

## Chapter 3. WHERE 및 ORDER BY

지난 장에서는 SELECT 문장을 이용하여 테이블의 모든 행을 검색할 수 있었다. 이번 장에서는 테이블 내의 모든 행을 검색하는 대신 검색 조건을 지정하여 사용자가 원하는 행들만 검색할 수 있는 방법을 살펴보고 출력 결과를 특정 컬럼 기준으로 정렬하는 방법을 알아본다.

### WHERE

데이터를 검색할 때 WHERE 조건을 지정하면, 조건에 만족하는 행들만 검색된다.

```
SELECT *|{[DISTINCT] column|expression [alias], ... }
FROM table
[WHERE condition(s)];
```

WHERE 구문 뒤에는 원하는 검색 조건을 지정한다. 예를 들어, 사원 테이블에서 부서코드가 30번인 사원들의 사번, 사원명, 업무, 부서코드를 검색하면 다음과 같다.

```
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.

WHERE 절에서 문자열과 날짜를 조건에 기술할 때는 반드시 ' '를 붙여야하며, 대소문자 구분이 되기 때문에 주의하여야 한다. 날짜 타입 데이터는 반드시 RR/MM/DD 형식으로 기술하여야 날짜 데이터로 올바르게 변환된다.

사원 테이블에서 사원명이 'KING'인 사원을 검색하면 다음과 같다.

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

EMPNO	ENAME	JOB	DEPTNO
7839	KING	PRESIDENT	10

문자열은 대소문자 구별이 되므로 원하는 결과를 얻기 위해서는 정확히 기술하여야 한다.

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

선택된 레코드가 없습니다.

사원 테이블에서 입사일이 1981년 11월 17인 사원을 검색하면 다음과 같다.

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

EMPNO	ENAME	JOB	HIREDATE
7839	KING	PRESIDENT	81/11/17

## 비교 연산자

WHERE 조건에 사용할 수 있는 비교 연산자는 다음과 같다.

표 3-1 비교 연산자 1

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

여기서, != 와 ^= 는 <> 와 동일한 의미를 갖는다.

사원 테이블에서 급여가 1000 이하인 사원을 검색하면 다음과 같다.

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

EMPNO	ENAME	SAL
7369	SMITH	800
7900	JAMES	950

다음은 일반적인 연산자는 아니지만 자주 사용되는 연산자들이다.

표 3-2. 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

### ■ BETWEEN ... AND ...

사원 테이블에서 급여가 1000 이상이고 2000 이하인 사원을 검색하면 다음과 같다.

```
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

BETWEEN을 사용 할 때 주의할 점은 하한값이 상한값보다 먼저 기술되어야 한다는 점과 하한 및 상한값이 모두 조건에 포함된다는 것이다. 위 문장은 다음 문장과 동일한 결과를 보여준다.

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

### ■ IN (set)

사원 테이블에서 사번이 7876, 7844, 7839인 사원을 검색하면 다음과 같다.

```
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

위 문장은 아래 문장과 동일한 결과를 보여준다.

```
SQL> SELECT EMPNO, ENAME, JOB
2 FROM EMP
3 WHERE EMPNO = 7839 OR EMPNO = 7844 OR EMPNO = 7876;
```

### ■ LIKE

검색하고자 하는 문자열을 정확히 알 수 없는 경우, LIKE 연산자와 패턴 매치 문자열을 이용하면 와일드카드(Wildcard) 탐색 방식을 수행 할 수 있다. 다음은 패턴 매치 문자열에서 사용되는 기호를 나타낸다.

표 3-3. 패턴 매치 문자열에서 사용되는 기호

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

사원 테이블에서 사원명이 'A'로 시작하는 사원을 검색하면 다음과 같다.

SQL> SELECT EMPNO, ENAME, JOB		
2 FROM EMP		
3 WHERE ENAME LIKE 'A%';		
EMPNO	ENAME	JOB
-----		
7499	ALLEN	SALESMAN
7876	ADAMS	CLERK

사원 테이블에서 사원명이 'N'으로 끝나는 사원을 검색하면 다음과 같다.

SQL> SELECT EMPNO, ENAME, JOB		
2 FROM EMP		
3 WHERE ENAME LIKE '%N';		
EMPNO	ENAME	JOB
-----		
7499	ALLEN	SALESMAN
7654	MARTIN	SALESMAN

사원 테이블에서 사원명에 'T'가 포함된 사원을 검색하면 다음과 같다.

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

패턴 매치열에 %와 \_를 같이 사용할 수도 있다.

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

참고로 검색하고자 하는 컬럼에 패턴매치 기호 %, \_가 포함되어 있고, 이러한 기호를 검

색하고자 하는 경우는 ESCAPE 옵션을 사용하면 된다. 예를 들어, JOB 컬럼의 데이터에 \_이 포함되어 있다고 가정하고 \_가 포함된 행을 검색하려면 다음과 같이 하면 된다.

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

## ■ NULL

사원 테이블에서 커미션이 NULL인 사원을 검색하면 다음과 같다.

```
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	

10 개의 행이 선택되었습니다.

위와 같은 경우, 절대 아래와 같이 작성하면 원하는 결과를 얻을 수 없다.

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

선택된 레코드가 없습니다.

