

SQL trong Python

I. Một số lệnh cơ bản trong Python để tạo cơ sở dữ liệu:

1. Kết nối với 1 cơ sở dữ liệu:

```
dbase=sqlite3.connect('mydata.db')
```

Mở cơ sở dữ liệu có tên mydata.db. Nếu nó không tồn tại nó sẽ tự động tạo ra.

2. `dbase.close()`: đóng kết nối cơ sở dữ liệu
3. `cursor = dbase.cursor()`: tạo một đối tượng cursor để thao tác với cơ sở dữ liệu
4. Tạo bảng trong cơ sở dữ liệu:

Cú pháp:

```
cursor.execute("""CREATE TABLE [IF NOT EXISTS] table_name(  
    col1 datatypes,  
    col2 datatypes,  
    col3 datatypes,  
    -----  
    )""
```

Ví dụ:

```
cursor.execute("""CREATE TABLE IF NOT EXISTS employees (  
    id INTEGER PRIMARY KEY,  
    name TEXT,  
    age INTEGER  
    )""
```

5. Chèn dữ liệu vào bảng:

Cú pháp:

```
cursor.execute(""" INSERT INTO ten_bang (cot1, cot2, cot3, ...) VALUES (?, ?, ?, ...) """,  
(gia_tri1, gia_tri2, gia_tri3, ...))
```

Trong đó:

`INSERT INTO ten_bang (cot1, cot2, cot3, ...)` là câu lệnh SQL để chèn dữ liệu vào bảng `ten_bang`.

`("?, ?, ?, ...")` là chuỗi placeholder để đại diện cho giá trị cần chèn.

(gia_tri1, gia_tri2, gia_tri3, ...) là một tuple chứa giá trị thực sự tương ứng với các placeholder.

Ví dụ:

```
cursor.execute("INSERT INTO employees (name, age) VALUES (?, ?) ", ("John Doe", 30))
```

6. Lệnh lưu thay đổi vào cơ sở dữ liệu:

```
dbase.commit()
```

7. Lệnh thực hiện trong Python:

```
cursor.execute(query)
```

Trong đó: query là câu lệnh truy vấn trong ngôn ngữ SQL

8. Lấy các dòng dữ liệu sau khi đã được truy vấn:

```
rows = cursor.fetchall()
```

9. Lấy dữ liệu từ bảng với điều kiện:

Cú pháp:

```
cursor.execute(" SELECT cot1, cot2, cot3, ...  
  
FROM ten_bang  
  
WHERE dieu_kien ")
```

Ví dụ:

```
data = cursor.execute("SELECT * FROM employee_records WHERE ID=1")
```

10. Cập nhật dữ liệu từ bảng:

Cú pháp:

```
cursor.execute("UPDATE ten_bang SET cot1 = gia_tri1, cot2 = gia_tri2 WHERE dieu_kien")
```

Ví dụ:

```
cursor.execute(" UPDATE employee_records set STARS=3 WHERE ID=5 ")
```

11. Xóa dữ liệu từ bảng:

Cú pháp:

```
cursor.execute("DELETE FROM ten_bang WHERE dieu_kien")
```

Ví dụ:

```
cursor.execute(" DELETE from employee_records WHERE ID = 1 ")
```

12. Đọc dữ liệu từ cơ sở dữ liệu vào một DataFrame:

```
df = pd.read_sql_query(query,dbase)
```

