

Class 4. SQL 기초

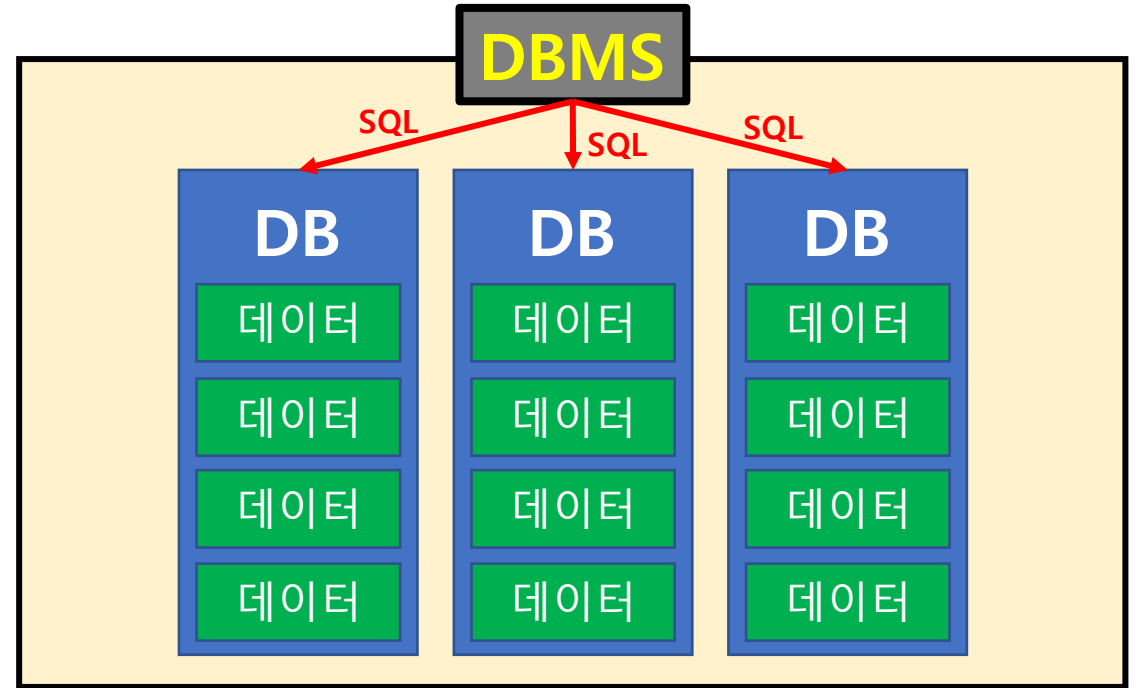
데이터베이스란?

데이터(data, 자료): 숫자, 문자, 소리, 이미지, 영상 등의 형태로 된 의미 단위.

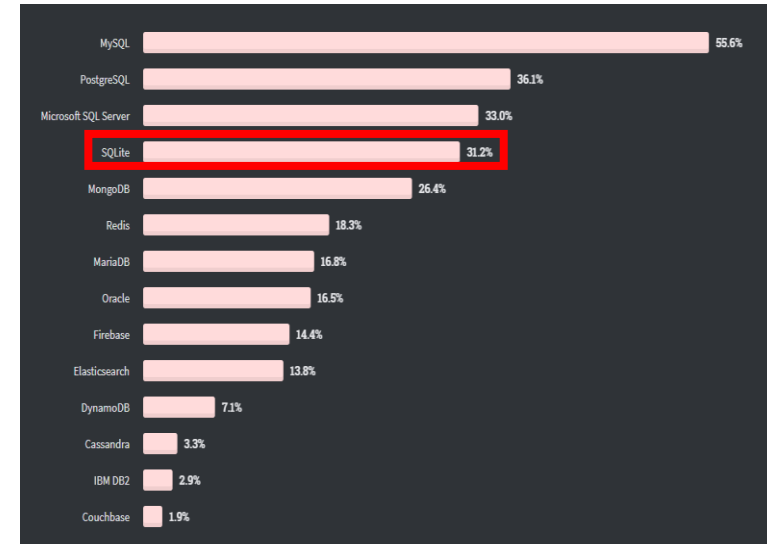
데이터베이스(database, DB): 데이터가 저장되어 있는 공간. 데이터를 모아둔 집합.

