

Basic Python - Work with SQLite Database

Hoàng-Nguyên Vũ



1. Mô tả: Làm quen với Cơ sở dữ liệu SQLite

- **SQLite** là một hệ quản trị cơ sở dữ liệu hay còn gọi là hệ thống cơ sở dữ liệu quan hệ nhỏ gọn, khác với các hệ quản trị khác như MySQL, SQL Server, Oracle, PostgreSQL... SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine truyền thống, không cần mô hình client-server nên rất nhỏ gọn. SQLite được sử dụng vào rất nhiều chương trình từ desktop đến mobile hay là website..
- Ngoài những lý do trên thì không thể không kể đến những ưu điểm khi sử dụng SQLite, sau đây là phần ưu điểm của SQLite. **Ưu điểm của SQLite:**
 - **Nhẹ và dễ sử dụng:** SQLite không yêu cầu cài đặt máy chủ riêng biệt và có thể được nhúng trực tiếp vào ứng dụng Python của bạn.
 - **Hiệu suất cao:** SQLite cung cấp hiệu suất truy vấn nhanh và có thể xử lý lượng dữ liệu lớn.
 - **Đa nền tảng:** SQLite có sẵn trên hầu hết các nền tảng, bao gồm Windows, MacOS, Linux và Android.
 - **Miễn phí và mã nguồn mở:** SQLite là phần mềm miễn phí và mã nguồn mở, có nghĩa là bạn có thể sử dụng và sửa đổi nó cho mục đích của riêng bạn.

2. Làm quen với các câu lệnh SQL trong SQLite Database:

- Ví dụ chúng ta cần quản lý thông tin khách hàng gồm các trường dữ liệu cơ bản như sau:

Bảng 1: Bảng quản lý thông tin của khách hàng

CUSTOMER		
Email	Name	Phone

Phân tích sơ lược, chúng ta dễ thấy mỗi khách hàng chỉ có duy nhất 1 email, nên email sẽ đóng vai trò là khóa chính, nhằm để xác định thông tin của một khách hàng trong hệ thống. Để quản lý các thông tin trên và truy vấn chúng ta sẽ có những thao tác chính như sau: Tạo bảng (**CREATE TABLE**), lấy dữ liệu từ bảng (**SELECT**), thêm mới dữ liệu (**INSERT**), cập nhật dữ liệu (**UPDATE**), xóa dữ liệu (**DELETE**).

+ **Tạo bảng mới (CREATE TABLE):** Để tạo bảng trên ta sẽ sử dụng như sau:

```
1 -- Tạo Bảng CUSTOMERS gồm 3 cột email, name, phone với email là
  khóa chính
2 CREATE TABLE customers (
3   email TEXT PRIMARY KEY,
4   name TEXT NOT NULL,
5   phone TEXT
6 );
```

+ **Lấy dữ liệu (SELECT):** Chúng ta có 2 cách lấy dữ liệu từ bảng trong cơ sở dữ liệu bao gồm: lấy hết dữ liệu của toàn bộ cột trong bảng và lấy dữ liệu từ các cột được chỉ định. Cách truy vấn (Query) có cấu trúc tổng quát như sau:

```
1 SELECT column1, column2, ...
2 FROM table_name;
```

A. Lấy toàn bộ dữ liệu của tất cả cột trong bảng:

```
1 -- Lấy toàn bộ dữ liệu của tất cả cột trong bảng CUSTOMERS
2 SELECT *
3 FROM CUSTOMERS;
```

B. Lấy dữ liệu từ các cột được chỉ định trong bảng:

```
1 -- Lấy dữ liệu từ các cột được chỉ định trong bảng CUSTOMERS
2 SELECT NAME, PHONE
3 FROM CUSTOMERS;
```

C. Lấy dữ liệu theo điều kiện trong bảng (Filter):

```
1 -- Lấy dữ liệu theo điều kiện email = giá trị truyền vào từ bảng
  CUSTOMERS
2 SELECT NAME, PHONE
3 FROM CUSTOMERS
4 WHERE 1 = 1
5 AND EMAIL = 'aivietnam@aivietnam.edu.vn';
```

+ **Thêm dữ liệu (INSERT):** Để thêm dữ liệu vào bảng chúng ta có thể thực thi với câu lệnh có cấu trúc tổng quát như sau:

```
1 INSERT INTO table_name (column1, column2, column3, ...)
2 VALUES (value1, value2, value3, ...);
```

