

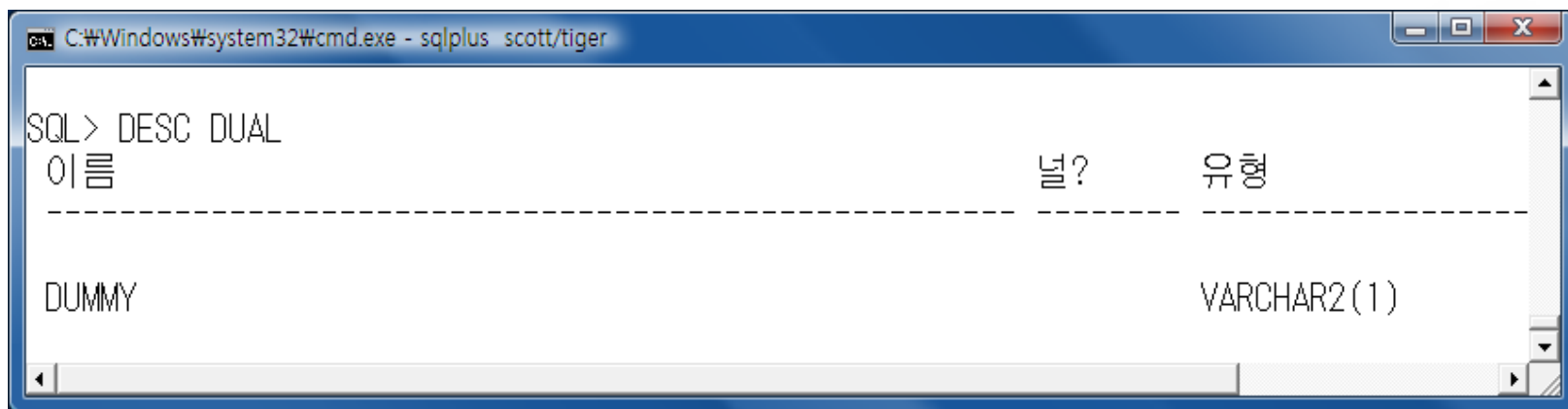
오라클 단일 행 SQL 함수

1. DUAL 테이블

- ❖ DUAL 테이블은 오라클에서 출력을 위해 사용하는 가상의 테이블
- ❖ DUAL 테이블은 DUMMY라는 단 하나의 컬럼으로 구성
- ❖ 이 컬럼의 최대 길이는 1

예

DESC DUAL



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 "DESC DUAL". The output is a table with three columns: "이름" (Name), "널?" (Null?), and "유형" (Type). The output shows a single row for the column "DUMMY" with a null value and a type of "VARCHAR2(1)".

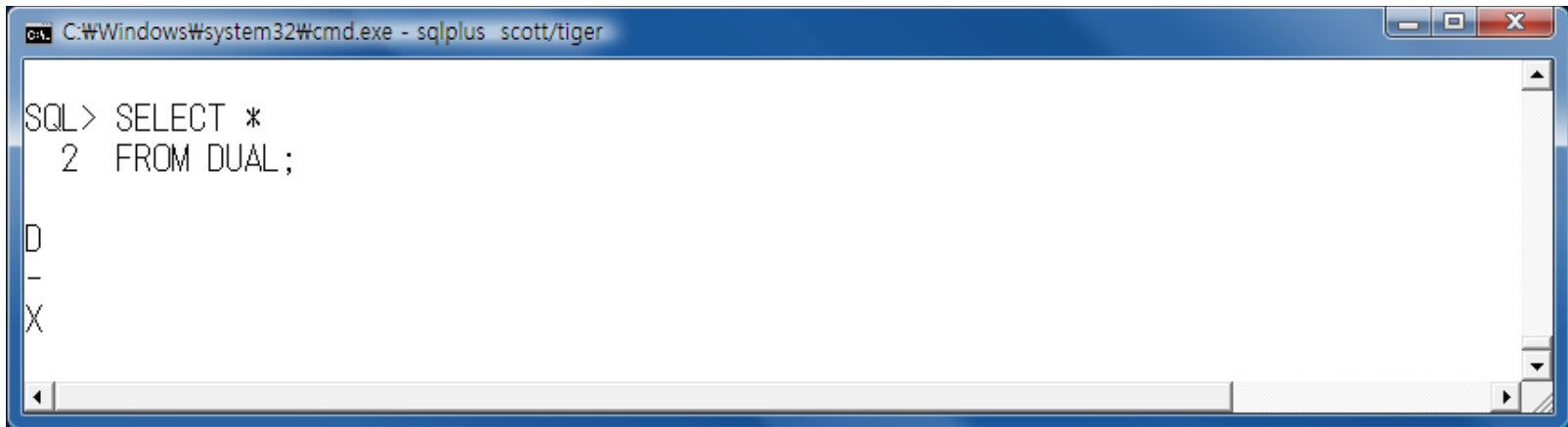
```
SQL> DESC DUAL
이름                                널?       유형
-----
DUMMY                                YES        VARCHAR2(1)
```

1. DUAL 테이블

- ❖ DUAL 테이블은 DUMMY라는 단 하나의 컬럼에 X라는 단 하나의 행을 저장하고 있으나 이 값은 아무런 의미가 없음
- ❖ 쿼리문의 수행 결과가 하나의 행으로 출력되도록 하기 위해서 단 하나의 행을 구성하고 있음

예

```
SELECT *  
FROM DUAL;
```



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 *
2 FROM DUAL;". The output displayed is a single row with the value "X" under the column header "D".

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
SQL> SELECT *  
2 FROM DUAL;  
  
D  
-  
X
```

1. DUAL 테이블

예

```
SELECT 60 * 60 * 24  
FROM DUAL;
```

예

```
SELECT sysdate  
FROM DUAL;
```



2. 숫자 함수

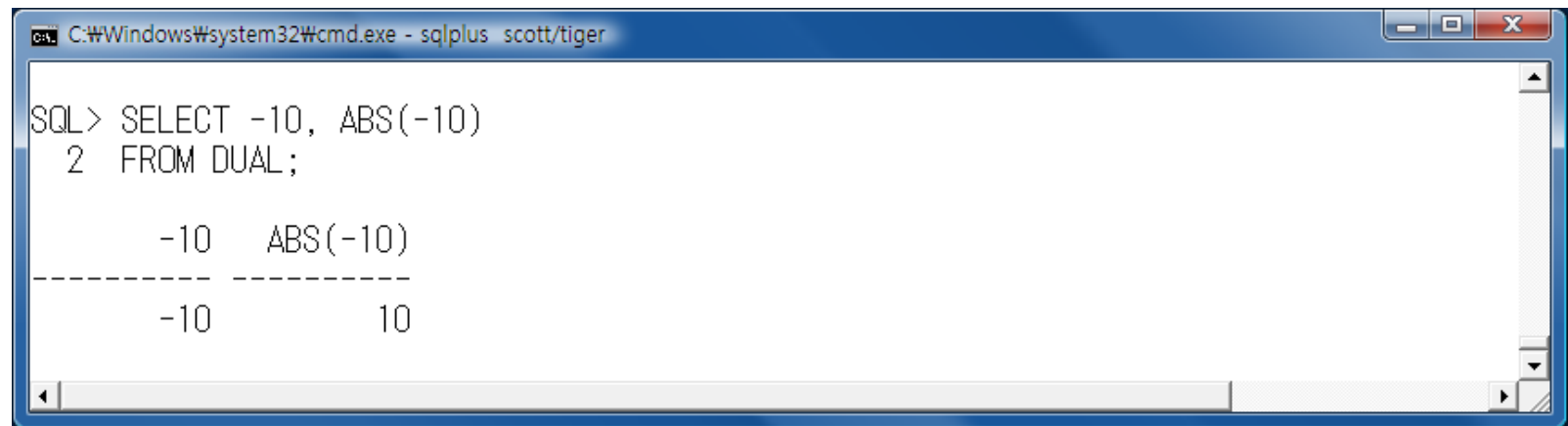
구분	설명
ABS	절대값 리턴
COS	COSINE 값을 반환
EXP	$e(2.71828183\cdots)$ 의 n 승을 반환
FLOOR	소수점 아래를 버림
LOG	LOG값을 반환
POWER	POWER(m, n) m 의 n 승을 반환
SIGN	SIGN (n) $n < 0$ 이면 -1 , $n = 0$ 이면 0 , $n > 0$ 이면 1 을 반환
SIN	SINE값을 반환
TAN	TANGENT값을 반환
ROUND	특정 자릿수에서 반올림
TRUNC	특정 자릿수에서 버림
MOD	입력 받은 수를 나눈 나머지 값을 반환

절대값 구하는 ABS 함수

- ❖ ABS 함수는 절대값을 구합니다. 절대값은 방향은 없고 크기만 있는 것으로서 주어진 데이터가 음수일 경우 양수로 표현
- ❖ -10 에 대한 절대값

예

```
SELECT -10, ABS(-10)
FROM DUAL;
```



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 command "SELECT -10, ABS(-10) FROM DUAL;". The output is displayed as a table with two columns: "-10" and "ABS(-10)". The first row shows the values "-10" and "10".

```
SQL> SELECT -10, ABS(-10)
      2  FROM DUAL;

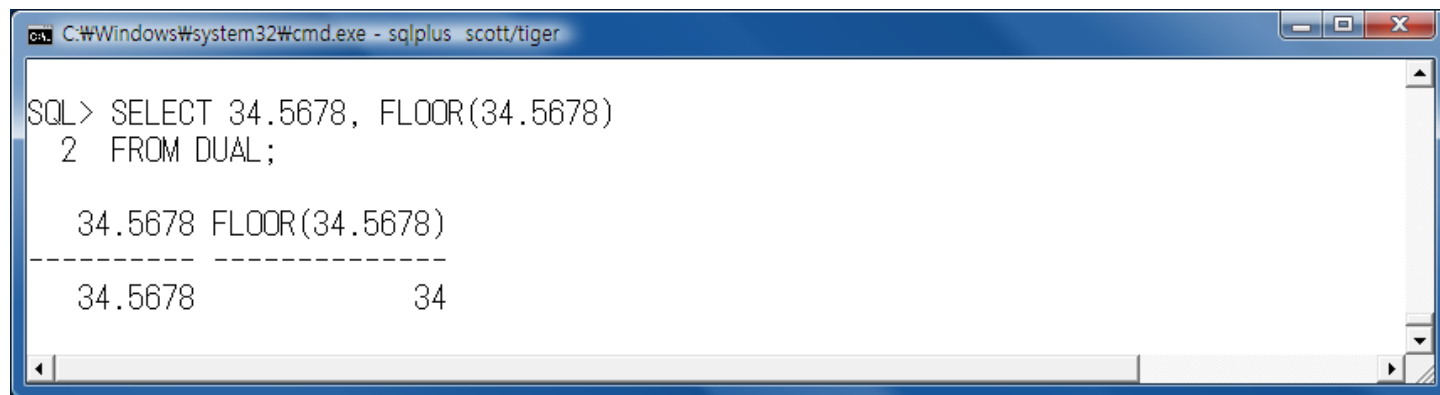
      -10    ABS(-10)
-----
      -10         10
```

소수를 버리는 FLOOR 함수

- ❖ FLOOR 함수는 소수점 아래를 버리는 함수로 34.5678를 FLOOR 함수에 적용하면 34
- ❖ 소수 버리기

예

```
SELECT 34.5678, FLOOR(34.5678)
FROM DUAL;
```



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 command "SELECT 34.5678, FLOOR(34.5678) FROM DUAL;". The output is displayed as follows:

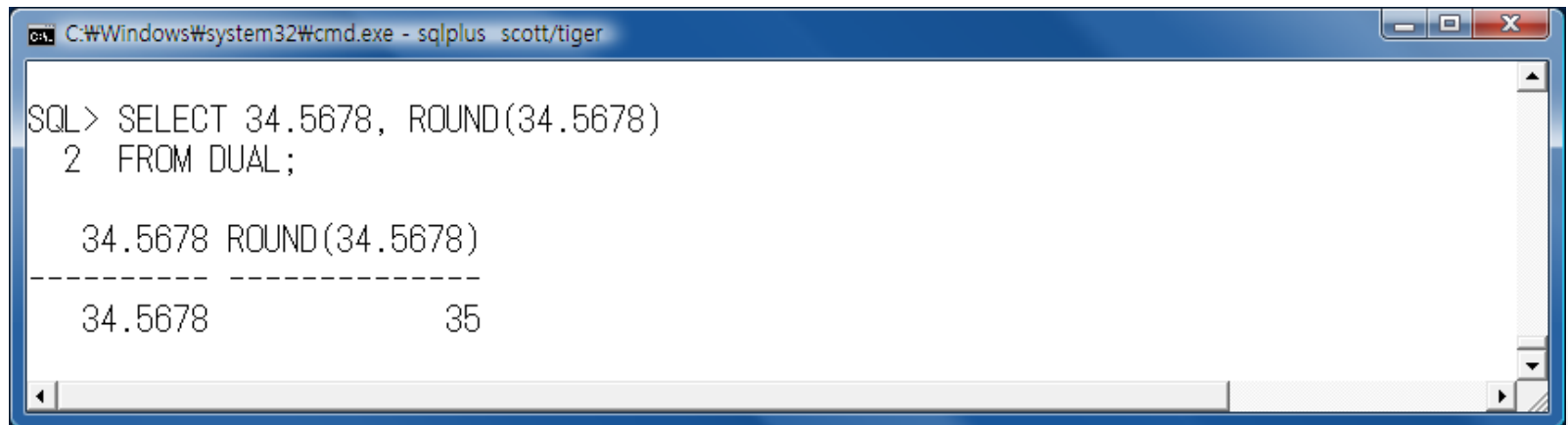
34.5678	FLOOR(34.5678)
34.5678	34

반올림하는 ROUND 함수

- ❖ 반올림한 결과를 구하기 위한 함수로 오라클에서는 ROUND가 제공

예

