

5.1 WHERE

■ WHERE 기능

- 테이블내의 모든 행을 검색하는 대신 검색 조건을 지정하여 사용자가 원하는 행들만 검색하는 기능.

■ 기본적인 SELECT ~ WHERE 문법

```
SELECT [DISTINCT] { *, column [alias],... }  
FROM table  
[WHERE 조건식];
```

```
SQL> SELECT EMPNO, ENAME, JOB, DEPTNO  
2 FROM EMP  
3 WHERE DEPTNO = 30;
```

EMPNO	ENAME	JOB	DEPTNO
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7654	MARTIN	SALESMAN	30
7698	BLAKE	MANAGER	30
7844	TURNER	SALESMAN	30
7900	JAMES	CLERK	30

6 개의 행이 선택되었습니다.

```
SQL> SELECT EMPNO, ENAME, JOB, DEPTNO  
2 FROM EMP  
3 WHERE JOB = 'SALESMAN';
```

EMPNO	ENAME	JOB	DEPTNO
7499	ALLEN	SALESMAN	30
7521	WARD	SALESMAN	30
7654	MARTIN	SALESMAN	30
7844	TURNER	SALESMAN	30

```
SQL> SELECT EMPNO , ENAME , JOB, DEPTNO  
2 FROM EMP  
3 WHERE HIREDATE = '81/11/17';
```

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10

5.2 비교 연산자 1

연산자	의미
=	같다
>	보다 크다
>=	보다 크거나 같다
<	보다 작다
<=	보다 작거나 같다
<>	다르다

여기서, != 와 ^= 는 <> 와 동일한 의미를 갖는다.

```
SQL> SELECT EMPNO, ENAME, SAL
2 FROM EMP
3 WHERE SAL <= 1000;
```

EMPNO	ENAME	SAL
7369	SMITH	800
7900	JAMES	950

5.3 비교 연산자 2

연산자	의미
BETWEEN ... AND ...	두 값의 범위에 포함되는
IN (set)	괄호 안의 값과 일치하는
LIKE	문자의 조합이 같은
IS NULL	널 값

```
SQL> SELECT EMPNO, ENAME, SAL
2 FROM EMP
3 WHERE SAL BETWEEN 1000 AND 2000;
```

EMPNO	ENAME	SAL
7499	ALLEN	1600
7521	WARD	1250
7654	MARTIN	1250
7844	TURNER	1500
7876	ADAMS	1100
7934	MILLER	1300

```
SQL> SELECT EMPNO, ENAME, JOB
2 FROM EMP
3 WHERE EMPNO IN (7839, 7844, 7876);
```

EMPNO	ENAME	JOB
7876	ADAMS	CLERK
7844	TURNER	SALESMAN
7839	KING	PRESIDENT

```
SQL> SELECT EMPNO, ENAME, COMM
2 FROM EMP
3 WHERE COMM IS NULL;
```

EMPNO	ENAME	COMM
7369	SMITH	
7566	JONES	
7698	BLAKE	
7782	CLARK	
7788	SCOTT	
7839	KING	
7876	ADAMS	
7900	JAMES	
7902	FORD	
7934	MILLER	

5.4 비교 연산자 2

- LIKE 연산자

- 검색하고자 하는 문자열을 정확히 알 수 없는 경우에 사용.
- 패턴 매칭 연산자 이용

기호	설명
%	0 글자 이상의 임의 문자를 대표한다.
_	1 글자의 임의 문자를 대표한다

```
SQL> SELECT EMPNO, ENAME, JOB  
2 FROM EMP  
3 WHERE ENAME LIKE 'A%';
```

EMPNO	ENAME	JOB
7499	ALLEN	SALESMAN
7876	ADAMS	CLERK

```
SQL> SELECT EMPNO, ENAME, JOB  
2 FROM EMP  
3 WHERE ENAME LIKE '%T%';
```

EMPNO	ENAME	JOB
7369	SMITH	CLERK
7654	MARTIN	SALESMAN
7788	SCOTT	ANALYST
7844	TURNER	SALESMAN

```
SQL> SELECT EMPNO, ENAME, JOB  
2 FROM EMP  
3 WHERE ENAME LIKE '_L%';
```