Ví dụ:

```
df = pd.read_sql_query("SELECT * FROM danh_ba", conn)
```

13. Xuất dữ liệu từ DataFrame ra file *.csv:

```
df.to_csv("filename.csv",index=False)
```

II. Một số đoạn code mẫu về database trong Python:

1. Tạo 1 cơ sở dữ liệu:

```

#create_database.py
import sqlite3
dbase = sqlite3.connect('Our_data.db') # Open a database File
print ('Database opened')
dbase.close()
print (' Database Closed')
2. Tạo 1 bảng:
#create_table.py
import sqlite3
dbase = sqlite3.connect('Our_data.db') # Open a database File
print ('Database opened')
dbase.execute(""" CREATE TABLE IF NOT EXISTS employee_records(
                ID INT PRIMARY KEY NOT NULL,
                NAME TEXT NOT NULL,
                DIVISION TEXT NOT NULL,
                STARS INT NOT NULL) """)
print ('Table created')
dbase.close()
print (' Database Closed')
3. Thêm dữ liệu vào bảng:
#insert_record.py
import sqlite3
dbase = sqlite3.connect('Our_data.db') # Open a database File
print ('Database opened')
dbase.execute(""" CREATE TABLE IF NOT EXISTS employee_records(
                ID INT PRIMARY KEY NOT NULL,
                NAME TEXT NOT NULL,
                DIVISION TEXT NOT NULL,
                STARS INT NOT NULL) """)
print ('Table created')
dbase.execute(""" INSERT INTO employee_records(ID,NAME,DIVISION,STARS)
                VALUES(5,'James','Maintenance',4)
                """)
dbase.commit()
print ('REcord inserted')
dbase.close()
print (' Database Closed')
4. Sử dụng hàm để thêm dữ liệu vào bảng:
# using_func.py
import sqlite3
dbase = sqlite3.connect('Our_data.db') # Open a database File
print ('Database opened')
dbase.execute(""" CREATE TABLE IF NOT EXISTS employee_records(
                ID INT PRIMARY KEY NOT NULL,
                NAME TEXT NOT NULL,

```

```

        DIVISION TEXT NOT NULL,
        STARS INT NOT NULL) ")
print ('Table created')
def insert_record(ID,NAME,DIVISION,STARS):
    dbase.execute(""" INSERT INTO employee_records(ID,NAME,DIVISION,STARS)
        VALUES(?,?,?,?)""",(ID,NAME,DIVISION,STARS))
    dbase.commit()
    print ('REcord inserted')
    insert_record(6,'Bob','Hardware',4)
    dbase.close()
    print (' Database Closed')
5. Đọc dữ liệu từ database:
#reading_data.py
import sqlite3
dbase = sqlite3.connect('Our_data.db') # Open a database File
print ('Database opened')
dbase.execute(""" CREATE TABLE IF NOT EXISTS employee_records(
    ID INT PRIMARY KEY NOT NULL,
    NAME TEXT NOT NULL,
    DIVISION TEXT NOT NULL,
    STARS INT NOT NULL) """)
print ('Table created')
def insert_record(ID,NAME,DIVISION,STARS):
    dbase.execute(""" INSERT INTO employee_records(ID,NAME,DIVISION,STARS)
        VALUES(?,?,?,?)""",(ID,NAME,DIVISION,STARS))
    dbase.commit()
    print ('REcord inserted')
    ##insert_record(6,'Bob','Hardware',4)
def read_Data():
    #from math import *
    data = dbase.execute( """ SELECT * FROM employee_records ORDER BY NAME""")
    for record in data:
        print ('ID : '+str(record[0]))
        print ('NAME : '+str(record[1]))
        print ('DIVISION : '+str(record[2]))
        print ('STARS : '+str(record[3])+'\n')
read_Data()
dbase.close()
print (' Database Closed')
6. Cập nhật 1 record trong database:
#update_record.py
import sqlite3

dbase = sqlite3.connect('Our_data.db') # Open a database File
print ('Database opened')

```

```

dbase.execute(""" CREATE TABLE IF NOT EXISTS employee_records(
    ID INT PRIMARY KEY NOT NULL,
    NAME TEXT NOT NULL,
    DIVISION TEXT NOT NULL,
    STARS INT NOT NULL) """)

print ('Table created')

def insert_record(ID,NAME,DIVISION,STARS):
    dbase.execute(""" INSERT INTO mployee_records(ID,NAME,DIVISION,STARS)
    VALUES(?,?,?,?)""",(ID,NAME,DIVISION,STARS))

    dbase.commit()
    print ('REcord inserted')

##insert_record(6,'Bob','Hardware',4)

def read_Data():
    #from math import *
    data = dbase.execute(""" SELECT * FROM employee_records ORDER BY
NAME""")
    for record in data:
        print ('ID : '+str(record[0]))
        print ('NAME : '+str(record[1]))
        print ('DIVISION : '+str(record[2]))
        print ('STARS : '+str(record[3])+'\n')

read_Data()

def update_record():
    dbase.execute(""" UPDATE employee_records set STARS=3 WHERE ID=5 """)
    dbase.commit()
    print ('Updated')

update_record()
print ('-----')
read_Data()

dbase.close()
print (' Database Closed')

