

4장 SELECT로 특정 데이터를 추출하기

4.1 WHERE 조건과 비교 연산자

- 원하는 로우만 얻으려면 제한하는 조건을 SELECT 문에 WHERE 절을 추가하여 제시해야 한다.

```
SELECT *  
FROM EMP  
WHERE SAL >= 3000;
```

4.1.1 비교 연산자

연산자	의 미	예 제
=	같다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL=3000;
>	보다 크다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL>3000;
<	보다 작다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<3000;
>=	보다 크거나 같다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL>=3000;
<=	보다 작거나 같다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<=3000;
<>, !=, ^=	다르다.	SELECT EMPNO, ENAME, SAL FROM EMP WHERE SAL<>3000;

4.1.2 문자 데이터 조회

- SQL에서 문자열이나 날짜는 반드시 작은따옴표(')안에 표시해야 한다.

```
SELECT EMPNO, ENAME, SAL  
FROM EMP  
WHERE ENAME='FORD';  
  
SELECT EMPNO, ENAME, SAL  
FROM EMP  
WHERE ENAME='ford'; -- 테이블 내에 저장된 데이터 값은 대소문자를 구분한다.
```

4.1.3 날짜 데이터 조회

- 날짜는 문자열과 마찬가지로 단일 따옴표 안에 기술해야 한다.

```
SELECT *  
FROM EMP  
WHERE HIREDATE<='1982/01/01';
```

4.2 논리 연산자

연산자	의미
AND	두 가지 조건을 모두 만족해야만 검색할 수 있다. SELECT * FROM emp WHERE deptno=10 AND job='MANAGER';
OR	두 가지 조건 중에서 한 가지만 만족하더라도 검색할 수 있다. SELECT * FROM emp WHERE deptno=10 OR job='MANAGER';
NOT	조건에 만족하지 못하는 것만 검색한다. SELECT * FROM emp WHERE NOT deptno=10;

4.2.1 AND 연산자

- 두 가지 조건을 모두 만족할 경우에만 검색할 수 있도록 하기 위해서는 AND 연산자를 사용한다.

```
SELECT *  
FROM EMP  
WHERE DEPTNO=10 AND JOB='MANAGER';
```

4.2.2 OR 연산자

- 두 가지 조건 중에서 한 가지만 만족하더라도 검색할 수 있도록 하기 위해서는 OR 연산자를 사용한다.

```
SELECT *  
FROM EMP  
WHERE DEPTNO=10 OR JOB='MANAGER';
```

4.2.3 NOT 연산자

- NOT 연산자는 참은 거짓으로 거짓은 참으로 즉 반대되는 논리값을 구하는 연산자이다.

```
SELECT *  
FROM EMP  
WHERE NOT DEPTNO=10;  
  
SELECT *  
FROM EMP  
WHERE DEPTNO<>10;
```

4.3 BETWEEN AND 연산자

- 오라클에서는 특정 범위의 값을 조회하기 위해서는 BETWEEN AND 연산자를 사용할 수 있다.

```
SELECT *
```

```

FROM EMP
WHERE SAL >= 2000 AND SAL <= 3000;

SELECT *
FROM EMP
WHERE SAL BETWEEN 2000 AND 3000;

SELECT *
FROM EMP
WHERE SAL NOT BETWEEN 2000 AND 3000;

SELECT *
FROM EMP
WHERE HIREDATE BETWEEN '1987/01/01' AND '1987/12/31';

```

4.4 IN 연산자

- 동일한 필드가 여러 개의 값 중에 하나인 경우인지를 살펴보기 위해서 비교 연산자와 논리 연산자 OR를 사용하여 복잡하게 쿼리문을 작성하지 않고 IN 연산자를 사용하여 훨씬 간단하게 표현할 수 있다.

```

SELECT *
FROM EMP
WHERE COMM=300 OR COMM=500 OR COMM=1400;

SELECT *
FROM EMP
WHERE COMM IN(300, 500, 1400);

SELECT *
FROM EMP
WHERE COMM <> 300 AND COMM <> 500 AND COMM <> 1400;

SELECT *
FROM EMP
WHERE COMM NOT IN(300, 500, 1400);

```

4.5 LIKE 연산자와 와일드카드

- LIKE 연산자는 검색하고자 하는 값을 정확히 모를 경우에도 검색 가능하도록 하기 위해서 와일드카드와 함께 사용하여 원하는 내용을 검색하도록 한다.
- LIKE 다음에는 pattern을 기술해야 하는데 pattern에 다음과 같이 두 가지 와일드카드가 사용된다.

와일드카드	의미
%	문자가 없거나, 하나 이상의 문자가 어떤 값이 와도 상관없다.
_	하나의 문자가 어떤 값이 와도 상관없다.