```
SELECT 34.5678, ROUND(34.5678)
FROM DUAL;
```



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

SQL> SELECT 34.5678, ROUND(34.5678)
      2  FROM DUAL;

      34.5678 ROUND(34.5678)
-----
      34.5678              35
```


반올림하는 ROUND 함수

- ❖ ROUND 함수는 지정한 자릿수에서 반올림을 할 수 있도록 자릿수를 지정할 수 있음

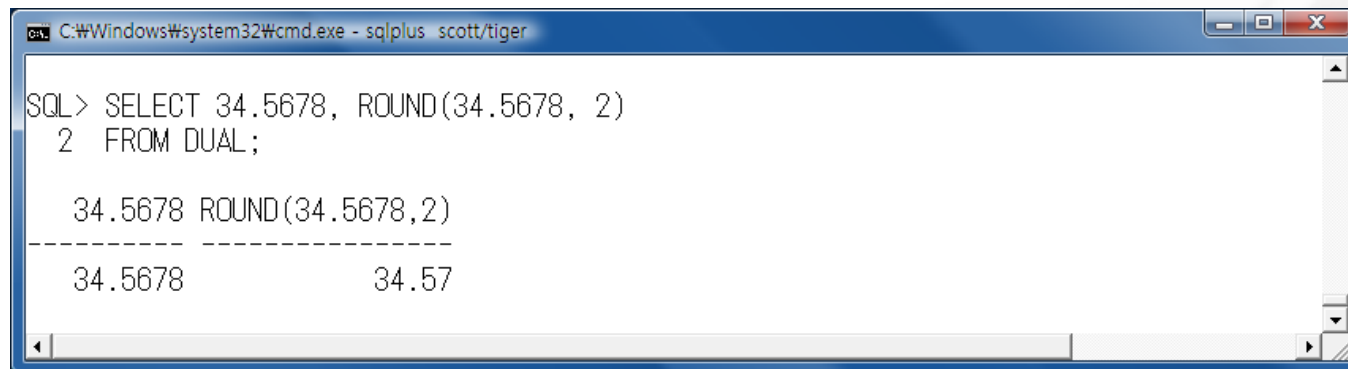
형식

ROUND(대상, 자릿수)

- ❖ ROUND 함수의 두 번째 인자 값이 2이면 소수점 이하 세 번째 자리에서 반올림하여 소수점 이하 두 번째 자리까지 표시
- ❖ ROUND(34.5678, 2)의 결과는 34.57

예

```
SELECT 34.5678, ROUND(34.5678, 2)
FROM DUAL;
```



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 command: "SELECT 34.5678, ROUND(34.5678, 2) FROM DUAL;". The output is displayed in two columns. The first column contains "34.5678" and the second column contains "ROUND(34.5678,2)". A dashed line separates the header from the data row. The data row shows "34.5678" in the first column and "34.57" in the second column.

```
SQL> SELECT 34.5678, ROUND(34.5678, 2)
2 FROM DUAL;

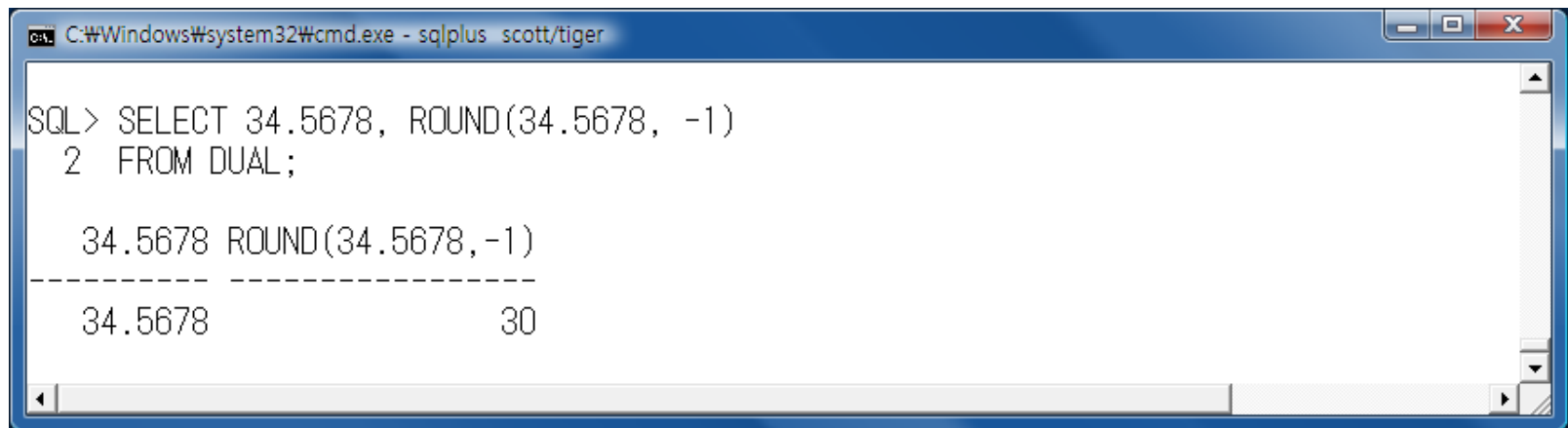
34.5678 ROUND(34.5678,2)
-----
34.5678          34.57
```

반올림하는 ROUND 함수

- ❖ 자릿수에 음수를 지정할 수 있는데 이럴 경우 소수점 이하가 아니라 반대쪽인 일단위, 십단위, 백단위 순으로 거슬러 올라가게 되며, 인자 값이 양수일 때와는 달리 해당 자리에서 반올림이 발생
- ❖ ROUND(34.5678, -1)의 결과는 30

예

```
SELECT 34.5678, ROUND(34.5678, -1)  
FROM DUAL;
```



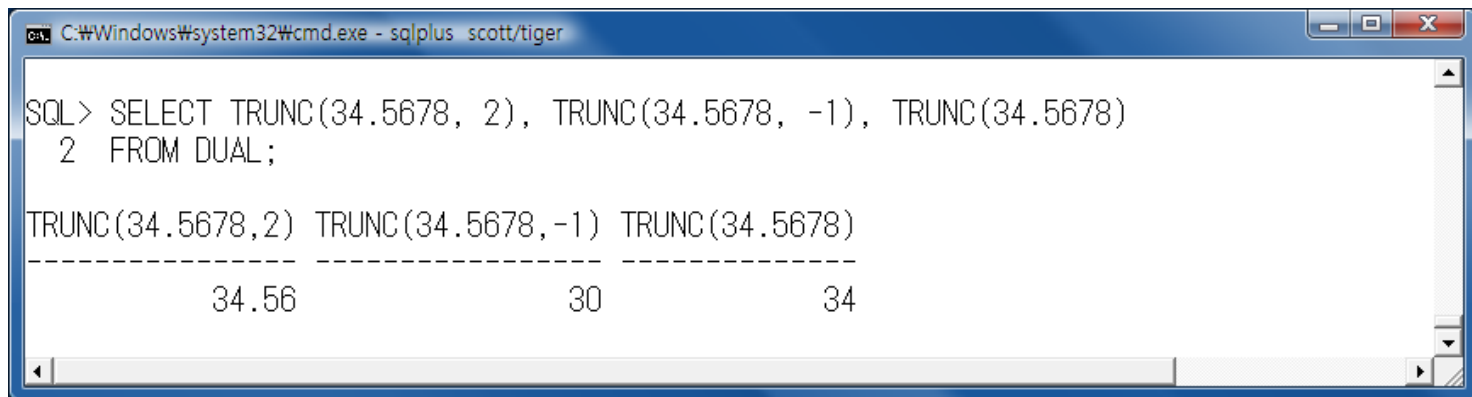
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT 34.5678, ROUND(34.5678, -1)  
2  FROM DUAL;  
  
34.5678 ROUND(34.5678, -1)  
-----  
34.5678                30
```

특정자리에서 잘라내는 TRUNC

- ❖ TRUNC 함수는 지정한 자리 수 이하를 버린 결과를 구해주는 함수

예

```
SELECT TRUNC(34.5678, 2), TRUNC(34.5678, -1), TRUNC(34.5678)  
FROM DUAL;
```



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 TRUNC(34.5678, 2), TRUNC(34.5678, -1), TRUNC(34.5678) FROM DUAL;". The output is displayed in a table format with three columns. The first column is labeled "TRUNC(34.5678,2)", the second is "TRUNC(34.5678,-1)", and the third is "TRUNC(34.5678)". The results are "34.56", "30", and "34" respectively. There is a dashed line above the results.

TRUNC(34.5678,2)	TRUNC(34.5678,-1)	TRUNC(34.5678)
34.56	30	34

- ✓ TRUNC 함수의 두 번째 인자 값이 2이면 소수점 이하 세 번째 자리에서 버림 연산을 하여 소수점 이하 두 번째 자리까지 표시
- ✓ 두 번째 인자 값이 0인 경우에는 소수점자리에서 버림 연산을 하고 -1인 경우는 일의 자리에서 버림 연산을 하며 두 번째 인자가 주어지지 않은 경우 0으로 간주 되어 소수점자리에서 버림 연산을 수행

나머지 구하는 MOD 함수

- ❖ MOD 함수는 나누기 연산을 한 후에 구한 몫이 아닌 나머지를 결과로 되돌려주는 함수

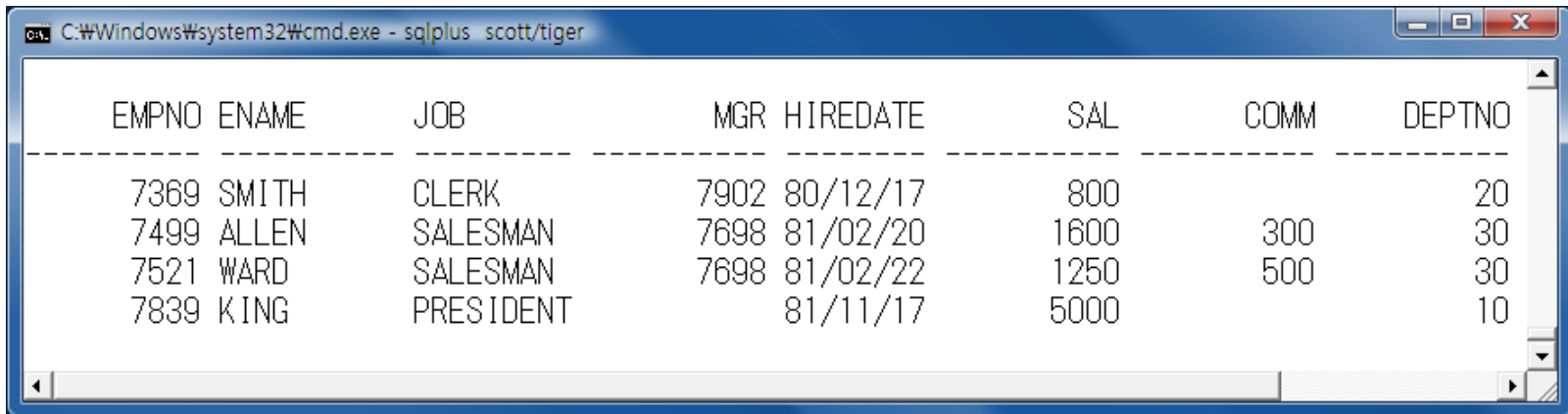
예

```
SELECT MOD (27, 2), MOD (27, 5), MOD (27, 7)  
FROM DUAL;
```

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT MOD (27, 2), MOD (27, 5), MOD (27, 7)  
2  FROM DUAL;  
  
MOD(27,2)  MOD(27,5)  MOD(27,7)  
-----  
1          2          6
```

<연습문제>

- ❖ EMP 테이블에서 EMPNO가 홀수인 데이터의 모든 컬럼을 검색



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
7839	KING	PRESIDENT		81/11/17	5000		10

