

이 장에서 다룰 내용



1 WHERE 조건과 비교 연산자

2 논리 연산자

3 BETWEEN AND 연산자

4 IN 연산자

5 LIKE 연산자와 와일드카드

이 장에서 다룰 내용



6 NULL을 위한 연산자

7 정렬을 위한 ORDER BY 절

01. WHERE 조건과 비교 연산자



- ❖ 원하는 로우만 얻으려면 다음과 같이 로우를 제한하는 조건을 SELECT 문에 WHERE 절을 추가하여 제시해야 합니다.

형식	<code>SELECT * [column1, column2, .. ,columnn] FROM table_name WHERE <u>조건절</u>;</code>
----	---

- ❖ 조건절은 다음의 세부분으로 구성이 됩니다.

조건절의 구성	<code>WHERE SAL >= 3000;</code> ① 컬럼 ② 연산자 ③ 비교대상값
---------	---

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;

3.2 비교 연산자



사원 테이블에서 부서번호가 10번인 사원
조회해 보세요

예

```
SELECT *  
FROM EMP  
WHERE DEPTNO=10;
```

C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger

```
SQL> SELECT *  
2 FROM EMP  
3 WHERE DEPTNO=10;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7839	KING	PRESIDENT		81/11/17	5000		10
7934	MILLER	CLERK	7782	82/01/23	1300		10

〈탄탄히 다지기〉

1. 급여가 1500 이하인 사원의 사원번호, 사원 이름, 급여를 출력하는 SQL 문을 작성해 보시오.

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

7 rows selected.

<힌트> 사원 정보가 저장된 테이블의 이름은 EMP이고, 사원번호 컬럼은 EMPNO, 사원이름 컬럼은 ENAME, 급여 컬럼은 SAL입니다.

1.2 문자 데이터 조회



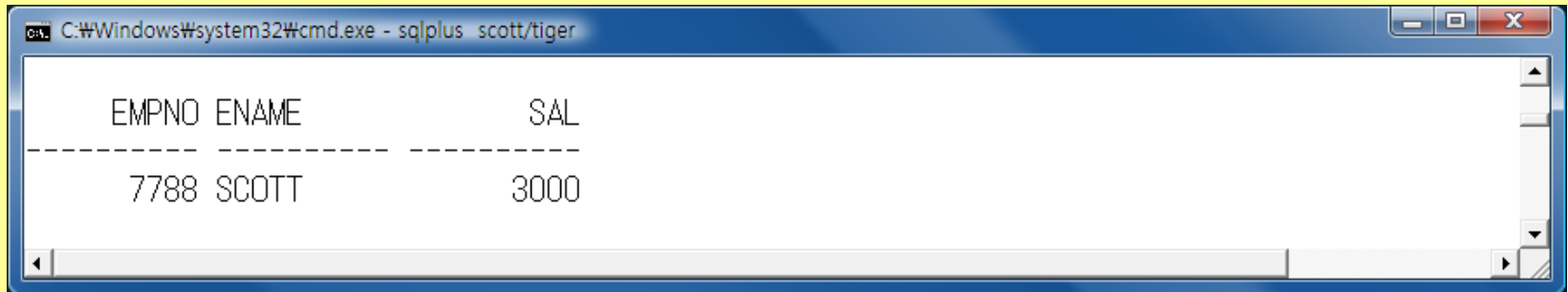
- ❖ 이전 예제에서 비교 연산자를 하기 위해서 다루었던 컬럼들은 수치 형태로 선언되었습니다. 이번에는 급여가 아닌 사원 이름 같은 문자 데이터를 조회해 봅시다.
- ❖ 다음은 이름이 FORD인 사원의 사원번호(EMPNO)과 사원이름(ENAME)과 급여(SAL)을 출력하는 예제입니다.

예	<pre>SELECT EMPNO, ENAME, SAL FROM EMP WHERE ENAME='FORD';</pre>
---	--

- ❖ SQL에서 문자열이나 날짜는 반드시 **단일 따옴표(single quotation)** 안에 표시해야 합니다.
- ❖ SQL문에 사용되는 키워드인 SELECT 나 FROM 이나 WHERE 등은 대소문자를 구별하지 않지만 테이블 내에 저장된 데이터 값은 대소문자를 구분하기에 WHERE ENAME='ford'와 같이 기술하면 사원이름이 FORD 인 사원을 찾을 수 없습니다.

<탄탄히 다지기>

2. 사원이름이 SCOTT 인 사원의 사원번호, 사원 이름, 급여를 출력하는 SQL 문을 작성해 보시오.



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The window displays the output of a SQL query, which is a table with three columns: EMPNO, ENAME, and SAL. The table has one data row showing the employee with EMPNO 7788, ENAME SCOTT, and SAL 3000.

EMPNO	ENAME	SAL
7788	SCOTT	3000

<힌트> 사원 정보가 저장된 테이블의 이름은 EMP이고, 사원번호 컬럼은 EMPNO, 사원이름 컬럼은 ENAME, 급여 컬럼은 SAL입니다.

1.3 날짜 데이터 조회

- ❖ 1982년 1월 1일 이전에 입사한 사원을 조회하려면 어떻게 해야할까요? 날짜는 문자열과 마찬가지로 단일 따옴표 안에 기술해야 합니다.
- ❖ 다음은 1982년 1월 1일 이전에 입사한 사원을 출력하는 예제입니다.



예

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

C:\Windows\system32\cmd.exe - sqlplus scott/tiger

```
SQL> SELECT *  
2 FROM EMP  
3 WHERE HIREDATE<='1982/01/01';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7566	JONES	MANAGER	7839	81/04/02	2975		20
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7900	JAMES	CLERK	7698	81/12/03	950		30
7902	FORD	ANALYST	7566	81/12/03	3000		20

11 개의 행이 선택되었습니다.

02. 논리 연산자



❖ 오라클에서 사용가능한 논리 연산자 AND나 OR나 NOT가 있습니다.

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

2.1 AND 연산자



- ❖ 두 가지 조건을 모두 만족할 경우에만 검색할 수 있도록 하기 위해서는 AND 연산자를 사용합니다.
- ❖ 다음은 AND 연산자가 조건에 따라 어떤 결과가 출력되는지를 나타내는 표입니다.
- ❖ 두 조건이 모두 만족할 경우에만 결과가 참이고, 조건 중 하나라도 만족하지 않으면 결과가 거짓입니다.

조건 1	조건2	&&
참	참	참
참	거짓	거짓
거짓	참	거짓
거짓	거짓	거짓

2.1 AND 연산자



- ❖ 10번 부서 소속인 직원들 중에서 직급이 MANAGER인 사람을 검색하여 사원명, 부서번호, 직급을 출력하려고 한다면 두 가지 조건을 제시해야 합니다.

