



03강.oracle

- 연산자 사용 (+, -, *, /)
- 별칭 as, 중복제거 Distinct
 - Where 조건문

1 연산자 사용

산술연산자

종류	예
+	SELECT sal + comm FROM emp;
-	SELECT sal - 100 FROM emp;
*	SELECT sal * 12 FROM emp;
/	SELECT sal / 2 FROM emp;

❖ 급여로 연봉 계산을 해보도록 합시다. 일반적으로 연봉은 급여를 12번 곱한 것이므로 연봉을 구하기 위해서 산술 연산자를 사용합시다.

예	SELECT ename, sal, sal*12 FROM emp;
---	--

1 연산자 사용

Null 데이터

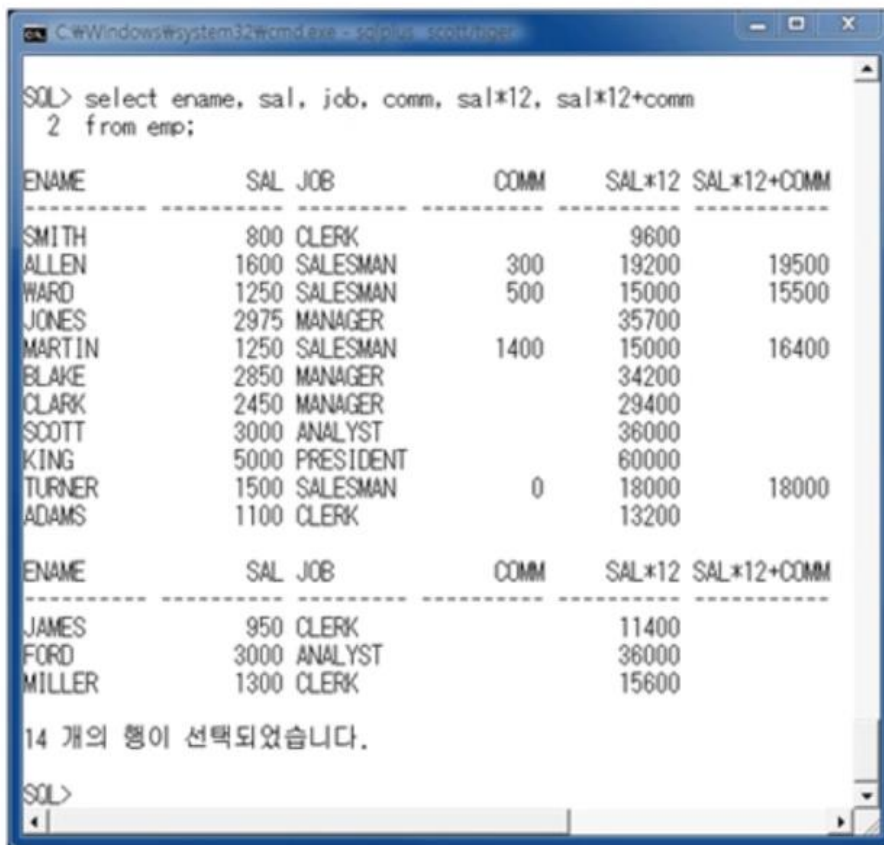
- ❖ 오라클에서의 널은 매우 중요한 데이터입니다. 왜냐하면 오라클에서는 컬럼에 널값이 저장되는 것을 허용하는데 널 값을 제대로 이해하지 못한 채 쿼리문을 사용하면 원하지 않는 결과를 얻을 수 있기 때문입니다.
- ❖ 다음은 널에 대한 이해를 돕기 위해서 다양한 널의 정의를 살펴본 것입니다.
 1. 0(zero)도 아니고
 2. 빈 공간도 아니다.
 3. 미확정(해당 사항 없음), 알 수 없는(unknown) 값을 의미한다.
 4. 어떤 값인지 알 수 없지만 어떤 값이 존재하고 있다.
 5. ? 혹은 ∞ 의 의미이므로
 6. 연산, 할당, 비교가 불가능하다.

