FLASK WEB STUDY - SESSION 01

웹 개발의 시작
데이터베이스(DB) 기초 이론

데이터의 저장, 관리 그리고 Python과의 연동

Presented by
APS_김영빈

Today's Agenda (120min)

PART 1. 준비 및 이론 (30분)

- 실습 환경 설정 (SQLite)
- DB vs File, RDBMS vs NoSQL
- SQL 명령어 종류 (DDL, DML)

PART 2. [실습] SQL 완전 정복 (40분)

- 테이블 설계 및 생성 (CREATE)
- C.R.U.D 데이터 조작 실습

PART 3. [Challenge] DB 설계 (30분)

- **Thinking Time:** 게시판 테이블 설계하기
- PK와 날짜 자동 생성의 비밀
- Join의 개념 (간단히)

PART 4. Python 연동 (20분)

- `sqlite3` 라이브러리 활용
- Python 코드로 DB 제어하기

실습 환경 설정

가볍고 강력한 SQLite를 설치해 봅시다.

Windows 사용자 (추천)

PowerShell 또는 CMD 창을 열고 아래 명령어를 입력하세요.

```
winget install SQLite.SQLite
```

Copy

* 설치 후 터미널을 재시작하고 `sqlite3`를 입력했을 때 실행되면 성공!

Alternative (설치가 안 될 경우)

1. sqliteonline.com 접속
2. 별도 설치 없이 웹 브라우저에서 바로 실습 가능

왜 데이터베이스(DB)인가?

파일 시스템 (File)

- ✕ 데이터 중복 발생
- ✕ 동시 접속 시 파일 깨짐
- ✕ 보안 관리의 어려움

데이터베이스 (DB)

- ✓ **무결성:** 정확한 데이터 유지
- ✓ **동시성:** 여러 명 동시 사용 가능
- ✓ **표준 언어(SQL):** 체계적인 관리

SQL 명령어의 두 가지 축

DDL

Data Definition Language

데이터를 담을 **그릇(구조)**을 만드는 언어

`CREATE` : 생성

`ALTER` : 구조 변경

`DROP` : 삭제 (주의!)

DML

Data Manipulation Language

그릇 안에 담긴 **내용물(데이터)**을 다루는 언어

`INSERT` : 넣기 (C)

`SELECT` : 꺼내기 (R) ★

`UPDATE` : 고치기 (U)

`DELETE` : 버리기 (D)

Step 1. 테이블 만들기 (DDL)

테이블 설계도 (Schema)

Column Name	Type	Constraint
id	INTEGER	PRIMARY KEY (주민번호 역할)
username	TEXT	NOT NULL (빈칸 금지)
age	INTEGER	

⌨ 실습 미션

터미널에 `sqlite3 test.db` 입력 후 아래 SQL을 작성하세요.

```
CREATE TABLE users (
    id INTEGER PRIMARY KEY,
    username TEXT NOT NULL,
    age INTEGER
);
```

데이터를 다루는 4가지 동작: C.R.U.D

C

Create

데이터 생성

`INSERT`

R

Read

데이터 조회

`SELECT`

U

Update

데이터 수정

`UPDATE`

D

Delete

데이터 삭제

`DELETE`

"게시판의 글쓰기(C), 글읽기(R), 글수정(U), 글삭제(D)와 100% 일치합니다."

Step 2. 데이터 채우기 (C - INSERT)

```
-- 기본 문법
INSERT INTO 테이블명 (컬럼1, 컬럼2)
VALUES (값1, 값2);
```

- ✓ 문자열은 반드시 **작은따옴표(')** 사용
- ✓ 숫자는 따옴표 없이 입력
- ✓ 모든 컬럼을 넣을 땐 컬럼명 생략 가능

⌨ 실습 미션

다음 3명의 데이터를 추가해 보세요.