[조건1] 10번 부서 소속인 직원 : DEPTNO=10

[조건2] 직급이 MANAGER인 직원 : JOB='MANAGER'

예

```
SELECT ename, deptno, job
FROM EMP
WHERE DEPTNO=10 AND JOB='MANAGER';
```

ENAME	DEPTNO	JOB
CLARK	10	MANAGER

2.2 OR 연산자



- ❖ 두 가지 조건 중에서 한 가지만 만족하더라도 검색할 수 있도록 하기 위해서는 OR 연산자를 사용합니다.
- ❖ 다음은 OR 연산자가 조건에 따라 어떤 결과가 출력되는지를 나타내는 표입니다.
- ❖ OR 연산자는 두 조건에 모두 만족하지 않을 경우는 결과가 거짓이고 제시한 조건에 한 가지라도 만족하면 결과가 참입니다.

조건 1	조건2	
참	참	참
참	거짓	참
거짓	참	참
거짓	거짓	거짓

2.2 OR 연산자



- ❖ 10번 부서에 소속된원이거나 직급이 MANAGER인 사람을 검색하여 사원명, 부서번호, 직급을 출력합시다.

[조건1] 10번 부서 소속인 사원 : DEPTNO=10

[조건2] 직급이 MANAGER인 사원 : JOB='MANAGER'

```
예 SELECT ename, deptno, job
    FROM EMP
    WHERE DEPTNO=10 OR JOB='MANAGER';
```

ENAME	DEPTNO	JOB
JONES	20	MANAGER
BLAKE	30	MANAGER
CLARK	10	MANAGER
KING	10	PRESIDENT
MILLER	10	CLERK

2.3 NOT 연산자



- ❖ 10번 부서에 소속된 직원만 제외하고 나머지 직원의 정보를 출력하려면 어떻게 해야 할까요? 이러한 조건을 제시하기 위해서 사용하는 논리 연산자가 바로 NOT 입니다.
- ❖ NOT 연산자는 참은 거짓으로 거짓은 참으로 즉 반대되는 논리값을 구하는 연산자입니다.
- ❖ 다음은 NOT 연산자가 논리값에 의해서 어떤 결과가 출력되는지를 나타내는 표입니다.

조건	NOT
참	거짓
거짓	참

2.3 NOT 연산자



- ❖ 이 조건 앞에 NOT을 붙이면 부서번호가 10번이 아닌 직원들에 대해서만 검색하게 됩니다.
- ❖ 다음은 부서번호가 10번이 아닌 직원의 직원이름, 부서번호, 직급을 출력해 봅시다.

예

```
SELECT ENAME, DEPTNO, JOB  
FROM EMP  
WHERE NOT DEPTNO=10;
```

예

```
SELECT ENAME, DEPTNO, JOB  
FROM EMP  
WHERE DEPTNO<>10;
```

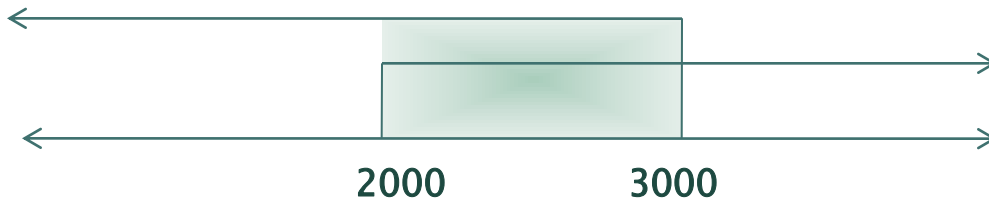

2.4 논리 연산자의 다양한 활용



❖ 2000에서 3000 사이의 급여를 받는 사원을 조회해 봅시다.

[조건1] 급여가 2000 이상 : $sal \geq 2000$

[조건2] 급여가 3000 이하 : $sal \leq 3000$



예	<pre>SELECT * FROM EMP WHERE SAL >= 2000 AND SAL <= 3000;</pre>
---	---

2.4 논리 연산자의 다양한 활용



❖ 커미션이 300 이거나 500 이거나 1400 인 사원을 검색해 봅시다.

[조건1] 커미션이 300 : COMM=300

[조건2] 커미션이 500 : COMM=500

[조건3] 커미션이 1400 : COMM=1400

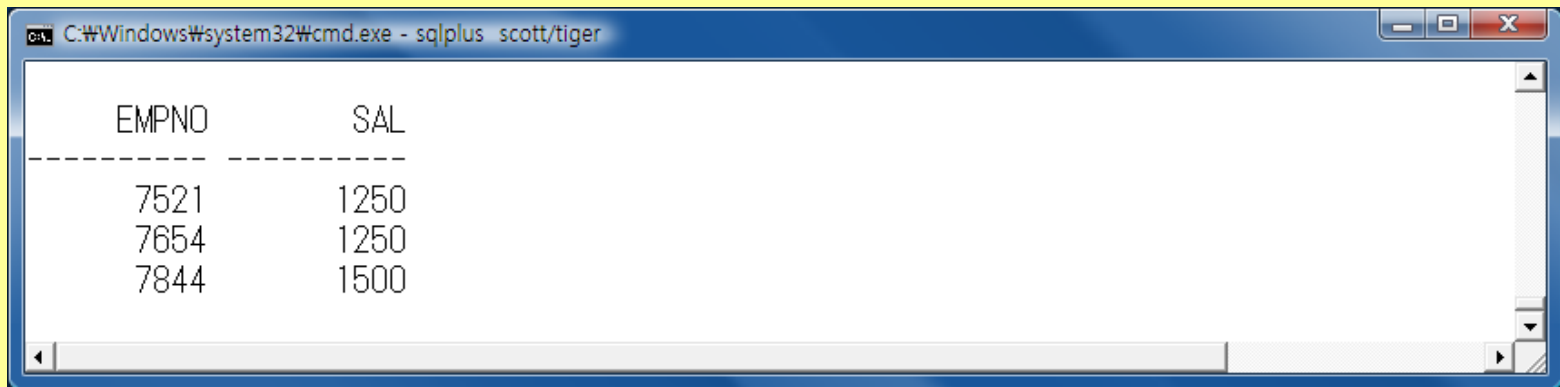
예

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

탄탄히 다지기

3. 사원 번호가 7521 이거나 7654 이거나 7844 인 사원들의 사원 번호와 급여를 검색하는 쿼리문을 비교 연산자와 OR 논리 연산자 사용하여 작성하시오.