DBMS(database management system): 데이터베이스를 관리하고 운영하기 위한 시스템.

SQL(structured query language, 구조화된 질의 언어): 데이터베이스를 구축하고 관리하고 활용하기 위해서 사용되는 언어.



세상에는 많은 종류의 DBMS가 존재.
하지만 SQL을 알면 Oracle, MySQL, MariaDB, SQLite,
Google Cloud SQL, Amazon RDS, MS SQL Server, PostgreSQL
등의 DBMS를 다룰 수 있음.



출처: Stack Overflow Developer Survey, 2020

DB는 여러 개의 테이블로 구성되어 있다

people 테이블: 선수 데이터

pitching 테이블: 투수 데이터

ex) 레먼 데이터베이스

salaries 테이블: 연봉 데이터

fielding 테이블: 수비 데이터

batting 테이블: 타자 데이터

테이블은 여러 개의 행(row)과 열(column)로 구성된다

idx	id	password	name	phone	age
1	tlarygns0211	12345678**	심교훈	010-1234-5678	35
2	rlaroehd77	ashdjfkhk!	김개똥	010-9876-5432	29
3	ghdrlehd01	pohuhh000	홍길동	010-3333-4444	88
4	young89su	000000susu1	영수	010-5678-1234	34
			...		

2차원 배열(표)의 형태:
인류가 발명한 최고의 데이터 정리 형태
ex) 엑셀, Sqlite3, 판다스 데이터프레임 등

SQLite3 특징

일반적으로 다른 SQL 데이터베이스 엔진들은 static data type을 제공하지만, SQLite는 dynamic data type을 제공한다
컬럼에 지정된 타입과 저장된 데이터의 타입이 다를 수 있다는 뜻

데이터 타입	설명
NULL	NULL 값
INTEGER	정수
REAL	실수(부동 소수점 숫자)
TEXT	텍스트
BLOB	binary large object 입력 데이터를 그대로 저장

컬럼 데이터 타입	설명
TEXT	NULL, TEXT, BLOB 허용
NUMERIC	모든 타입 가능
INTEGER	NUMERIC과 동일 단, 실수(12.4)는 정수(12)로 변환해서 저장
REAL	NUMERIC과 동일 단, 정수(12)는 실수(12.0)로 변환해서 저장
NONE	특정 타입을 선호하지 않음. 강제 변환 X.

우분투에 SQLite3 설치하기

우리는 DBMS 중에 하나인 SQLite3로 실습 진행

SQLite: "에스큐엘라이트" 또는 "시퀼라이트"라고 부름

터미널에 다음과 같이 입력

```
sudo apt-get install sqlite3 libsqlite3-dev
```

- 1) sudo: super user do의 약자. 가장 강력한 권한을 가진 존재가 처리한다는 뜻.
- 2) apt-get: 우분투(Ubuntu)와 같은 데비안 계열의 리눅스 운영체제에서 사용되는 패키지 관리 도구.

터미널에 `sqlite3 --version` 를 입력했을 때 다음과 같은 결과가 나온다면 제대로 설치된 것

```
root@goorm:/workspace/KUSF_data# sqlite3 --version
3.22.0 2018-01-22 18:45:57 0c55d179733b46d8d0ba4d88e01a25e10677046ee3da1d5b1581e86726f2a1t1
```