사원 테이블에서 커미션이 NULL이 아닌 경우를 검색하면 다음과 같다.

```
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

## 논리 연산자

논리 연산자는 WHERE 절에 부여할 조건이 여러 개인 경우, 조건을 결합시키는데 사용한

다. 논리 연산자의 종류는 다음과 같다.

표 3-4. 논리 연산자

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

사원 테이블에서 업무가 'SALESMAN'이고 급여가 1500이상인 사원을 검색하면 다음과 같다.

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

사원 테이블에서 업무가 'SALESMAN' 또는 급여가 1500이상인 사원을 검색하면 다음과 같다.

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

사원 테이블에서 업무가 'SALESMAN', 'ANALYST', 'MANAGER' 가 아닌 사원을 검색하면 다음과 같다.

```
SQL> SELECT EMPNO, ENAME, JOB, SAL
2 FROM EMP
3 WHERE JOB NOT IN ('SALESMAN', 'ANALYST', 'MANAGER');
```

EMPNO	ENAME	JOB	SAL
7369	SMITH	CLERK	800
7839	KING	PRESIDENT	5000
7876	ADAMS	CLERK	1100
7900	JAMES	CLERK	950
7934	MILLER	CLERK	1300

논리 연산자는 NOT, AND, OR 순의 우선순위가 있으며 우선순위를 임의로 지정하고자 하는 경우에는 괄호를 사용하면 된다. 다음 예제를 보고 어떤 조건이 먼저 적용될 것인지를 생각해보자.

```
SQL> SELECT EMPNO, ENAME, JOB, SAL
2 FROM EMP
3 WHERE JOB = 'SALESMAN'
4 OR JOB = 'MANAGER'
5 AND SAL >= 2000;
```

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
7844	TURNER	SALESMAN	1500

7 개의 행이 선택되었습니다.

위의 경우는 논리 연산자의 우선순위에 따라 다음과 같이 연산된다.

```
SQL> SELECT EMPNO, ENAME, JOB, SAL
2 FROM EMP
3 WHERE JOB = 'SALESMAN'
4 OR (JOB = 'MANAGER'
5 AND SAL >= 2000);
```

만약, OR 조건부터 적용된 후 AND 조건이 적용되도록 하려면 다음과 같이 하면 된다.

```
SQL> SELECT EMPNO, ENAME, JOB, SAL
2 FROM EMP
3 WHERE (JOB = 'SALESMAN'
4 OR JOB = 'MANAGER')
5 AND SAL >= 2000;
```

EMPNO	ENAME	JOB	SAL
7566	JONES	MANAGER	2975
7698	BLAKE	MANAGER	2850
7782	CLARK	MANAGER	2450

## ORDER BY

SELECT 문장에 의해 검색된 결과를 정렬하려면 ORDER BY 문장을 추가하면 된다.

```
SELECT *|{[DISTINCT] column|expression [alias], ... }
FROM table
[WHERE condition(s)]
[ORDER BY {column, expr} [ASC|DESC]];
```

ORDER BY 구문 뒤에는 정렬의 기준이 되는 컬럼을 기술하고 정렬방식을 지정하는데, 오름차순으로 정렬을 하려면 ASC, 내림차순으로 정렬을 하려면 DESC를 기술하면 된다. 지정하지 않으면 ASC가 생략된 것으로 오름차순으로 정렬된다.

사원 테이블의 사번, 사원명, 입사일을 입사일이 빠른 순서로 정렬하면 다음과 같다.

```
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
...		

14 개의 행이 선택되었습니다.

사원 테이블의 사번, 사원명, 입사일을 입사일이 늦은 순서로 정렬하면 다음과 같다.

```
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
...		

14 개의 행이 선택되었습니다.

컬럼 별칭을 ORDER BY 구문 뒤에 기술할 수도 있다.



```
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
...		

14 개의 행이 선택되었습니다.

또한, SELECT 구문 뒤의 컬럼 순번을 대신 기술할 수도 있다.

```
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
...		

14 개의 행이 선택되었습니다.

사원 테이블의 사번, 사원명, 급여를 급여가 높은 순서대로 정렬하고, 급여가 같은 사원의 경우에는 사번이 빠른 순서로 정렬하면 다음과 같다.

```
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
7844	TURNER	1500
7934	MILLER	1300
7521	WARD	1250
7654	MARTIN	1250
7876	ADAMS	1100
7900	JAMES	950
7369	SMITH	800

14 개의 행이 선택되었습니다.

위 문장에서 사번에는 ASC가 생략되었음을 주의해야 한다.

## 복습

1. 사원 테이블에서 입사일이 81년도인 직원의 사번, 사원명, 입사일, 업무, 급여를 검색하시오.
2. 사원 테이블에서 입사일이 81년이고 업무가 'SALESMAN'이 아닌 직원의 사번, 사원명, 입사일, 업무, 급여를 검색하시오.
3. 사원 테이블의 사번, 사원명, 입사일, 업무, 급여를 급여가 높은 순으로 정렬하고, 급여가 같으면 입사일이 빠른 사원으로 정렬하시오.
4. 사원 테이블에서 사원명의 세 번째 알파벳이 'N'인 사원의 사번, 사원명을 검색하시오.
5. 사원 테이블에서 연봉(SAL\*12)이 35000 이상인 사번, 사원명, 연봉을 검색하시오.