1 연산자 사용

Null 데이터

예

```
SELECT ename, sal, job, comm, sal*12, sal*12+comm  
FROM emp;
```



```
SQL> select ename, sal, job, comm, sal*12, sal*12+comm  
2 from emp;
```

ENAME	SAL	JOB	COMM	SAL*12	SAL*12+COMM
SMITH	800	CLERK		9600	
ALLEN	1600	SALESMAN	300	19200	19500
WARD	1250	SALESMAN	500	15000	15500
JONES	2975	MANAGER		35700	
MARTIN	1250	SALESMAN	1400	15000	16400
BLAKE	2850	MANAGER		34200	
CLARK	2450	MANAGER		29400	
SCOTT	3000	ANALYST		36000	
KING	5000	PRESIDENT		60000	
TURNER	1500	SALESMAN	0	18000	18000
ADAMS	1100	CLERK		13200	
JAMES	950	CLERK		11400	
FORD	3000	ANALYST		36000	
MILLER	1300	CLERK		15600	

14 개의 행이 선택되었습니다.

```
SQL>
```

영업직인 경우 커미션(comm) 컬럼에 값이 저장되어 있으므로 제대로 연봉 계산을 하게 된다.

영업직인 경우 무능력하여 받을 커미션(comm)이 없더라도 0으로 저장되어 있으므로 연봉 계산이 제대로 된다.

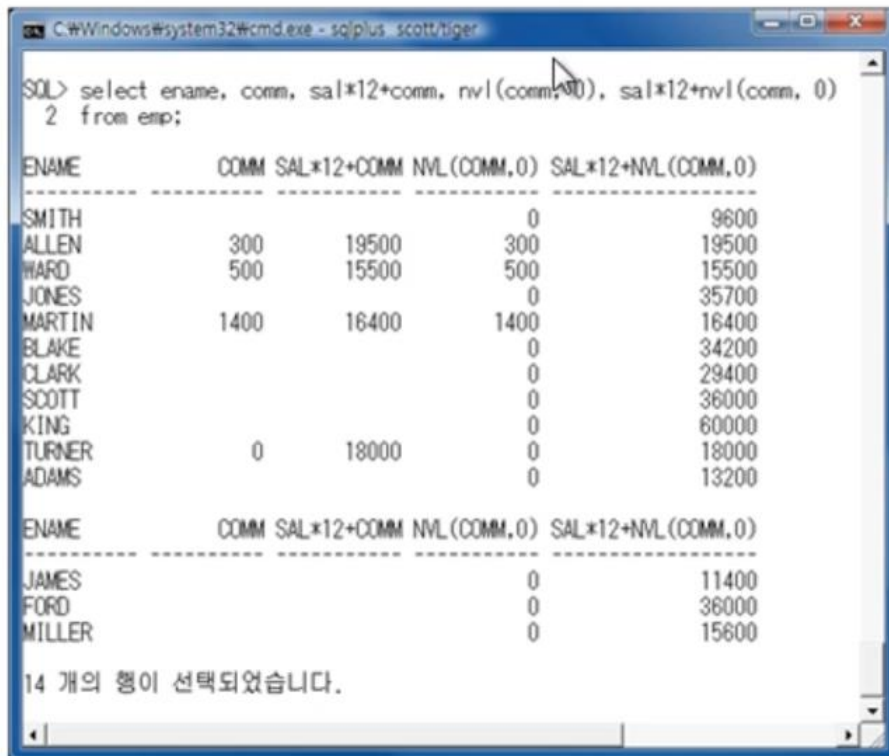
영업직인 아닌 경우에는 커미션에 널 값이 저장되어 있어서 연봉 계산 결과도 널값으로 구해지는 모순이 발생한다.