3. 문자 처리 함수

구 분	설 명
LOWER	소문자로 변환
UPPER	대문자로 변환
INITCAP	첫 글자만 대문자로 나머지 글자는 소문자로 변환
CONCAT	문자를 연결
SUBSTR	문자를 잘라 추출(한글 한 개를 1로 계산)
SUBSTRB	문자를 잘라 추출(한글 한 개를 2Byte 이상으로 계산)
LENGTH	문자의 길이를 반환(한글 한 개를 1로 계산)
LENGTHB	문자의 길이를 반환 (한글 한 개를 2Byte 이상으로 계산)



3. 문자 처리 함수

구 분	설 명
LPAD, RPAD	입력 받은 문자열과 기호를 정렬하여 특정 길이의 문자열로 반환
TRIM(LTRIM, RTRIM)	잘라내고 남은 문자를 표시
CONVERT	CHAR SET을 변환
CHR	ASCII 코드 값으로 변환
ASCII	ASCII 코드 값을 문자로 변환
REPLACE	문자열에서 특정 문자를 변경

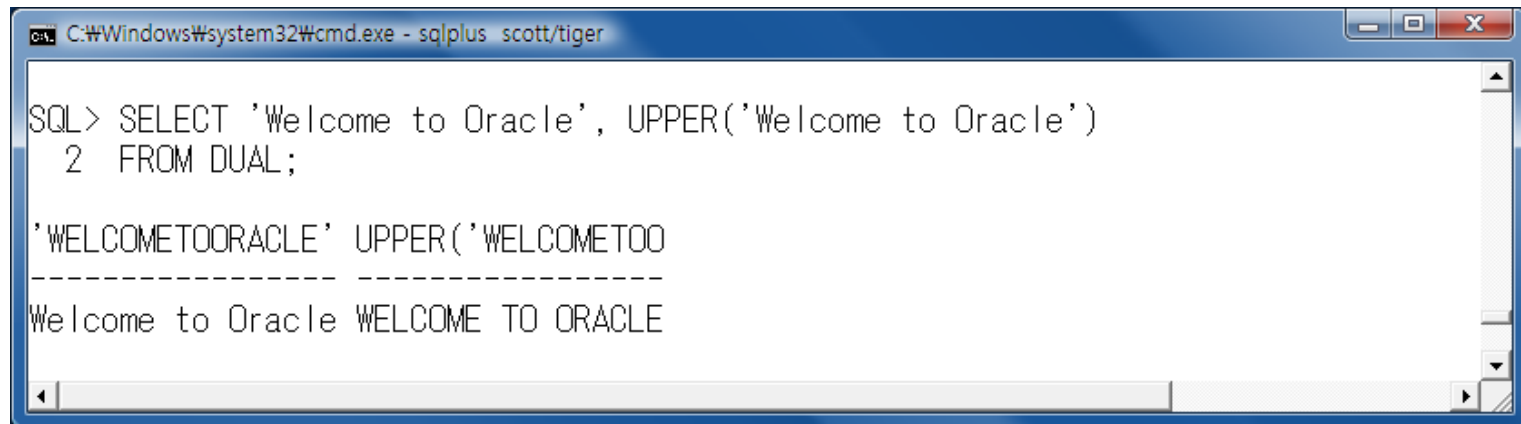


대문자로 변환하는 UPPER

- ❖ UPPER 함수는 입력한 문자값을 대문자로 변환하는 함수

예

```
SELECT 'Welcome to Oracle', UPPER('Welcome to Oracle')  
FROM DUAL;
```



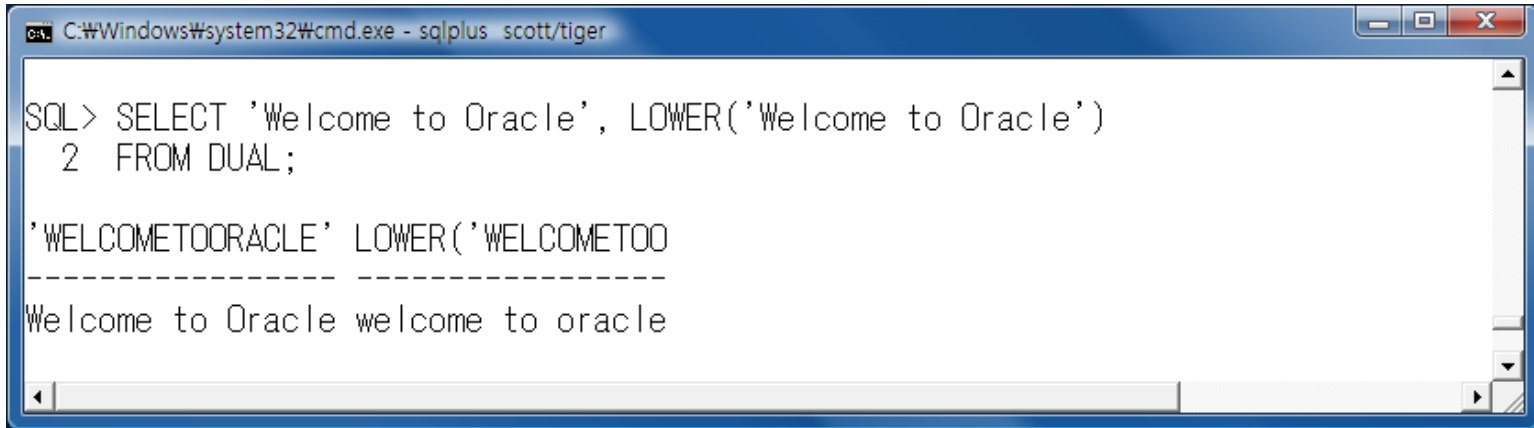
The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL>". The user has entered the SQL query: `SELECT 'Welcome to Oracle', UPPER('Welcome to Oracle')` followed by a new line and `2 FROM DUAL;`. The output of the query is displayed below the prompt: `'WELCOMETOORACLE' UPPER('WELCOMETOO` on the first line, a dashed separator line `-----` on the second line, and `Welcome to Oracle WELCOME TO ORACLE` on the third line. The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

소문자로 변환하는 LOWER

- ❖ LOWER 함수는 문자열을 모두 소문자로 변경

예

```
SELECT 'Welcome to Oracle', LOWER('Welcome to Oracle')  
FROM DUAL;
```



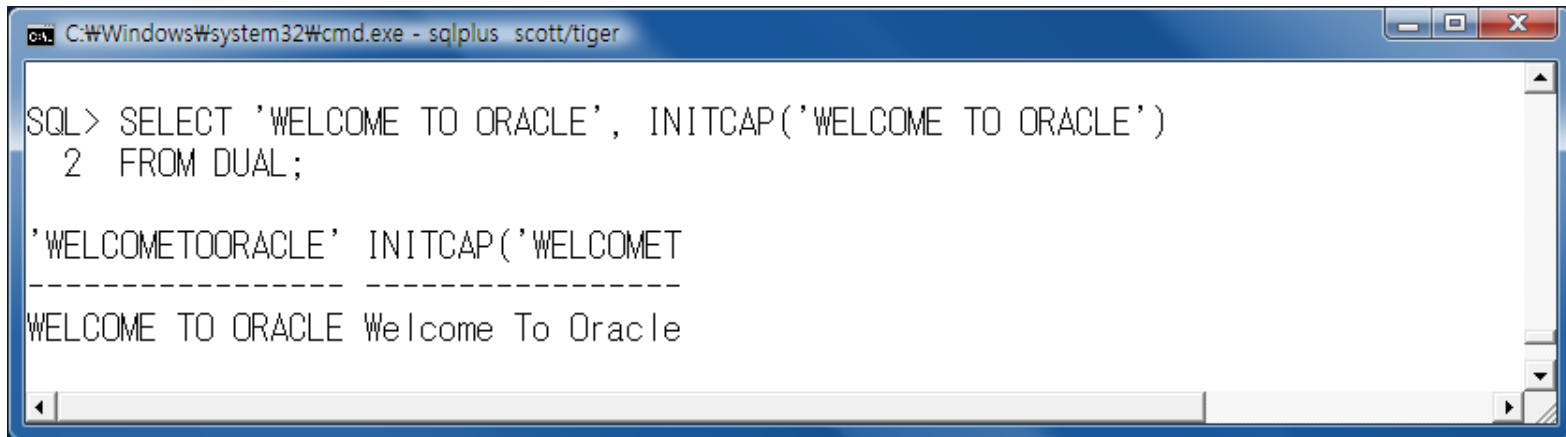
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT 'Welcome to Oracle', LOWER('Welcome to Oracle')  
2 FROM DUAL;  
  
'WELCOMETOORACLE' LOWER('WELCOMETOORACLE')  
-----  
Welcome to Oracle welcome to oracle
```

첫글자만 변환하는 INITCAP

- ❖ INITCAP 함수는 문자열의 이니셜만 대문자로 변경

예

```
SELECT 'WELCOME TO ORACLE',  
       INITCAP('WELCOME TO ORACLE')  
FROM DUAL;
```



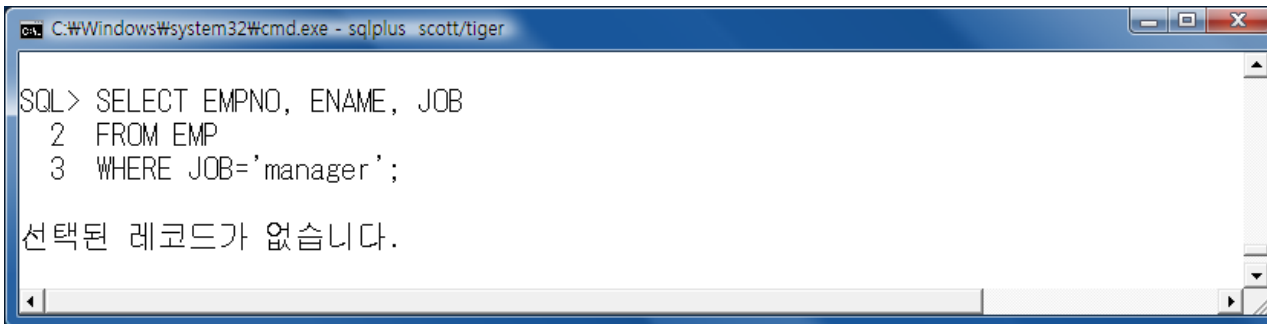
The screenshot shows a Windows command prompt window with the title bar "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The command prompt displays the following SQL query and its output:

```
SQL> SELECT 'WELCOME TO ORACLE', INITCAP('WELCOME TO ORACLE')  
2 FROM DUAL;  
  
'WELCOMETOORACLE' INITCAP('WELCOMET  
-----  
WELCOME TO ORACLE Welcome To Oracle
```

대소문자 변환

예

```
SELECT EMPNO, ENAME, JOB  
FROM EMP  
WHERE JOB='manager';
```



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

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

The output of the query is displayed as:

```
선택된 레코드가 없습니다.
```

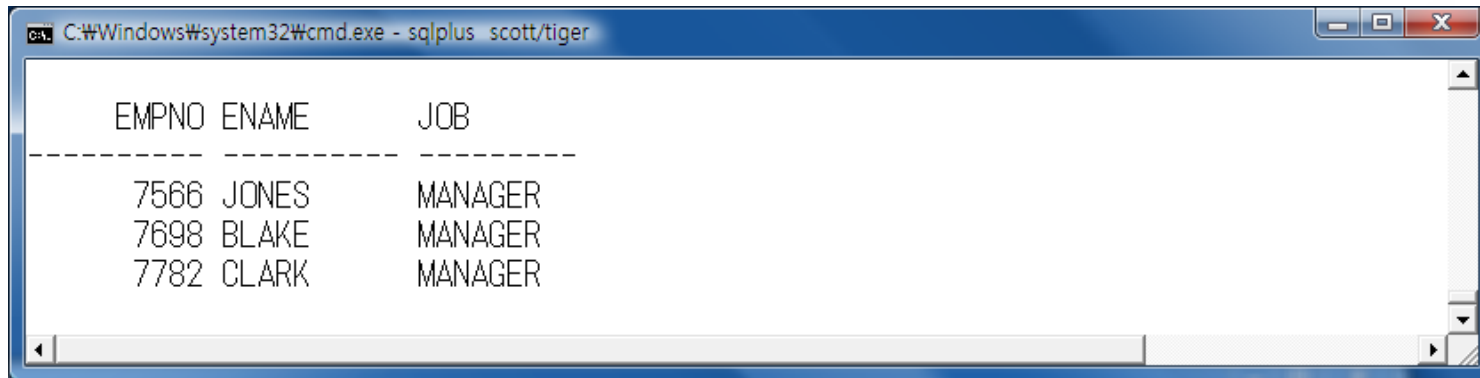
The background of the slide features a faint image of a fountain pen resting on a document.

연습문제

- ❖ 소문자('manager')로 데이터를 검색해야 한다면 LOWER 함수를 사용해서 원하는 결과를 얻을 수 있음
- ❖ JOB 컬럼에 LOWER 함수를 적용하여 JOB 컬럼에 저장된 값을 소문자로 변경한 후에 검색에 참여하도록 해야 함

예