<힌트> 사원 정보가 저장된 테이블의 이름은 EMP이고, 사원번호 컬럼은 EMPNO, 급여 컬럼은 SAL입니다.



EMPNO	SAL
7521	1250
7654	1250
7844	1500

03. BETWEEN AND 연산자



- ❖ 2000에서 3000 사이의 급여를 받는 사원과 같이 특정 범위 내에 속하는 데이터인지를 알아보기 위해서 비교연산자와 논리 연산자를 결합하여 표현할 수 있습니다.

예

```
SELECT *  
FROM EMP  
WHERE SAL >= 2000 AND SAL <= 3000;
```

03. BETWEEN AND 연산자



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

형식	<i>column_name</i> BETWEEN A AND B
----	------------------------------------

- ❖ 다음은 2000에서 3000 사이의 급여를 받는 사원을 조회하기 위해서 BETWEEN AND 연산자를 사용한 예입니다.

예	<pre>SELECT * FROM EMP WHERE SAL BETWEEN 2000 AND 3000;</pre>
---	---

03. BETWEEN AND 연산자



- ❖ 급여가 2000 미만이거나 3000 초과인 사원을 검색하기 위해서 비교 연산자와 OR 연산자로 다음과 같이 표현할 수 있습니다.

예

```
SELECT *  
FROM EMP  
WHERE SAL < 2000 OR SAL > 3000;
```

예

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

03. BETWEEN AND 연산자



- ❖ BETWEEN AND 연산자는 숫자형뿐만 아니라 문자형, 날짜형에도 사용할 수 있습니다.
- ❖ 주의할 점은 비교 대상이 되는 값을 단일 따옴표로 둘러싸야 한다는 점입니다.
- ❖ 1987년에 입사한 사원을 출력해 봅시다.

예

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

C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger

```
SQL> SELECT *  
2 FROM EMP  
3 WHERE HIREDATE BETWEEN '1987/01/01' AND '1987/12/31';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7876	ADAMS	CLERK	7788	87/05/23	1100		20

04. IN 연산자



- ❖ 커미션이 300 이거나 500 이거나 1400 인 사원을 검색하기 위해서 다음과 같이 쿼리문을 작성했습니다.

예	<pre>SELECT * FROM EMP WHERE COMM=300 OR COMM=500 OR COMM=1400;</pre>
---	---

- ❖ 비교 연산자와 논리 연산자 OR를 사용해야 합니다.
- ❖ 위 예제를 보면 OR 연산자로 묶인 비교 연산자 내의 컬럼은 COMM으로 동일합니다.

04. 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 있습니다.

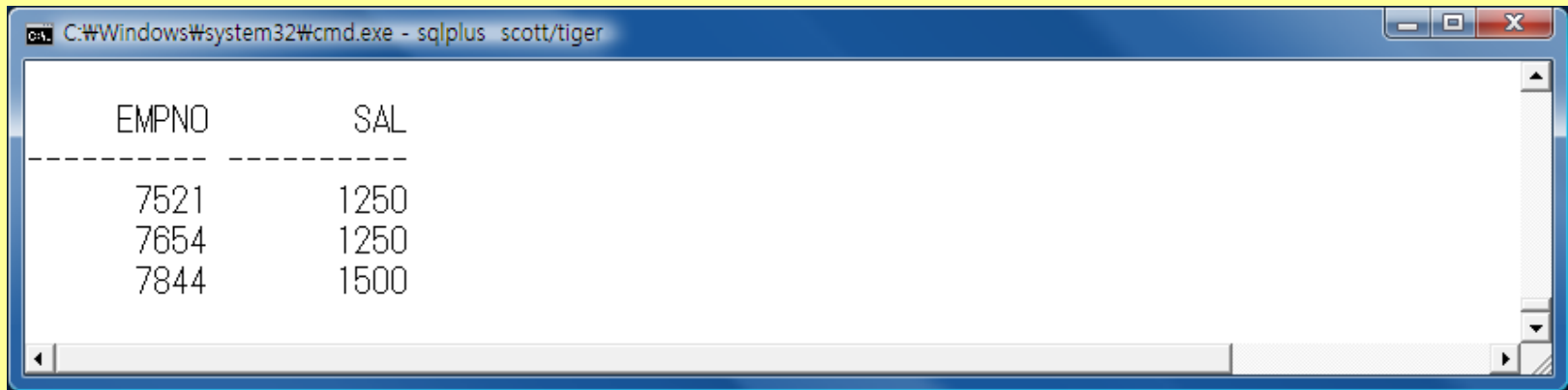
형식	<code>column_name IN(A, B, C)</code>
----	--------------------------------------

- ❖ 특정 필드의 값이 A이거나 B이거나 C 중에 어느 하나만 만족하더라도 출력하도록 하는 표현을 IN 연산자를 사용하여 할 수 있습니다.
- ❖ 이번에는 커미션이 300 이거나 500 이거나 1400 인 사원을 검색하기 위해서 IN 연산자를 사용해 봅시다.

예	<pre>SELECT * FROM EMP WHERE COMM IN(300, 500, 1400);</pre>
---	---

<탄탄히 다지기>

4. 7521 이거나 7654 이거나 7844 인 직원들을 검색하는 쿼리문을 비교 연산자와 논리 연산자 OR로 작성할 수 있지만, 이번에는 IN 연산자를 사용하여 작성해 보시오.



EMPNO	SAL
7521	1250
7654	1250
7844	1500

04. IN 연산자



- ❖ 이번에는 COMM이 300, 500, 1400이 모두 아닌 사원을 검색해 봅시다. <> 연산자를 사용하여 다음과 같이 표현할 수 있습니다.

예

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

- ❖ COMM이 300, 500, 1400이 모두 아닌 사원을 **NOT IN** 연산자를 사용하면 다음과 같이 간단하게 조회할 수 있습니다.

예

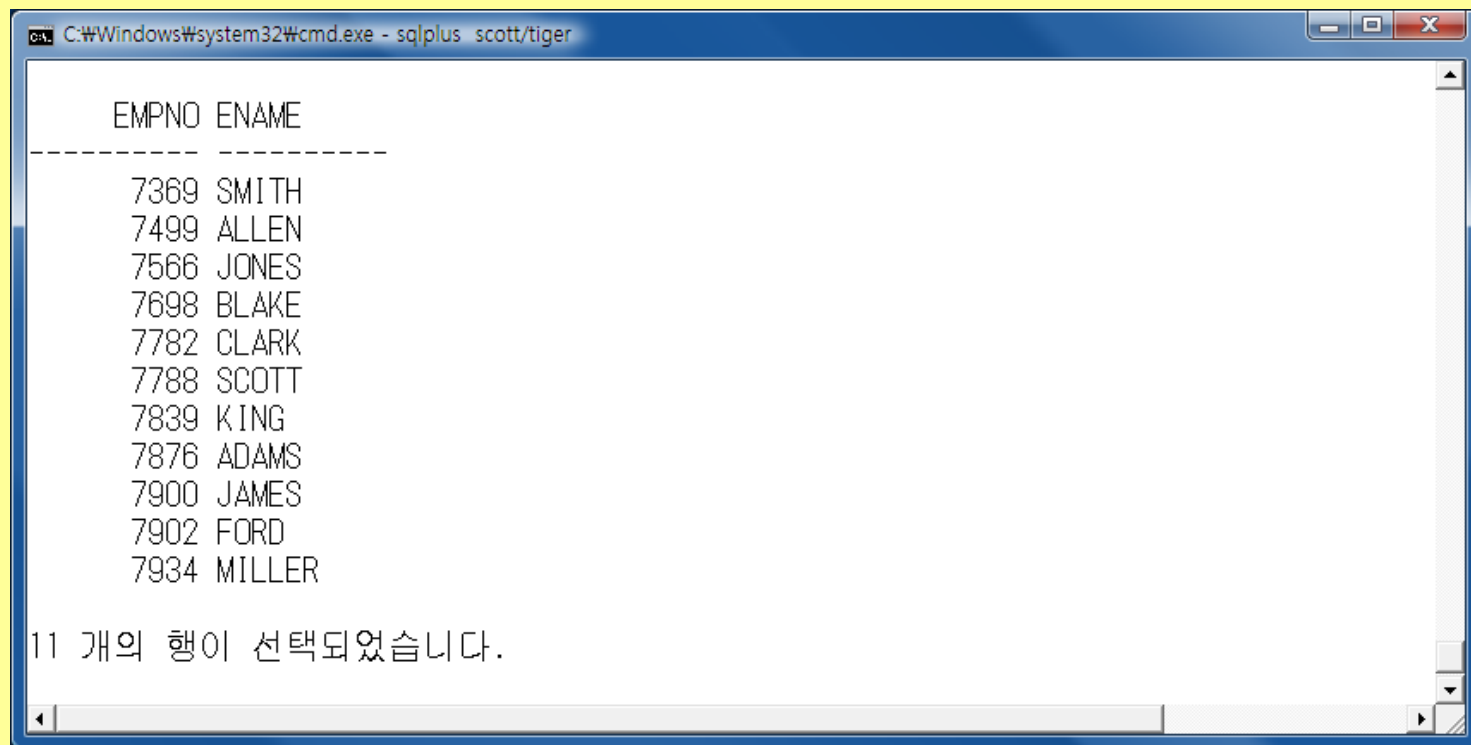
```
SELECT *  
FROM EMP  
WHERE COMM NOT IN(300, 500, 1400);
```

<탄탄히 다지기>

5. 사원 번호가 7521도 아니고 7654도 아니고 7844도 아닌 사원들을 검색하는 쿼리문을 작성하시오.

5.1. 비교 연산자와 논리 연산자 AND를 사용하여 작성하기

5.2. NOT IN 연산자를 사용하여 작성하기



The screenshot shows a SQL*Plus window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The window displays the following text:

