Bassen', 'F', '17-NOV-1894')
```

Method-2

```
(1007, 'Will', 'Olsen', 'M', '24-SEP-1865')
(1008, 'Rkb', 'Boss', 'M', '27-NOV-1864')
(1098, 'Sak', 'Rose', 'F', '27-DEC-1864')
(1908, 'Royal', 'Bassen', 'F', '17-NOV-1894')
```

Note: This piece of code may not work on online interpreters, due to permission privileges to create/write database.



Piyush Doorwar

Check out this Author's [contributed articles](#).

If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](https://www.geeksforgeeks.org/contribute) or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.



GATE CS Corner Company Wise Coding Practice

Python SQL

[Login to Improve this Article](#)

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Recommended Posts:

[MD5 hash in Python](#)

[SHA in Python](#)

[Fetch top 10 starred repositories of user on GitHub | Python](#)

[How to print without newline in Python?](#)

[Python String | ljust\(\), rjust\(\), center\(\)](#)

[Part of Speech Tagging with Stop words using NLTK in python](#)

[Python Basics](#)

[numpy.hypot\(\) in Python](#)

[Django Introduction | Set 2 \(Creating a Project\)](#)

Replacing strings with numbers in Python for Data Analysis

([Login](#) to Rate)

0

Average Difficulty : **0/5.0**
No votes yet.

☐

Add to TODO List

☐

Mark as DONE

Basic

Easy

Medium

Hard

Expert

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments

Share this post!

@geeksforgeeks, Some rights reserved

Contact Us!

About Us!

Careers!

Privacy

Policy