```
SELECT EMPNO, ENAME, JOB  
FROM EMP  
WHERE ①_____='manager';
```



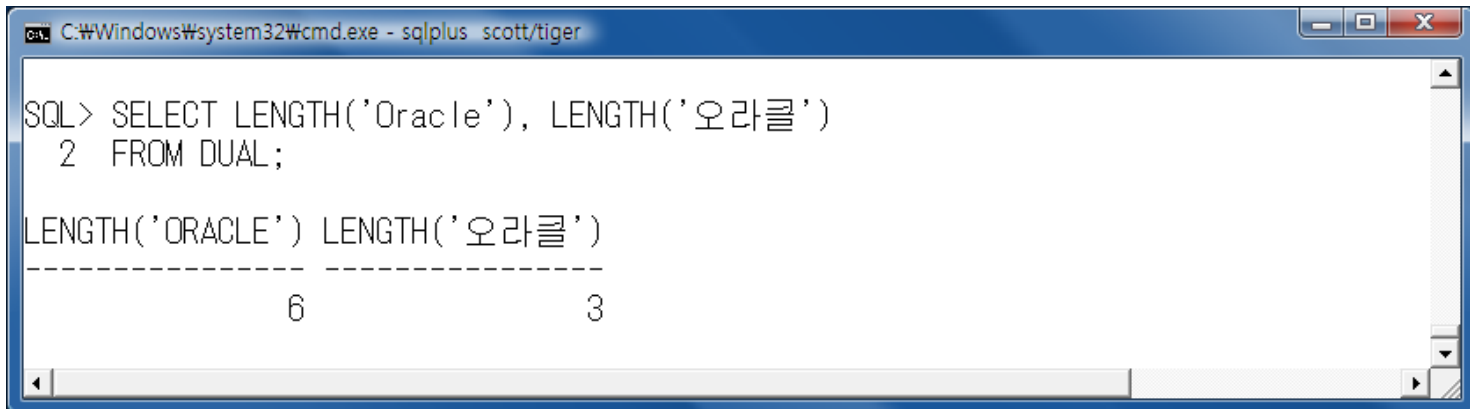
EMPNO	ENAME	JOB
7566	JONES	MANAGER
7698	BLAKE	MANAGER
7782	CLARK	MANAGER

문자 길이를 구하는 LENGTH

- ❖ LENGTH 함수는 컬럼에 저장된 데이터 값이 몇 개의 문자로 구성되었는지 길이를 알려주는 함수
- ❖ 한글도 1글자로 처리
- ❖ 영문자와 한글의 길이

예

```
SELECT LENGTH('Oracle'), LENGTH('오라클')  
FROM DUAL;
```



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The window shows the execution of an SQL query. The prompt "SQL>" is followed by the query "SELECT LENGTH('Oracle'), LENGTH('오라클') FROM DUAL;". The output shows two columns: "LENGTH('ORACLE')" and "LENGTH('오라클')". The first column has a value of 6, and the second column has a value of 3. The output is formatted with a dashed line above the values.

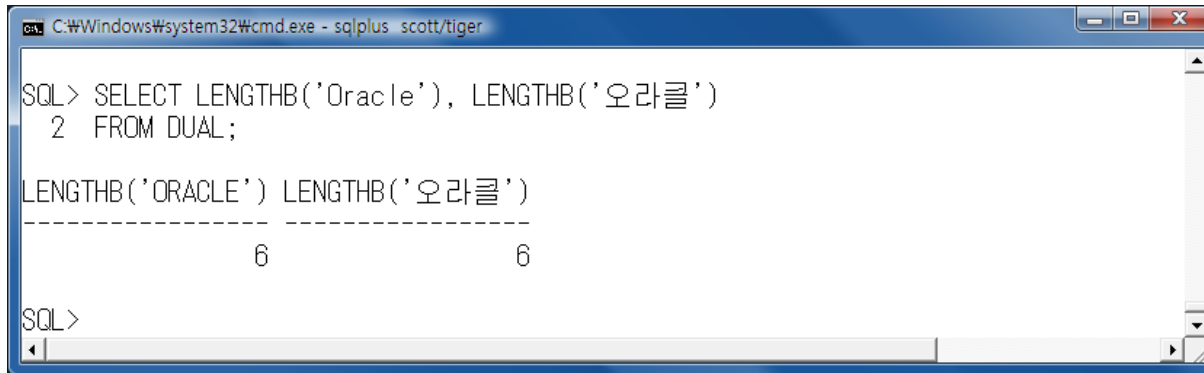
```
SQL> SELECT LENGTH('Oracle'), LENGTH('오라클')  
2  FROM DUAL;  
  
LENGTH('ORACLE') LENGTH('오라클')  
-----  
6                3
```

바이트 수를 구하는 LENGTHB

- ❖ LENGTHB 함수는 바이트 수를 알려주는 함수
- ❖ 인코딩 방식에 따라 다른 결과를 리턴

예

```
SELECT LENGTHB('Oracle'), LENGTHB('오라클')  
FROM DUAL;
```



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

```
SQL> SELECT LENGTHB('Oracle'), LENGTHB('오라클')  
2 FROM DUAL;
```

The output of the query is displayed as follows:

LENGTHB('ORACLE')	LENGTHB('오라클')
6	6

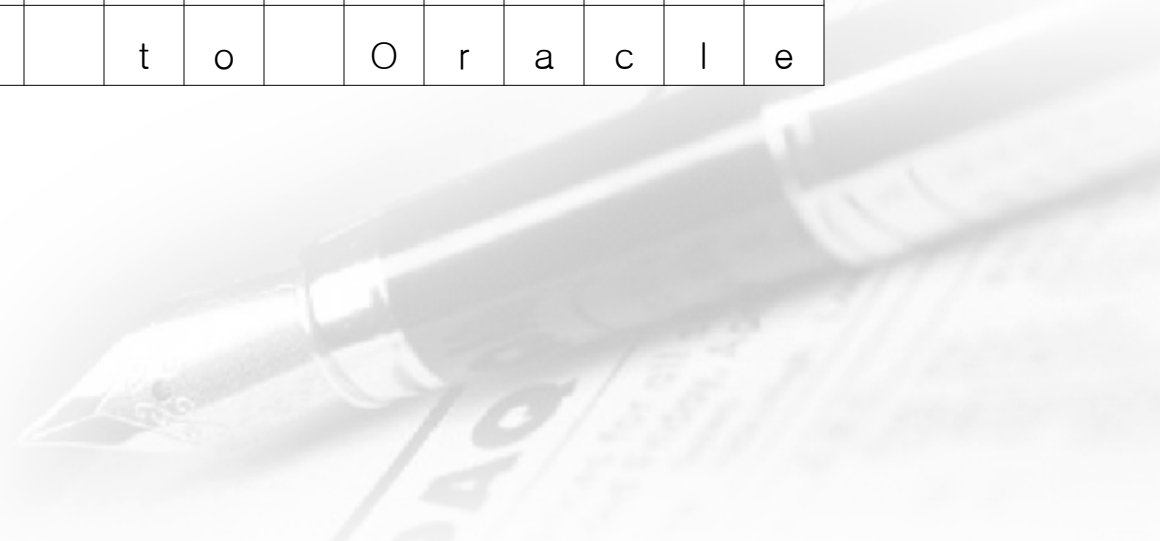
The window also shows the prompt "SQL>" at the bottom, indicating the end of the command.

문자열 일부만 추출-SUBSTR

- ❖ SUBSTR 과 SUBSTRB 함수는 대상 문자열이나 칼럼의 자료에서 시작위치부터 선택 개수만큼의 문자를 추출
- ❖ SUBSTRB 함수도 같은 형식이지만 명시된 개수만큼의 문자가 아닌 바이트 수를 잘라낸다는 점에서만 차이가 있음

형식	SUBSTR(대상, 시작위치, 추출할 개수)
예	SELECT SUBSTR('Welcome to Oracle', 4, 3) FROM DUAL;

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
W	e	l	c	o	m	e		t	o		O	r	a	c	l	e

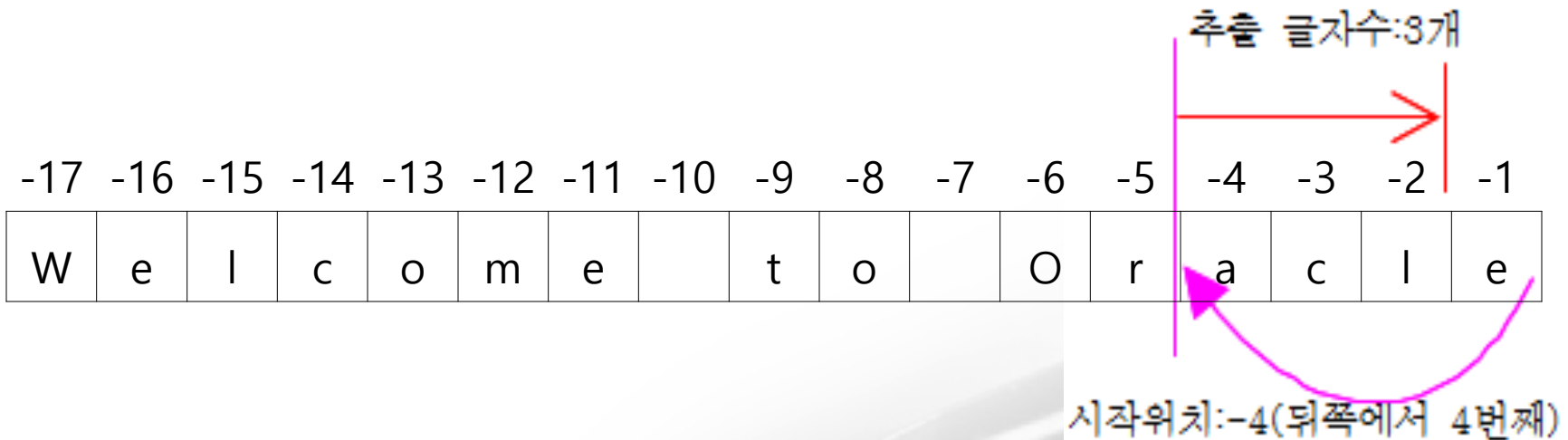


문자열 일부만 추출-SUBSTR

- ❖ 시작 위치 인자 값을 음수 값으로 줄 수 있는데 이때는 문자열의 앞쪽이 아닌 뒤 쪽에서부터 세어서 시작 위치를 설정

예

```
SELECT SUBSTR('Welcome to Oracle', -4, 3)  
FROM DUAL;
```

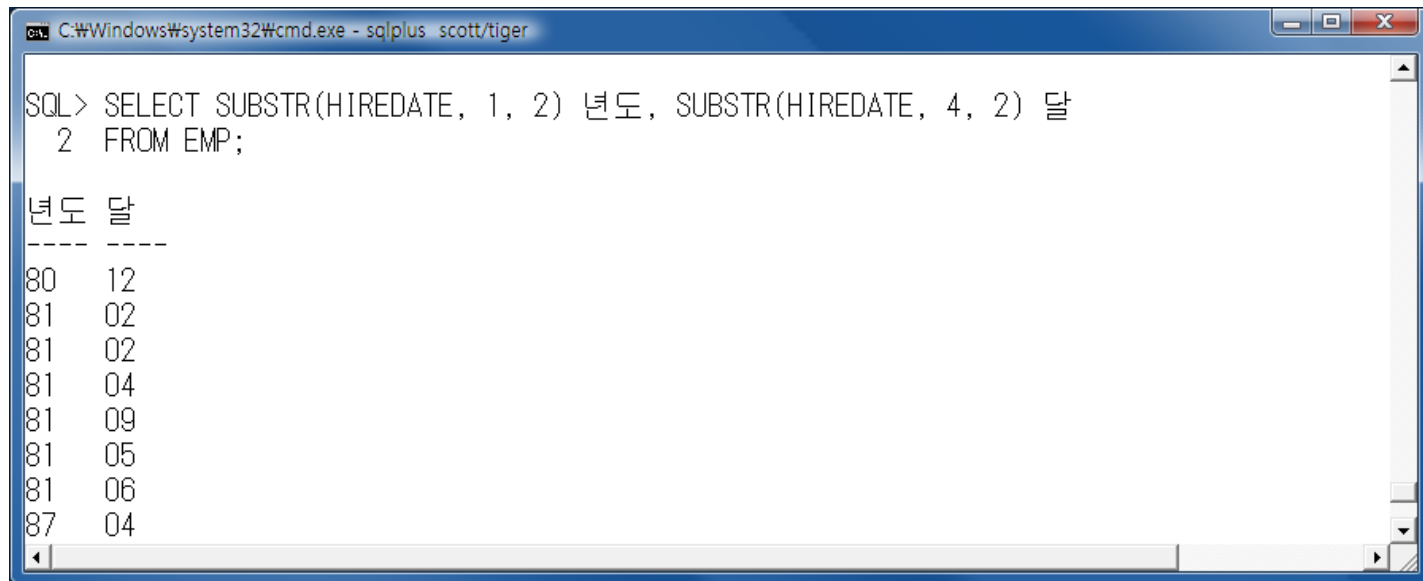


문자열 일부만 추출-SUBSTR

- ❖ 직원들의 입사 년도만 출력하려면 SUBSTR 함수를 이용해서 입사일을 저장하고 있는 HIREDATE 컬럼에서 첫 글자부터 2개를 추출
- ❖ 입사한 달만 출력하려면 HIREDATE 컬럼에서 네번째 글자부터 2개를 추출

예

