

02장. SQL의 기본

이 장에서 다룰 내용

- 1 데이터 디렉터리 TAB
- 2 테이블 구조를 살펴보기 위한 DESC
- 3 오라클의 데이터 형
.....
- 4 데이터를 조회하기 위한 SELECT 문
- 5 특정 데이터만 보기

이 장에서 다룰 내용

6 산술 연산자

7 NULL도 데이터이다.

8 컬럼 이름에 별칭 지정하기

9 Concatenation 연산자의 정의와 사용

10 DISTINCT 키워드

01. 데이터 디렉터리 TAB

- ❖ 오라클을 설치하면 제공되는 사용자인 SCOTT은 학습을 위해서 테이블들이 제공됩니다. SCOTT이 소유하고 있는 테이블을 살펴보기 위해서 다음과 같은 명령을 입력합니다.

```
예 SELECT * FROM TAB;
```

- ❖ TAB은 TABLE의 약자로서 SCOTT 사용자가 소유하고 있는 테이블의 정보를 알려주는 데이터 디렉터리입니다.

02. 테이블 구조를 살펴보기 위한 DESC

- ❖ 테이블에서 데이터를 조회하기 위해서는 테이블의 구조를 알아야 합니다. 테이블의 구조를 확인하기 위한 명령어로는 DESCRIBE가 있습니다.

형식	DESC[RIBE] 테이블명
----	-----------------

- DESC 명령어는 테이블의 컬럼 이름, 데이터 형, 길이와 NULL 허용 유무 등과 같은 특정 테이블의 정보를 알려줍니다.

02. 테이블 구조를 살펴보기 위한 DESC

- ❖ 오라클을 설치하면 학습용으로 제공되는 DEPT 테이블은 부서의 정보를 저장하고 있으며, 이에 대한 구조를 살펴보기 위해서는 desc 명령어를 사용해야 합니다.

예

DESC DEPT ----- DEPT 테이블의 구조 살피기

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

SQL> desc dept
이름                                널?       유형
-----
DEPTNO                             NOT NULL  NUMBER(2)
DNAME                              VARCHAR2(14)
LOC                                VARCHAR2(13)
```

- DESC 명령어는 테이블의 컬럼 이름, 데이터 형, 길이와 NULL 허용 유무 등과 같은 특정 테이블의 정보를 알려줍니다.

02. 테이블 구조를 살펴보기 위한 DESC

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

SQL> desc dept
이름                널?       유형
-----
DEPTNO              NOT NULL  NUMBER(2)
DNAME                VARCHAR2(14)
LOC                  VARCHAR2(13)
```

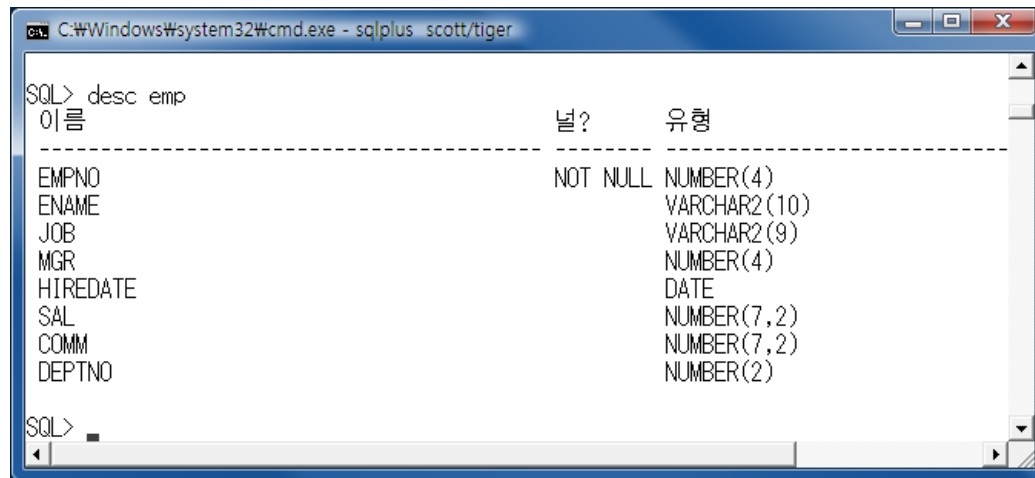
- ❖ DESC 명령어로 DEPT 테이블을 살펴보면 3개의 컬럼으로 구성되어 있음을 살펴볼 수 있습니다.

컬럼 이름	의미
DEPTNO	부서번호
DNAME	부서명
LOC	지역명

02. 테이블 구조를 살펴보기 위한 DESC

예

DESC EMP ----- EMP테이블의 구조 살피기



```
SQL> desc emp
이름              널?       유형
-----
EMPNO            NOT NULL  NUMBER(4)
ENAME            VARCHAR2(10)
JOB              VARCHAR2(9)
MGR              NUMBER(4)
HIREDATE         DATE
SAL              NUMBER(7,2)
COMM             NUMBER(7,2)
DEPTNO           NUMBER(2)

SQL>
```

칼럼 이름	의미	칼럼 이름	의미
EMPNO	사원번호	HIREDATE	입사일
ENAME	사원이름	SAL	급여
JOB	담당 업무	COMM	커미션
MGR	해당 사원의 상사 사원번호	DEPTNO	부서번호

03. 오라클의 데이터 형

❖ NUMBER

- NUMBER 데이터 형은 숫자 데이터를 저장하기 위해서 제공됩니다.