1 연산자 사용

Nvl(comm,0) -> nvl은 comm에 null 이 있으면 0을 입력

예

```
select ename, comm, sal*12+comm,  
       nvl(comm, 0), sal*12+nvl(comm, 0)  
from emp;
```



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

```
SQL> select ename, comm, sal*12+comm, nvl(comm,0), sal*12+nvl(comm, 0)  
2 from emp;
```

ENAME	COMM	SAL*12+COMM	NVL(COMM,0)	SAL*12+NVL(COMM,0)
SMITH			0	9600
ALLEN	300	19500	300	19500
WARD	500	15500	500	15500
JONES			0	35700
MARTIN	1400	16400	1400	16400
BLAKE			0	34200
CLARK			0	29400
SCOTT			0	36000
KING			0	60000
TURNER	0	18000	0	18000
ADAMS			0	13200
JAMES			0	11400
FORD			0	36000
MILLER			0	15600

14 개의 행이 선택되었습니다.

연봉 계산을 위해 사원 테이블에서 급여와 커미션 칼럼을 살펴본 결과 영업사원이 아닌 직원들의 커미션은 NULL로 지정되어 있으므로 연봉을 올바르게 계산하기 위해서는 커미션이 NULL인 경우 0으로 변경하여 계산에 참여하도록 해야 합니다.

오라클에서는 NULL을 0 또는 다른 값으로 변환하기 위해서 사용하는 함수로 NVL을 제공합니다. 커미션에 널이 저장되어 있더라도 널을 다른 값으로 변환하는 NVL 함수를 사용하면 제대로 된 계산 결과를 얻을 수 있습니다.

1 As 컬럼 별칭

As라는 키워드 후 별칭명 기술

- ❖ 컬럼 이름 대신 별칭을 출력하고자 하면 컬럼을 기술한 바로 뒤에 AS 라는 키워드를 쓴 후 별칭을 기술합니다.

예

```
select ename, sal*12+nvl(comm, 0) as Annsal  
from emp;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The user has entered the SQL command: `SQL> select ename, sal*12+nvl(comm, 0) as Annsal from emp;`. The output displays a table with two columns: `ENAME` and `ANNSAL`. The data rows are: SMITH (9600), ALLEN (19500), WARD (15500), JONES (35700), and MARTIN (16400).

ENAME	ANNSAL
SMITH	9600
ALLEN	19500
WARD	15500
JONES	35700
MARTIN	16400

Below the command window, there is a detailed breakdown of the calculation for each employee. It shows the `NAME`, `COMM`, `SAL*12+COMM`, `NVL(COMM,0)`, and `SAL*12+NVL(COMM,0)` values. The `SAL*12+NVL(COMM,0)` column is highlighted with a red box. The data is as follows:

NAME	COMM	SAL*12+COMM	NVL(COMM,0)	SAL*12+NVL(COMM,0)
SMITH			0	9600
ALLEN	300	19500	300	19500
WARD	500	15500	500	15500
JONES			0	35700
MARTIN	1400	16400	1400	16400
SLAKE			0	34200
SLARK			0	29400
SCOTT			0	36000
TING			0	60000
TURNER	0	18000	0	18000
DAMS			0	13200

Below this table, another partial table is shown with columns `NAME`, `COMM`, `SAL*12+COMM`, `NVL(COMM,0)`, and `SAL*12+NVL(COMM,0)`. The data rows are: JAMES (11400), FORD (36000), and MILLER (15600).

NAME	COMM	SAL*12+COMM	NVL(COMM,0)	SAL*12+NVL(COMM,0)
JAMES			0	11400
FORD			0	36000
MILLER			0	15600

At the bottom of the screenshot, it says "4 개의 행이 선택되었습니다." (4 rows selected).

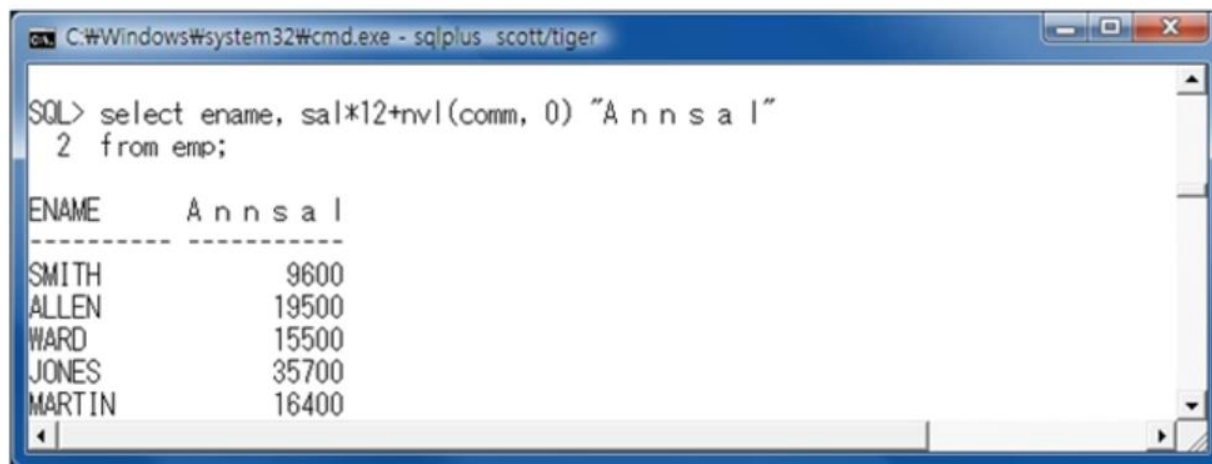
1 As 컬럼 별칭

“ ” 별칭 부여

- ❖ 위 예를 살펴보면 별칭을 부여 할 때에는 대소문자를 섞어서 기술하였는데 출력 결과를 보면 일괄적으로 대문자로 출력된 것을 확인할 수 있습니다.
- ❖ 대소문자를 구별하고 싶으면 “ ” 을 사용합니다.
- ❖ “ ” 을 사용하여 별칭을 부여할 경우에는 별칭에 공백문자나 \$,_, #등 특수 문자를 포함시킬 수 있습니다.

예