Ví dụ:

```

1 -- Thêm mới 1 dòng trong bảng CUSTOMERS
2 INSERT INTO CUSTOMERS(EMAIL, NAME, PHONE)
3 VALUES('nguyen@nguyen.com', 'Nguyen', '123456789');
4
5 -- Thêm mới nhiều dòng trong bảng CUSTOMERS
6 INSERT INTO CUSTOMERS(EMAIL, NAME, PHONE)
7 VALUES
8     ('nguyen@aivietnam.edu.vn', 'Nguyen', '123456789'),
9     ('admin@aivietnam.edu.vn', 'Vinh', '1122334455');

```

- + **Cập nhật dữ liệu (UPDATE):** Để cập nhật dữ liệu vào bảng chúng ta có thể thực thi với câu lệnh có cấu trúc tổng quát như sau:

```

1 UPDATE table_name
2 SET column1 = value1, column2 = value2, ...
3 WHERE condition;

```

Ví dụ: Cập nhật tên của email là `nguyen@aivietnam.edu.vn` thành `Hoang Nguyen`.

```

1 -- Cập nhật tên của email là nguyen@aivietnam.edu.vn thành Hoang
  Nguyen
2 UPDATE CUSTOMERS
3 SET NAME = 'Hoang Nguyen'
4 WHERE 1 = 1
5 AND EMAIL = 'nguyen@aivietnam.edu.vn';

```

- + **Xóa dữ liệu (DELETE):** Để xóa dữ liệu khỏi bảng chúng ta có thể thực thi với câu lệnh có cấu trúc tổng quát như sau:

```

1 DELETE FROM table_name
2 WHERE condition;

```

Ví dụ: Xóa email `nguyen@aivietnam.edu.vn` khỏi bảng `CUSTOMERS`

```

1 -- Xóa email: nguyen@aivietnam.edu.vn khỏi bảng CUSTOMERS
2 DELETE FROM CUSTOMERS
3 WHERE 1 = 1
4 AND EMAIL = 'nguyen@aivietnam.edu.vn';

```

3. Sử dụng SQLite trong Python:

- **Tạo kết nối đến CSDL:** Để sử dụng SQLite và tương tác với cơ sở dữ liệu SQLite trong python, chúng ta cần tạo kết nối đến CSDL bằng câu lệnh như sau:

```

1 import sqlite3
2 # Tạo kết nối tới CSDL có tên là database.sqlite
3 # Nếu database.sqlite chưa tồn tại trong hệ thống thì nó sẽ
4 # tự tạo mới
5 connection = sqlite3.connect('database.sqlite')
6 cursor = connection.cursor()

```

→ Sau khi tạo kết nối đến cơ sở dữ liệu xong, chúng ta có Object có tên *cursor* nhằm giúp chúng ta thực thi và tương tác các câu lệnh truy vấn (SELECT, INSERT, UPDATE, DELETE,...) đến cơ sở dữ liệu.

- **Tạo bảng:** Để tạo bảng mới xong cơ sở dữ liệu chúng ta sẽ thực hiện như sau:

```
1 # Tạo bảng CUSTOMERS:
2 cursor.execute("""
3 CREATE TABLE CUSTOMERS (
4     EMAIL TEXT PRIMARY KEY,
5     NAME TEXT NOT NULL,
6     PHONE TEXT NOT NULL
7 );
8 """)
```

- **Thêm dữ liệu vào bảng (INSERT):** Sau khi thực hiện tạo bảng ở bước trên, lúc này bảng CUSTOMERS của chúng ta hoàn toàn không có dữ liệu, nên chúng ta sẽ thêm dữ liệu vào bảng bằng cách thực thi câu SQL như sau:

```
1 # Insert data mới
2 cursor.execute("""
3 INSERT INTO CUSTOMERS(EMAIL, NAME, PHONE)
4 VALUES
5     ('nguyen@aivietnam.edu.vn', 'Nguyen', '123456789'),
6     ('admin@aivietnam.edu.vn', 'Vinh', '1122334455');
7 """)
8
9 connection.commit()
```

→ **Mở rộng:** Khi thực hiện thao tác INSERT/UPDATE/DELETE, khi thực hiện cursor.execute() trên cơ sở dữ liệu SQLite, những thay đổi này chỉ được lưu trữ tạm thời trong bộ nhớ. Nên chúng ta cần phải thực bước commit để lưu những thay đổi này vào CSDL, đảm bảo chúng được lưu trữ vĩnh viễn.