```
SELECT SUBSTR(HIREDATE, 1, 2) 년도, SUBSTR(HIREDATE, 4, 2) 달  
FROM EMP;
```



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

```
SQL> SELECT SUBSTR(HIREDATE, 1, 2) 년도, SUBSTR(HIREDATE, 4, 2) 달  
2 FROM EMP;
```

The output of the query is displayed as follows:

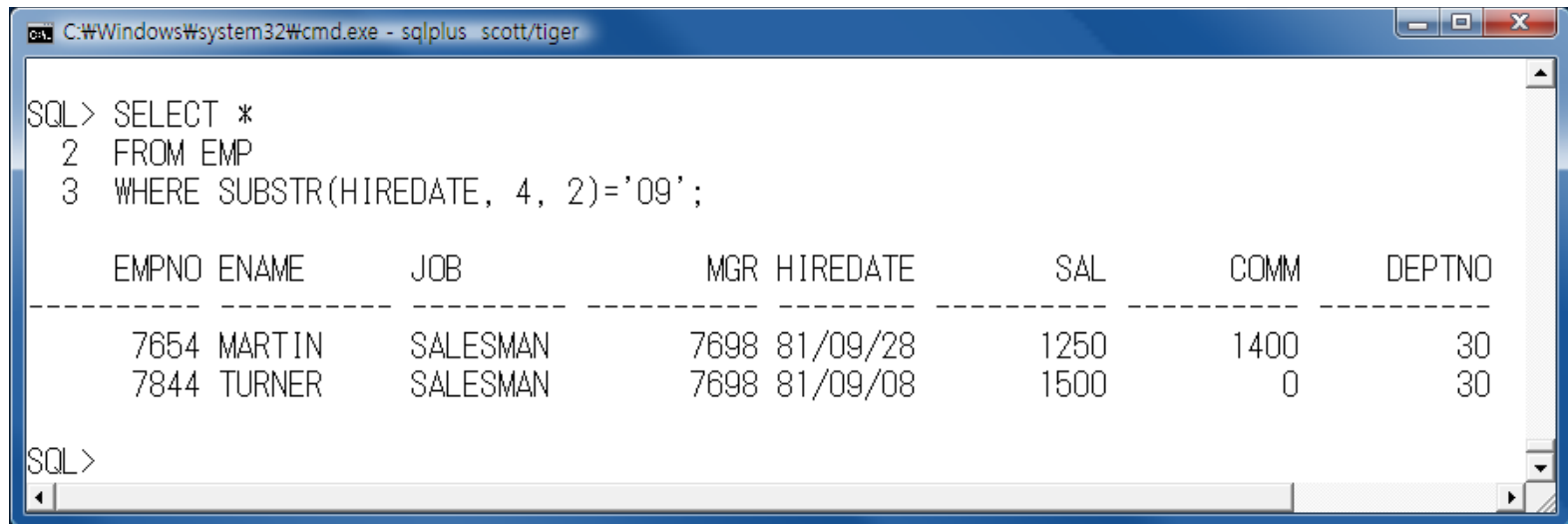
년도	달
80	12
81	02
81	02
81	04
81	09
81	05
81	06
87	04

문자열 일부만 추출-SUBSTR

- ❖ 12월에 입사한 사원을 조회

예

```
SELECT *  
FROM EMP  
WHERE SUBSTR(HIREDATE, 4, 2)='12';
```



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

```
SQL> SELECT *  
2 FROM EMP  
3 WHERE SUBSTR(HIREDATE, 4, 2)='09';
```

The query returns two rows of data. The columns are EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, and DEPTNO. The results are as follows:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7654	MARTIN	SALESMAN	7698	81/09/28	1250	1400	30
7844	TURNER	SALESMAN	7698	81/09/08	1500	0	30

The command prompt shows the prompt "SQL>" at the bottom, indicating the query has been executed.

연습문제

- ❖ EMP 테이블에서 Hiredate의 년도가 1981인 사원의 모든 정보를 조회
 - ✓ (단 substr 함수를 이용해서 WHERE절을 작성)
- ❖ EMP 테이블에서 ename이 E로 끝나는 사원의 모든 정보를 조회
 - ✓ (단 substr 함수를 이용해서 WHERE절을 작성)

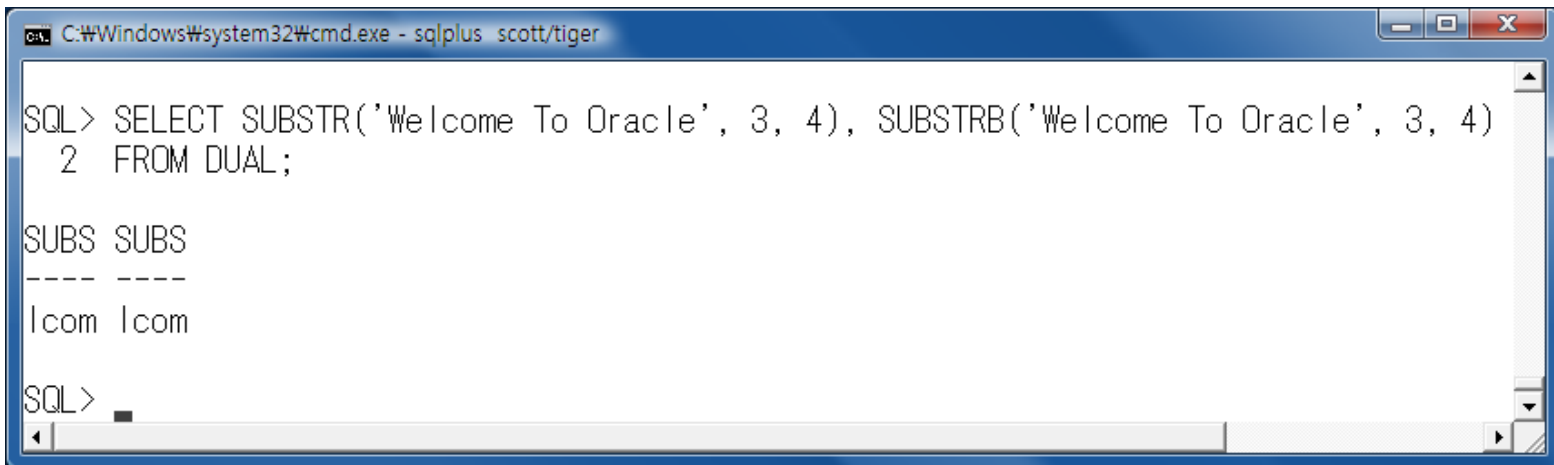


바이트 수를 추출-SUBSTRB

- ❖ SUBSTRB 함수는 문자열을 추출하기 위해서 문자를 셀 때 문자의 개수가 아닌 그 문자가 메모리에 저장되는 바이트 수로 계산
- ❖ 영문자는 1자는 메모리에 1바이트로 저장되기 때문에 SUBSTR 함수와 SUBSTRB 함수 어떤 함수를 사용하여도 결과가 동일

예

```
SELECT SUBSTR('Welcome To Oracle', 3, 4),  
       SUBSTRB('Welcome To Oracle', 3, 4)  
FROM DUAL;
```



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 SQL query: "SELECT SUBSTR('Welcome To Oracle', 3, 4), SUBSTRB('Welcome To Oracle', 3, 4) FROM DUAL;". The output shows two columns, both labeled "SUBS", with the value "lcom" displayed under each. The prompt "SQL>" is visible at the bottom.

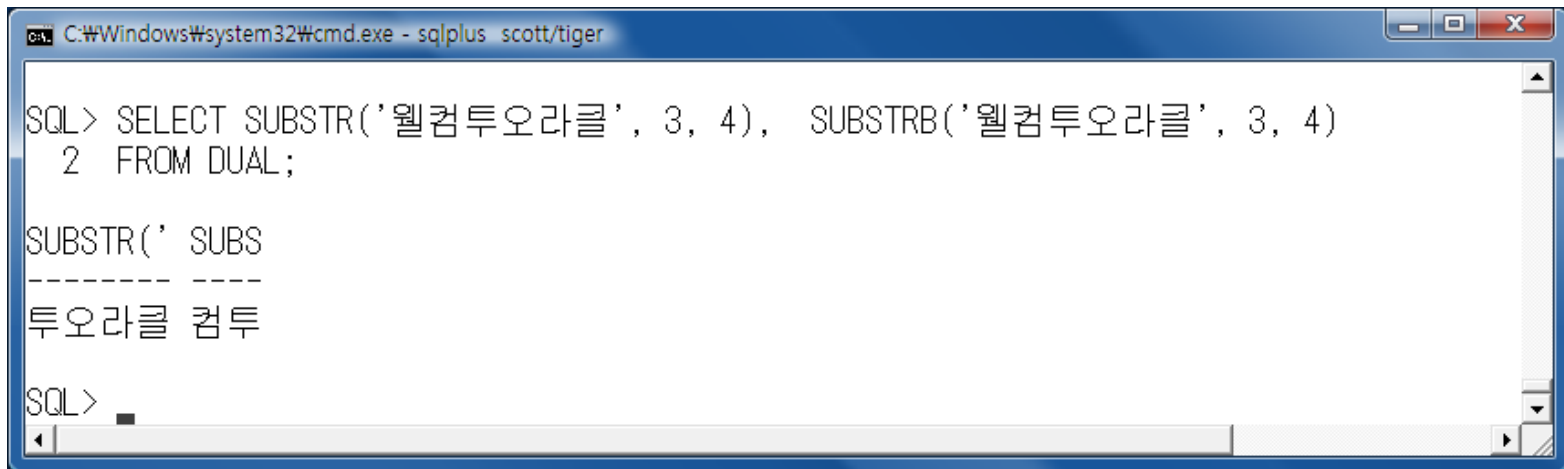
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
SQL> SELECT SUBSTR('Welcome To Oracle', 3, 4), SUBSTRB('Welcome To Oracle', 3, 4)  
2  FROM DUAL;  
  
SUBS SUBS  
-----  
lcom lcom  
  
SQL>
```

바이트 수를 추출-SUBSTRB

- ❖ 한글 1자는 2바이트 이상을 차지하기 때문에 SUBSTR 함수와 SUBSTRB 함수의 결과가 다름

예

```
SELECT SUBSTR('웰컴투오라클', 3, 4), SUBSTRB('웰컴투오라클', 3, 4)  
FROM DUAL;
```



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

```
SQL> SELECT SUBSTR('웰컴투오라클', 3, 4), SUBSTRB('웰컴투오라클', 3, 4)  
2 FROM DUAL;
```

The output of the query is displayed as follows:

```
SUBSTR(' SUBS  
-----  
투오라클 컴투
```

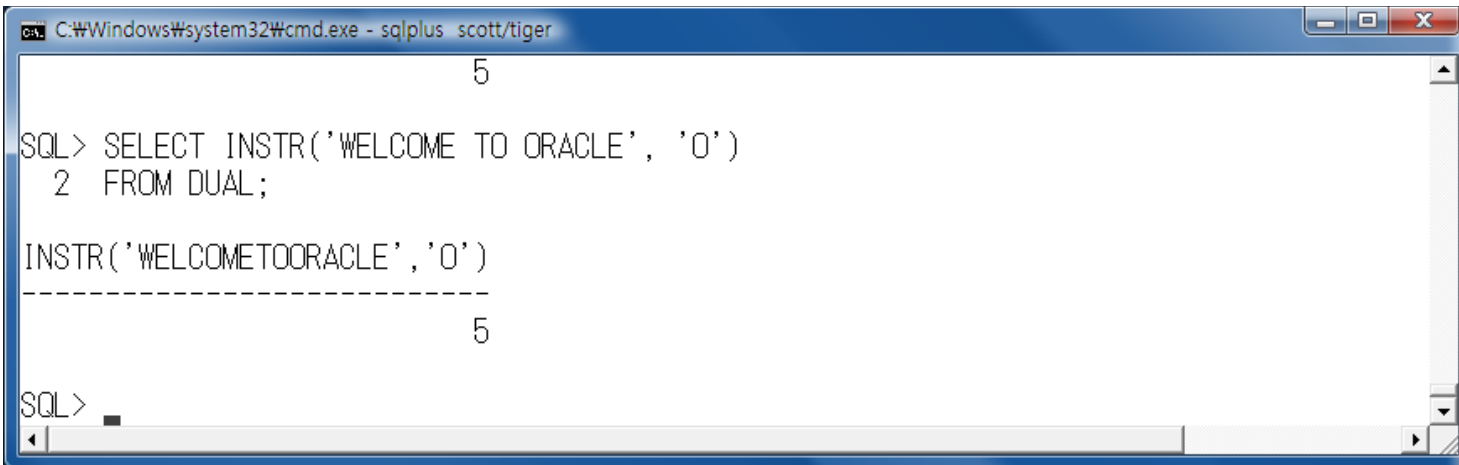
The output shows that the SUBSTR function returns 'SUBS' (4 characters) and the SUBSTRB function returns '투오라클 컴투' (8 bytes, 4 Korean characters). This demonstrates that in Korean text, each character is represented by 2 bytes.