형식	NUMBER(precision, scale)
----	--------------------------

- precision은 소수점을 포함한 전체 자리수를 의미하며 scale은 소수점 이하 자리수를 지정합니다.
- scale을 생략한 채 precision만 지정하면 소수점 이하는 반올림되어 정수 값만 저장됩니다.
- precision과 scale을 모두 생략하면 입력한 데이터 값만큼 공간이 할당됩니다.

03. 오라클의 데이터 형

❖ DATE

- DATE는 세기, 년, 월, 일, 시간, 분, 초의 날짜 및 시간 데이터를 저장하기 위한 데이터 형입니다.
- 이렇듯 날짜 타입 안에는 세기, 년, 월, 일, 시, 분, 초, 요일 등 여러 가지 정보가 들어 있지만 별다른 설정이 없으면 년, 월, 일만 출력합니다.
- 기본 날짜 형식은 "YY/MM/DD" 형식으로 “년/월/일” 로 출력됩니다.
- 2005년 12월 14일은 “05/12/14” 로 출력됩니다.

03. 오라클의 데이터 형

❖ DATE

- DATE는 세기, 년, 월, 일, 시간, 분, 초의 날짜 및 시간 데이터를 저장하기 위한 데이터 형입니다.
- 이렇듯 날짜 타입 안에는 세기, 년, 월, 일, 시, 분, 초, 요일 등 여러 가지 정보가 들어 있지만 별다른 설정이 없으면 년, 월, 일만 출력합니다.
- 기본 날짜 형식은 "YY/MM/DD" 형식으로 “년/월/일” 로 출력됩니다.
- 2005년 12월 14일은 “05/12/14” 로 출력됩니다.

03. 오라클의 데이터 형

❖ CHAR

- 문자 데이터를 저장하기 위한 자료형으로 CHAR가 있습니다. CHAR는 고정 길이 문자 데이터를 저장합니다.
- 입력된 자료의 길이와는 상관없이 정해진 길이만큼 저장 영역 차지하며 최소 크기는 1입니다.
- 주소를 저장하기 위해서 address 란 컬럼을 생성하되 저장될 데이터의 최대 크기를 고려해서 CHAR(20)이라고 주었고, 'seoul' 이란 데이터를 저장하였다고 합시다.

address

s	e	o	u	l															
---	---	---	---	---	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

- CHAR는 주어진 크기만큼 저장공간이 할당되므로 편차가 심한 데이터를 입력할 경우 위의 예와 같이 저장공간의 낭비를 초래합니다.

03. 오라클의 데이터 형

❖ VARCHAR2

- VARCHAR2 데이터 형은 가변적인 길이의 문자열을 저장하기 위해서 제공합니다.
- 이번에는 주소를 저장하기 위해서 address 란 컬럼의 데이터형을 VARCHAR2(20)이라고 설정하고, 'seoul' 이란 데이터를 저장하였다고 합시다.

address

s	e	o	u	l
---	---	---	---	---

- VARCHAR2는 저장되는 데이터에 의해서 저장공간이 할당되므로 메모리 낭비를 줄일 수 있습니다.

탄탄히 다지기

1. emp 테이블을 구성하는 각 컬럼의 데이터 형태에 대해서 설명하시오.

컬럼 이름	자료형 설명
EMPNO	①
ENAME	②
JOB	③
MGR	④
HIREDATE	⑤
SAL	⑥
COMM	⑦
DEPTNO	⑧

```

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

SQL> desc emp
이름                널?       유형
-----
EMPNO                NOT NULL   NUMBER(4)
ENAME                VCHAR2(10)
JOB                  VCHAR2(9)
MGR                  NUMBER(4)
HIREDATE              DATE
SAL                  NUMBER(7,2)
COMM                 NUMBER(7,2)
DEPTNO               NUMBER(2)

SQL>
  
```

04. 데이터를 조회하기 위한 SELECT 문

❖ SELECT 문은 데이터를 조회하기 위한 SQL 명령어입니다.

형식	<pre>SELECT [DISTINCT] {*, column[Alias], . . .} FROM table_name;</pre>
----	---

- SQL 명령어는 하나의 문장으로 구성되어야 하는데 여러 개의 절이 모여서 문장이 되는 것이고 이러한 문장들은 반드시 세미콜론(;)으로 마쳐야 합니다.
- SELECT 문은 반드시 SELECT와 FROM 이라는 2개의 키워드로 구성되어야 합니다.
- SELECT절은 출력하고자 하는 컬럼 이름을 기술합니다.
- 특정 컬럼 이름 대신 * 를 기술할 수 있는데, * 는 테이블 내의 모든 컬럼을 출력하고자 할 경우 사용합니다.
- FROM절 다음에는 조회하고자 하는 테이블 이름을 기술합니다.
- SQL 문에서 사용하는 명령어들은 대문자와 소문자를 구분하지 않는다는 특징이 있습니다.

04. 데이터를 조회하기 위한 SELECT 문

❖ 다음은 부서 테이블의 내용을 살펴보기 위한 쿼리문입니다.

예

```
SELECT *  
FROM dept;
```

- SELECT는 데이터베이스 내에 저장되어있는 테이블을 조회하기 위한 명령어입니다.
- SELECT 다음에는 보고자 하는 대상의 컬럼 이름을 기술합니다. SELECT 다음에 *을 기술하면 지정된 테이블(dept)의 모든 컬럼을 조회합니다.
- FROM 다음에는 보고자 하는 대상의 테이블 이름을 기술합니다. 위 예에서는 dept를 기술하였기에 dept 테이블에 등록된 부서의 정보를 살펴볼 수 있었습니다.

