

# SQLITE3 w PYTHON

## I. Đặc điểm :

- Nhẹ & Nhanh
  - Zero Configuration
  - Cross-platform
  - Transactional
  - Able to download file & share
- 

## II. SQL code : = SQL database

---

## III. SQLite3 w Python

- Python have sqlite3 package → download it → `pip install pysqlite3 (version 3)`  
`pip install pysqlite`
- If you have it → `import package` → `import sqlite3`

import sqlite3

MEMORY DATABASE

if \_\_name\_\_ == '\_\_main\_\_':

# create connection

con = sqlite3.connect(":memory:") # tạo cơ sở dữ liệu trong RAM (tạm thời)

con = sqlite3.connect("\_.db") (o lưu file vật lý)

# create cursor

cur = con.cursor()

# tạo bảng

cur.execute("CREATE TABLE test(id INTEGER PRIMARY KEY, msg TEXT);")

# chèn dữ liệu vào bảng

for i in range(10):

cur.execute("INSERT INTO test VALUES (?, ?);", (i, f"Hello {i}"))

→ thay dữ liệu thật số để tránh lỗi SQL inject vào

# lưu thay đổi → vào DB

con.commit()

# truy vấn dữ liệu

cur.execute("SELECT \* FROM test")

for row in cur:

print(row) # in ra từng dòng dữ liệu

# sao lưu cơ sở dữ liệu sang file trên ổ đĩa

backup = sqlite3.connect("../resources/sample.db") # đường dẫn lưu file CSDL

with backup:

con.backup(backup, pages=1) # sao lưu 1 trang mỗi lần (có thể tăng lên nếu cần)

# đóng kết nối

con.close()

→ CSDL sẽ mất khi chương trình kết thúc → lưu ra file (an toàn)

# DATABASE

import sqlite3

→ Tạo tệp

# 1. Kết nối tới file CSDL "tutorial.db" (sẽ tạo nếu chưa tồn tại)

con = sqlite3.connect("tutorial.db")

cur = con.cursor() # Tạo con trỏ để thực hiện truy vấn

# 2. Xóa bảng 'movie' nếu đã tồn tại

cur.execute("DROP TABLE IF EXISTS movie")

# 3. Tạo bảng mới với 3 cột: title, year, score

cur.execute("CREATE TABLE movie(title, year, score)")

# 4. Thêm hai bản ghi đầu tiên vào bảng

cur.execute("""

INSERT INTO movie

VALUES ('Monty Python and the Holy Grail', 1975, 8.2),

('And Now for Something Completely Different', 1971, 7.5)

""")

con.commit() # Lưu thay đổi

# 5. Truy vấn và in tất cả bản ghi hiện có

res = cur.execute("SELECT \* FROM movie")

for row in res.fetchall(): # Truy vấn toàn bộ kết quả của truy vấn trên  
print(row)

# 6. Thêm dữ liệu bằng danh sách

data = [

('Monty Python Live at the Hollywood Bowl', 1982, 7.9),

('Monty Python's The Meaning of Life', 1983, 7.5),

('Monty Python's Life of Brian', 1979, 8.0),

]

cur.executemany("INSERT INTO movie VALUES(?, ?, ?)", data)

con.commit() # Lưu thay đổi

# 7. Truy vấn các bản ghi, sắp xếp theo năm, chỉ lấy title và year

for row in cur.execute("SELECT year, title FROM movie ORDER BY year"):  
print(row)

## Format

1. Import thư viện: `import sqlite3`

2. Kết nối tới Database (nếu chưa có sẽ tạo): `con = sqlite3.connect("ten_file.db")`  
"memory": tạo db ở RAM

3. Tạo cursor để thực thi câu lệnh: `cur = con.cursor()`

4. Thực thi câu lệnh: `cur.execute(sql, params)`

+ Tạo bảng: `cur.execute("CREATE TABLE IF NOT EXISTS table_name (col1 TYPE, ...)" )`

+ Chèn dữ liệu:

. Chèn 1 dòng: `cur.execute("INSERT INTO table_name VALUES (?, ?, ...)", (val1, val2, ...))`

. Chèn nhiều dòng:

a) `cur.execute("..." .. "...")`: multi-line string.

Ex: `cur.execute("..." INSERT VALUES table_names VALUES (val1, val2, val3, ...),  
(val1, val2, ...),  
... "...")`

b) `cur.execute many (sql, list of tuples)` → Tránh bị SQL INJECTION

Ex: `data = [(val1, val2, ...),  
(...)]`  
`cur.executemany("INSERT INTO table_names VALUES (?, ?, ?)", data)`

c) Use for:

Ex: `data = [(val1, val2, ...),  
(...)]`  
for row in data:  
`cur.execute("INSERT INTO table_names VALUES (?, ?, ?, ...)", row)`

5. Commit thay đổi (Lưu thay đổi vào DB) : `con.commit()`

6. Truy vấn dữ liệu :

```
res = cur.execute ("SELECT ____ FROM take_name")
```

+1 Trả về toàn bộ kết quả truy vấn SELECT : `cursor.fetchall()`

```
for row in res.fetchall():  
    print (row)
```

+1 Trả về 1 dòng kết quả : `cursor.fetchone()`

```
row1 = res.fetchone()  
print (row1)
```

#### FETCHONE()

- mỗi lần gọi lấy 1 dòng tiếp theo
- Tuple or None
- xử lý từng dòng

#### FETCHALL()

- lấy tất cả dòng còn lại
- List các tuple
- xử lý toàn bộ kết quả một lần.

7. Sao lưu DB trên cái ổ đĩa : `con.backup (backup-db)`

Ex: `backup = sqlite3.connect ("... | ... | resources / ____ .db")` # on disk db  
with backup :

`con.backup (backup, pages=1)` # sao lưu mỗi trang 1 lần → chargeable

8. Đóng kết nối : `con.close()`