EMPNO	ENAME	JOB
7499	ALLEN	SALESMAN
7698	BLAKE	MANAGER
7782	CLARK	MANAGER

- 검색하고자 하는 문자열에 패턴 매칭 연산자가 포함되어 있을 때는 ESCAPE 옵션을 사용한다.

```
SQL> SELECT EMPNO, ENAME, JOB  
2 FROM EMP  
3 WHERE ENAME LIKE '%W%' ESCAPE 'W';
```

5.5 논리 연산자

- WHERE 절에 부여할 조건이 여러 개인 경우에 사용한다.

연산자	의미
AND	두개의 조건이 TRUE이면 TRUE를 리턴
OR	두개의 조건중 하나의 조건이 TRUE이면 TRUE를 리턴
NOT	조건이 FALSE이면 TRUE를 리턴

```
SQL> SELECT EMPNO, ENAME, JOB, SAL
2 FROM EMP
3 WHERE JOB = 'SALESMAN'
4 AND SAL >= 1500;
```

EMPNO	ENAME	JOB	SAL
7499	ALLEN	SALESMAN	1600
7844	TURNER	SALESMAN	1500

```
SQL> SELECT EMPNO, ENAME, COMM
2 FROM EMP
3 WHERE COMM IS NOT NULL;
```

EMPNO	ENAME	COMM
7499	ALLEN	300
7521	WARD	500
7654	MARTIN	1400
7844	TURNER	0

```
SQL> SELECT EMPNO, ENAME, JOB, SAL
2 FROM EMP
3 WHERE JOB = 'SALESMAN'
4 OR SAL >= 1500
```

EMPNO	ENAME	JOB	SAL
7499	ALLEN	SALESMAN	1600
7521	WARD	SALESMAN	1250
7566	JONES	MANAGER	2975
7654	MARTIN	SALESMAN	1250
7698	BLAKE	MANAGER	2850
7782	CLARK	MANAGER	2450
7788	SCOTT	ANALYST	3000
7839	KING	PRESIDENT	5000
7844	TURNER	SALESMAN	1500
7902	FORD	ANALYST	3000

5.6 정렬

- SELECT 문장에 의해 검색된 결과를 정렬.

```
SELECT [DISTINCT] { *, column [alias],... }  
FROM   table  
[WHERE 조건식]  
[ORDER BY {column , exp } [ASC|DESC] ];
```

```
SQL> SELECT EMPNO, ENAME, HIREDATE  
2 FROM EMP  
3 ORDER BY HIREDATE;
```

EMPNO	ENAME	HIREDATE
7369	SMITH	80/12/17
7499	ALLEN	81/02/20
7521	WARD	81/02/22
7566	JONES	81/04/02
...		

```
SQL> SELECT EMPNO, ENAME, HIREDATE  
2 FROM EMP  
3 ORDER BY HIREDATE DESC;
```

EMPNO	ENAME	HIREDATE
7876	ADAMS	87/05/23
7788	SCOTT	87/04/19
7934	MILLER	82/01/23
7900	JAMES	81/12/03
...		

```
SQL> SELECT EMPNO, ENAME, SAL * 12 ANNUAL  
2 FROM EMP  
3 ORDER BY ANNUAL;
```

EMPNO	ENAME	ANNUAL
7369	SMITH	9600
7900	JAMES	11400
7876	ADAMS	13200
7521	WARD	15000
...		

```
SQL> SELECT EMPNO, ENAME, SAL * 12 ANNUAL  
2 FROM EMP  
3 ORDER BY 3;
```

EMPNO	ENAME	ANNUAL
7369	SMITH	9600
7900	JAMES	11400
7876	ADAMS	13200
7521	WARD	15000
...		

```
SQL> SELECT EMPNO, ENAME, SAL  
2 FROM EMP  
3 ORDER BY SAL DESC, EMPNO;
```

EMPNO	ENAME	SAL
7839	KING	5000
7788	SCOTT	3000
7902	FORD	3000
7566	JONES	2975
7698	BLAKE	2850
7782	CLARK	2450
7499	ALLEN	1600