탄탄히 다지기

1. 사원에 관련된 정보를 모두 살펴보기 위한 쿼리문을 작성해 보시오.
<힌트> 사원 테이블의 이름은 emp입니다.

05. 특정 데이터만 보기

- ❖ **사원번호에 해당되는 컬럼 이름은 empno이고 사원명에 해당되는 컬럼 이름은 ename입니다. empno와 ename을 출력하기 위해서는 출력하고자 하는 순서대로 기술하되 컬럼과 컬럼 사이에 콤마를 기술합니다.**

예

```
select empno, ename  
from emp;
```

탄탄히 다지기

2. 사원의 이름과 급여와 입사일자만을 출력하는 SQL 문을 작성해 보시오.
<힌트> 사원 정보가 저장된 테이블의 이름은 EMP이고, 사원이름 칼럼은 ENAME, 급여 칼럼은 SAL, 입사일자 칼럼은 HIREDATE입니다.

06. 산술 연산자

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;
---	--

07. NULL도 데이터이다.

- ❖ 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.

07. NULL도 데이터이다.

- ❖ **널은 ? 혹은 ∞ 의 의미이기 때문에 연산, 할당, 비교가 불가능합니다.**

$$100 + ? = ?$$

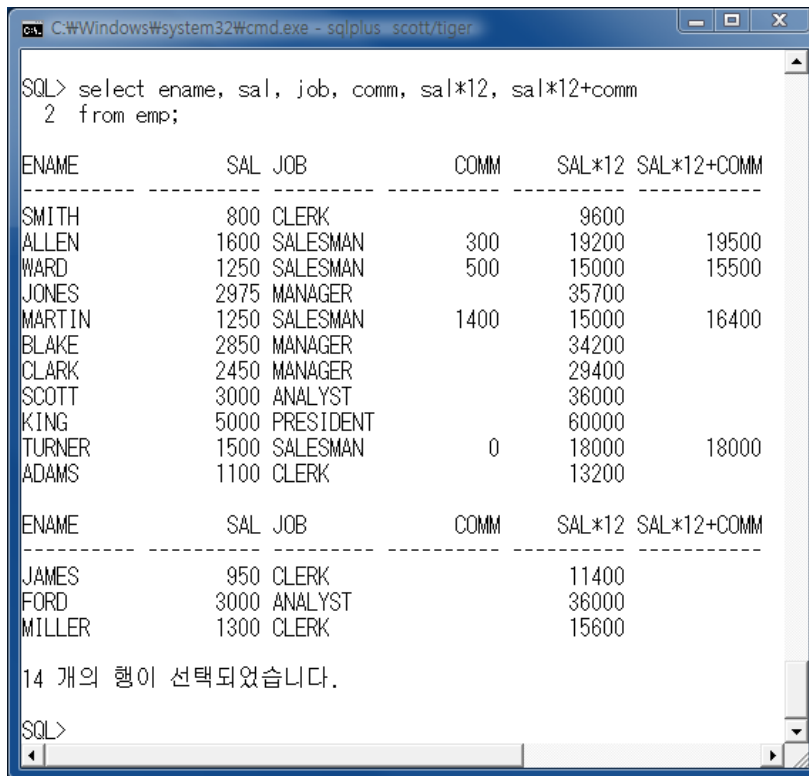
$$100 + \infty = \infty$$

- ❖ **다음은 산술 연산자를 이용해서 연봉을 계산하는 쿼리문으로 앞에서 구했던 예제에서 커미션을 연봉 계산에 추가해 본 것입니다. 커미션의 경우에는 널값을 가진 행도 있으므로 널 값을 가진 데이터와 산술 연산하면 어떤 결과가 나오는지 확인할 수 있는 좋은 예제입니다.**

07. NULL도 데이터이다.

예

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



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

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	
ENAME	SAL	JOB	COMM	SAL*12	SAL*12+COMM
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.

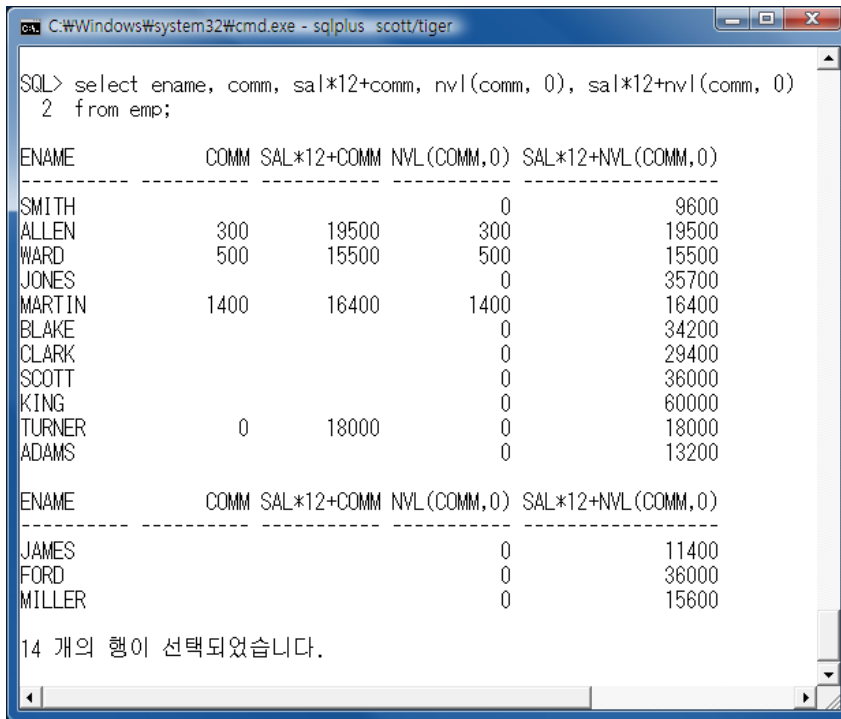
영업직인 경우 무능력하여 받을 커미션(comm)이 없더라도 0으로 저장되어 있으므로 연봉 계산이 제대로 된다.