4.5.1 와일드카드(%) 사용하기

- 검색하고자 하는 값을 정확히 모를 경우 즉, 특정 문자 포함되지만 하고 그 이전이나 이후에 어떤 문자가 몇 개가 오든지 상관없다는 의미를 표현하기 위해서는 LIKE 연산자와 함께 %를 사용해야 한다.

```

SELECT *
FROM EMP
WHERE ENAME LIKE 'F%';

SELECT *
FROM EMP
WHERE ENAME LIKE '%A%';

SELECT *
FROM EMP
WHERE ENAME LIKE '%N';

```

4.5.2 와일드카드(_) 사용하기

- _ 역시 %와 마찬가지로 어떤 문자가 오든 상관없다는 의미로 사용되는 와일드카드이다.
- 차이점은 %는 몇 개의 문자가 오든 상관없지만 _ 는 단 한 문자에 대해서만 와일드카드 역할을 한다.

```

SELECT *
FROM EMP
WHERE ENAME LIKE '_A%';

SELECT *
FROM EMP
WHERE ENAME LIKE '_ _A%';

```

4.5.3 NOT LIKE 연산자

```

SELECT *
FROM EMP
WHERE ENAME NOT LIKE '%A%';

```

4.6 NULL을 위한 연산자

- 어떤 컬럼을 NULL 즉, 모르는 값과 같다(=)라는 것은 의미상으로 말이 되지 않기 때문에 = 대신 IS NULL 연산자를 사용해야 한다.

```

SELECT *
FROM EMP
WHERE COMM=NULL; -- NULL 값을 = 연산자로 판단할 수 없다.

SELECT *
FROM EMP
WHERE COMM IS NULL; -- EMP 테이블에서 COMM 컬럼이 NULL 사원을 출력한다.

SELECT *
FROM EMP
WHERE COMM IS NOT NULL;

```

4.7 정렬을 위한 ORDER BY절

- 정렬이란 크기 순서대로 나열하는 것을 의미한다.
- 오름차순(ascending) 정렬 방식
 - 작은 것이 위에 출력되고 아래로 갈수록 큰 값이 출력
- 내림차순(descending) 정렬 방식
 - 큰 값이 위에 출력되고 아래로 갈수록 작은 값이 출력
- 로우를 정렬하기 위해서는 SELECT 문에 ORDER BY 절을 추가하고 어떤 컬럼을 기준으로 어떤 정렬을 할 것인지를 결정해야 한다.

	ASC(오름차순)	DESC(내림차순)
숫자	작은 값부터 정렬	큰 값부터 정렬
문자	사전 순서로 정렬	사전 반대 순서로 정렬
날짜	빠른 날짜 순서로 정렬	늦은 날짜 순서로 정렬
NULL	가장 마지막에 나온다.	가장 먼저 나온다.

4.7.1 오름차순 정렬을 위한 ASC

- 오름차순 정렬은 작은 값부터 큰 값으로 정렬하는 것을 의미한다.(예:1~9, 'A'~'Z') 이를 위해서는 ASC를 컬럼 다음에 기술해야 하는데 만일 생략하게 되면 디폴트로 ASC로 지정되어 있기 때문에 오름차순으로 출력된다.

```
SELECT *  
FROM EMP  
ORDER BY SAL ASC;
```

```
SELECT *  
FROM EMP  
ORDER BY SAL;
```

```
SELECT *  
FROM EMP  
ORDER BY ENAME; -- 문자 데이터의 경우 아스키 코드 값으로 저장되므로 아스키 코드 값을 기준으로 정렬된다.
```

4.7.2 내림차순 정렬을 위한 DESC

- 내림차순 정렬은 큰 값부터 작은 값으로 정렬을 하는 것이다.(예:9~1, Z~A)

```
SELECT *  
FROM EMP  
ORDER BY SAL DESC;
```

```
SELECT *  
FROM EMP  
ORDER BY HIREDATE DESC;
```

```
SELECT *  
FROM EMP  
ORDER BY SAL DESC, ENAME ASC;
```

[과제] 과제-04-01.TXT

SQL> CONN SCOTT/TIGER 로 접속하여 SQL문을 작성하세요.

1. 사원 테이블(EMP)에서 가장 최근에 입사한 사원부터 출력하되, 동일한 입사일인 경우에는 사원번호(EMPNO)를 기준으로 오름차순으로 정렬해서 출력하는 SQL문을 작성하세요?

<정답>

2. 와일드 카드를 사용하여 사원중에서 이름이 K로 시작하는 사원의 사원번호와 이름을 출력하세요?

<정답>

3. 와일드 카드를 사용하여 이름중에서 K를 포함하는 사원의 사원번호와 이름을 출력 하세요?

<정답>

4. 와일드 카드를 사용하여 이름중에서 끝에서 두번째 글자가 K로 끝나는 사원의 사원번호와 이름을 출력 하세요?

<정답>