```

7. Xóa 1 record trong database:

```
#delete_record.py
```

```

import sqlite3

dbase = sqlite3.connect('Our data.db') # Open a database File
print ('Database opened')

dbase.execute(""" CREATE TABLE IF NOT EXISTS employee_records(
    ID INT PRIMARY KEY NOT NULL,
    NAME TEXT NOT NULL,
    DIVISION TEXT NOT NULL,
    STARS INT NOT NULL) """)

print ('Table created')

def insert_record(ID,NAME,DIVISION,STARS):
    dbase.execute(""" INSERT INTO employee_records(ID,NAME,DIVISION,STARS)
    VALUES(?,?,?,?) """,(ID,NAME,DIVISION,STARS))
    dbase.commit()
    print ('REcord inserted')

##insert_record(6,'Bob','Hardware',4)

def read_Data():
    #from math import *
    data = dbase.execute(""" SELECT * FROM employee_records ORDER BY
NAME """)
    for record in data:
        print ('ID : '+str(record[0]))
        print ('NAME : '+str(record[1]))
        print ('DIVISION : '+str(record[2]))
        print ('STARS : '+str(record[3])+'\n')
    read_Data()

def update_record():
    dbase.execute(""" UPDATE employee_records set STARS=3 WHERE ID=2 """)
    dbase.commit()
    print ('Updated')

##update_record()
##print '-----'
##read_Data()

def delete_record():
    dbase.execute(""" DELETE from employee_records WHERE ID = 1 """)

```

```
dbase.commit()
print ('Deleted')
```

```
delete_record()
print ('-----')
read_Data()
```

```
dbase.close()
print (' Database Closed')
```

III. Bài tập về cơ sở dữ liệu:

Bài 1:

- Tạo 1 CSDL mydatabase.db và 1 bảng employees gồm có các cột: ID(khóa chính), name, age.
- Thêm 2 record vào bảng
- Đọc dữ liệu từ bảng và in thông tin lên màn hình.

Bài 2:

- Tạo 1 CSDL contacts.db và 1 bảng contacts gồm có các cột: ID(khóa chính), name, email, phone.
- Viết các hàm thêm dữ liệu vào CSDL, cập nhật thông tin, và tìm kiếm, xóa theo tên trong bảng.
- Thêm 1 số record vào cơ sở dữ liệu.
- Tìm kiếm và in ra thông tin.
- Xóa dữ liệu theo tên.

Bài 3: Cho câu truy vấn sau:

```
SELECT employee_id, first_name, hire_date, strftime('%Y', 'now') - strftime('%Y',
hire_date) AS tham_nien
FROM employees
ORDER BY tham_nien DESC;
```

Dùng Python để tạo cơ sở dữ liệu có thêm dữ liệu vào. Sau đó truy vấn và kết quả sẽ in ra màn hình để kiểm chứng.

Bài 4: Cho câu truy vấn sau:

```
SELECT *
FROM employees
ORDER BY salary DESC
LIMIT 1;
```

Dùng Python để tạo cơ sở dữ liệu có thêm dữ liệu vào. Sau đó truy vấn và kết quả sẽ in ra màn hình để kiểm chứng.

Bài 5:

Tạo cơ sở dữ liệu SQLite3 với một bảng để lưu trữ thông tin về người liên hệ. Danh bạ này sẽ lưu trữ thông tin về các người liên hệ, bao gồm tên, số điện thoại, địa chỉ email và địa chỉ.

Chương trình sẽ tạo menu:

- Thêm dữ liệu vào danh bạ
- Cập nhật dữ liệu danh bạ theo số điện thoại.
- Xóa danh bạ theo số điện thoại

4. Tìm danh bạ theo tên
5. Xem toàn bộ danh bạ
6. Xuất danh bạ ra file *.csv

SQL trong Python(tiếp tục)

I. Tạo CSDL bằng cách import các file *.csv:

Giả sử ta có 2 file departments.csv, employees.csv.