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

07. NULL도 데이터이다.

예

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



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
ENAME	COMM	SAL*12+COMM	NVL (COMM,0)	SAL*12+NVL (COMM,0)
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 사원들의 커미션은 NULL로 지정되어 있으므로 연봉을 올바르게 계산하기 위해서는 커미션이 NULL인 경우 0으로 변경하여 계산에 참여하도록 해야 합니다.

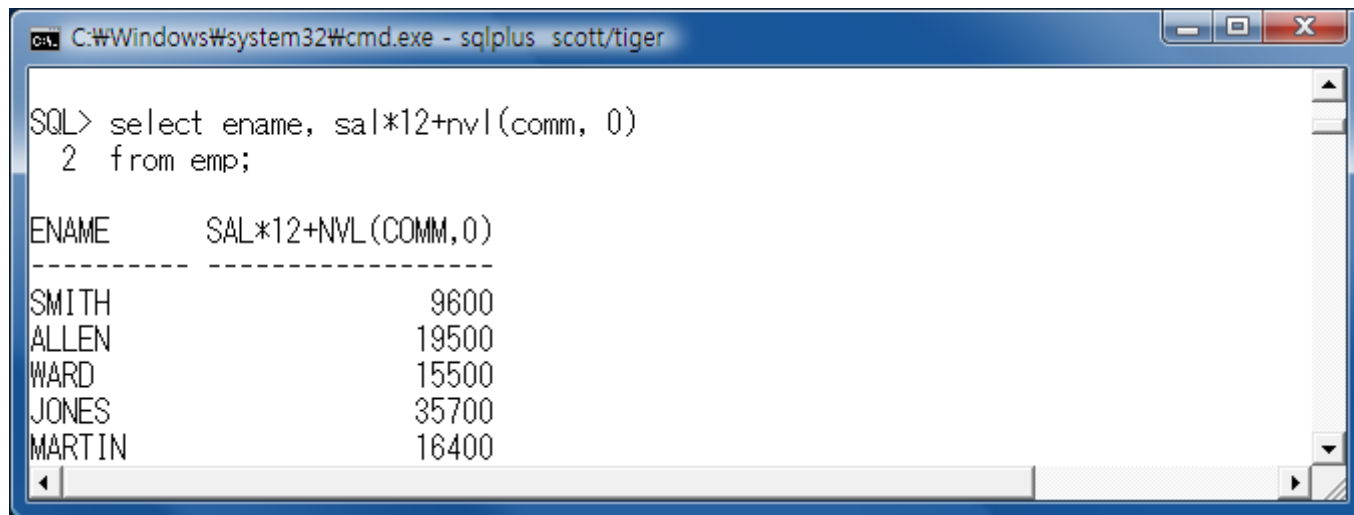
오라클에서는 NULL을 0 또는 다른 값으로 변환하기 위해서 사용하는 함수로 NVL을 제공합니다. 커미션에 널이 저장되어 있더라도 널을 다른 값으로 변환하는 NVL 함수를 사용하면 제대로 된 계산 결과를 얻을 수 있습니다.

08. 컬럼 이름에 별칭 지정하기

- ❖ SQL에서 쿼리문의 결과가 출력될 때, 컬럼 이름이 컬럼에 대한 헤딩(heading)으로 출력됩니다.

예

```
select ename, sal*12+nvl(comm, 0)
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, sal*12+nvl(comm, 0) 2 from emp;". The output is a table with two columns: "ENAME" and "SAL*12+NVL(COMM,0)". The data rows are: SMITH (9600), ALLEN (19500), WARD (15500), JONES (35700), and MARTIN (16400).

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

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

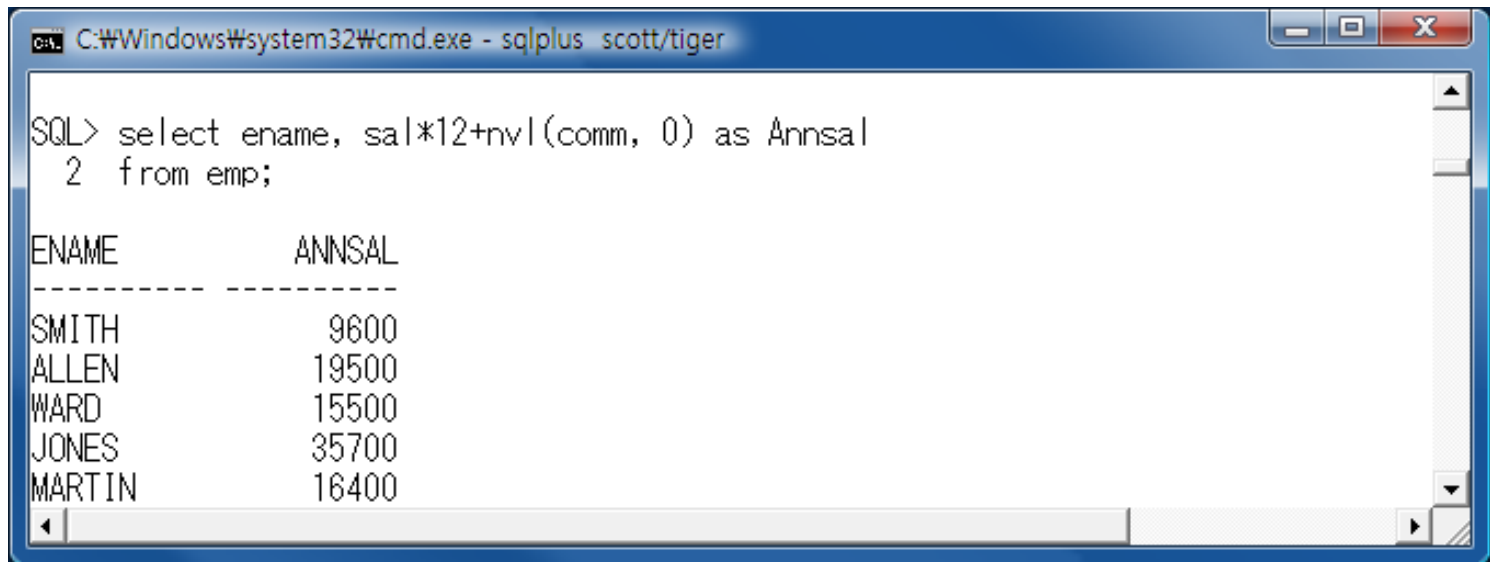
ENAME          SAL*12+NVL(COMM,0)
-----
SMITH              9600
ALLEN             19500
WARD              15500
JONES             35700
MARTIN            16400
```