```
EMPNO ENAME
-----
7369 SMITH
7499 ALLEN
7566 JONES
7698 BLAKE
7782 CLARK
7788 SCOTT
7839 KING
7876 ADAMS
7900 JAMES
7902 FORD
7934 MILLER
```

At the bottom of the window, it says "11 개의 행이 선택되었습니다."

05. LIKE 연산자와 와일드카드



- ❖ LIKE 연산자는 검색하고자 하는 값을 정확히 모를 경우에도 **검색** 가능하도록 하기 위해서 와일드카드와 함께 사용하여 원하는 내용을 검색하도록 합니다. 다음은 LIKE 연산자의 형식입니다.

형식 *column_name* LIKE *pattern*

- ❖ LIKE 다음에는 pattern을 기술해야 하는데 pattern에 다음과 같이 두 가지 와일드카드가 사용됩니다.

와일드카드는 패턴 매칭이나 검색에서 일부 문자열을 대체하는 데 사용되는 문자입니다.

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

5.1 와일드카드(%) 사용하기



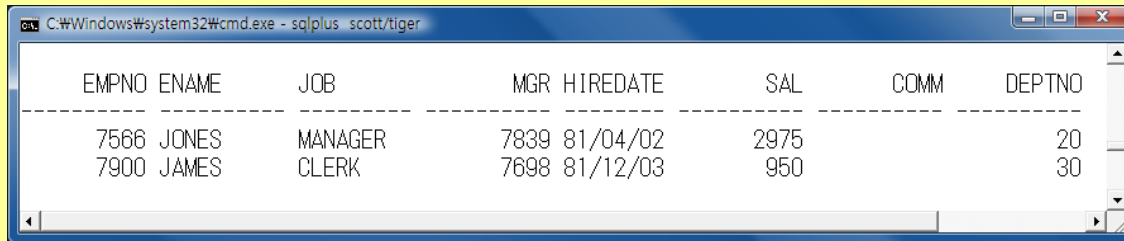
- ❖ 찾으려는 이름이 F로 시작 하는 것은 알지만 그 뒤의 문자는 모를 경우 `ename = 'F'`로 검색하게 되면 될까요?
- ❖ `ename = 'F'` 표현은 이름이 정확히 F인 사람만을 검색하겠다는 의미이기에 이름이 'F' 로 시작하는 사원을 검색하지 못합니다.
- ❖ 검색하고자 하는 값을 정확히 모를 경우 즉, 특정 문자 포함되기만 하고 그 이전이나 이후에 어떤 문자가 몇 개가 오든지 상관없다는 의미를 표현하기 위해서는 LIKE 연산자와 함께 %를 사용해야 합니다.

예

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

<탄탄히 다지기>

6. 사원들 중에서 이름이 J로 시작하는 사람만을 찾는 쿼리문을 작성해 보시오.



The screenshot shows a SQL*Plus window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The window displays the output of a query, which is a table with 8 columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The table contains two rows of data, both of which start with the letter 'J' in the ENAME column: JONES (MANAGER) and JAMES (CLERK).

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7566	JONES	MANAGER	7839	81/04/02	2975		20
7900	JAMES	CLERK	7698	81/12/03	950		30

5.1 와일드카드(%) 사용하기



- ❖ 이번에는 이름 중 A를 포함하는 사원을 검색해봅시다.
- ❖ 문자 A 앞뒤에 %를 기술하면 문자열 중간에 A 문자만 있으면 앞뒤에 어떤 문자열이 몇 개가 오는 상관없이 찾습니다.

예

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

C:\Windows\system32\cmd.exe - sqlplus scott/tiger

```
SQL> SELECT *  
2 FROM EMP  
3 WHERE ENAME LIKE '%A%';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7876	ADAMS	CLERK	7788	87/05/23	1100		20
7900	JAMES	CLERK	7698	81/12/03	950		30

7 개의 행이 선택되었습니다.

