



05장 (WHERE절과 연산자)

WHERE절

- SELECT문으로 데이터를 조회할 때 특정 조건을 기술해주는 역할

```
SELECT [조회할 열1 이름], [열2 이름], ..., [열N 이름]
FROM   [조회할 테이블 이름]
WHERE  [조회할 행을 선별하기 위한 조건식]; -1
```

기본 형식

WHERE절의 조건식

- 개수는 사실상 제한이 없다고 보아도 무방
- 실무에서는 **OR연산자**보다 **AND연산자**를 더 많이 사용
- 산술연산자, 비교연산자, 등가 비교연산자 사용가능

연산자	사용법	설명
>	A > B	A 값이 B 값을 초과할 경우 true
>=	A >= B	A 값이 B 값 이상일 경우 true
<	A < B	A 값이 B 값 미만일 경우 true
<=	A <= B	A 값이 B 값 이하일 경우 true

연산자	사용법	의미
=	A = B	A 값이 B 값과 같을 경우 true, 다를 경우 false 반환
!=	A != B	A 값과 B 값이 다를 경우 true, 같을 경우 false 반환
<>	A <> B	
^=	A ^= B	

NOT (논리 부정 연산자)

- 논리 부정 연산자
- **IN, BETWEEN, IS NULL 연산자**와 복합적으로 사용하는 경우가 많음
- **AND, OR**로 묶여 있는 상태에서 정반대 결과를 얻고자 할 때 유용

IN 연산자

- OR 연산자를 여러개 사용 할 경우 사용하면 용이

```
SELECT [조회할 열1 이름], [열2 이름], ..., [열N 이름]
FROM   [조회할 테이블 이름]
WHERE  열 이름 IN (데이터1, 데이터2, ... 데이터N); -1
```

기본 형식

BETWEEN A AND B 연산자

- 특정 범위를 지정할 때 사용하는 연산자

```
SELECT [조회할 열1 이름], [열2 이름], ..., [열N 이름]
FROM   [조회할 테이블 이름]
WHERE  열 이름 BETWEEN 최솟값 AND 최댓값; -1
```

기본 형식

LIKE 연산자와 와일드 카드

- 일부 문자열이 포함된 데이터를 조회할 때 사용
- SQL의 성능을 고려해 불만함(아직 의견이 분분하다)

종류	의미
_	어떤 값이든 상관없이 한 개의 문자 데이터를 의미
%	길이와 상관없이(문자 없는 경우도 포함) 모든 문자 데이터를 의미

- **ESCAPE절**

- 와일드 카드 문자가 데이터의 일부일 경우 사용
- 와일드 카드 기호가 아닌 데이터의 일부분으로써 문자로 다루는 것이 가능
- 해당 문자 바로뒤에 있는 문자를 데이터에 포함된 문자로 인식하라는 의미
 - 실무에서는 그리 많이 사용하지는 않는다고함

```
SELECT *
  FROM SOME_TABLE
 WHERE SOME_COLUMN LIKE 'A%W_A%' ESCAPE 'W';
```

IS NULL 연산자

- 숫자 '0'과 달리 비어있는 Data
 - 숫자 '0'은 존재하는 Data
- 현재 무슨 값인지 확정되지 않은 상태 or 값 자체가 존재하지 않는 상태를 나타내는 데이터에 사용
- **IS NULL, IS NOT NULL**은 매우 자주 사용됨

```
WHERE COMM = NULL;
//NULL인 행이 나와야 할 것 같지만 실제로 출력되는 데이터는 없다.
WHERE COMM IS NULL;
//이렇게 출력해야만 NULL값인 데이터들이 출력됨
```

AND연산자

피연산자 2	true	false	NULL
피연산자 1			
true	true	false	NULL
false	false	false	false
NULL	NULL	false	NULL

OR연산자

피연산자 2	true	false	NULL
피연산자 1			
true	true	true	true
false	true	false	NULL
NULL	true	NULL	NULL

집합 연산자

- SELECT문을 통해 데이터를 조회한 결과를 하나의 집합과 같이 다룰 수 있는 연산자

- 두 개 이상의 SELECT문의 결과 값을 연결할 때 사용
- **출력하려는 열 개수와 각 열의 자료형이 순서별로 일치해야 사용 가능**
- 종류

종류	설명
UNION	연결된 SELECT문의 결과 값을 합집합으로 묶어 줍니다. 결과 값의 중복은 제거됩니다.
UNION ALL	연결된 SELECT문의 결과 값을 합집합으로 묶어 줍니다. 중복된 결과 값도 제거 없이 모두 출력됩니다.
MINUS	먼저 작성한 SELECT문의 결과 값에서 다음 SELECT문의 결과 값을 차집합 처리합니다. 먼저 작성한 SELECT문의 결과 값 중 다음 SELECT문에 존재하지 않는 데이터만 출력됩니다.
INTERSECT	먼저 작성한 SELECT문과 다음 SELECT문의 결과 값이 같은 데이터만 출력됩니다. 교집합과 같은 의미입니다.

연산자 우선순위

우선순위	연산자	설명
↑ (높음)	*, /	산술 연산자 곱하기, 나누기
	+, -	산술 연산자 더하기, 빼기
	=, !=, ^=, <>, >, >=, <, <=	대소 비교 연산자
	IS (NOT) NULL, (NOT) LIKE, (NOT) IN	(그 외) 비교 연산자
	BETWEEN A AND B	BETWEEN 연산자
	NOT	논리 부정 연산자 NOT
(낮음) ↓	AND	논리 연산자 AND
	OR	논리 연산자 OR