

# SQL 명령어

- Karns -

# DDL(Data Definition Language)

- 데이터베이스와 테이블을 정의, 수정, 삭제하는 구문
- CREATE
  - 데이터베이스와 테이블을 생성하는 명령어
- ALTER
  - 테이블을 수정하는 명령어
- DROP
  - 데이터베이스와 테이블을 삭제하는 명령어

- 데이터베이스 관련 명령 예시

# 데이터베이스 생성

```
CREATE DATABASE dbname;
```

# 데이터베이스 목록 보기

```
SHOW DATABASES;
```

# dbname 데이터베이스 선택

```
USE dbname;
```

# dbname 데이터베이스 삭제

```
DROP DATABASE IF EXISTS dbname;
```

- 테이블 생성 예시

```
# 학생 테이블 생성
CREATE TABLE student (
    student_id INT UNSIGNED auto_increment,
    student_name varchar(10) not null,
    student_address varchar(50) null,
    PRIMARY KEY(student_id)
);
```

- 테이블 조회 및 수정, 삭제 예시

# 데이터베이스안의 테이블들 조회

```
SHOW TABLES;
```

# 테이블 정보 조회

```
desc mytable;
```

# 테이블 수정 > 새로운 컬럼 추가

```
ALTER TABLE mytable ADD COLUMN new_column varchar(10) NOT NULL;
```

# 테이블 삭제

```
DROP TABLE IF EXISTS mytable;
```

# DCL(Data Control Language)

- 데이터의 보안, 무결성, 회복 등을 제어하는 구문
- grant
  - 사용자 계정에 대한 권한을 주는 명령어
- revoke
  - 사용자 계정에 대한 권한을 회수하는 명령어
- commit
  - `insert, update, delete` 대한 데이터베이스에 실제 반영
- rollback
  - `insert, update, delete` 대한 복구

- 사용자 관련 명령 예시

```
# 사용자 확인
```

```
use mysql;
```

```
SELECT * FROM user;
```

```
# 모든 IP에서 접속 가능한 계정 생성 예시
```

```
CREATE USER 'userid'@'%' IDENTIFIED BY 'userpw';
```

```
# 계정 삭제
```

```
DROP USER 'userid'@'%';
```

## • 권한 부여 예시

# 모든 DB와 테이블에 모든 권한 부여

```
GRANT ALL PRIVILEGES ON *.* TO 'userid'@'%';  
FLUSH PRIVILEGES; # 새로운 권한 적용
```

# 특정 데이터베이스의 특정 테이블의 조회/수정/추가 권한 부여

```
GRANT select,insert,update ON 데이터베이스.테이블 to 'userid'@'%';  
FLUSH PRIVILEGES; # 새로운 권한 적용
```

# 권한 회수

```
REVOKE SELECT,INSERT,DELETE,CREATE,UPDATE,ALTER,DROP ON *.* FROM 'userid'@'%';  
FLUSH PRIVILEGES; # 새로운 권한 적용
```



- commit & rollback 예시

```
# tb_user 테이블의 데이터를 삭제후 커밋  
DELETE FROM tb_user;  
COMMIT; # 되돌릴 수 없음
```

```
# tb_user 테이블의 데이터를 삭제후 롤백  
DELETE FROM tb_user;  
ROLLBACK; # tb_user 테이블의 데이터를 복구
```

# DML(Data Manipulation Language)

- 테이블의 데이터를 삽입, 조회, 수정, 삭제하는 구문
- INSERT
  - 테이블에 데이터를 삽입하는 명령어
- UPDATE
  - 테이블에 데이터를 수정하는 명령어
- DELETE
  - 테이블에 데이터를 삭제하는 명령어
- SELECT
  - 테이블에 데이터를 조회하는 명령어

- 테이블의 데이터를 삽입, 수정, 삭제 예시

# 테이블의 데이터 삽입

```
INSERT INTO 테이블명(col1, col2, ...) VALUES(value1, value2, ...);
```

# 테이블의 데이터 수정

```
UPDATE 테이블명 SET 컬럼1 = '수정할 값' WHERE 컬럼2 = '값';
```

# 테이블의 데이터 삭제

```
DELETE FROM 테이블명 WHERE 컬럼1 = '값';
```

- 테이블의 데이터를 조회하는 예시

# 테이블의 전체 데이터 조회

```
SELECT * FROM 테이블명 ;
```

# 테이블의 전체 데이터를 컬럼1 기준으로 내림차순으로 정렬해서 조

```
SELECT * FROM 테이블명 order by 컬럼1 DESC ;
```

# 테이블의 데이터를 조건을 걸어 일부만 조회

```
SELECT * FROM 테이블명 WHERE 컬럼1 = '값'
```

# MySQL 주요 자료형

- 자료형(Data Types)은 데이터를 어떻게 저장하고 처리할지를 정의
- MySQL의 자료형은 크게 세 가지 범주 나눌 수 있음
  - 숫자 자료형
  - 문자열 자료형
  - 날짜 및 시간 자료형

# 숫자(Numeric) 자료형

- 정수형(Integer Types)
  - TINYINT(1바이트)
    - -128 ~ 127
    - 0 ~ 255 (UNSIGNED)
  - SMALLINT(2바이트)
    - -32,768 ~ 32,767
    - 0 ~ 65,535 (UNSIGNED)

- 정수형(Integer Types)
  - MEDIUMINT(3바이트)
    - 8,388,608 ~ 8,388,607
      - 0 ~ 16,777,215 (UNSIGNED)
  - INT(4바이트)
    - -2,147,483,648 ~ 2,147,483,647
    - 0 ~ 4,294,967,295 (UNSIGNED)
  - BIGINT(8바이트)
    - 9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
    - 0 ~ 18,446,744,073,709,551,615 (UNSIGNED)

- 실수형(Floating-Point Types)

- FLOAT(4바이트)

- 소수점 이하 자리수가 적은 실수를 표현

- DOUBLE(8바이트)

- 더 높은 정밀도의 실수를 표현



# 문자열(String) 자료형

- CHAR(M)
  - 고정 길이 문자열. 길이가 M으로 지정된 문자열을 저장. M은 0 ~ 255까지 지정할 수 있음.
- VARCHAR(M)
  - 가변 길이 문자열. M은 최대 길이를 의미하며, 실제 저장되는 문자열 길이에 따라 공간을 차지함.
- TEXT
  - 긴 텍스트를 저장. 최대 65,535자까지 저장 가능.

- TINYTEXT
  - 최대 255자까지 저장 가능한 짧은 텍스트.
- MEDIUMTEXT
  - 최대 16,777,215자까지 저장 가능한 텍스트.
- LONGTEXT
  - 최대 4,294,967,295자까지 저장 가능한 긴 텍스트.

# 날짜 및 시간(Date and Time) 자료형

- DATE
  - 날짜를 'YYYY-MM-DD' 형식으로 저장.
- TIME
  - 시간을 'HH:MM:SS' 형식으로 저장.
- DATETIME
  - 날짜와 시간을 'YYYY-MM-DD HH:MM:SS' 형식으로 저장.