데이터베이스 열기

sqlite3 DB파일명

ex) sqlite3 test.sqlite

데이터베이스 내 테이블 확인

.tables

자세한 내용은 강사 블로그 확인
<https://bskyvision.com/958>

SQL 기초: 테이블 생성

테이블 생성

```
CREATE TABLE 테이블명 (컬럼명1 데이터타입, 컬럼명2 데이터타입,...);
```

ex) CREATE TABLE people (idx INTEGER, name TEXT, height REAL, weight REAL);

```
sqlite> CREATE TABLE people (idx INTEGER, name TEXT, height REAL, weight REAL);
```

```
sqlite> .tables  
people
```

people 테이블 생성 되었음

테이블 구조 확인

```
.schema 테이블명
```

ex) .schema people

```
sqlite> .schema people  
CREATE TABLE people (idx INTEGER, name TEXT, height REAL, weight REAL);
```

people 테이블 구조

idx	name	height	weight

SQL 기초: 데이터 입력

데이터 입력

INSERT INTO 테이블명 VALUES (데이터1, 데이터2, ...);

ex) INSERT INTO people VALUES (1, '홍길동', 179.2, 78.5);

```
sqlite> INSERT INTO people VALUES (1, '홍길동', 179.2, 78.5);  
sqlite> INSERT INTO people VALUES (2, '임꺽정', 175.1, 70.4);  
sqlite> INSERT INTO people VALUES (3, '장영실', 169.8, 73.2);  
sqlite> INSERT INTO people VALUES (4, '이순신', 185.5, 89.6);
```

입력 컬럼 순서를 다르게 하려면?

INSERT INTO 테이블명 (컬럼명3, 컬럼명1,...) VALUES (데이터3, 데이터1, ...);

ex) INSERT INTO people (name, height, idx, weight) VALUES ('김개똥', 189.2, 5, 95.9);

```
sqlite> INSERT INTO people (name, height, idx, weight) VALUES ('김개똥', 189.2, 5, 95.9);  
sqlite> 
```

에러 메시지 없이 sqlite> 가 나오면 잘 입력된 것

SQL 기초: 데이터 조회

데이터 조회

SELECT 컬럼명1, 컬럼명2, ... FROM 테이블명;

ex) SELECT idx, name, height, weight FROM people;

```
sqlite> SELECT idx, name, height, weight FROM people;  
1|홍길동|179.2|78.5  
2|임꺽정|175.1|70.4  
3|장영실|169.8|73.2  
4|이순신|185.5|89.6
```



컬럼명도 같이 나오게 하려면?

.header on

```
sqlite> .header on  
sqlite> SELECT idx, name, height, weight FROM people;  
idx|name|height|weight  
1|홍길동|179.2|78.5  
2|임꺽정|175.1|70.4  
3|장영실|169.8|73.2  
4|이순신|185.5|89.6
```

조회 결과 출력 형식
ascii
column
csv
html
Insert
list
quote
tabs
tcl

기본 출력 모드

컬럼 별로 정리되어 출력되게 하려면?

출력 모드를 column으로 설정

.mode column

```
sqlite> .mode column  
sqlite> SELECT idx, name, height, weight FROM people;  
idx      name      height    weight  
-----  
1        홍길동    179.2     78.5  
2        임꺽정    175.1     70.4  
3        장영실    169.8     73.2  
4        이순신    185.5     89.6
```

SQL 기초: 데이터 조회

데이터 조회

모든 컬럼 데이터를 보고 싶다면? *는 모든 컬럼을 뜻함

```
SELECT * FROM 테이블명;
```

ex) SELECT * FROM people;

```
sqlite> SELECT * FROM people;
```

idx	name	height	weight
1	홍길동	179.2	78.5
2	임꺽정	175.1	70.4
3	장영실	169.8	73.2
4	이순신	185.5	89.6

조건을 가지고 데이터 조회

where 사용

```
SELECT * FROM people WHERE 조건내용;
```

ex) SELECT * FROM people WHERE height >= 180;

idx	name	height	weight
4	이순신	185.5	89.6

키가 180이상인 사람의 데이터만 출력되었음

SQL 기초: 데이터 수정

데이터 수정

UPDATE 테이블명 SET 컬럼명1 = 새데이터1, 컬럼명2 = 새데이터2 WHERE 조건;

ex)