```
select ename, sal*12+nvl(comm,0) "A n n s a l"  
from emp;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The user has entered the SQL command: `SQL> select ename, sal*12+nvl(comm,0) "A n n s a l" 2 from emp;`. The output is a table with two columns: `ENAME` and `A n n s a l`. The data rows are: `SMITH 9600`, `ALLEN 19500`, `WARD 15500`, `JONES 35700`, and `MARTIN 16400`.

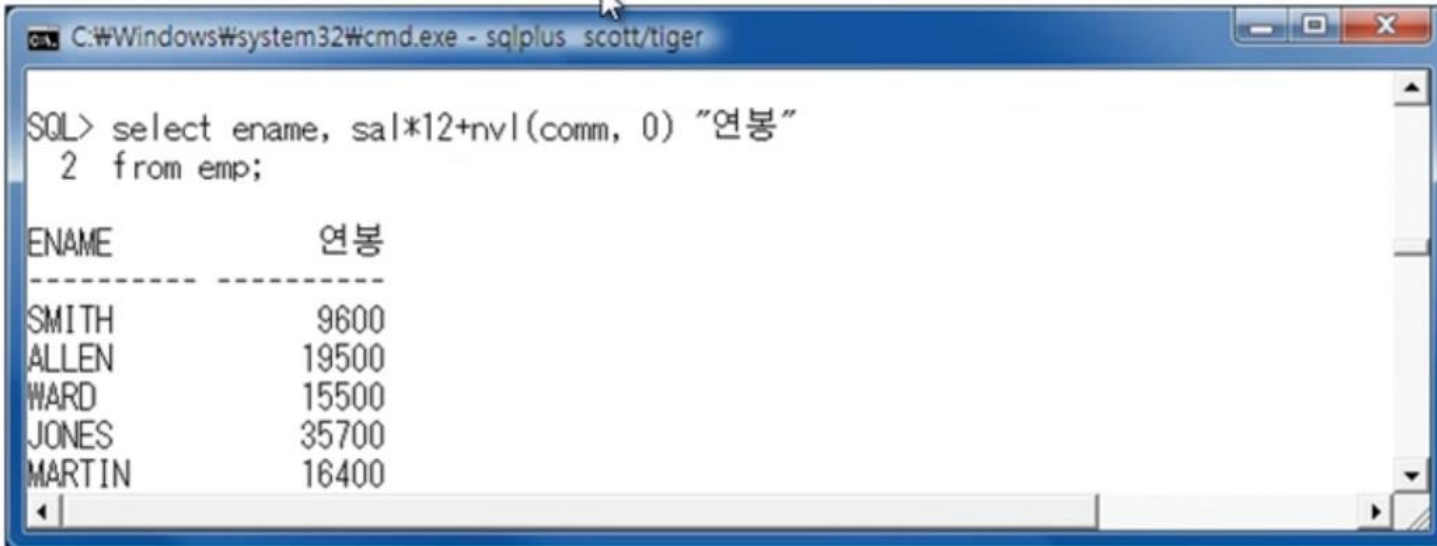
ENAME	A n n s a l
SMITH	9600
ALLEN	19500
WARD	15500
JONES	35700
MARTIN	16400