5.1 와일드카드(%) 사용하기



- ❖ 이번에는 문자열의 앞에 어떤 문자열이 몇 개가 오는 상관없이 N으로 끝나는 데이터를 찾아봅시다.

예

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

C:\Windows\system32\cmd.exe - sqlplus scott/tiger

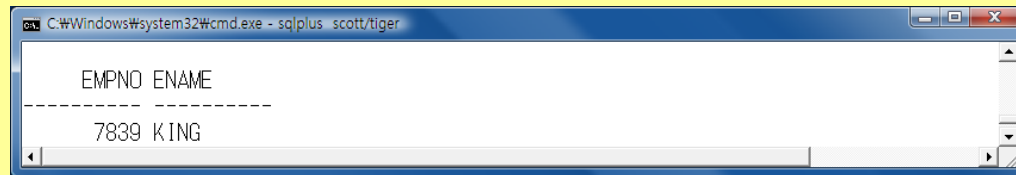
```
SQL> SELECT *  
2 FROM EMP  
3 WHERE ENAME LIKE '%N';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	81/02/20	1600	300	30
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30

<탄탄히 다지기>

7. 와일드 카드를 사용하여 다음 조건에 만족하는 쿼리문을 작성하시오.

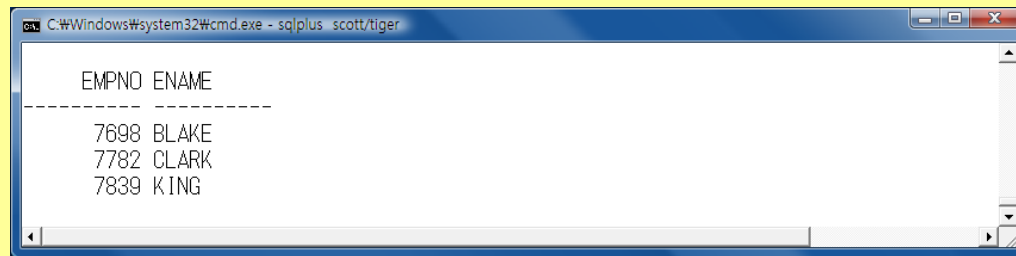
7.1. 사원들 중에서 이름이 K로 시작하는 사원의 사원번호와 사원이름을 출력하시오.



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger

EMPNO ENAME
-----
7839 KING
```

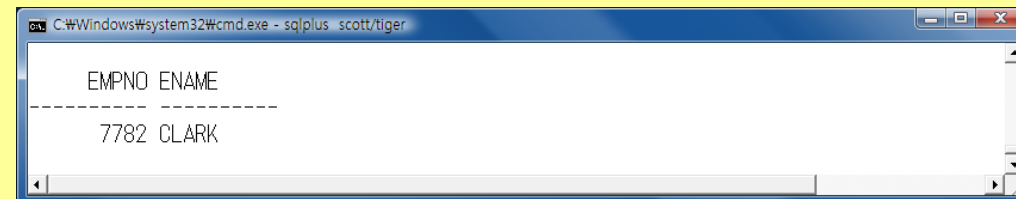
7.2. 이름 중에 K를 포함하는 사원의 사원번호와 사원이름을 출력하시오.



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger

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

7.3. 이름이 K로 끝나는 사원의 사원번호와 사원이름을 출력하시오.



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger

EMPNO ENAME
-----
7782 CLARK
```

5.2 와일드카드(_) 사용하기



- ❖ _ 역시 %와 마찬가지로 어떤 문자가 오는 상관없다는 의미로 사용되는 와일드카드입니다.
- ❖ 차이점은 %는 몇 개의 문자가 오는 상관없지만 _ 는 단 **한 문자**에 대해서만 와일드카드 역할을 합니다.
- ❖ 다음은 이름의 두 번째 글자가 A인 사원을 찾는 예제입니다.

예

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

C:\Windows\system32\cmd.exe - sqlplus scott/tiger

```
SQL> SELECT *  
2 FROM EMP  
3 WHERE ENAME LIKE '_A%';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7521	WARD	SALESMAN	7698	81/02/22	1250	500	30
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7900	JAMES	CLERK	7698	81/12/03	950		30

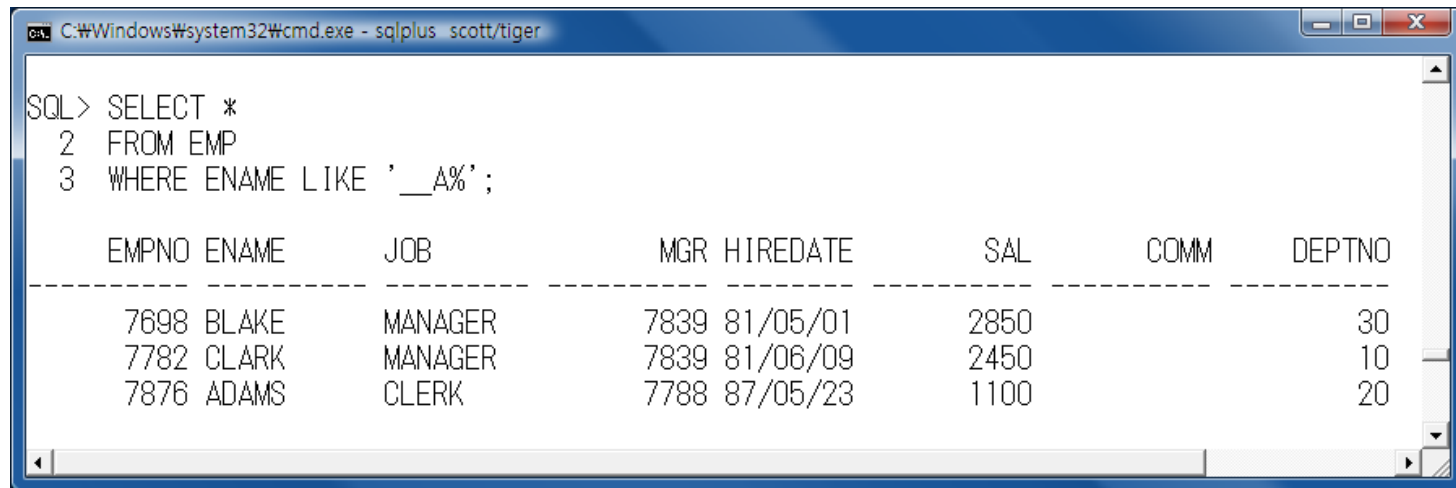
5.2 와일드카드(_) 사용하기



- ❖ 세 번째 글자가 A인 자료를 검색하려면 `_ _ A%`처럼 기술해야 합니다.