8.1 AS 로 컬럼에 별칭 부여하기

- ❖ 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 prompt is "SQL>". The user has entered the query: "select ename, sal*12+nvl(comm, 0) as Annsal 2 from emp;". The output is a table with two columns: "ENAME" and "ANNSAL". The data rows are: SMITH (9600), ALLEN (19500), WARD (15500), JONES (35700), and MARTIN (16400).

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

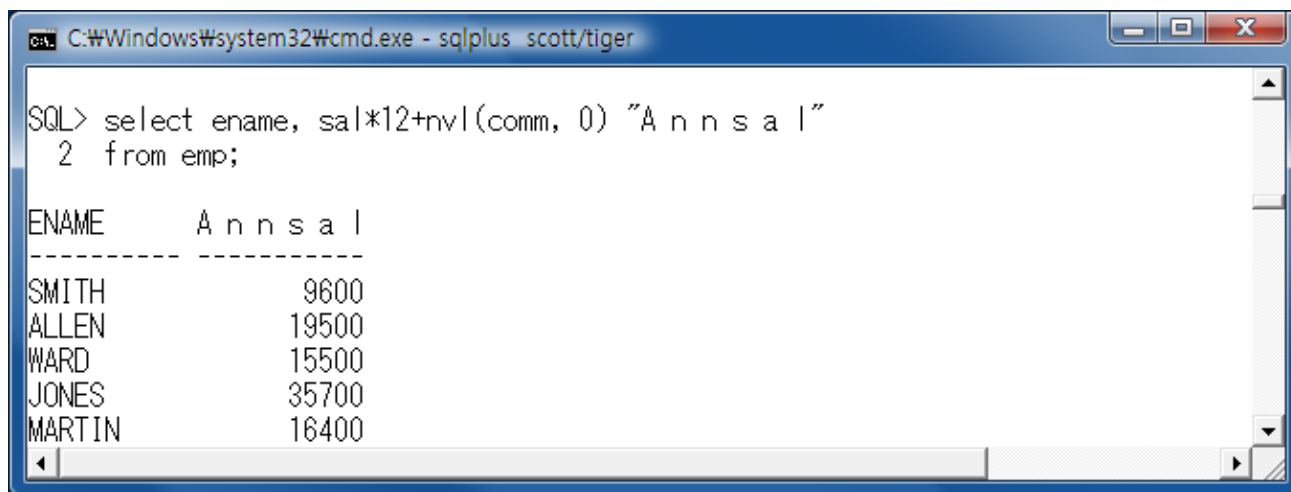
ENAME	ANNSAL
SMITH	9600
ALLEN	19500
WARD	15500
JONES	35700
MARTIN	16400

8.3 “ ” 로 별칭 부여하기

- ❖ 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 following SQL query:

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

The output of the query is displayed as follows:

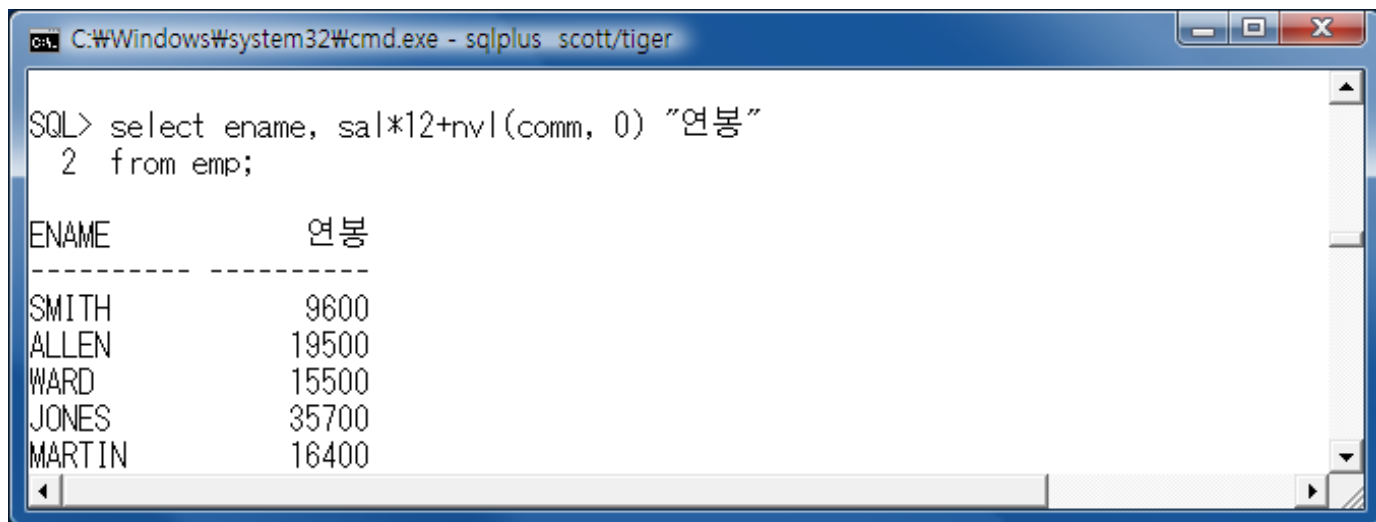
ENAME	A n n s a l
SMITH	9600
ALLEN	19500
WARD	15500
JONES	35700
MARTIN	16400