"" 한글 별칭 부여

- ❖ 영어가 아닌 한글로 별칭을 부여해 봅시다.
- ❖ 오라클에서 한글을 지원하므로 별칭이 아닌 테이블을 생성할 때 컬럼을 설정하면서 컬럼 이름도 한글로 부여할 수 있습니다.

예

```
select ename, sal*12+nvl(comm, 0) "연봉"  
from emp;
```

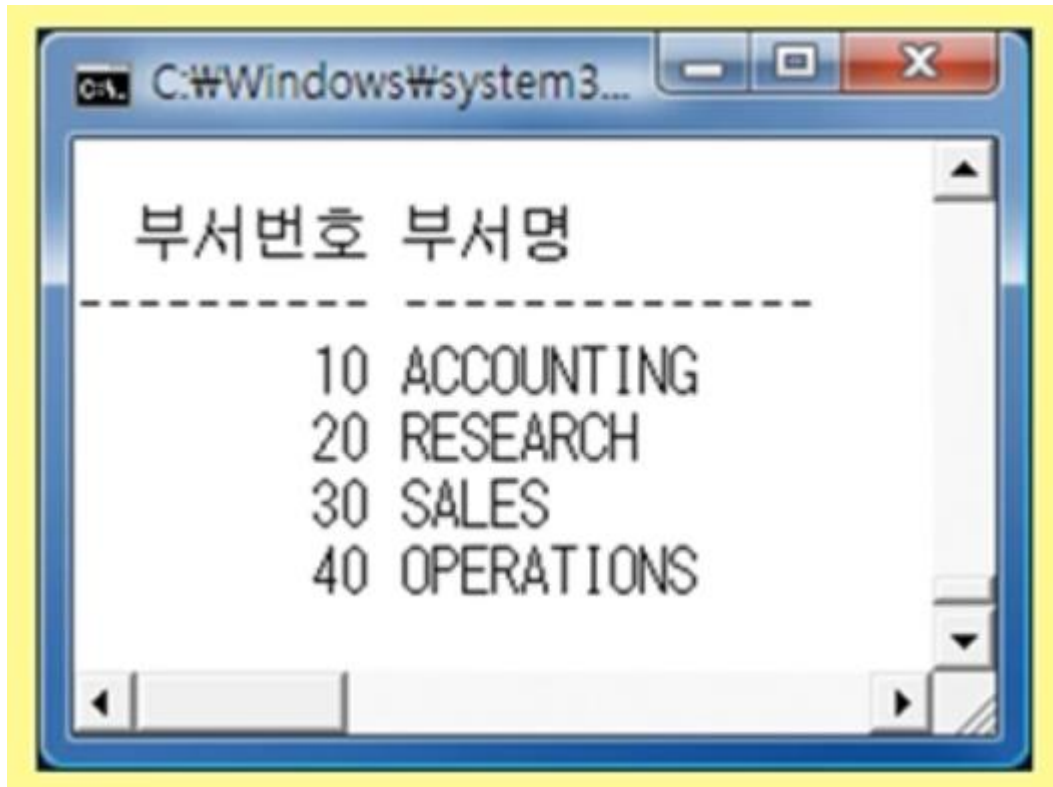


The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The user has entered the SQL command: `SQL> select ename, sal*12+nvl(comm, 0) "연봉" 2 from emp;`. The output displays a table with two columns: "ENAME" and "연봉". The data rows are: SMITH (9600), ALLEN (19500), WARD (15500), JONES (35700), and MARTIN (16400).