예

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



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL>". The user has entered the following SQL query:

```
SQL> SELECT *  
2 FROM EMP  
3 WHERE ENAME LIKE ' _ _ A%';
```

The query results are displayed in a table with the following columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The results are as follows:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7698	BLAKE	MANAGER	7839	81/05/01	2850		30
7782	CLARK	MANAGER	7839	81/06/09	2450		10
7876	ADAMS	CLERK	7788	87/05/23	1100		20

5.4 NOT LIKE 연산자



❖ 이름에 A를 포함하지 않은 사람만을 검색하려고 할 경우에

NOT LIKE 연산자를 사용합니다.

예

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

C:\Windows\system32\cmd.exe - sqlplus scott/tiger

```
SQL> SELECT *  
2 FROM EMP  
3 WHERE ENAME NOT LIKE '%A%';
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	80/12/17	800		20
7566	JONES	MANAGER	7839	81/04/02	2975		20
7788	SCOTT	ANALYST	7566	87/04/19	3000		20
7839	KING	PRESIDENT		81/11/17	5000		10
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30
7902	FORD	ANALYST	7566	81/12/03	3000		20
7934	MILLER	CLERK	7782	82/01/23	1300		10

7 개의 행이 선택되었습니다.

06. NULL을 위한 연산자



- ❖ 사원 테이블의 커미션 컬럼에 널이 저장되어 있으므로 = 연산자로 커미션을 받지 않는 사원에 대한 검색해 봅시다.

예

```
SELECT *  
FROM EMP  
WHERE COMM=NULL;
```

- ❖ NULL 값을 가진 데이터와 비교 연산한 결과는 다음과 같습니다. 왜냐하면 NULL이 저장되어 있는 경우에는 = 연산자로 판단할 수 없기 때문입니다.

06. NULL을 위한 연산자

```
SQL> SELECT *  
  2  FROM EMP  
  3  WHERE COMM=300  
  4      AND COMM=NULL;  
no rows selected
```

```
SQL> SELECT EMPNO, ENAME, JOB  
  2  FROM EMP  
  3  WHERE COMM=300  
  4      OR COMM=NULL;
```

EMPNO	ENAME	JOB
7499	ALLEN	SALESMAN

6.1 IS NULL과 IS NOT NULL



- ❖ 어떤 컬럼을 NULL 즉, 모르는 값과 같다(=)라는 것은 의미상으로 말이 되지 않기 때문에 = 대신 IS NULL 연산자를 사용해야 합니다.
- ❖ IS NULL 연산자 역시 조건절에 사용되면 대상 컬럼과 연산자, 비교할 값 세부분으로 구성되어야 합니다.

형식	대상컬럼 IS (연산자) NULL(비교값)
----	-------------------------

- ❖ 이번에는 IS NULL 연산자를 사용하여 커미션을 받지 않는 사원을 검색해 봅시다.

예	<pre>SELECT * FROM EMP WHERE COMM IS NULL;</pre>
---	--

6.1 IS NULL과 IS NOT NULL



- ❖ 이번에는 커미션을 받는 사원에 대해 검색해 봅시다. 문장대로 해석하면 커미션(COMM) 칼럼이 NULL 아닌 자료만 추출하면 되므로 IS NOT NULL 연산자를 사용하면 됩니다.

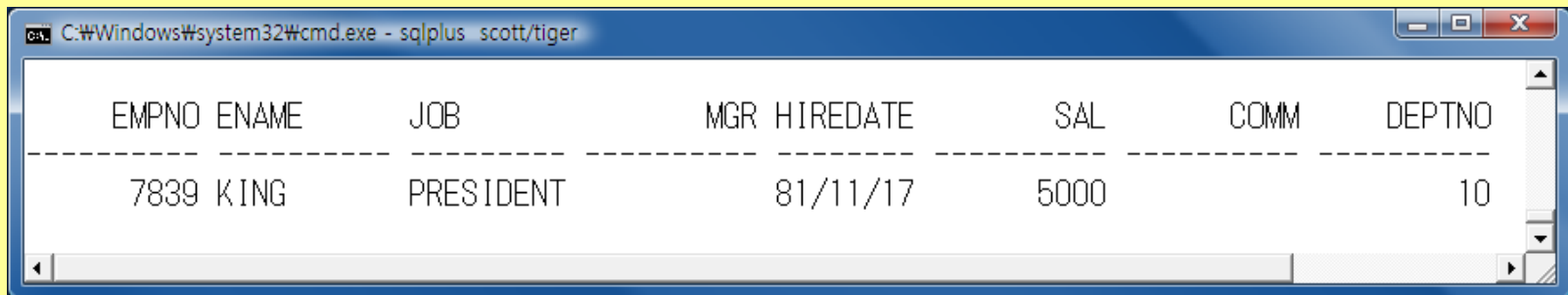
형식