8.4 별칭으로 한글 사용하기

- ❖ 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 prompt is "SQL>". The user has entered the query: "select ename, sal*12+nvl(comm, 0) '연봉' from emp;". The output is 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

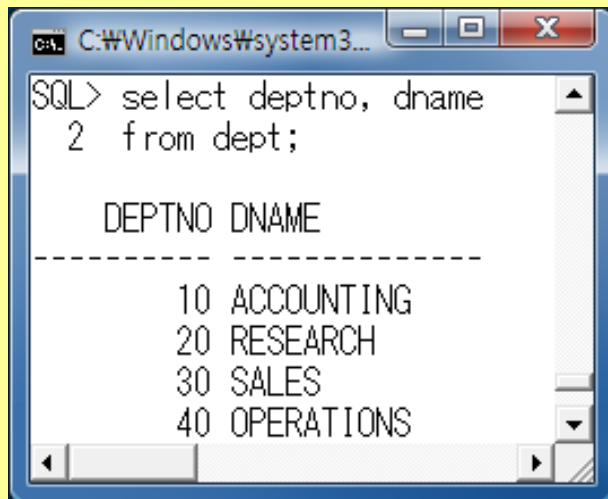
탄탄히 다지기

3. 부서번호는 DEPTNO로, 부서명은 DNAME으로 정해져 있으므로 다음과 같은 쿼리문의 결과는 왼쪽 그림과 같이 컬럼 헤딩에도 역시 DEPTNO, DNAME으로 출력됩니다.

쿼리문

```
select deptno, dname  
from dept;
```

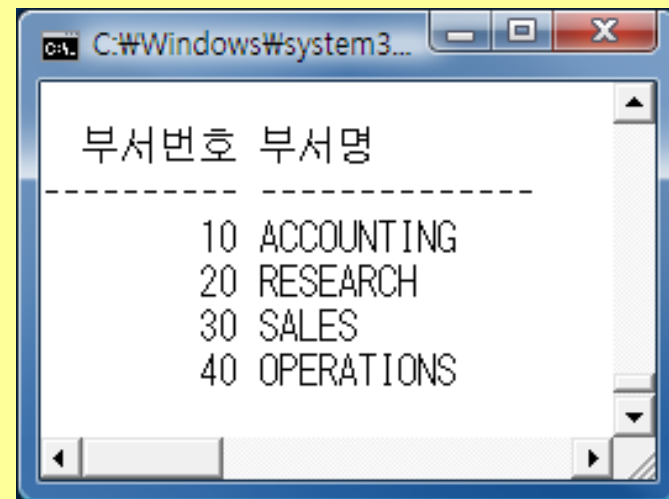
오른쪽 그림과 같이 컬럼 이름 출력되도록 하기 위한 별칭을 지정하는 쿼리문을 작성해 보시오.



The screenshot shows a SQL command window with the following text:

```
SQL> select deptno, dname  
2  from dept;
```

DEPTNO	DNAME
10	ACCOUNTING
20	RESEARCH
30	SALES
40	OPERATIONS



The screenshot shows a SQL command window with the following text:

```
부서번호 부서명
```

10	ACCOUNTING
20	RESEARCH
30	SALES
40	OPERATIONS

09. Concatenation 연산자의 정의와 사용

- ❖ 오라클에서는 Concatenation 연산자를 제공해 줍니다.
- ❖ Concatenation 의 사전적인 의미는 연결입니다.

concatenation

con·cate·na·tion (kɒnkætəˈneɪʃən | kɒn-) n. 《문어》 [U.C] 연쇄;(사건 등의) 연결, 연속, 연관 《of》