문자의 위치를 구하는 INSTR

- ❖ INSTR 함수는 대상 문자열이나 칼럼에서 특정 문자가 나타나는 위치를 리턴
- ❖ 문자열 'Welcome to Oracle'에 'O'가 저장된 위치가 얼마인지 알고 싶을 때에는 INSTR 함수를 사용하여 다음과 같이 쿼리문을 작성할 수 있음

예

```
SELECT INSTR('WELCOME TO ORACLE', 'O')  
FROM DUAL;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The window contains the following text:

```
5  
SQL> SELECT INSTR('WELCOME TO ORACLE', 'O')  
2 FROM DUAL;  
  
INSTR('WELCOMETOORACLE', 'O')  
-----  
5  
  
SQL>
```

The output shows the result of the INSTR function is 5, indicating the position of the first 'O' in the string 'WELCOME TO ORACLE'.

문자의 위치를 구하는 INSTR

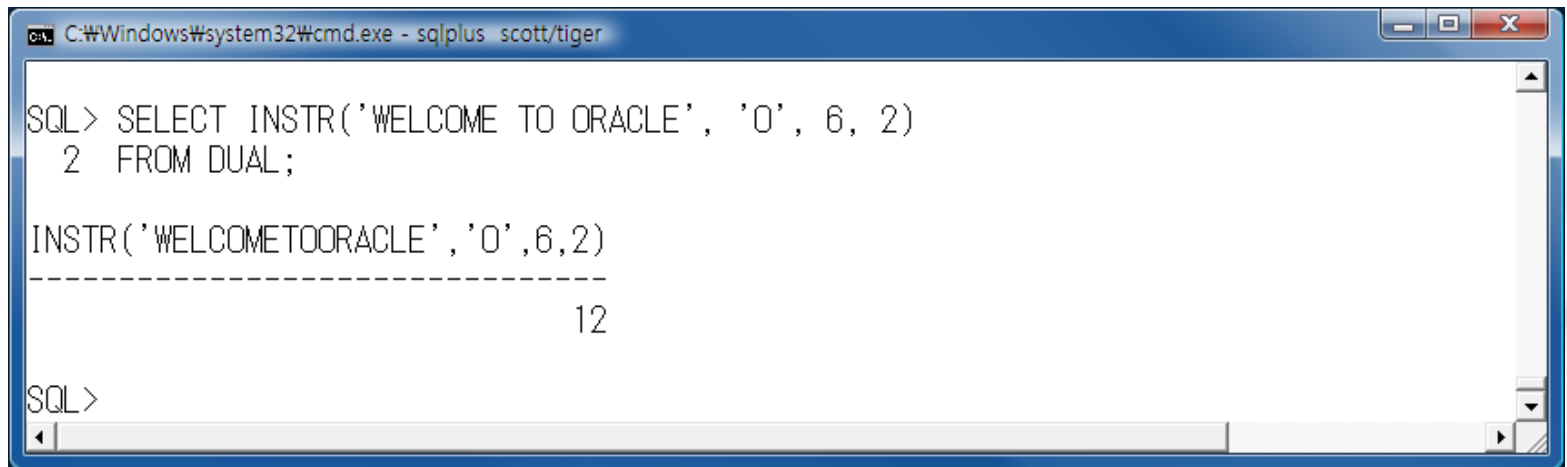
❖ INSTR 함수의 기본 형식

형식	INSTR(대상, 찾을글자, 시작위치, 몇_번째_발견)
----	--------------------------------

- ❖ 구문에서 보듯이 앞선 예제에서는 '시작위치'와 '몇 번째 발견'을 생략한 채 사용한 것으로 이들 값을 생략하면 모두 1로 간주되므로 시작 위치도 1이고 첫 번째 발견된 위치를 반환

예

```
SELECT INSTR('WELCOME TO ORACLE', 'O', 6, 2)
FROM DUAL;
```



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

SQL> SELECT INSTR('WELCOME TO ORACLE', 'O', 6, 2)
      2  FROM DUAL;

INSTR('WELCOMETOORACLE', 'O', 6, 2)
-----
                                12

SQL>
```

INSTRB

- ❖ INSTRB 함수 역시 SUBSTRB 함수에서와 마찬가지로 문자의 위치를 알아내기 위한 기준으로 바이트를 사용
- ❖ 영문자는 1 글자가 1 바이트이므로 INSTR 함수와 INSTRB 함수의 결과가 동일
- ❖ 한글은 다르게 동작

예

```
SELECT INSTR('데이터베이스', '이', 3, 1),
       INSTRB('데이터베이스', '이', 3, 1)
FROM DUAL;
```

A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The window contains the following text:

```
SQL> SELECT INSTR('데이터베이스', '이', 3, 1), INSTRB('데이터베이스', '이', 3, 1)
      2 FROM DUAL;
```


The output shows two columns separated by a dashed line. The first column contains the value 5, and the second column contains the value 3.

```
INSTR(' 데이터베이스','이',3,1) INSTRB(' 데이터베이스','이',3,1)
-----
                                5                                3
```


At the bottom, there is another prompt line:

```
SQL> 
```


연습문제

- ❖ emp 테이블에서 ename의 세 번째 자리가 N인 사원의 모든 정보를 조회

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

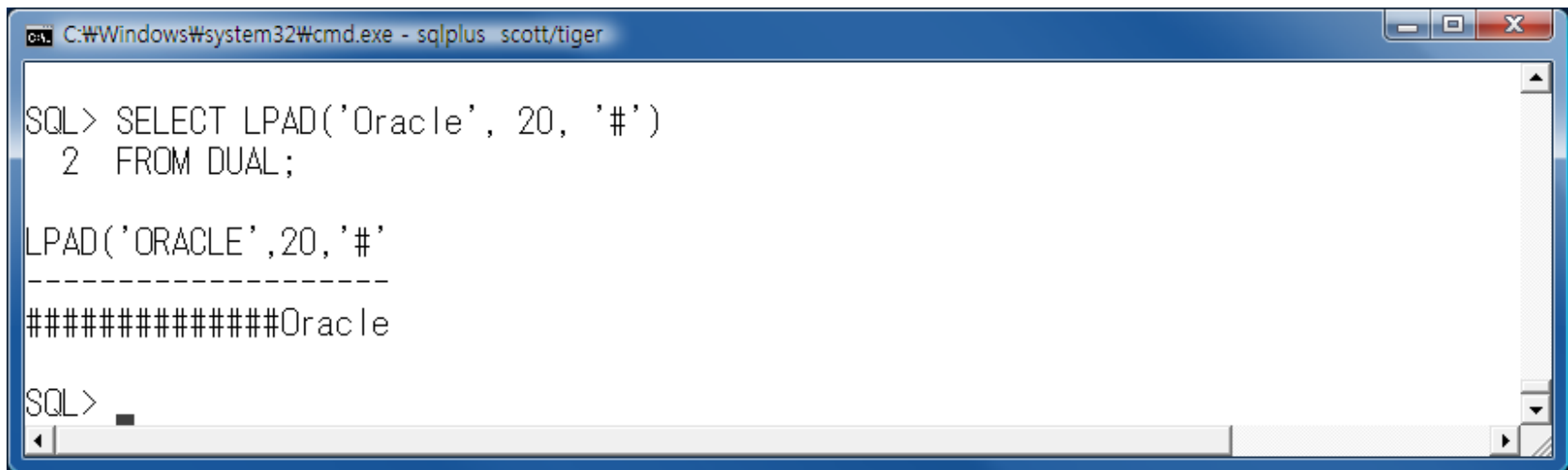


LPAD/RPAD

- ❖ LPAD(LEFT PADDING) 함수는 칼럼이나 대상 문자열을 명시된 자릿수에서 오른쪽에 나타내고, 남은 왼쪽 자리를 특정 기호로 채움

예

```
SELECT LPAD('Oracle', 20, '#')  
FROM DUAL;
```



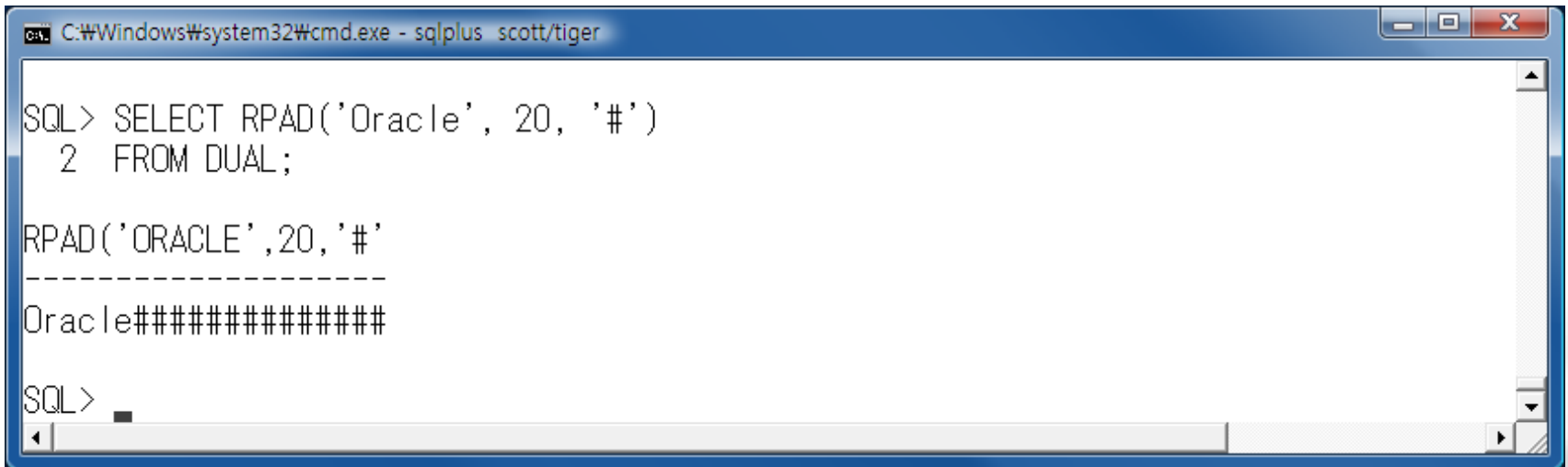
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT LPAD('Oracle', 20, '#')  
2 FROM DUAL;  
  
LPAD('ORACLE',20,'#'  
-----  
#####Oracle  
  
SQL>
```

LPAD/RPAD

- ❖ RPAD(RIGHT PADDING) 함수는 반대로 칼럼이나 대상 문자열을 명시된 자릿수에서 왼쪽에 나타내고, 남은 오른쪽 자리를 특정 기호로 채움

예

```
SELECT RPAD('Oracle', 20, '#')  
FROM DUAL;
```



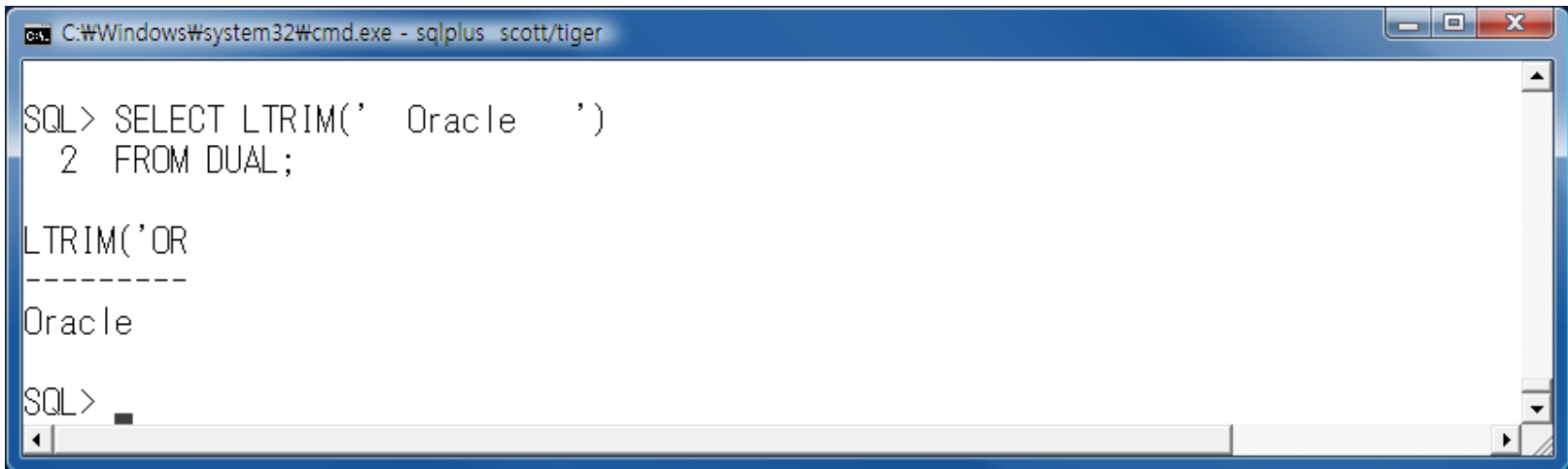
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT RPAD('Oracle', 20, '#')  
2  FROM DUAL;  
  
RPAD('ORACLE',20,'#'  
-----  
Oracle#####  
  
SQL>
```

