

Chapter 2. 테이블에서 데이터 검색

04강. Hello World 실행하기

예제 데이터베이스 다운로드 후 진행

1) 'SELECT * FROM 테이블명' 실행

mysql 클라이언트에서 SQL 명령을 입력하여 실행할 수 있다

SQL 명령의 마지막에는 **세미콜론(;)**을 붙인다!

2) SELECT 명령 구문

- DML의 일종
- SELECT는 질의, 쿼리라고 불리기도 함

```
SELECT * FROM 테이블명;
```

- *(애스터리스크) : '모든 열'을 의미하는 메타문자
- FROM : 처리 대상 테이블 지정 키워드
 - FROM 뒤에 테이블명
- SQL 명령은 몇 개의 구로 구성

3) 예약어와 데이터베이스 객체명

- 예약어: SELECT와 FROM은 구를 결정하는 키워드이자 예약어
 - 객체명은 예약어와 동일한 이름 X
- 예약어와 데이터베이스 객체명은 **대소문자 구별 X**

4) Hello World를 실행한 결과 = 테이블

- SELECT 명령 실행 → 표 형식의 데이터 : 행(레코드)과 열(컬럼/필드)로 구성
- 셀: 행과 열이 만나는 부분, 하나의 데이터 값이 저장
- 데이터 종류
 - 수치형 데이터: 숫자만으로 구성된 데이터, 오른쪽 정렬
 - 문자형 데이터: 임의의 문자로 구성된 데이터, 왼쪽 정렬
 - 날짜시간형 데이터: 날짜, 시각을 나타내는 데이터, 왼쪽 정렬

| 데이터는 자료형으로 분류 가능, 열은 하나의 자료형만 가짐

5) 값이 없는 데이터 = NULL

- NULL은 아무 것도 저장 되어있지 않는 상태

05강. 테이블 구조 참조하기

1) DECS 명령

```
DESC 테이블명;
```

- 테이블에 어떤 열이 정의되어있는지 알 수 있음
- DESC는 SQL 명령이 아님
- 실행결과

Field	Type	Null	Key	Default	Extra
no	int	YES		NULL	
name	varchar(20)	YES		NULL	
birthday	date	YES		NULL	
address	varchar(40)	YES		NULL	

- Type: 자료형
- Null: Null값을 허용할 것인지 아닌지를 나타내는 제약사항
 - YES: 널 값 허용
- Key: 해당 열이 키로 지정되어있는지
- Default : 그 열에 주어진 기본값; 생략했을 경우 적용되는 값

2) 자료형

종류	설명
INTEGER 형	정수값 저장, 소수점 포함 불가
CHAR 형	<ul style="list-style-type: none"> - 문자열을 저장할 수 있는 자료형 - 열의 최대 길이를 정해야함 - 고정된 길이로 데이터 저장(Vs. VARCHAR) => 최대 길이보다 작은 문자열 저장 시, 공백문자로 나머지를 채움
VARCHAR 형	<ul style="list-style-type: none"> - 문자열을 저장할 수 있는 자료형 - 가변 길이 문자열 자료형
DATE 형	날짜값 저장 자료형
TIME 형	시간 저장 자료형

06강. 검색 조건 지정하기

행/열 지정하여 검색

- WHERE 구: 행을 선택할 때
- SELECT 구: 열을 선택할 때

1) SELECT 구에서 열 지정하기

```
SELECT 열1, 열2 ... FROM 테이블명;
```

- 콤마(,)를 이용하여 여러 개 지정

2) WHERE 구에서 행 지정하기

```
SELECT 열 FROM 테이블명 WHERE 조건식;
```

- 구의 순서와 생략
 - 구의 순서대로 적어야 함
 - WHERE 구 생략 시, 테이블 내 모든 행이 검색 대상
- WHERE 구
 - 조건식 : 열과 연산자, 상수로 구성되는 식
- 조건식
 - 조건식이 참인 행만 결과값 반환
 - 비교 연산자 사용
- 값이 서로 다른 경우 '<>'

3) 문자열형의 상수

- 싱글쿼트(' ')로 둘러싸 표기
 - 이는 날짜시간형의 경우도 동일
 - 연월일은 하이픈(-), 시분초는 콜론(:)
- 리터럴 : 자료형에 맞게 표기한 상수값

4) NULL값 검색

```
SELECT * FROM sample21 WHERE 열 IS NULL;
```

- NULL값 비교할 때 '=' 사용 X, IS NULL 사용!

5) 비교 연산자

종류	설명
= 연산자	좌변과 우변의 값이 같을 경우 참
<> 연산자	좌변과 우변의 값이 같지 않을 경우 참
> 연산자	좌변의 값이 우변의 값보다 클 경우 참
>= 연산자	좌변의 값이 우변의 값보다 크거나 같을 경우 참
< 연산자	좌변의 값이 우변의 값보다 작을 경우 참
<= 연산자	좌변의 값이 우변의 값보다 작거나 같을 경우 참

07강. 조건 조합하기

논리 연산자 AND, OR, NOT

1) AND로 조합하기

조건식1 AND 조건식2

#예제

```
SELECT * FROM sample24 WHERE a<>0 AND b<>0;
```

- 모든 조건을 만족하는 경우 조건식은 참
- '교집합'

2) OR로 조합하기

조건식1 OR 조건식 2

#예제

```
SELECT * FROM sample24 WHERE a<>0 OR b<>0;
```

- 어느 쪽이든 하나만 참이 되면 조건식은 참
- '합집합'

3) AND와 OR를 사용할 경우 주의할 점

- 연산자의 우선 순위
 - 우선순위 : AND > OR
- 조건이 잘못될 것을 방지해 괄호로 묶어두면 좋음

4) NOT으로 조합

- 오른쪽에만 항목을 지정하는 '단항 연산자'
- 오른쪽에 지정한 조건식의 반대 값을 반환

#예제

```
SELECT * FROM sample24 WHERE NOT(a<>0 OR b<>0);
```

08강. 패턴 매칭에 의한 검색

'LIKE'술어 사용 : 문자열의 일부분을 비교하는 '부분 검색'('패턴 매칭')

1) LIKE로 패턴 매칭하기

열명 LIKE '패턴'

- 부분적으로 일치하는 경우에도 참
- 패턴 정의 시, '&_'과 같은 메타문자 사용
 - % : 임의의 문자열
 - _ : 임의의 문자 하나(패턴 정의 시, 메타문자 여러개 사용 가능)

- 메타문자: 와일드 카드라고 불림. 패턴 매칭 시, 임의의 문자 또는 문자열에 매칭하는 부분을 지정하기 위해 쓰인 특수 문자
 - * 는 LIKE에서 사용 불가
- 예제

```
#전방일치 : 'SQL'로 시작하는 행 검색
SELECT * FROM sample25 WHERE text LIKE 'SQL%';
```

```
#중간일치 : 문자열 도중에 존재하는 'SQL'검색
SELECT * FROM sample25 WHERE text LIKE '%SQL%';
```

2) LIKE로 %를 검색하기

행에 메타문자(%, _)와 동일한 문자를 검색할 시

- 이스케이프(\) 문자 사용
 - % 검색 → '\%'

```
#예제
SELECT * FROM sample25 WHERE text LIKE '%\%%';
```

3) 문자열 상수 '의 이스케이프!

- ' 을 문자열 상수 안에 포함할 경우는 '를 2개 연속해서 기술한다!
 - 'It's ' → 'It''s'