

## 제2장

### SQL 기초-1



# 1. SELECT문

## ❖ <SELECT... FROM>

- 원하는 데이터를 가져와 주는 기본적인 구문
- 가장 많이 사용되는 구문
- 데이터베이스 내 테이블에서 원하는 정보 추출하는 기능

# 1. SELECT문

## ❖ SELECT의 구문 형식

- 복잡한 형식을 실제 사용되는 형태로 요약

```
SELECT select_expr  
  [FROM table_references]  
  [WHERE where_condition]  
  [GROUP BY {col_name | expr | position}]  
  [HAVING where_condition]  
  [ORDER BY {col_name | expr | position}]
```



```
SELECT 열이름  
FROM 테이블이름  
WHERE 조건
```

# 1. SELECT문

## ❖ USE 구문

- **SELECT문 학습 위해 사용할 데이터베이스 지정**
- 지정해 놓은 후 특별히 다시 USE문 사용하거나 다른 DB를 사용하겠다고 명시하지 않는 이상 모든 SQL문은 지정 DB에서 수행

```
USE 데이터베이스_이름;
```

- **Workbench 에서 직접 선택해서 사용도 가능**
  - [Navigator]의 [Schemas] 탭
    - **employees 데이터베이스를 더블 클릭하면 진한 글자 전환**
    - 왼쪽 아래 'Active schema changed to employees' 메시지

# 1. SELECT문

## ❖ SELECT와 FROM

### ■ SELECT \*

- 선택된 DB가 employees 라면 다음 두 쿼리는 동일

```
SELECT * FROM employees.titles;  
SELECT * FROM titles;
```

### ■ SELECT 열 이름

- 테이블에서 필요로 하는 열만 가져오기 가능
- 여러 개의 열을 가져오고 싶을 때는 콤마로 구분
- 열 이름의 순서는 출력하고 싶은 순서대로 배열 가능

# 1. SELECT문

❖ DB, TABLE, 열의 이름이 확실하지 않을 때 조회하는 법

- 현재 서버에 어떤 DB가 있는지 보기

- SHOW DATABASES;

- 현재 서버에 어떤 TABLE이 있는지 보기

- 데이터베이스 employees 에 있는 테이블 정보 조회

- SHOW TABLE STATUS;

- 테이블 이름만 간단히 보기

- SHOW TABLES;

- employees 테이블의 열이 무엇이 있는지 확인

- DESCRIBE employees;

- DESC employees;

# 1. SELECT문

## ❖ 문제가 생긴 DB 초기화하기

### ■ DB가 존재한다면 지우고 다시 만들기

```
DROP DATABASE IF EXISTS sqlDB; -- 만약 sqlDB가 존재하면 우선 삭제한다.  
CREATE DATABASE sqlDB;
```

- 계속 사용할 쿼리는 SQL 파일로 저장해서 재사용 가능하게 만들기
- 파일 내용을 불러다 쓰기 전에 모든 쿼리 창을 닫도록 함

### ■ DB 내용을 입력하는 예제(실습)

감사합니다.