ENAME	연봉
SMITH	9600
ALLEN	19500
WARD	15500
JONES	35700
MARTIN	16400

1 실습.

아래와 같이 출력하시오.



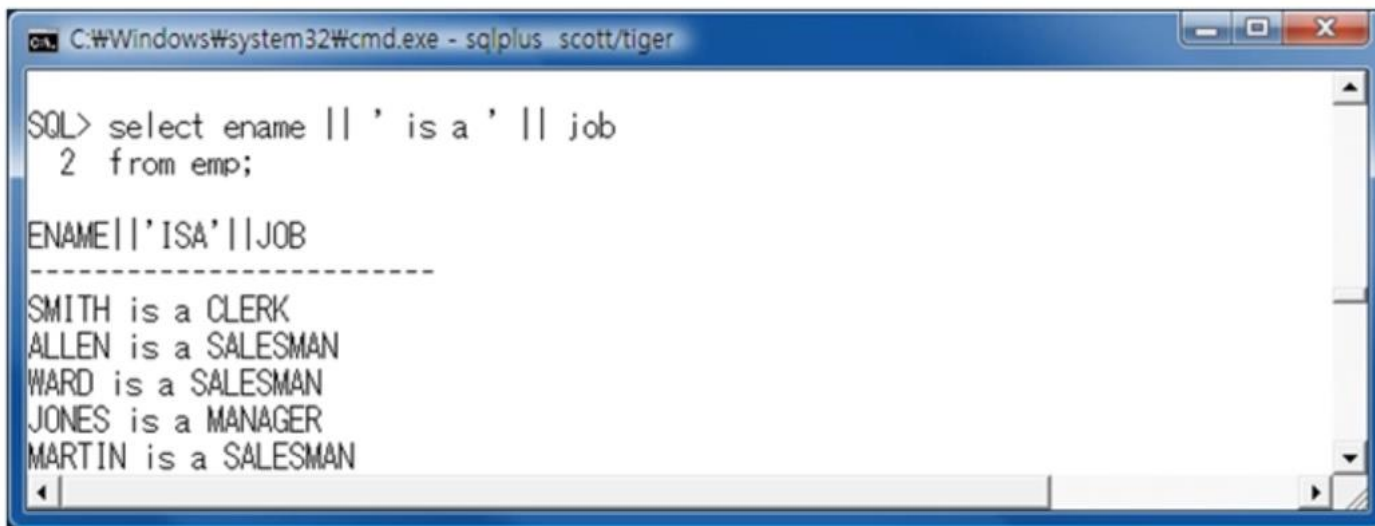
1 Concatenation 연산자의 정의와 사용

||

- ❖ 결과를 살펴보면 컬럼과 특정 값 사이에 공백이 생기는 것을 확인할 수 있습니다. 정말 영 문장처럼 보이도록 하기 위해서 “||” 를 컬럼과 문자열 사이에 기술하여 하나로 연결하여 출력하면 됩니다.

예

```
select ename || ' is a ' || job  
from emp;
```



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 query: "select ename || ' is a ' || job from emp;". The output shows a header line "ENAME||'ISA' ||JOB" followed by a dashed separator line. Below the separator, the results are listed: "SMITH is a CLERK", "ALLEN is a SALESMAN", "WARD is a SALESMAN", "JONES is a MANAGER", and "MARTIN is a SALESMAN".