왼쪽 공백을 삭제 - LTRIM

- ❖ LTRIM 함수는 문자열의 왼쪽(앞)의 공백 문자들을 삭제

예

```
SELECT LTRIM(' Oracle ')
FROM DUAL;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". Inside the window, the following SQL command is entered and executed:

```
SQL> SELECT LTRIM(' Oracle ')
2 FROM DUAL;
```

The output of the query is displayed as follows:

```
LTRIM('OR
-----
Oracle
SQL>
```

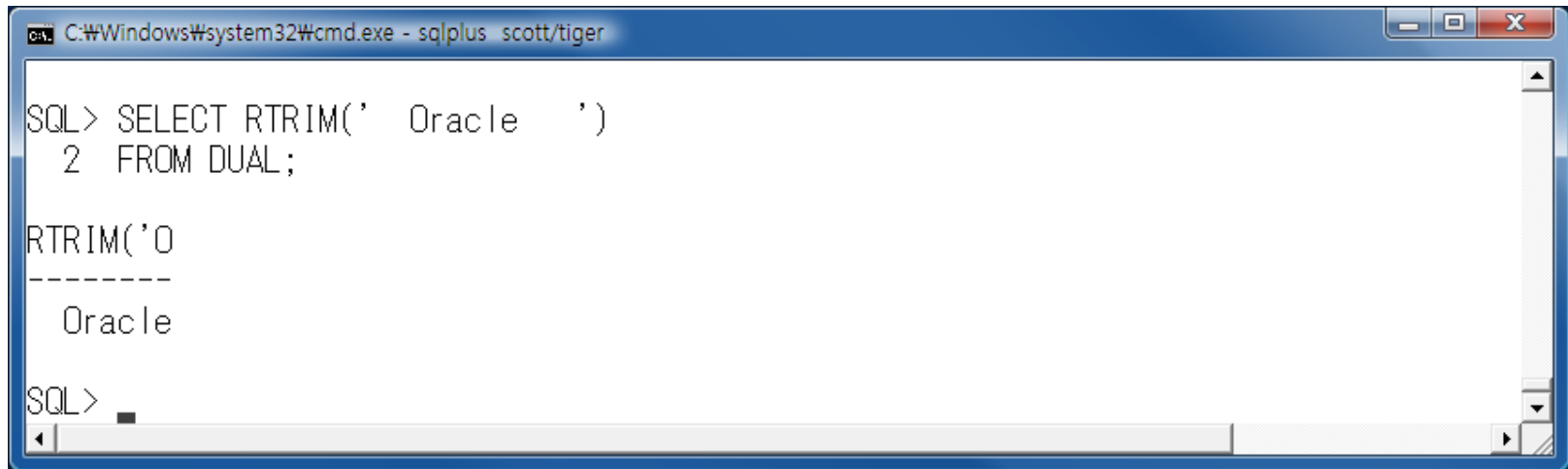
The output shows the string "Oracle" with a dashed line above it, indicating the result of the LTRIM function applied to the string " Oracle ".

오른쪽 공백을 삭제 - RTRIM

- ❖ RTRIM 함수 오른쪽 공백 문자를 잘라냄

예

```
SELECT RTRIM(' Oracle ')  
FROM DUAL;
```



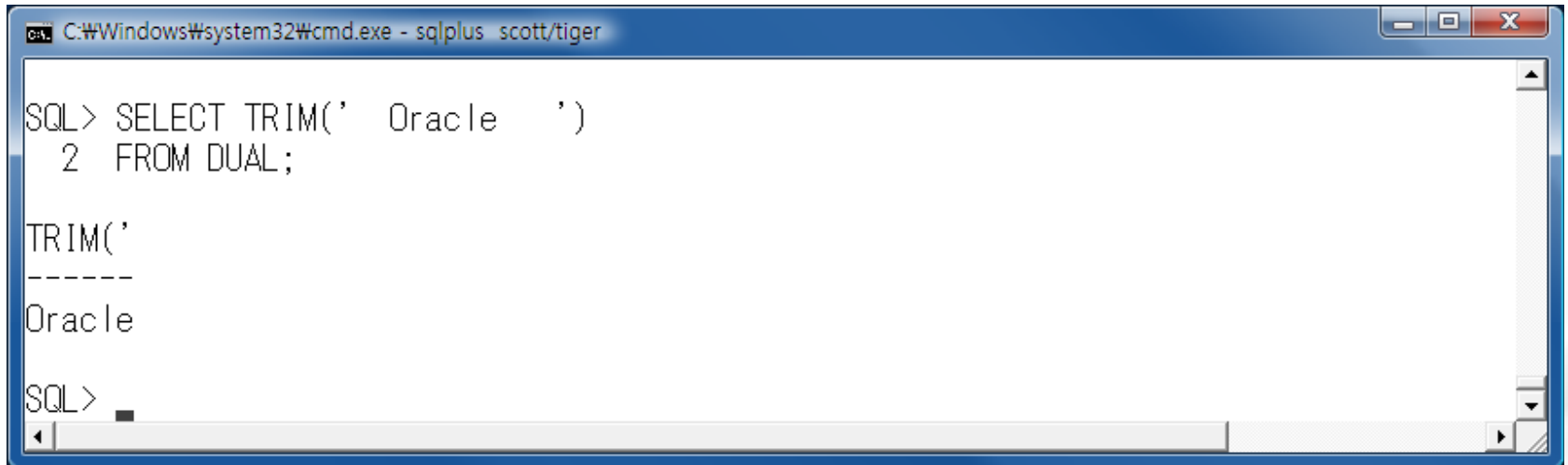
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT RTRIM(' Oracle ')  
2 FROM DUAL;  
  
RTRIM('O  
-----  
Oracle  
  
SQL>
```

좌우 문자를 삭제 - TRIM

- ❖ 다음과 같이 기술하면 시작하는 부분과 끝나는 부분의 공백 문자를 잘라냄

예

```
SELECT TRIM(' Oracle ')  
FROM DUAL;
```



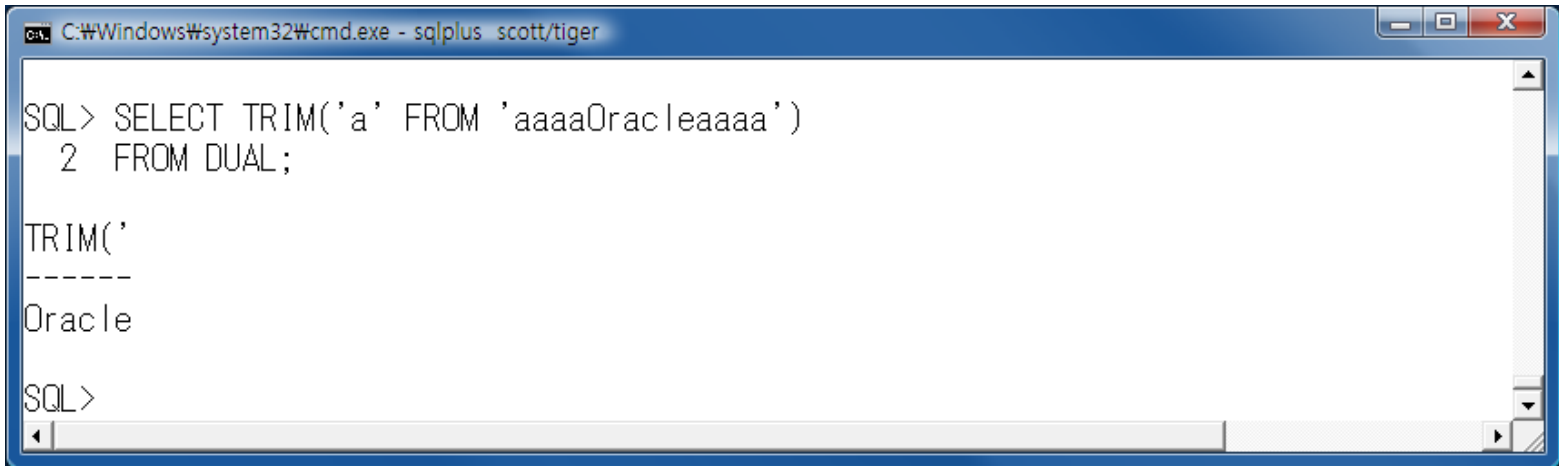
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT TRIM(' Oracle ')  
2 FROM DUAL;  
  
TRIM('-----  
Oracle  
  
SQL>
```

좌우 문자를 삭제 - TRIM

- ❖ TRIM 함수는 칼럼이나 대상 문자열에서 특정 문자가 첫 번째 글자이거나 마지막 글자이면 잘라내고 남은 문자열만 반환

예

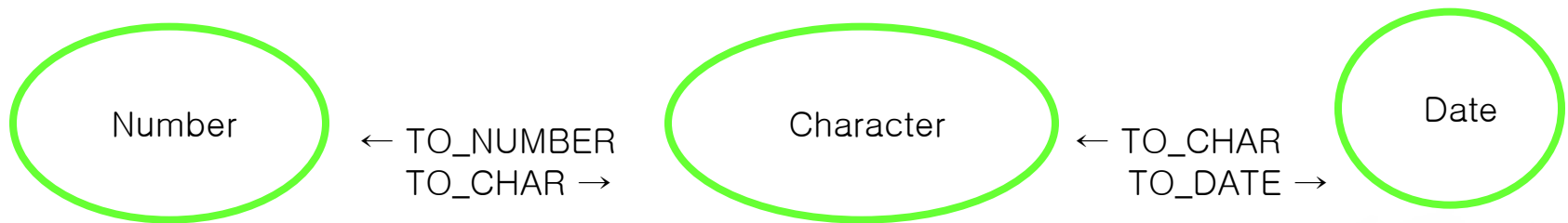
```
SELECT TRIM('a' FROM 'aaaaOracleaaaa')  
FROM DUAL;
```



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT TRIM('a' FROM 'aaaaOracleaaaa')  
2 FROM DUAL;  
  
TRIM('-----  
Oracle  
  
SQL>
```

형 변환 함수

- ❖ 오라클을 사용하다 보면 숫자, 문자, 날짜의 데이터 형을 다른 데이터형으로 변환해야 하는 경우가 발생
- ❖ 이때 사용하는 함수가 형 변환 함수
- ❖ 형 변환 함수로는 TO_NUMBER, TO_CHAR, TO_DATE 가 있음



구 분	설 명
TO_CHAR	날짜형 혹은 숫자형을 문자형으로 변환
TO_DATE	문자형을 날짜형으로 변환
TO_NUMBER	문자형을 숫자형으로 변환