```
SELECT *  
FROM EMP  
WHERE COMM IS NOT NULL;
```

<탄탄히 다지기>

8. 상관이 없는 사원(CEO 가 되겠지요!)을 검색하기 위한 SQL 문을 작성해 보시오.

<힌트> 사원 정보가 저장된 테이블의 이름은 EMP이고, 사원 테이블에는 상관의 사원번호를 저장하는 MGR 컬럼이 존재합니다. MGR 컬럼에 NULL이 저장되어 있으면 CEO가 되겠지요.^^



The screenshot shows a SQL*Plus window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". It displays the contents of the EMP table. The table has columns: EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The data shown is for the employee with EMPNO 7839, ENAME KING, JOB PRESIDENT, MGR NULL, HIREDATE 81/11/17, SAL 5000, COMM NULL, and DEPTNO 10.

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7839	KING	PRESIDENT		81/11/17	5000		10

07. 정렬을 위한 ORDER BY 절



- ❖ 정렬이란 크기 순서대로 나열하는 것을 의미합니다.
- ❖ 오름차순(ascending) 정렬 방식
 - 작은 것이 위에 출력되고 아래로 갈수록 큰 값이 출력
- ❖ 내림차순(descending) 정렬 방식이
 - 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 합니다.

	ASC(오름차순)	DESC(내림차순)
숫자	작은 값부터 정렬	큰 값부터 정렬
문자	사전 순서로 정렬	사전 반대 순서로 정렬
날짜	오래된 -> 최근	최근 -> 오래된
NULL	가장 마지막에 나온다.	가장 먼저 나온다.

7.1 오름차순 정렬을 위한 ASC



- ❖ 오름차순 정렬은 작은 값부터 큰 값으로 정렬하는 것을 의미합니다.(예:1~9, 'A' ~ 'Z') 이를 위해서는 ASC를 칼럼 다음에 기술해야 하는데 만일 생략하게 되면 디폴트로 ASC로 지정되어 있기 때문에 오름차순으로 출력됩니다.
- ❖ 다음은 급여 컬럼을 기준으로 오름차순으로 정렬한 예입니다.

예

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

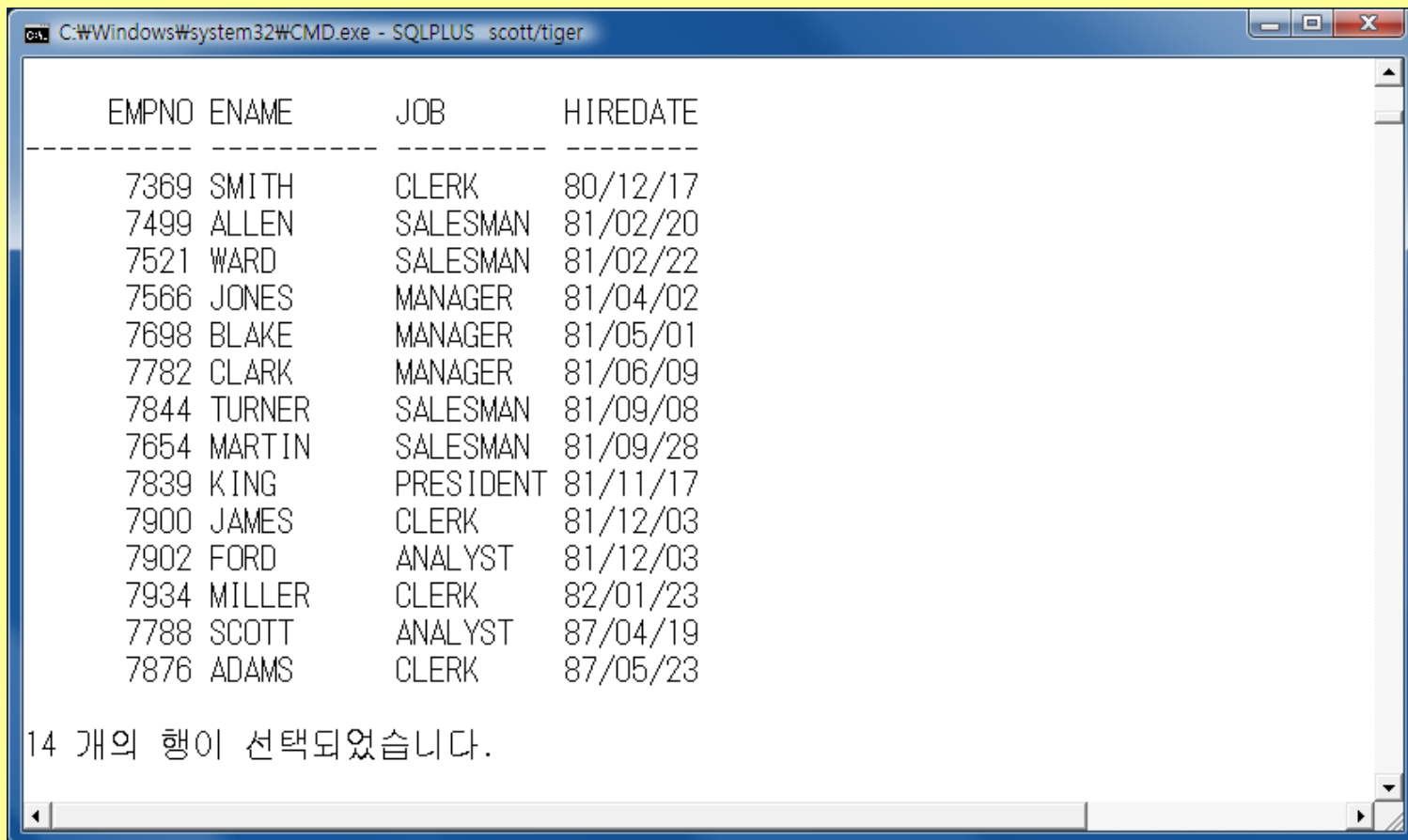
- ❖ 정렬방식을 지정하지 않은 경우에는 디폴트로 오름차순으로 정렬합니다.

예

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

<탄탄히 다지기>

9. EMP 테이블의 자료를 입사일을 오름차순으로 정렬하고 사원번호, 사원명, 직급, 입사일 칼럼을 출력하는 쿼리문을 작성하시오.



EMPNO	ENAME	JOB	HIREDATE
7369	SMITH	CLERK	80/12/17
7499	ALLEN	SALESMAN	81/02/20
7521	WARD	SALESMAN	81/02/22
7566	JONES	MANAGER	81/04/02
7698	BLAKE	MANAGER	81/05/01
7782	CLARK	MANAGER	81/06/09
7844	TURNER	SALESMAN	81/09/08
7654	MARTIN	SALESMAN	81/09/28
7839	KING	PRESIDENT	81/11/17
7900	JAMES	CLERK	81/12/03
7902	FORD	ANALYST	81/12/03
7934	MILLER	CLERK	82/01/23
7788	SCOTT	ANALYST	87/04/19
7876	ADAMS	CLERK	87/05/23

14 개의 행이 선택되었습니다.

7.2 내림차순 정렬을 위한 DESC



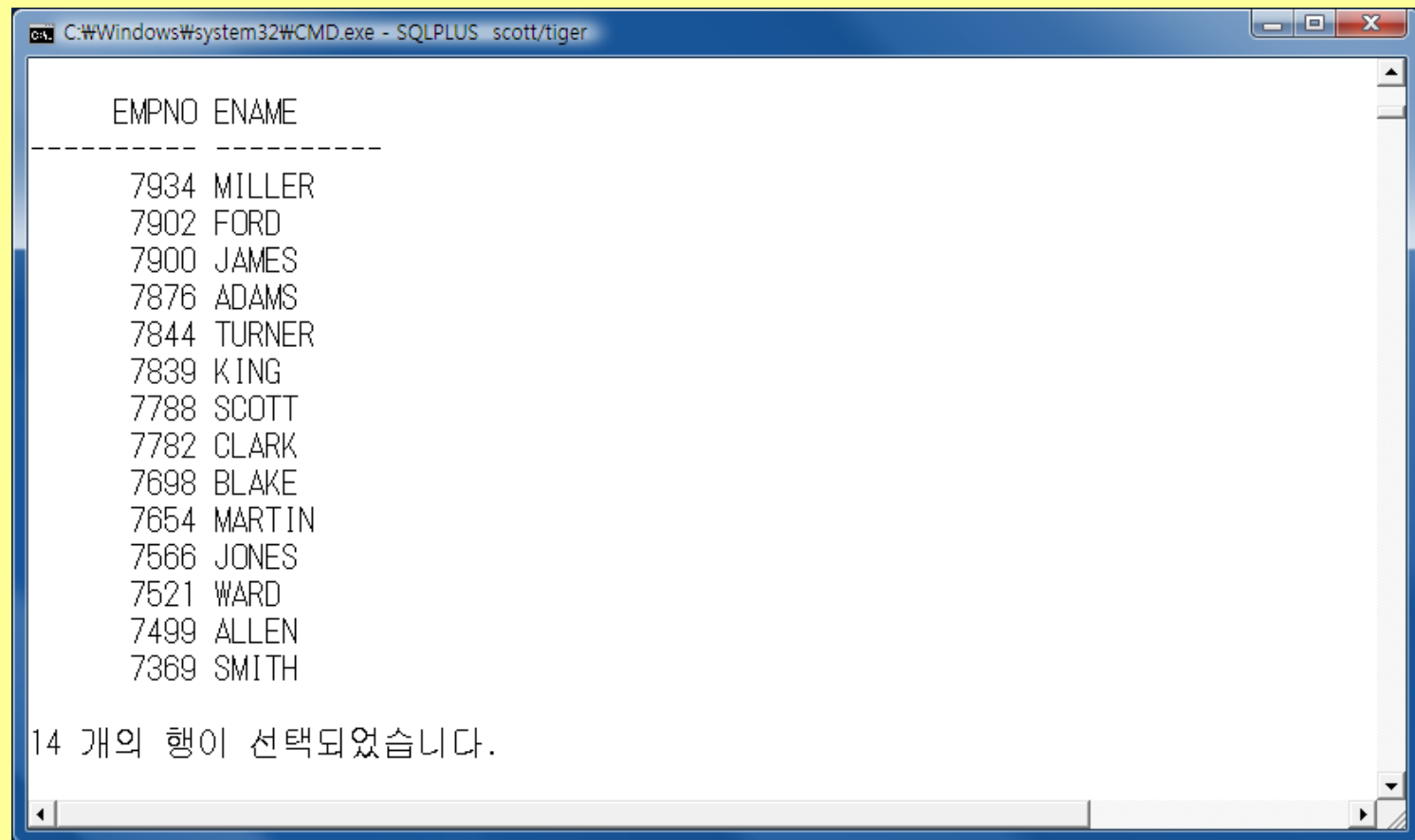
- ❖ 내림차순 정렬은 큰 값부터 작은 값으로 정렬을 하는 것이다.(예:9~1, Z~A)
- ❖ 이번에는 급여를 많이 받는 사람부터 적게 받는 사람 순으로 순차적으로 출력해 봅시다.

예	<pre>SELECT * FROM EMP ORDER BY SAL DESC;</pre>
---	---

- ❖ 큰 값이 위에 출력되고 아래로 갈수록 작은 값이 출력되도록 하려면 내림차순(descending) 으로 정렬해야 하기 때문에 칼럼 다음에 DESC를 기술해야 합니다.

<탄탄히 다지기>

10. EMP 테이블의 자료를 사원번호를 기준으로 내림차순으로 정렬하여 사원 번호와 사원명 칼럼을 표시하시오.



