

SELECT문은 데이터베이스로부터 저장되어 있는 데이터를 검색하는데 사용 한다.

SELECT 문법

```
SELECT [DISTINCT] {*, column [alias], ...}
FROM   table_name
[WHERE condition]
[ORDER BY {column, expression} [ASC | DESC]];
```

- **DISTINCT** : 중복되는 행을 제거하는 옵션.
- ***** : 테이블의 모든 column을 출력.
- **alias** : 해당 column에 대해서 다른 이름을 부여할 때 사용.
- **table_name** : 질의 대상 테이블 명
- **WHERE** : 조건을 만족하는 행들 만 검색
- **condition** : column, 표현식, 상수 및 비교 연산자
- **ORDER BY** : 질의 결과 정렬을 위한 옵션(ASC:오름차순(Default),DESC내림차순)

SQL문의 작성 방법

- SQL 문장은 대소문자를 구별하지 않는다.
- SQL 문장은 한 줄 또는 여러 줄에 입력될 수 있다.
- 일반적으로 키워드는 대문자로 입력한다. 다른 모든 단어, 즉 테이블 이름, 열 이름은 소문자로 입력한다.(권장)
- 가장 최근의 명령어 한 개가 SQL buffer에 저장 된다.
- SQL문 마지막 절의 끝에 ";"를 기술하여 명령의 끝을 표시 한다.

```
-- empno와 ename 은 각각 사번과 성명이라는 컬럼 별칭(alias)으로 만들어 출력
-- alias를 사용할 때 as라는 키워드를 사용해도 되고, 생략 할 수도 있다.
```

```
SQL> SELECT empno 사번, ename 성명
      FROM emp
      WHERE deptno = 10
```

사번	성명
7782	CLARK
7839	KING
7934	MILLER

WHERE절에 사용될 수 있는 SELECT 연산자

연산자	설 명
BETWEEN a AND b	a와 b사이의 데이터를 출력.(a, b값 포함)
IN (list)	list의 값 중 어느 하나와 일치하는 데이터를 출력
LIKE	문자 형태로 일치하는 데이터를 출력(% ,_사용)

IS NULL	NULL값을 가진 데이터를 출력 합니다.
NOT BETWEEN a AND b	a와 b사이에 있지않은 데이터를 출력(a, b값 포함하지 않음)
NOT IN (list)	list의 값과 일치하지 않는 데이터를 출력
NOT LIKE	문자 형태와 일치하지 않는 데이터를 출력
IS NOT NULL	NULL값을 갖지 않는 데이터를 출력

IN, NOT IN 연산자

IN 연산자

```
-- 사번이 7900, 7934번인 사원의 사번과 성명 출력
SQL> SELECT empno, ename
FROM emp
WHERE empno IN (7900, 7934) ;
```

EMPNO	ENAME
7934	MILLER
7900	JAMES

NOT IN 연산자

```
-- 사번이 7900, 7934번이 아닌 사원의 사번과 성명 출력
SQL> SELECT empno, ename
FROM emp
WHERE empno NOT IN (7900, 7934) ;
```

EMPNO	ENAME
7369	SMITH
7499	ALLEN
7698	BLAKE
....	

BETWEEN 연산자

AND를 이용해 두 조건을 결합한 검색과 같은 결과값을 보여 준다.

```
-- 급여가 3000에서 5000사이인 사원만 조회
SQL> SELECT empno, ename
      FROM emp
      WHERE sal BETWEEN 3000 AND 5000;
```

```
EMPNO ENAME
-----
 7788 SCOTT
 7839 KING
 7902 FORD
```

LIKE 연산자

- 검색 STRING 값에 대한 와일드 카드 검색을 위해서 LIKE 연산자를 사용.
- %: 여러 개의 문자열을 나타내는 와일드 카드
- _: 단 하나의 문자를 나타내는 와일드 카드
- ESCAPE : 와일드 카드 문자를 일반문자 처럼 사용하고 싶은 경우에 사용.
ex) WHERE name LIKE '%aW_y%' ESCAPE 'W'
- LIKE 연산자는 대소문자를 구분 한다.
- UPPER() 함수를 이용해 대소문자 구분없이 출력 할 수 있다.(인덱스 성능문제 발생, 함수기반 인덱스 사용..)

구 분	설 명
LIKE 'A%'	'A'로 시작하는 데이터만 검색
LIKE '%A'	'A'로 끝나는 데이터들만 검색
LIKE '%KIM%'	'KIM' 문자가 있는 데이터 들만 검색
LIKE '%K%I%'	'K' 문자와 'I' 문자가 있는 데이터 들만 검색
LIKE '_A%'	'A'문자가 두 번째 위치한 데이터 들만 검색

LIKE 연산자 예제

```
-- 'K' 문자가 들어있는 사원 정보 조회
-- UPPER() 함수는 k를 대문자로 변환하여 'K'로 인식
SQL> SELECT empno, ename
FROM emp
WHERE UPPER(ename) LIKE '%K%';
```

```
EMPNO  ENAME
-----
7698   BLAKE
7782   CLARK
7839   KING
```

```
-- '_'를 이용한 LIKE검색
SQL> SELECT empno, ename
FROM emp
WHERE UPPER(ename) LIKE '_I%'
```

```
EMPNO  ENAME
-----
7839   KING
7934   MILLER
```

ORDER BY

ORDER BY 절은 데이터의 정렬을 위해 사용합니다.(**ASC**[오름차순], **DESC**[내림차순])

```
-- 아래 두 개의 쿼리는 동일한 결과를 반환 한다.
-- 이름을 ASC로 정렬
SQL> SELECT empno, ename
FROM emp
WHERE deptno = 30
ORDER BY ename ASC;
```

```
-- 두 번째 컬럼을 디폴트 (ASC) 순으로 정렬
SQL> SELECT empno, ename
FROM emp
WHERE deptno = 30
ORDER BY 2
```

```
EMPNO  ENAME
-----
7499   ALLEN
7698   BLAKE
7900   JAMES
...
```