

터미널 또는 프롬프트 창에서 SQL flights table 만들기!

(강의노트 출처 : <https://cs50.harvard.edu/web/2020/notes/4/>)

1. 새로운 프로젝트 만들기 - `django-admin startproject SecondProject` 입력

```
PS C:\Users\woo\Desktop\HNU\program\KDT> django-admin startproject SecondProject
```

2. `cd SecondProject` 로 경로 지정한 후, `python manage.py migrate` 입력

```
PS C:\Users\woo\Desktop\HNU\program\KDT> cd SecondProject
PS C:\Users\woo\Desktop\HNU\program\KDT\SecondProject> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
```

3. `sqlite3 flights.sql` 입력

```
PS C:\Users\woo\Desktop\HNU\program\KDT\SecondProject> sqlite3 flights.sql
SQLite version 3.36.0 2021-06-18 18:36:39
Enter ".help" for usage hints.
```

SQL Table 만들기

4. 아래 코드와 같이 입력 한 후, `.tables`로 만든 table flights 를 출력한다.

```
sqlite> CREATE TABLE flights(
...> id INTEGER PRIMARY KEY AUTOINCREMENT,
...> origin TEXT NOT NULL,
...> destination TEXT NOT NULL,
...> duration INTEGER NOT NULL);
sqlite> .table
flights
```

5. 전체 flights table 을 출력한다. 아직 table에 입력된 값이 없기 때문에 실행해도 아무것도 출력되지 않는다.

```
sqlite> select * from flights ;
sqlite> select * from flights;
```

6. 값 삽입하고 전체 flights table을 출력하면 아래와 같이 결과가 나온다.

```
sqlite> insert into flights(origin, destination, duration) values("New York", "London", 415);
sqlite> select * from flights;
1|New York|London|415
```

7. 나머지 값을 삽입하고, table을 확인하면 아래와 같이 나온다.

```
sqlite> NSERT INTO flights (origin, destination, duration) VALUES ("Shanghai", "Paris", 760);
Error: near "NSERT": syntax error
sqlite> INSERT INTO flights (origin, destination, duration) VALUES ("Shanghai", "Paris", 760);
sqlite> INSERT INTO flights (origin, destination, duration) VALUES ("Istanbul", "Tokyo", 700);
sqlite> INSERT INTO flights (origin, destination, duration) VALUES ("New York", "Paris", 435);
sqlite> INSERT INTO flights (origin, destination, duration) VALUES ("Moscow", "Paris", 245);
sqlite> INSERT INTO flights (origin, destination, duration) VALUES ("Lima", "New York", 455);
sqlite> select * from flights;
1|New York|London|415
2|Shanghai|Paris|760
3|Istanbul|Tokyo|700
4|New York|Paris|435
5|Moscow|Paris|245
6|Lima|New York|455
```

8. flights table 표스타일로 만들기

```
sqlite> .mode columns
sqlite> .headers yes
sqlite> .model columns
Error: unknown command or invalid arguments: "model". Enter ".help" for help
```

.model columns를 입력했을 때, unknown command or invalid arguments 오류가 나면 table의 스타일이 바뀌어져있다.

```
sqlite> select * from flights;
id  origin      destination  duration
--  -
1   New York    London       415
2   Shanghai    Paris        760
3   Istanbul    Tokyo        700
4   New York    Paris        435
5   Moscow      Paris        245
6   Lima        New York     455
```

SQL 맛보기!!

일부만 select 해서 불러오기

SELECT * FROM flights WHERE origin = "New York";

```
sqlite> SELECT * FROM flights WHERE origin = "New York";
id  origin    destination  duration
--  -
1   New York  London       415
4   New York  Paris        435
```

duration이 500 이상인 것만 출력

SELECT * FROM flights WHERE **duration** > 500;

```
sqlite> SELECT * FROM flights WHERE duration > 500;
id  origin    destination  duration
--  -
2   Shanghai  Paris        760
3   Istanbul  Tokyo        700
```

and, or 연산자

SELECT * FROM flights WHERE duration > 500 **AND** destination = "Paris";

SELECT * FROM flights WHERE duration > 500 **OR** destination = "Paris";

```
sqlite> SELECT * FROM flights WHERE duration > 500 AND destination = "Paris";
id  origin    destination  duration
--  -
2   Shanghai  Paris        760
sqlite> SELECT * FROM flights WHERE duration > 500 OR destination = "Paris";
id  origin    destination  duration
--  -
2   Shanghai  Paris        760
3   Istanbul  Tokyo        700
4   New York  Paris        435
5   Moscow    Paris        245
```

in 연산자

SELECT * FROM flights WHERE origin **IN** ("New York", "Lima");

```
sqlite> SELECT * FROM flights WHERE origin IN ("New York", "Lima");
id  origin    destination  duration
--  -
1   New York  London       430
4   New York  Paris        435
6   Lima      New York     455
```

like 연산자

SELECT * FROM flights WHERE origin **LIKE** "%a%";

```
sqlite> SELECT * FROM flights WHERE origin LIKE "%a%";
id  origin    destination  duration
--  -
2   Shanghai  Paris        760
3   Istanbul  Tokyo        700
6   Lima      New York     455
```

→ origin에 'a'가 포함되어 있는 것만 출력

update 연산자

UPDATE flights

SET duration = 430

WHERE origin = "New York"

AND destination = "London";

```
sqlite> UPDATE flights
...> SET duration = 430 ← 바꿀 값
...> WHERE origin = "New York"
...> AND destination = "London"; ] 바꿀 값의 위치

sqlite> select * from flights;
id  origin    destination  duration
--  -
1   New York  London      430
2   Shanghai  Paris       760
3   Istanbul  Tokyo       700
4   New York  Paris       435
5   Moscow    Paris       245
6   Lima      New York    455
```

제거하기

DELETE FROM flights **WHERE** destination = "Tokyo";

```
sqlite> DELETE FROM flights WHERE destination = "Tokyo";
sqlite> select * from flights;
id  origin    destination  duration
--  -
1   New York  London      430
2   Shanghai  Paris       760
4   New York  Paris       435
5   Moscow    Paris       245
6   Lima      New York    455
```

** sql에서 빠져나올 때, **ctrl + c** 하면 된다.

```
sqlite> (ctrl+c) ↵
PS C:\Users\woo\Desktop\HNU\program\KDT\SecondProject>
```

SQL문의 취약점

SQL을 사용하지 않고 Django ORL로 사용하는 이유!

>> 유저 아이디만 알면 비밀번호까지 입력하지 않고 로그인을 할 수 있기 때문에 해킹당할 문제가 있다.