```
C:\Windows\system32\CMD.exe - SQLPLUS scott/tiger

EMPNO ENAME
-----
7934 MILLER
7902 FORD
7900 JAMES
7876 ADAMS
7844 TURNER
7839 KING
7788 SCOTT
7782 CLARK
7698 BLAKE
7654 MARTIN
7566 JONES
7521 WARD
7499 ALLEN
7369 SMITH

14 개의 행이 선택되었습니다.
```

7.3 문자 순으로 출력



- ❖ 크기에 대한 비교는 수치 데이터 뿐만 아니라 문자 데이터나 날짜 데이터에 대해서도 가능합니다.
- ❖ 문자 데이터의 경우 아스키 코드 값으로 저장되므로 아스키 코드 값을 기준으로 정렬됩니다.
- ❖ 오름차순인 경우에는 A, B, . . . Z 순으로 출력되고 내림차순인 경우에는 Z, Y, . . . A 순으로 출력됩니다.
- ❖ 다음은 사원의 이름을 알파벳 순(오름차순)으로 출력하는 예제입니다.

예

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


7.4 날짜 순으로 출력



- ❖ 날짜의 경우에도 오름차순 혹은 내림차순으로 출력할 수 있습니다.
- ❖ 오름차순으로 지정하면 가장 오래된 과거의 시점이 가장 위에 출력되고 아래로 갈수록 최근 시점이 출력됩니다.
- ❖ 내림차순인 경우에는 최근 시점부터 출력합니다.
- ❖ 다음은 가장 최근에 입사한 사람부터 출력하는 예제입니다.

예

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

7.5 정렬 방식에 여러 가지 조건 제시



- ❖ 급여를 많이 받는 사람부터 적게 받는 사람 순으로 순차적으로 출력하는 결과 화면을 살펴보면 동일한 급여를 받는 사람이 존재합니다.
- ❖ 급여가 같은 사람이 존재할 경우 이름의 철자가 빠른 사람부터 출력되도록 하려면 정렬 방식을 여러 가지로 지정해야 합니다.

예

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

<탄탄히 다지기>

11. 부서 번호가 빠른 사원부터 출력하되

같은 부서내의 사원을 출력할 경우 최근에 입사한 사원부터 출력하도록 하세요.

단, 사원 번호, 입사일, 사원 이름, 급여를 출력하시오.

EMPNO	HIREDATE	ENAME	SAL
7934	82/01/23	MILLER	1300
7839	81/11/17	KING	5000
7782	81/06/09	CLARK	2450
7876	87/07/13	ADAMS	1100
7788	87/07/13	SCOTT	3000
7902	81/12/03	FORD	3000
7566	81/04/02	JONES	2975
7369	80/12/17	SMITH	800
7900	81/12/03	JAMES	950
7654	81/09/28	MARTIN	1250
7844	81/09/08	TURNER	1500
7698	81/05/01	BLAKE	2850
7521	81/02/22	WARD	1250
7499	81/02/20	ALLEN	1600