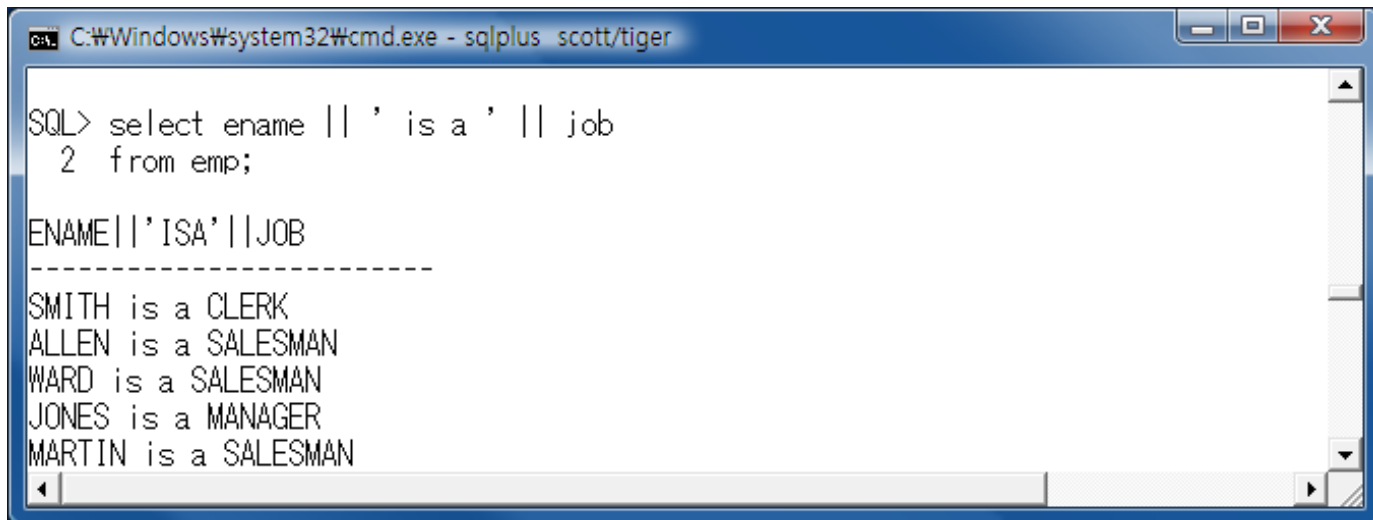
- ❖ 따라서 오라클에서의 Concatenation 연산자 역시 여러 개의 컬럼을 연결할 때 사용하는데 Concatenation 연산자로 “||” 수직바를 사용합니다.

09. 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 user has entered the SQL command: `SQL> select ename || ' is a ' || job
2 from emp;`. The output displays the employee names concatenated with their jobs, separated by " is a ". The output is as follows:

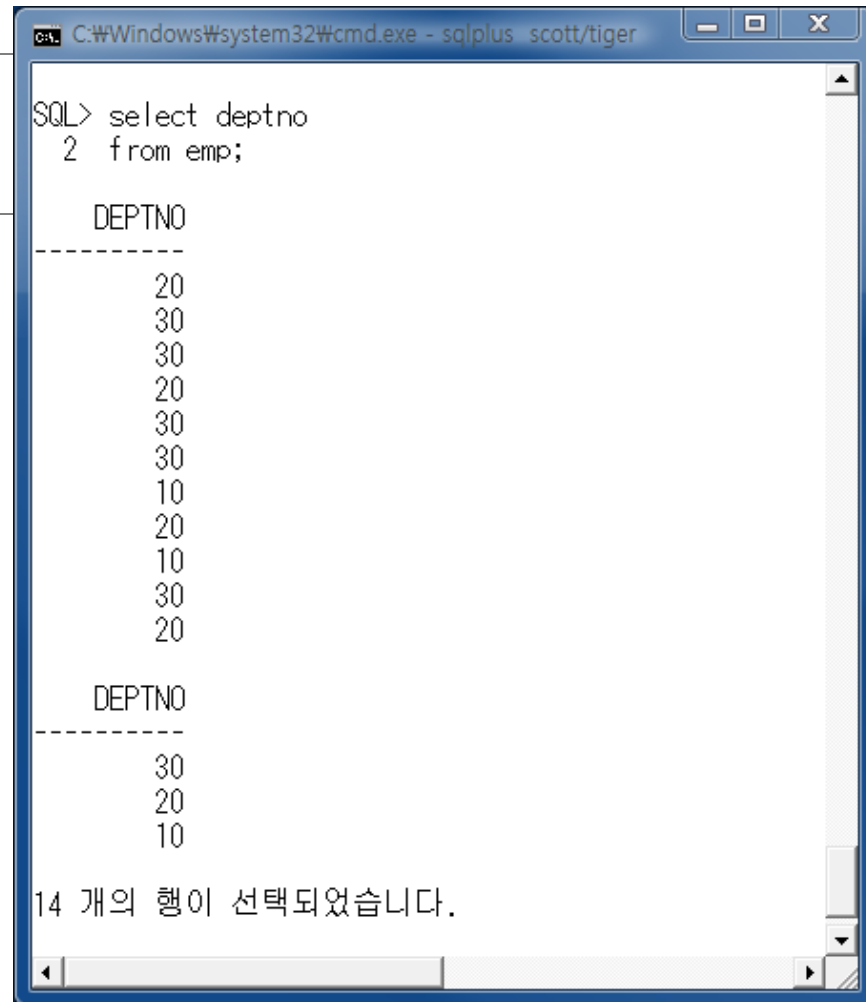
ENAME	JOB
SMITH	CLERK
ALLEN	SALESMAN
WARD	SALESMAN
JONES	MANAGER
MARTIN	SALESMAN

10. DISTINCT 키워드

- ❖ 다음은 사원들이 소속되어 있는 부서 번호를 출력하기 위한 예입니다.

예

```
select deptno  
from emp;
```



The screenshot shows a SQL*Plus window with the title bar "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The command prompt shows the SQL command: `SQL> select deptno
2 from emp;`. The output displays a list of department numbers (DEPTNO) with a dashed line separator. The numbers are: 20, 30, 30, 20, 30, 30, 10, 20, 10, 30, 20. Below this, the text "DEPTNO" is repeated, followed by a dashed line and the numbers 30, 20, 10. At the bottom, a message states "14 개의 행이 선택되었습니다." (14 rows selected).