만약 장영실의 키와 몸무게를 잘못 입력해서 수정하고 싶다면,

UPDATE people SET height = 159.8, weight = 63.2 WHERE name = '장영실';

```
sqlite> SELECT * FROM people;
idx      name      height  weight
-----
1        홍길동    179.2   78.5
2        임꺽정    175.1   70.4
3        장영실    169.8   73.2
4        이순신    185.5   89.6
sqlite> UPDATE people SET height = 159.8, weight = 63.2 WHERE name = '장영실';
sqlite> SELECT * FROM people;
idx      name      height  weight
-----
1        홍길동    179.2   78.5
2        임꺽정    175.1   70.4
3        장영실    159.8   63.2
4        이순신    185.5   89.6
```

키 169.8 => 159.8
몸무게 73.2 => 63.2
로 수정되었음

SQL 기초: 데이터 삭제

데이터 삭제

DELETE FROM 테이블명 WHERE 조건;

ex) DELETE FROM people WHERE idx = 2;

```
sqlite> DELETE FROM people WHERE idx = 2;
sqlite> SELECT * FROM people;
idx      name      height   weight
-----
1        홍길동     179.2    78.5
3        장영실     159.8    63.2
4        이순신     185.5    89.6
```

idx 가 2였던 임꺽정 데이터 행이 삭제되었음

모든 데이터 삭제

조건을 주지 않으면 모든 데이터가 삭제됨

DELETE FROM 테이블명;

```
sqlite> DELETE FROM people;
sqlite> SELECT * FROM people;
sqlite> 
```

조회했지만 아무런 데이터가 없음

SQL 기초: 테이블 삭제

테이블 삭제

DROP TABLE 테이블명;

ex) DROP TABLE people;

```
sqlite> .tables
```

```
people
```

```
sqlite> DROP TABLE people;
```

```
sqlite> .tables
```

```
sqlite> 
```

PRIMARY KEY AUTOINCREMENT

PRIMARY KEY로 설정된 컬럼의 값은 중복될 수 없음, 직접 입력하지 않아도 저절로 1씩 증가
AUTOINCREMENT를 설정해주면 행이 입력될 때마다 자동으로 1씩 증가

테이블 생성(primary key, autoincrement 활용)

```
CREATE TABLE people (idx INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT, height REAL, weight REAL);
```

데이터 입력

```
INSERT INTO people (name, height, weight) VALUES ('홍길동', 179.2, 78.5);  
INSERT INTO people (name, height, weight) VALUES ('임꺽정', 175.1, 78.4);  
INSERT INTO people (name, height, weight) VALUES ('장영실', 169.8, 73.2);  
INSERT INTO people (name, height, weight) VALUES ('이순신', 185.5, 89.6);
```

sqlite> SELECT * FROM people;			
idx	name	height	weight
1	홍길동	179.2	78.5
2	임꺽정	175.1	78.4
3	장영실	169.8	73.2
4	이순신	185.5	89.6

데이터 삭제 후 입력

```
DELETE FROM people WHERE idx = 4;  
INSERT INTO people (name, height, weight) VALUES ('김유신', 192.4, 81.7);
```

sqlite> SELECT * FROM people;			
idx	name	height	weight
1	홍길동	179.2	78.5
2	임꺽정	175.1	78.4
3	장영실	169.8	73.2
5	김유신	192.4	81.7

autoincrement가 없었으면 5가 아닌 4가 들어갔을 것

러너 학생 정보 테이블을 만들어보자

idx, 이름, 나이, 조번호, 조이름, 전공, 취미, MBTI, 좋아하는스포츠종목

idx	이름	나이	조번호	전공	취미	MBTI	좋아하는스포츠
1	심교훈	21	1	전자공학	코딩	ESFP	축구
2	손지현	20					
3							
4							

1. 컬럼마다 적절한 데이터타입을 지정해서 KUSF 학생 테이블을 생성해보세요. (primary key, autoincrement 활용할 것)
2. 20명 학생의 데이터를 전부 넣어보세요.
3. 자신이 속한 조 학생들의 데이터만 조회해보세요.
4. UPDATE를 활용하여 데이터 일부를 수정해보세요.
5. MBTI가 ISTJ, ESFP인 학생의 데이터를 삭제해보세요. (ISTJ, ESFP 미워하는 것 아닙니다...ㅠㅎㅎ 단지 예시로..)