File departments gồm có các tên cột: department_id, department_name, manager_id, location_id

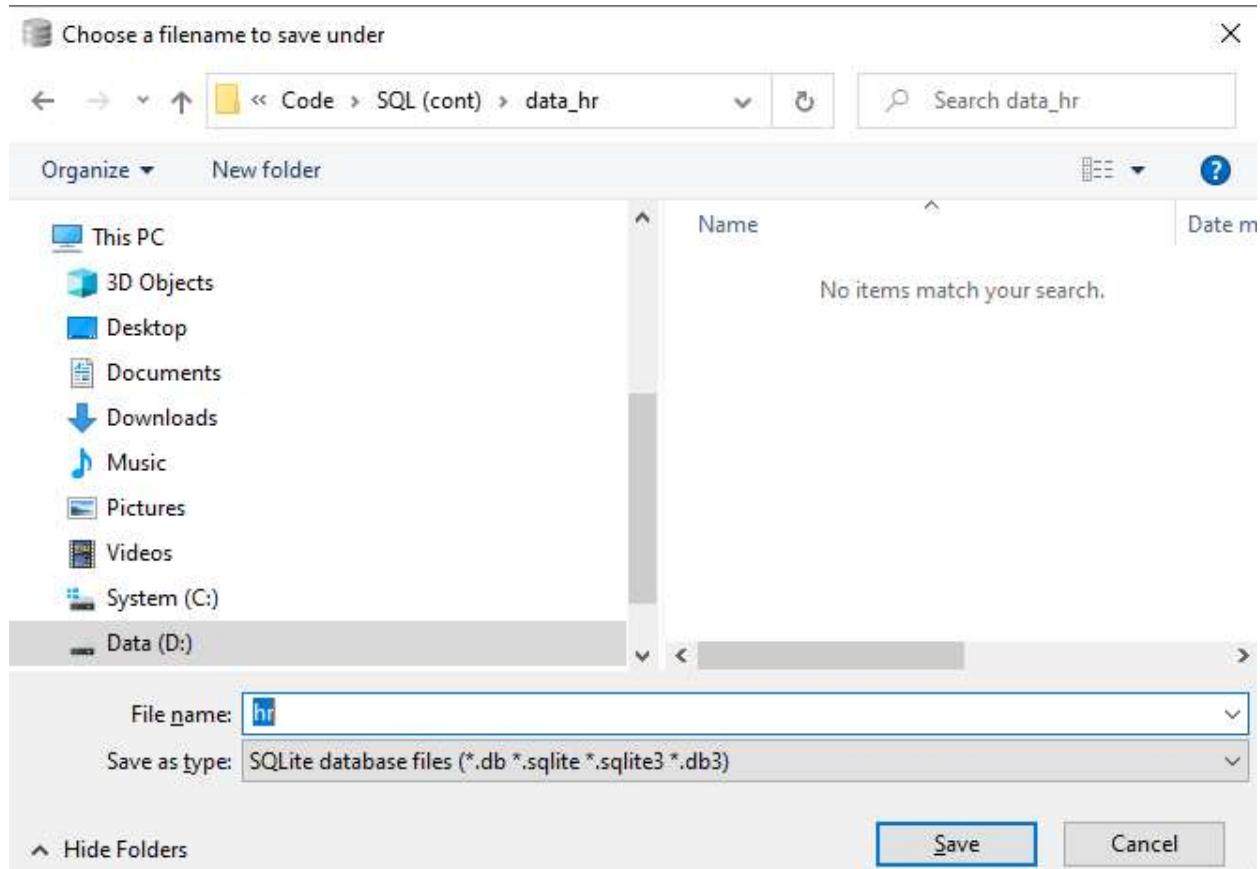
File employees gồm có các tên cột: employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id.

Bước 1:

Tạo file cơ sở dữ liệu:

Vào phần mềm DB Browser for SQL.

File- New Database-File Name ... → Save



Bước 2: Tạo các table departments, employees

Vào tên table departments-Add

Edit table definition

Table

departments

Advanced

Fields Constraints

Add Remove Move to top Move up Move down Move to bottom

Name	Type	NN	PK	AI	U	Default	Check
------	------	----	----	----	---	---------	-------

```
1 CREATE TABLE "departments" (  
2  
3 );
```

OK Cancel

Điền các tên cột và các kiểu dữ liệu

Edit table definition

?