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
SQL> select ename || ' is a ' || job  
2  from emp;  
  
ENAME||'ISA' ||JOB  
-----  
SMITH is a CLERK  
ALLEN is a SALESMAN  
WARD is a SALESMAN  
JONES is a MANAGER  
MARTIN is a SALESMAN
```

1

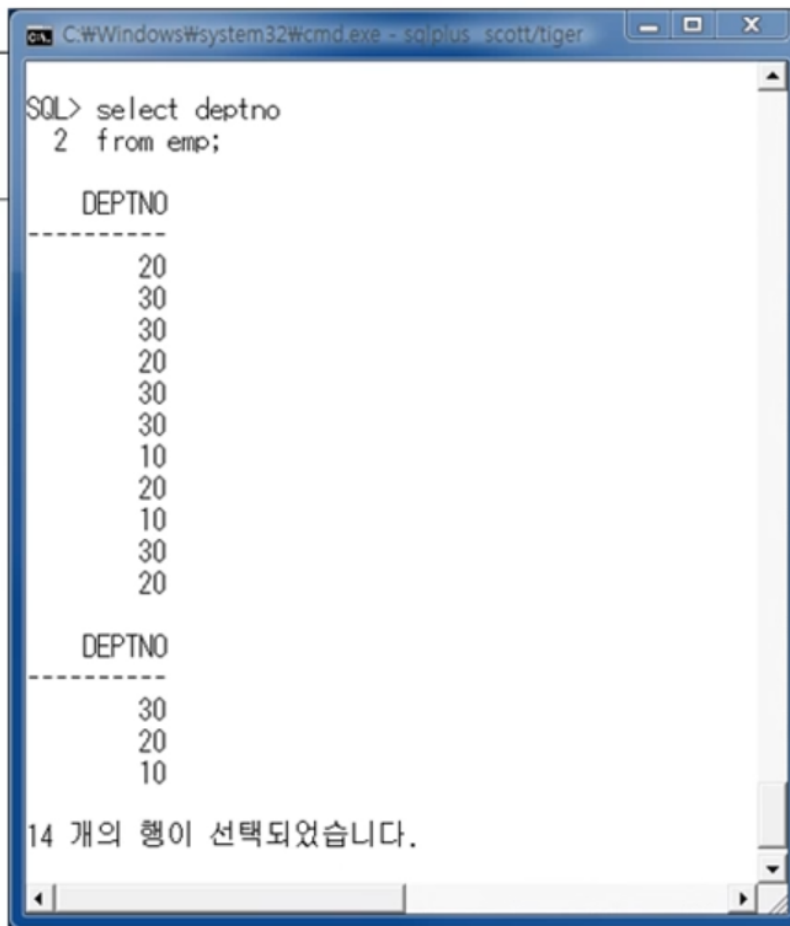
Distinct 키워드

중복 제거

❖ 다음은 사원들이 소속되어 있는 부서 번호를 출력하기 위한 예입니다.

예

```
select deptno  
from emp;
```



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
SQL> select deptno  
2 from emp;  
  
DEPTNO  
-----  
20  
30  
30  
20  
30  
30  
30  
10  
20  
10  
30  
20  
  
DEPTNO  
-----  
30  
20  
10  
  
14 개의 행이 선택되었습니다.
```

1 실습.