- **Lấy dữ liệu (SELECT):** Sau khi thực hiện bước tạo dữ liệu ở trên, chúng ta sẽ kiểm tra dữ liệu vừa thêm bằng cách thực thi câu SQL như sau:

```
1 import pandas as pd
2 # Lấy tất cả data từ bảng CUSTOMER
3 data=pd.read_sql_query("SELECT * FROM CUSTOMERS", connection)
4 print(data)
```

→ **Kết quả:**

```
[5]: import pandas as pd

[7]: data=pd.read_sql_query("SELECT * FROM CUSTOMERS", connection)
      print(data)
```

	EMAIL	NAME	PHONE
0	nguyen@aivietnam.edu.vn	Nguyen	123456789
1	admin@aivietnam.edu.vn	Vinh	1122334455

Hình 1: Kết quả sau khi thực hiện SELECT

- **Cập nhật dữ liệu (UPDATE):** Để thực hiện cập nhật dữ liệu trong CSDL, chúng ta sẽ thực thi câu SQL như sau:

Cập nhật tên của email : `nguyen@aivietnam.edu.vn` sang giá trị mới và kiểm tra kết quả sau khi cập nhật.

```

1 # Update name của email: nguyen@aivietnam.edu.vn
2 cursor.execute("""
3 UPDATE CUSTOMERS
4 SET NAME = 'Hoang Nguyen'
5 WHERE 1 = 1
6 AND EMAIL = 'nguyen@aivietnam.edu.vn';
7 """)
8
9 connection.commit()
10
11 data=pd.read_sql_query("SELECT * FROM CUSTOMERS", connection)
12 print(data)

```

→ Kết quả:

	EMAIL	NAME	PHONE
0	<code>nguyen@aivietnam.edu.vn</code>	Hoang Nguyen	123456789
1	<code>admin@aivietnam.edu.vn</code>	Vinh	1122334455

Hình 2: Kết quả sau khi thực hiện UPDATE

- **Xóa dữ liệu (DELETE):** Để thực hiện xóa dữ liệu trong CSDL, chúng ta sẽ thực thi câu SQL như sau:

Xóa email : `nguyen@aivietnam.edu.vn` khỏi bảng CUSTOMERS và kiểm tra kết quả sau khi cập nhật.

```

1 # Update name của email: nguyen@aivietnam.edu.vn
2 cursor.execute("""
3 DELETE FROM CUSTOMERS
4 WHERE 1 = 1
5 AND EMAIL = 'nguyen@aivietnam.edu.vn';
6 """)
7
8 connection.commit()
9
10 data=pd.read_sql_query("SELECT * FROM CUSTOMERS", connection)
11 print(data)

```

→ Kết quả:

	EMAIL	NAME	PHONE
0	<code>admin@aivietnam.edu.vn</code>	Vinh	1122334455

Hình 3: Kết quả sau khi thực hiện DELETE

4. Bài tập:

- Hãy tạo mới bảng có tên *PRODUCT* có các cột như sau:

Bảng 2: Bảng quản lý sản phẩm

PRODUCT		
Cột	Kiểu dữ liệu	Chú thích
ID	INTEGER	Khóa chính
NAME	TEXT	Not null
PRICE	INTEGER	Not null

Câu 1: Hãy thêm mới các dòng có giá trị như sau và kiểm tra kết quả:

PRODUCT		
ID	NAME	PRICE
1	iPhone 15	18000000
2	Galaxy Z-Fold 5	30000000

→ Kết quả:

	ID	NAME	PRICE
0	1	iPhone 15	18000000
1	2	Galaxy Z-Fold 5	30000000

Câu 2: Hãy cập nhật giá mới cho Galaxy Z-Fold 5 thành 50.000.000 và kiểm tra kết quả:

→ Kết quả:

	ID	NAME	PRICE
0	1	iPhone 15	18000000
1	2	Galaxy Z-Fold 5	50000000

Câu 3: Hãy xóa iPhone 15 ra khỏi CSDL và kiểm tra kết quả:

→ Kết quả:

	ID	NAME	PRICE
0	2	Galaxy Z-Fold 5	50000000