×

Table

departments

▼ Advanced

Fields

Constraints

Add
 Remove
 Move to top
 Move up
 Move down
 Move to bottom

Name	Type	NN	PK	AI	U	Default	Check
department_id	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
department_name	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
manager_id	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
location_id	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

< >

```

1 CREATE TABLE "departments" (
2     "department_id" INTEGER,
3     "department_name" TEXT NOT NULL UNIQUE,
4     "manager_id" INTEGER,
5     "location_id" INTEGER,
6     PRIMARY KEY("department_id")
7 );

```

OK

Cancel

Edit table definition

Table
employees

▼ Advanced

Fields Constraints

Add Remove Move to top Move up Move down Move to bottom

Name	Type	NN	PK	AI	U	Default	Check
employee_id	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
first_name	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
last_name	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
email	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
phone_number	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
hire_date	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
salary	NUMERIC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
commission_pct	NUMERIC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
manager_id	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
department_id	INTEGER	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

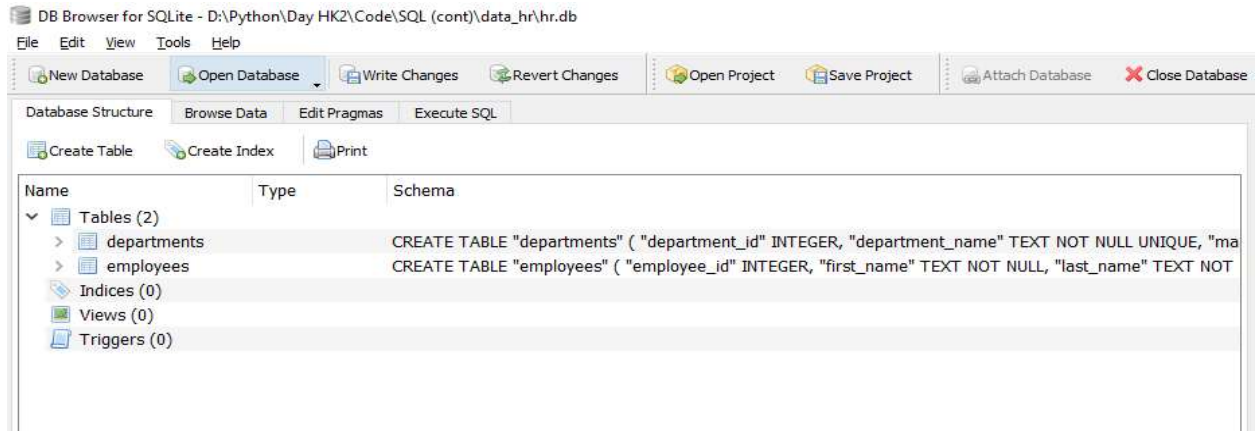
```

1 CREATE TABLE "employees" (
2     "employee_id"    INTEGER,
3     "first_name"      TEXT NOT NULL,
4     "last_name"       TEXT NOT NULL,
5     "email"           TEXT,
6     "phone_number"    TEXT,
7     "hire_date"       TEXT,
8     "salary"          NUMERIC

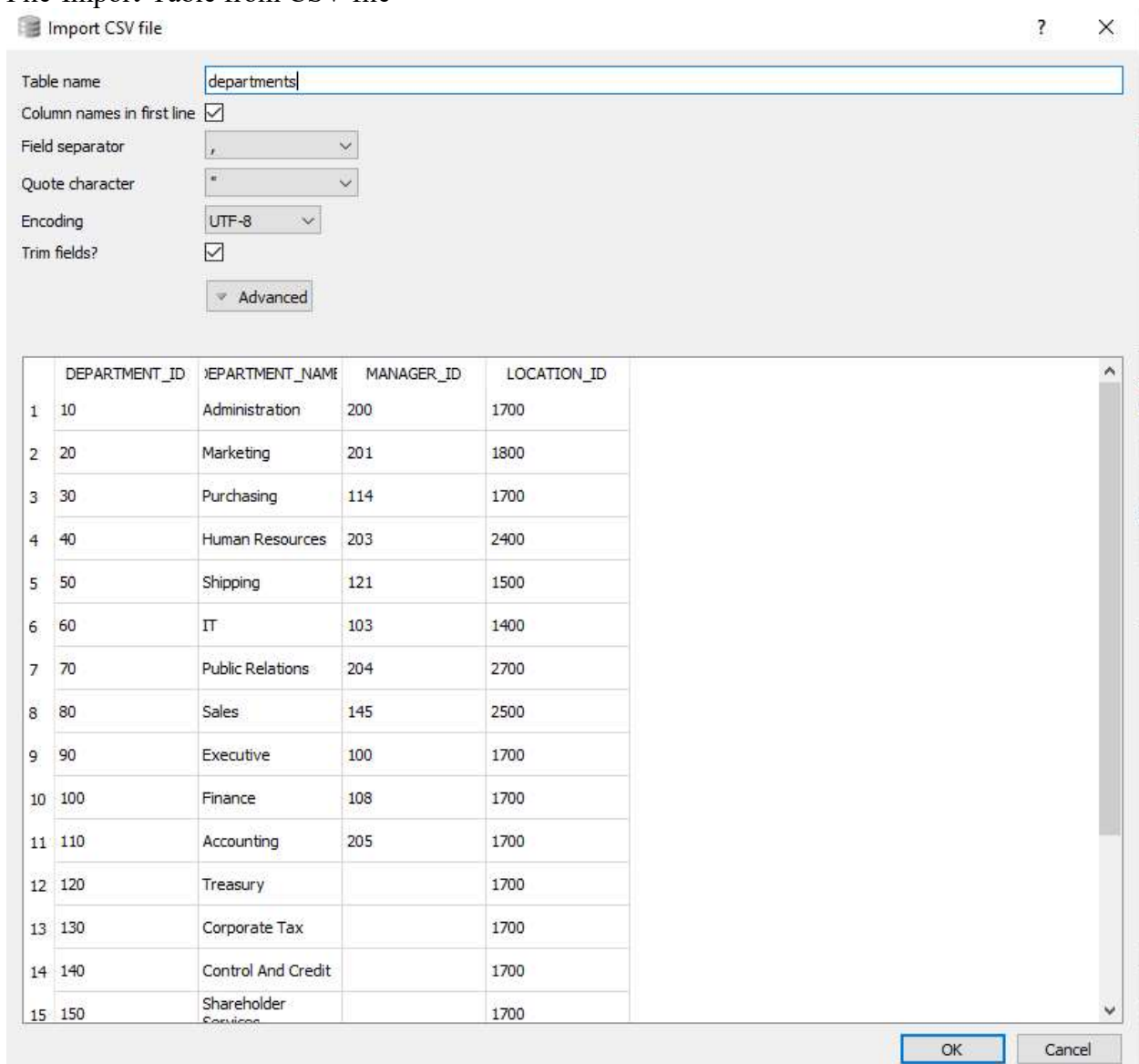
```

