

K-Digital Training

웹 풀스택 과정

# 데이터베이스2

WITH 팀 리처드



# SQL문

# SQL문

- Structured Query Language
- 구조적 질의 언어
- 데이터베이스를 **제어하고 관리할 수 있는 목적의 프로그래밍 언어**



DDL



DML



DCL

DDL

# SQL문 – DDL

- DDL ( Data **Definition** Language )
- 데이터베이스 or 테이블을 **정의**하는 언어

종류	역할
CREATE	데이터베이스, 테이블 등을 생성하는 역할을 한다.
ALTER	테이블을 수정하는 역할을 한다.
DROP	데이터베이스, 테이블을 삭제하는 역할을 한다.
TRUNCATE	테이블을 초기화 시키는 역할을 한다.

# DDL - CREATE

## 데이터베이스 만들기 + 한글 인코딩

한글인코딩?

프로그래밍 언어는 영어 기반이므로 한글을 사용할 수 있도록!!

```
CREATE DATABASE 이름 DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;
```

## 테이블 만들기

```
CREATE TABLE 테이블명 (
    필드1 값형식,
    필드2 값형식 |
);
```

주의) 명령어 끝에는 세미콜론(;)을 붙여야한다!!

문장이 끝났음을 알려주기 위함

# 문자형 데이터 형식

<u>CHAR(n)</u>	고정 길이 데이터 타입(최대 255byte)- 지정된 길이보다 짧은 데이터 입력될 시 나머지 공간 공백으로 채워진다.
<u>VARCHAR(n)</u>	가변 길이 데이터 타입(최대 65535byte)- 지정된 길이보다 짧은 데이터 입력될 시 나머지 공간은 채우지 않는다.
<u>TINYTEXT(n)</u>	문자열 데이터 타입(최대 255byte)
<u>TEXT(n)</u>	문자열 데이터 타입(최대 65535byte)
<u>MEDIUMTEXT(n)</u>	문자열 데이터 타입(최대 16777215byte)
<u>LONGTEXT(n)</u>	문자열 데이터 타입(최대 4294967295byte)

# 숫자형 데이터 형식

<u>TINYINT</u> (n)	정수형 데이터 타입(1byte) -128 ~ +127 또는 0 ~ 255수 표현 가능하다.
<u>SMALLINT</u> (n)	정수형 데이터 타입(2byte) -32768 ~ 32767 또는 0 ~ 65536수 표현 가능하다.
<u>MEDIUMINT</u> (n)	정수형 데이터 타입(3byte) -8388608 ~ +8388607 또는 0 ~ 16777215수 표현 가능하다.
<u>INT</u> (n)	정수형 데이터 타입(4byte) -2147483648 ~ +2147483647 또는 0 ~ 4294967295수 표현 가능하다.
<u>BIGINT</u> (n)	정수형 데이터 타입(8byte) - 무제한 수 표현 가능하다.
<u>FLOAT</u> (길이,소수)	부동 소수형 데이터 타입(4byte) -고정 소수점을 사용 형태이다.
<u>DECIMAL</u> (길이,소수)	고정 소수형 데이터 타입고정(길이+1byte) -소수점을 사용 형태이다.
<u>DOUBLE</u> (길이,소수)	부동 소수형 데이터 타입(8byte) -DOUBLE을 문자열로 저장한다



# 날짜형 데이터 형식

<u>DATE</u>	날짜(년도, 월, 일) 형태의 기간 표현 데이터 타입(3byte)
<u>TIME</u>	시간(시, 분, 초) 형태의 기간 표현 데이터 타입(3byte)
<u>DATETIME</u>	날짜와 시간 형태의 기간 표현 데이터 타입(8byte)
<u>TIMESTAMP</u>	날짜와 시간 형태의 기간 표현 데이터 타입(4byte) -시스템 변경 시 자동으로 그 날짜와 시간이 저장된다.
<u>YEAR</u>	년도 표현 데이터 타입(1byte)

# DDL - CREATE

```
CREATE TABLE member (  
    id VARCHAR(10) NOT NULL PRIMARY KEY,  
    name VARCHAR(10) NOT NULL,  
    birthday DATE NOT NULL  
);
```

# DDL - ALTER

- 테이블의 특정 컬럼(열)을 삭제하거나 추가, 변경할 때 사용하는 명령어

# 1. 컬럼 삭제

```
ALTER TABLE 테이블명 DROP COLUMN 컬럼명;
```

# 2. 컬럼 추가

```
ALTER TABLE 테이블명 ADD 컬럼명 타입
```

# 3. 컬럼 속성 변경

```
ALTER TABLE 테이블명 MODIFY 컬럼명 타입;
```

# DDL – DROP vs. TRUNCATE

```
DROP TABLE 테이블명;
```

- 테이블 삭제하기
- 테이블을 잘못 만들었거나 더 이상 필요 없는 경우

```
TRUNCATE TABLE 테이블명;
```

- 테이블 초기화하기 (테이블의 모든 행(row) 일괄 삭제)