1. 'Kim', 22세
2. 'Lee', 24세
3. 'Park', 20세

```
INSERT INTO users (username, age) VALUES ('Kim', 22);
INSERT INTO users (username, age) VALUES ('Lee', 24);
INSERT INTO users (username, age) VALUES ('Park', 20);
```

Step 3. 데이터 조회 (R - SELECT)

DB에서 가장 많이 쓰이는 명령어입니다.

1. 모든 데이터 보기

```
SELECT * FROM users;
```

2. 특정 컬럼만 보기

```
SELECT username FROM users;
```

3. 조건 검색 (Filtering) ★

```
SELECT * FROM users WHERE age >= 22;
```

도전 과제

Q1. 나이가 23살 이상인 사람만 찾아보세요.

```
SELECT * FROM users WHERE age >= 23;
```

Q2. 이름이 'Kim'인 사람의 나이는?

```
SELECT age FROM users WHERE username = 'Kim';
```

Step 4. 수정과 삭제 (U & D)

⚠ 경고: WHERE 절 없이 실행하면 모든 데이터가 수정/삭제됩니다!

UPDATE (수정)

```sql
UPDATE users
SET age = 25
WHERE username = 'Kim';
```

Mission: 'Kim'의 나이를 25살로 수정하세요.

DELETE (삭제)

```sql
DELETE FROM users
WHERE username = 'Park';
```

Mission: 'Park' 데이터를 삭제하세요.

Challenge Mode

Thinking Time: 게시판 설계하기

Q. 게시판(Board)을 만들려면?

사용자 정보는 `users` 테이블에 있습니다.
이제 게시글을 저장할 `posts` 테이블을 만들어야 합니다.

어떤 **컬럼(열)**들이 필요할까요?

- ✓ 게시글을 구별할 번호가 필요해 (ID)
- ✓ 글 제목과 내용은 당연히 있어야지
- ○ 누가 썼는지 어떻게 알지? (작성자)
- ○ 언제 썼는지 기록해야 하지 않을까? (작성일)

잠시 멈추고
종이나 메모장에 적어보세요.

Mission 05: 게시판 테이블 생성

권장 설계안 (Schema)

id: INTEGER PK (AUTOINCREMENT)

title: TEXT NOT NULL

content: TEXT

user_id: 작성자의 ID (Foreign Key 개념)

created_at: 작성일 (자동생성)

★ 꿀팁 1: AUTOINCREMENT
값을 안 넣으면 1, 2, 3... 번호표를 자동으로 뽑아줍니다.

★ 꿀팁 2: DEFAULT CURRENT_TIMESTAMP
값을 안 넣으면 '현재 시간'을 자동으로 찍어줍니다.

⌨ 최종 미션 SQL

```sql
CREATE TABLE posts (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    title TEXT NOT NULL,
    content TEXT,
    user_id INTEGER,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);

-- 데이터 넣어보기 (날짜, ID 생략 가능!)
INSERT INTO posts (title, content, user_id)
VALUES ('첫번째 글', '안녕하세요', 1);

SELECT * FROM posts;
```

Python에서 DB 다루기

```python
import sqlite3

# 1. 연결 (없으면 파일 생성됨)
conn = sqlite3.connect('test.db')
cur = conn.cursor()

# 2. SQL 실행 (우리가 배운 쿼리 그대로!)
cur.execute("SELECT * FROM users")

# 3. 결과 받아서 출력
for row in cur.fetchall():
    print(f"이름: {row[1]}, 나이: {row[2]}")

conn.close()
```

Connection

Python과 DB 파일 사이의 전화 연결

Cursor

SQL을 배달하고 데이터를 가져오는 심부름꾼

Tip: Flask에서는 이 과정을 더 편하게 해주는 도구(ORM)를 사용합니다.

강의 요약 & Next Step

DB & SQL

데이터 저장소의 구조와 CRUD 명령어를 익혔습니다.

Python 연동

코드로 데이터를 넣고 빼는 법을 배웠습니다.

Next: Flask

이제 이 DB를 웹페이지와 연결해볼 차례입니다.

수고하셨습니다!

Questions?