OK Cancel

Khi tạo xong ta sẽ thấy có 2 table.



Bước 3: Import dữ liệu csv vào 2 table:
File-Import-Table from CSV file



Import CSV file

Table name:

Column names in first line: ☒

Field separator:

Quote character:

Encoding:

Trim fields?: ☒

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	
1	100	Steven	King	SKING	515.123.4567	2003-06-17 00:00:00	AD_PRES	2400
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21 00:00:00	AD_VP	1700
3	102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13 00:00:00	AD_VP	1700
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03 00:00:00	IT_PROG	9000
5	104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21 00:00:00	IT_PROG	6000
6	105	David	Austin	DAUSTIN	590.423.4569	2005-06-25 00:00:00	IT_PROG	4800
7	106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05 00:00:00	IT_PROG	4800
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07 00:00:00	IT_PROG	4200
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	2002-08-17 00:00:00	FI_MGR	1200
10	109	Daniel	Faviet	DFAVIET	515.124.4169	2002-08-16 00:00:00	FI_ACCOUNT	9000
11	110	John	Chen	JCHEN	515.124.4269	2005-09-28 00:00:00	FI_ACCOUNT	8200
12	111	Ismael	Sciarra	ISCIARRA	515.124.4369	2005-09-30 00:00:00	FI_ACCOUNT	7700
13	112	Jose Manuel	Urman	JMURMAN	515.124.4469	2006-03-07 00:00:00	FI_ACCOUNT	7800
14	113	Luis	Popp	LPOPP	515.124.4567	2007-12-07 00:00:00	FI_ACCOUNT	6900

Nhớ check vào mục Column names in first line và Trim fields

Lưu ý: Nếu ta có file scripts SQL. Nội dung file scrip SQL ở trường hợp trên là:

CREATE TABLE departments

(
 department_id int primary key,
 department_name text not null unique,
 manager_id int,
 location_id int

);