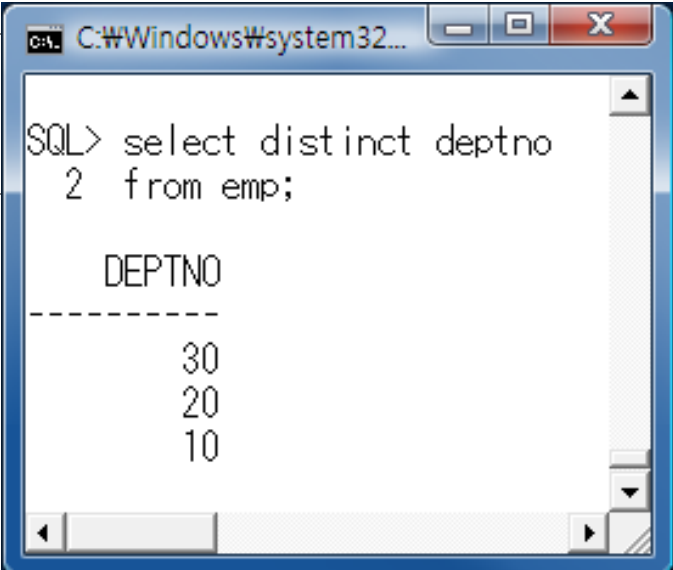
```
SQL> select deptno  
2 from emp;  
  
DEPTNO  
-----  
20  
30  
30  
20  
30  
30  
10  
20  
10  
30  
20  
  
DEPTNO  
-----  
30  
20  
10  
  
14 개의 행이 선택되었습니다.
```


10. DISTINCT 키워드

- ❖ 직원들이 소속되어 있는 부서의 목록을 얻기 위한 목적이라면 같은 부서의 번호가 중복되어 출력되는 것은 의미가 없습니다.
- ❖ 중복되는 부서 번호를 한 번씩만 출력하기 위해서는 키워드 DISTINCT를 사용합니다.

예

```
select distinct deptno  
from emp;
```



The screenshot shows a Windows command prompt window with the title bar 'C:\Windows\system32...'. The prompt is 'SQL>' and the user has entered the command 'select distinct deptno' followed by a new line and '2 from emp;'. The output shows the column header 'DEPTNO' followed by a dashed line and the values '30', '20', and '10' listed vertically.

```
SQL> select distinct deptno  
2  from emp;  
  
DEPTNO  
-----  
30  
20  
10
```

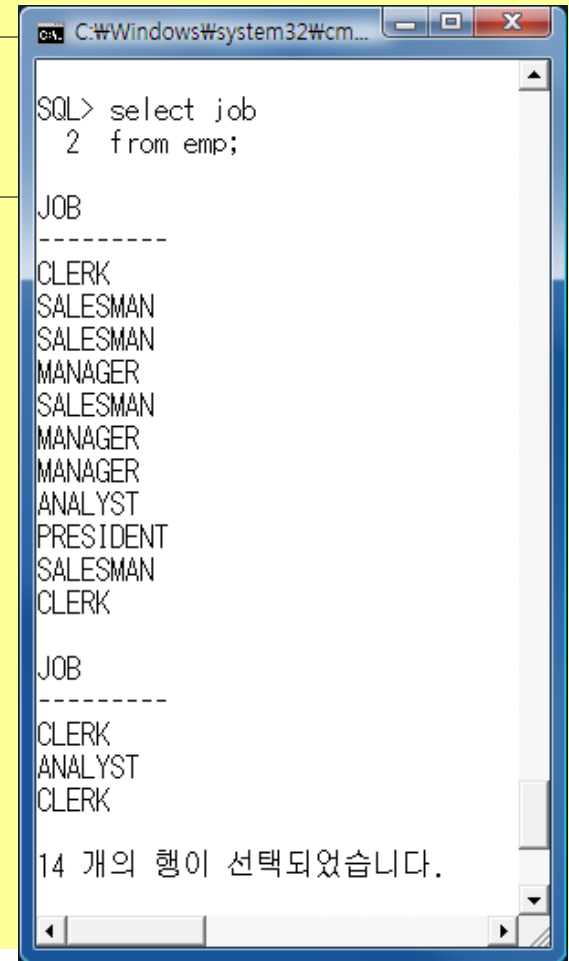
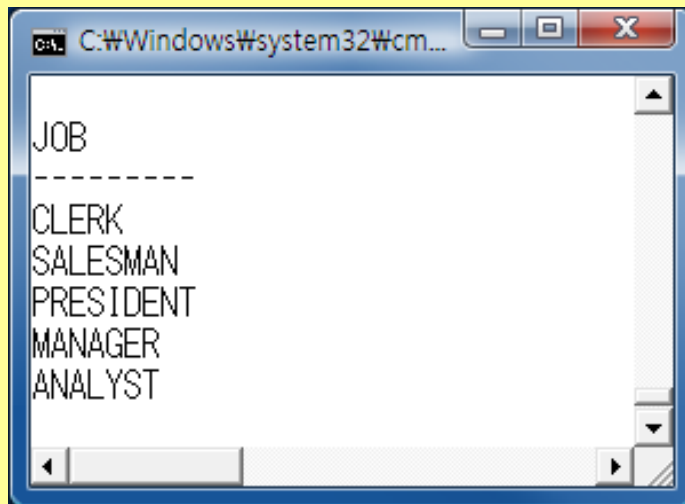
탄탄히 다지기

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.



Thank You !