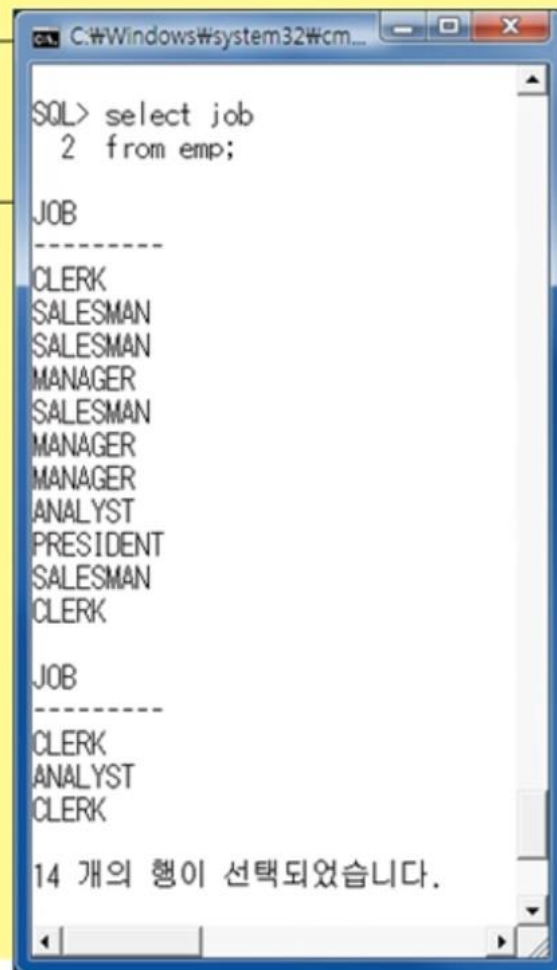
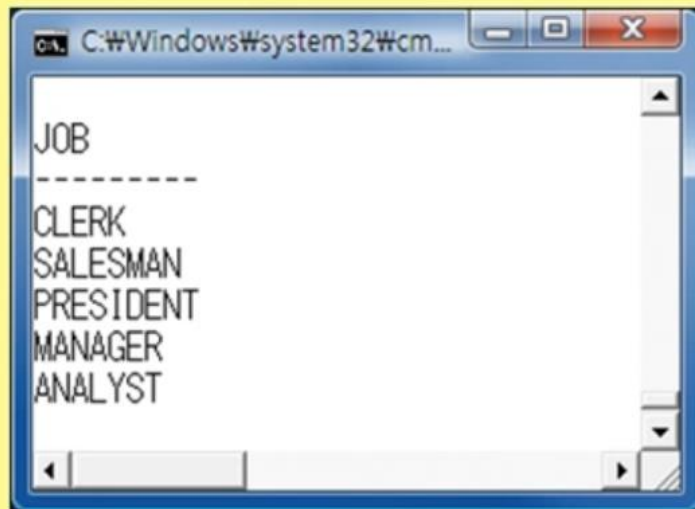
아래와 같이 출력하시오.

4. 사원 테이블에 존재하는 직급의 종류를 출력하기 위해서 다음과 같은 쿼리문을 수행하면 화면 왼쪽 그림과 같이 중복된 내용이 출력됩니다.

쿼리문

```
select job  
from emp;
```

아래 그림과 같이 직급이 중복되지 않고 한번씩 나열되도록 쿼리문을 작성해 보시오.



1 Where 절 쿼리

Where조건과 비교 연산자

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

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> ①컬럼 ②연산자 ③비교대상값
---------	---

1

Where 절 쿼리

연산자 우선순위

우선순위	연산자
1	산술연산자 (*, /, +, -)
2	연결 연산자 ()
3	비교연산자(<, >, <=, >=, <>, =)
4	IS NULL, LIKE, IN
5	BETWEEN
6	NOT 논리 연산자
7	AND 논리 연산자
8	OR 논리 연산자

1 Where 절 쿼리

연산자 우선순위

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;

1

Where 절 쿼리

Select * from emp where deptno=10;

예

```
SELECT *  
FROM EMP  
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

1

실습.

아래와 같이 출력하시오.

1. 급여가 4000 이하인 사원의 사원번호, 사원 이름, 급여를 출력하는 SQL 문을 작성해 보시오.

1 Where 절 쿼리

문자 데이터 조회

- ❖ 이전 예제에서 비교 연산자를 하기 위해서 다루었던 컬럼들은 수치 형태로 선언되었습니다. 이번에는 급여가 아닌 직원 이름 같은 문자 데이터를 조회해 봅시다.
- ❖ 다음은 이름이 FORD인 직원의 직원번호(EMPNO)과 직원이름(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 직원이름이 FORD 인 직원을 찾을 수 없습니다.


1 Where 절 쿼리

날짜 데이터 조회

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

예

```
SELECT *  
FROM EMP  
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.

THANK YOU