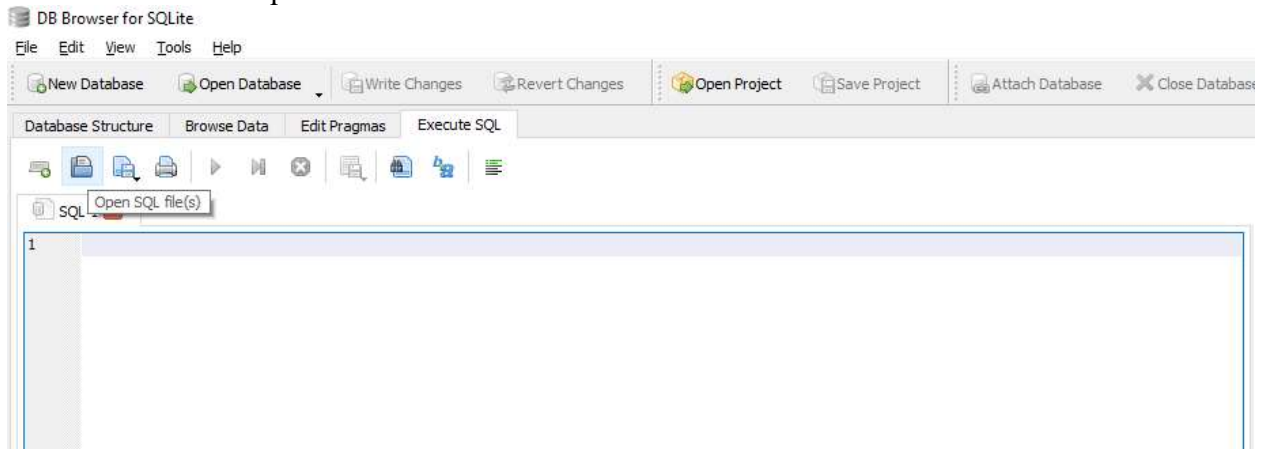
CREATE TABLE employees

(
 employee_id int primary key,
 first_name text not null,

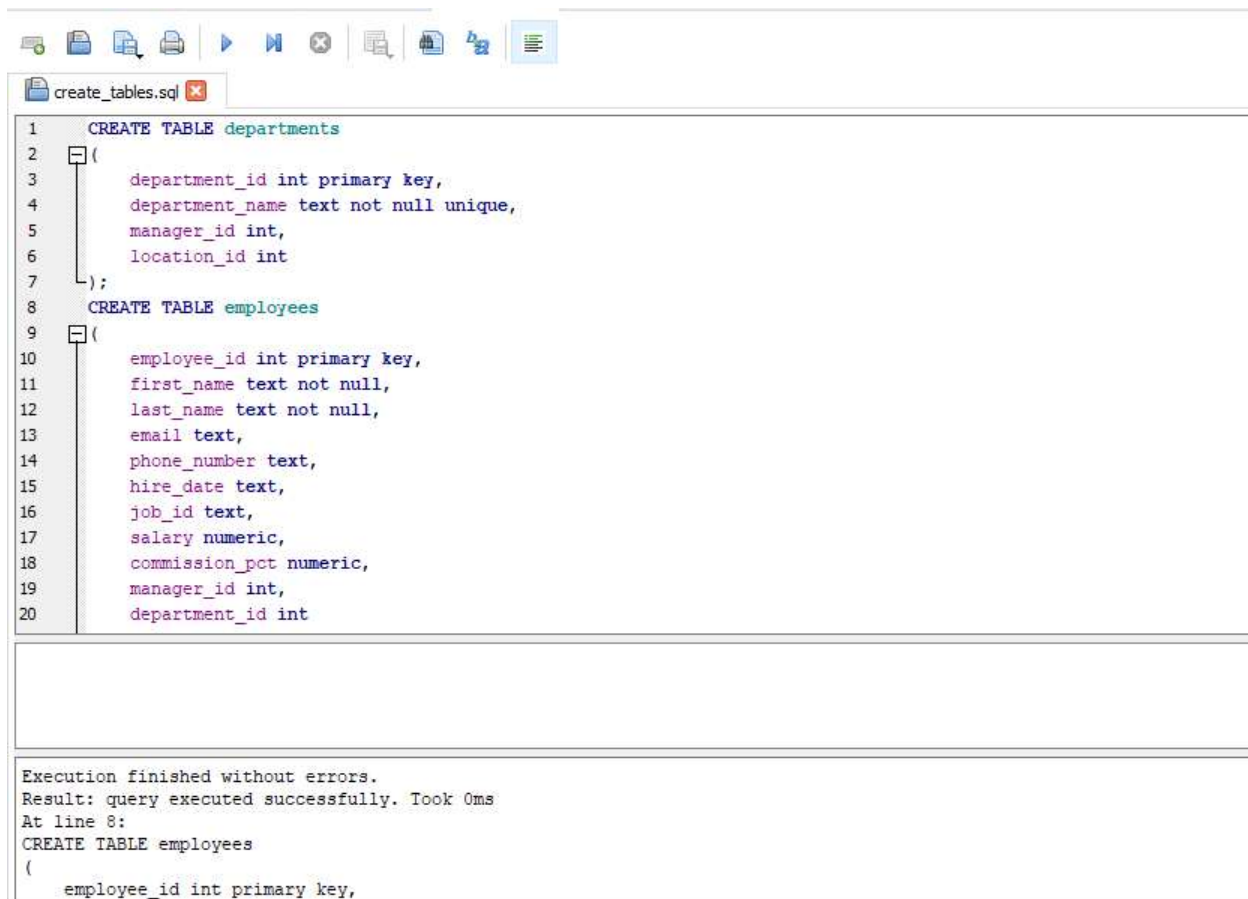
```
last_name text not null,  
email text,  
phone_number text,  
hire_date text,  
job_id text,  
salary numeric,  
commission_pct numeric,  
manager_id int,  
department_id int  
);
```