이름	성별	나이
홍길동	남	40
임꺽정	여	50
장발장	남	60



이름	성별	나이

DELETE 후



이름	성별	연락처
----	----	-----

TRUNCATE 후



DROP 후

이미지 출처: <https://lee-mandu.tistory.com/476>

# SQL 공통

# 데이터베이스 목록 보기

**SHOW DATABASES;**

# 데이터베이스 이용하기

**USE 데이터베이스명;**

# 테이블 목록 보기

**SHOW TABLES;**

# 테이블 구조 보기

**DESC 테이블명;**

DML

# SQL문 – DML

- DML ( Data **Manipulation** Language )
- 데이터베이스의 내부 데이터를 관리하기 위한 언어

종류	역할
SELECT	데이터베이스에서 데이터를 검색(조회)하는 역할을 한다.
INSERT	테이블에 데이터를 추가하는 역할을 한다.
UPDATE	테이블에서 데이터를 수정하는 역할을 한다.
DELETE	테이블에서 데이터를 삭제하는 역할을 한다.

# CRUD

- 대부분의 컴퓨터 소프트웨어가 가지는 기본적인 처리 기능
- **C**reate(생성)
- **R**ead(읽기)
- **U**ppdate(갱신)
- **D**delete(삭제)

이름	조작	SQL
Create	생성	INSERT
Read	읽기	SELECT
Update	갱신	UPDATE
Delete	삭제	DELETE



# DML - INSERT

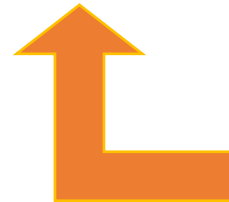
- 테이블에 데이터를 추가하기 위해 사용

방법1

```
INSERT INTO 테이블명 (필드1, 필드2, 필드3) VALUES (값1, 값2, 값3);
```

방법2

```
INSERT INTO 테이블명 VALUES(값1, 값2, 값3);
```



필드를 명시하지 않을 때는 테이블의 모든 컬럼에 값을 추가할 때만 사용할 수 있다.

# DML - SELECT

- 데이터를 검색(조회)하기 위해 사용

```
SELECT * FROM 테이블명;  
SELECT * FROM 테이블명 WHERE 필드1 = 조건값1;  
SELECT * FROM 테이블명 WHERE 필드1 = 조건값1 ORDER BY 필드1 ASC;  
SELECT 필드1, 필드2, 필드3 FROM 테이블명 WHERE 필드1 = 조건값1 ORDER BY 필드1 ASC;  
SELECT 필드1, 필드2, 필드3 FROM 테이블명 WHERE 필드1 = 조건값1 ORDER BY 필드1 ASC LIMIT 개수;
```

# WHERE 절 - 비교연산자

=	같다
>	보다 크다
>=	보다 크거나 같다
<	보다 작다
<=	보다 작거나 같다

# WHERE 절 – 부정연산자

!=	같지 않다.
^=	같지 않다.
<>	같지 않다.
NOT 컬럼명 =	~와 같지 않다.

# WHERE 절 – SQL 연산자

BETWEEN a AND b	a와 b의 값 사이에 있으면 참 ( a와 b 값도 포함 )
IN ( list )	리스트에 있는 값 중에서 어느 하나라도 일치하면 참
LIKE '비교문자열'	비교문자열과 형태가 일치하면 사용 ( %, _ 사용 ) <ul style="list-style-type: none"><li>• % : 0개 이상의 어떤 문자</li><li>• _ : 1개의 단일문자</li></ul>
IS NULL	NULL 값인 경우 true, 아니면 false

# WHERE 절 – 논리연산자

AND	앞에 있는 조건과 뒤에 오는 조건이 참(TRUE)가 되면 결과도 참(TRUE)
OR	앞에 있는 조건과 뒤에 오는 조건 중 하나라도 참(TRUE)면 결과는 참(TRUE)
NOT	뒤에 오는 조건과 반대되는 결과를 돌려준다.

# DML – UPDATE, DELETE

## UPDATE

- 데이터를 수정하기 위해 사용

```
UPDATE 테이블명 SET 필드1=값1 WHERE 필드2=조건2;
```

## DELETE

- 데이터를 삭제하기 위해 사용

```
DELETE FROM 테이블명 WHERE 필드1=값1;
```

DCL



# SQL문 – DCL

- DCL ( Data **Control** Language )
- 데이터베이스의 **사용권한과 보안을 관리하기 위한 언어**

종류	역할
GRANT	사용자에게 데이터베이스 객체(테이블, 뷰 등)에 대한 권한을 부여
REVOKE	사용자에게 부여된 데이터베이스 객체 권한을 취소

# DCL – GRANT, REVOKE

## GRANT

- 예시) employees 테이블에 대해 SELECT 권한을 사용자 kdt에게 부여

```
GRANT SELECT ON employees TO kdt;
```

## REVOKE

- 예시) kdt에게 부여된 employees 테이블의 SELECT 권한을 취소

```
REVOKE SELECT ON employees FROM kdt;
```