File này là create_tables.sql thì ở bước 2 sẽ làm như sau:

Bước 2: Mở file *.sql



Sau đó nhấn F5 (run)



```
1 CREATE TABLE departments
2 (
3     department_id int primary key,
4     department_name text not null unique,
5     manager_id int,
6     location_id int
7 );
8 CREATE TABLE employees
9 (
10     employee_id int primary key,
11     first_name text not null,
12     last_name text not null,
13     email text,
14     phone_number text,
15     hire_date text,
16     job_id text,
17     salary numeric,
18     commission_pct numeric,
19     manager_id int,
20     department_id int
21 );
```

Execution finished without errors.
Result: query executed successfully. Took 0ms
At line 8:
CREATE TABLE employees
(
 employee_id int primary key,

Nhấn vào Write_Changes để cập nhật thay đổi. Các bước khác làm tương tự như trên.

II. Code Python bằng cách sử dụng CSDL đã được import:

Dựa vào CSDL ở trên viết 1 số truy vấn cơ bản:

Ví dụ 1: Liệt kê các nhân viên gồm: mã nhân viên, tên (first_name, last_name), ngày vào làm, lương.

```
import sqlite3
import pandas as pd
dbase = sqlite3.connect("hr.db")
sql = """SELECT employee_id, first_name||' '||last_name AS Name, hire_date, salary FROM
employees;"""
df = pd.read_sql_query(sql, dbase)
df.head()
print(df.head())
dbase.close()
```

Ví dụ 2: Liệt kê các nhân viên có thêm cột biểu thức tham_nien (thâm niên = năm hiện tại - năm vào làm), sắp xếp giảm theo thâm niên.

```
import sqlite3
import pandas as pd
dbase = sqlite3.connect("hr.db")
```



```

sql = "SELECT employee_id, first_name, hire_date, strftime('%Y', 'now') - strftime('%Y',
hire_date) AS tham_nien
FROM employees
ORDER BY tham_nien DESC;"
df = pd.read_sql_query(sql, dbase)
df.head()
print(df.head())
dbase.close()

```

III. Bài tập

Với cơ sở dữ liệu như bài ví dụ ở trên dùng python thực hiện 1 số truy vấn sau:

Bài 1: Cho biết nhân viên có lương cao nhất.

Bài 2: Cho biết nhân viên có thâm niên lâu nhất của phòng 100

Bài 3: Liệt kê các nhân viên có lương cao nhất của mỗi phòng.

Bài 4: Liệt kê theo mã phòng và đếm số nhân viên, tổng lương theo phòng

Bài 5: Sắp xếp các phòng có tổng lương từ cao xuống thấp

Bài 6: Liệt kê theo mã phòng, tên phòng và đếm số nhân viên, tổng lương theo phòng

Bài 7: Liệt kê các nhân viên có lương > trung bình lương

Bài 8: Tìm các nhân viên làm việc tại vị trí phòng là 1700

Bài 9: Hiển thị tên trưởng phòng của phòng 50

Bài 10: Liệt kê các họ tên nhân viên có ngày vào làm việc vào thứ 2.