TO_CHAR

- ❖ DATE 형태의 데이터를 지정한 양식에 의해 VARCHAR2 형의 문자로 변환

형식	TO_CHAR (날짜 데이터, '출력형식')
----	--------------------------

- ❖ 다음은 날짜 출력 형식의 종류를 나열한 표

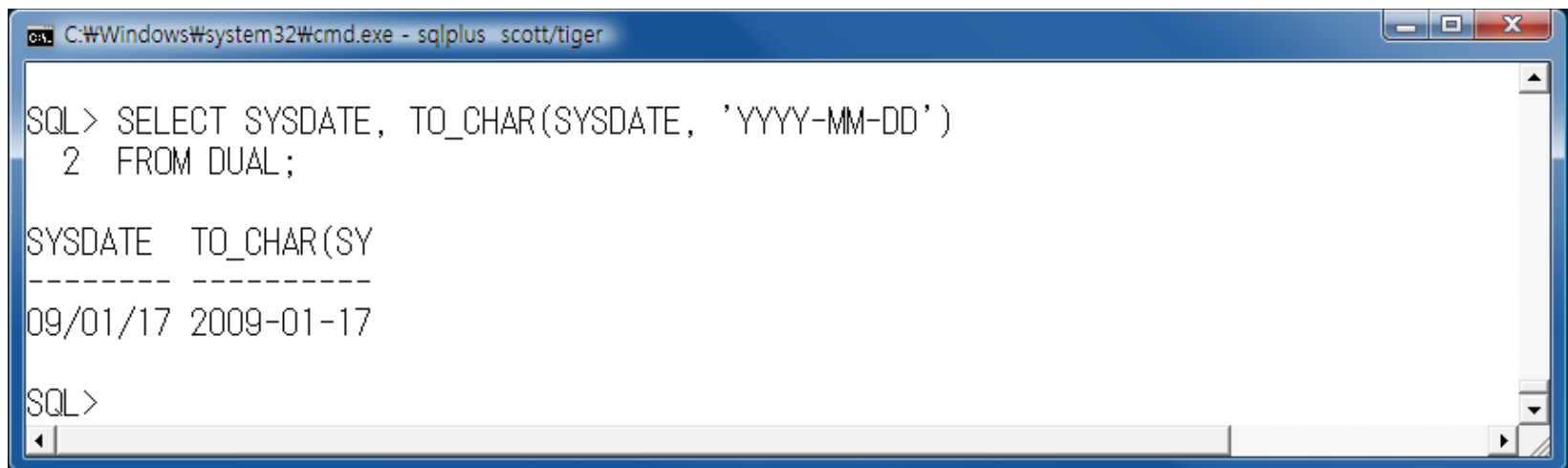
종류	의미
YYYY	년도 표현(4자리)
YY	년도 표현(2자리)
MM	월을 숫자로 표현
MON	월을 알파벳으로 표현
DAY	요일 표현
DY	요일을 약어로 표현

TO_CHAR

- ❖ 현재 날짜를 기본 형식과 다른 형태로 출력

예

```
SELECT SYSDATE, TO_CHAR(SYSDATE, 'YYYY-MM-DD')  
FROM DUAL;
```



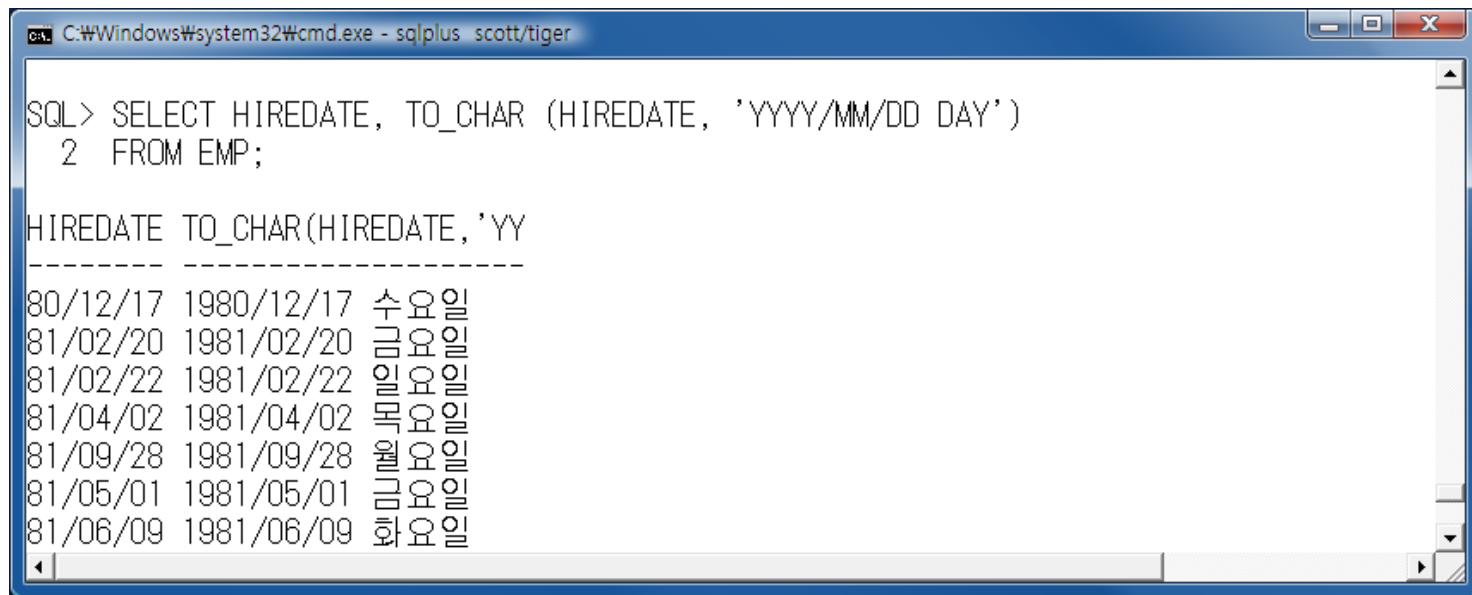
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT SYSDATE, TO_CHAR(SYSDATE, 'YYYY-MM-DD')  
2 FROM DUAL;  
  
SYSDATE    TO_CHAR(SY  
-----  
09/01/17   2009-01-17  
  
SQL>
```

TO_CHAR

- ❖ 사원들의 입사일을 출력하되 요일까지 함께 출력

예

```
SELECT HIREDATE, TO_CHAR (HIREDATE, 'YYYY/MM/DD DAY')  
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 HIREDATE, TO_CHAR (HIREDATE, 'YYYY/MM/DD DAY')
2 FROM EMP;`. The output displays a list of employee hire dates and their corresponding formatted strings, including the day of the week in Korean. The output is as follows:

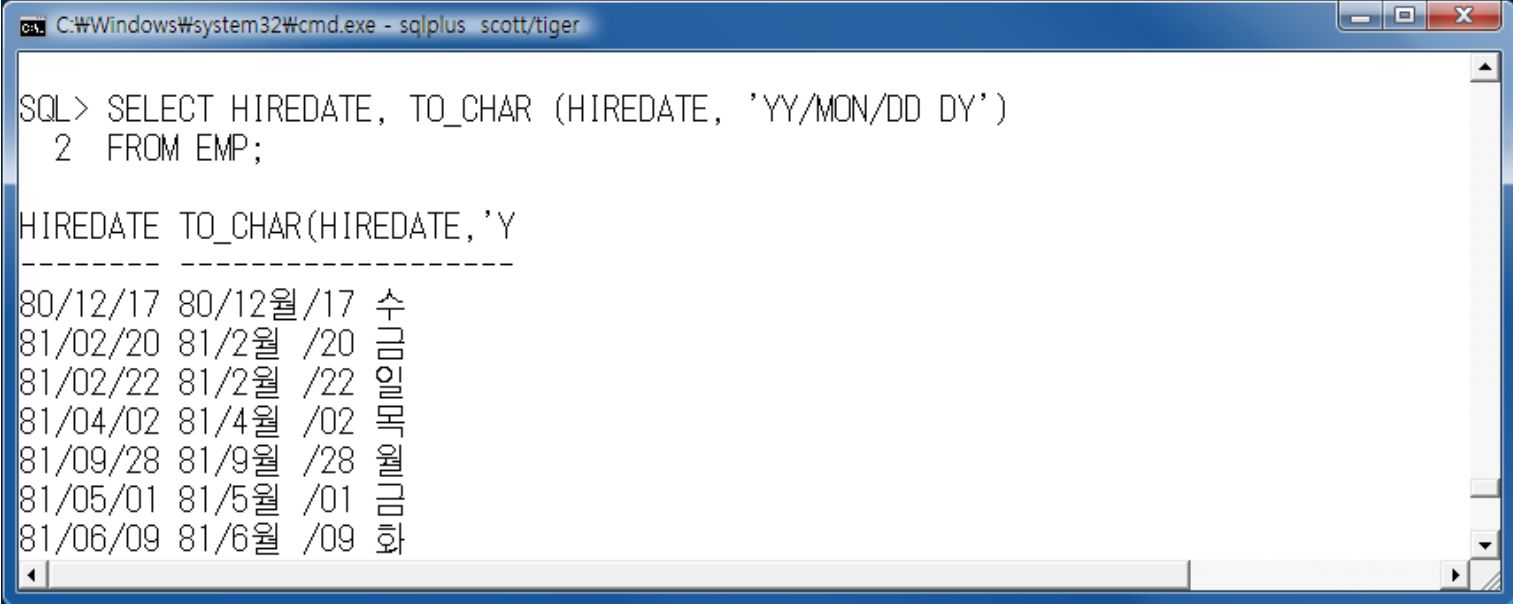
HIREDATE	TO_CHAR(HIREDATE, 'YYYY/MM/DD DAY')
80/12/17	1980/12/17 수요일
81/02/20	1981/02/20 금요일
81/02/22	1981/02/22 일요일
81/04/02	1981/04/02 목요일
81/09/28	1981/09/28 월요일
81/05/01	1981/05/01 금요일
81/06/09	1981/06/09 화요일

TO_CHAR

- ❖ 년도를 2자리(YY)로 출력하고 월은 문자(MON)로 표시하고 요일을 약어(DY)로 표시

예

```
SELECT HIREDATE, TO_CHAR (HIREDATE, 'YY/MON/DD DY')  
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 HIREDATE, TO_CHAR (HIREDATE, 'YY/MON/DD DY')
2 FROM EMP;`. The output displays a table with two columns: `HIREDATE` and `TO_CHAR(HIREDATE, 'YY/MON/DD DY')`. The data rows show dates in YY/MM/DD format followed by the Korean day of the week (e.g., 수, 금, 화, 목, 금, 화).

HIREDATE	TO_CHAR(HIREDATE, 'YY/MON/DD DY')
80/12/17	80/12월/17 수
81/02/20	81/2월/20 금
81/02/22	81/2월/22 화
81/04/02	81/4월/02 목
81/09/28	81/9월/28 금
81/05/01	81/5월/01 화
81/06/09	81/6월/09 화

TO_CHAR

❖ 시간 출력 형식의 종류를 나열한 표

종류	의미
AM 또는 PM	오전(AM), 오후(PM) 시각 표시
A.M 또는 P.M	오전(A.M), 오후(P.M) 시각 표시
HH또는 HH12	시간(1~12)
HH24	24시간으로 표현(0~23)
MI	분 표현
SS	초 표현

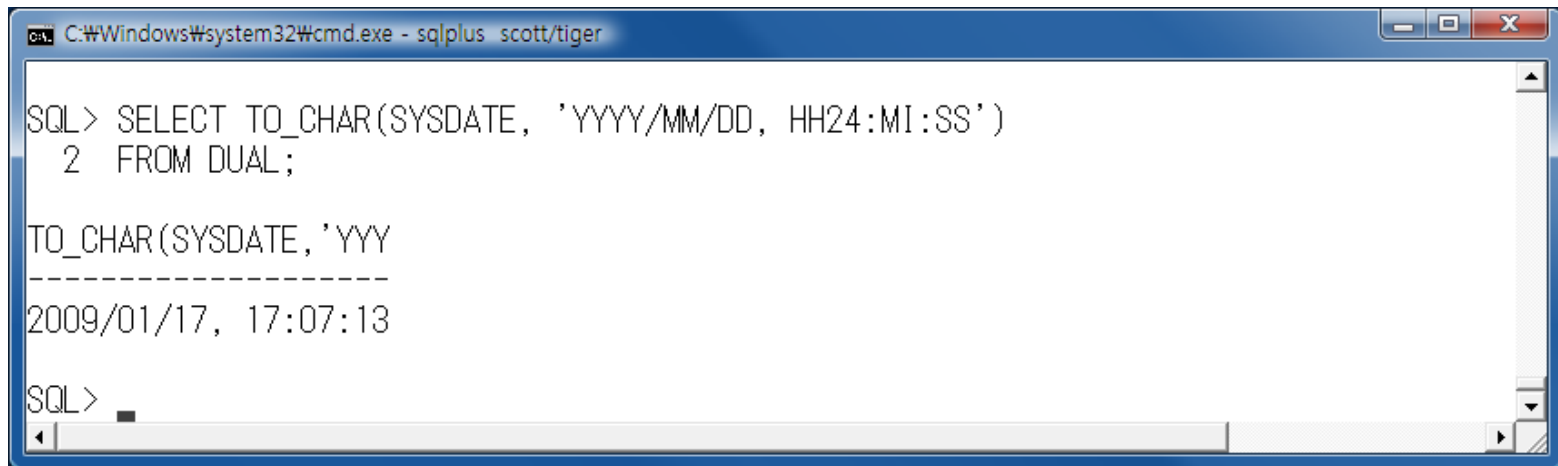


TO_CHAR

- ❖ 현재 날짜와 시간을 출력

예

```
SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD, HH24:MI:SS')  
FROM DUAL;
```



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 command: `SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD, HH24:MI:SS')` followed by a new line with `2 FROM DUAL;`. The output displayed is: `TO_CHAR(SYSDATE, 'YYY` followed by a dashed line separator and the result `2009/01/17, 17:07:13`. The prompt is now `SQL>` with a cursor.

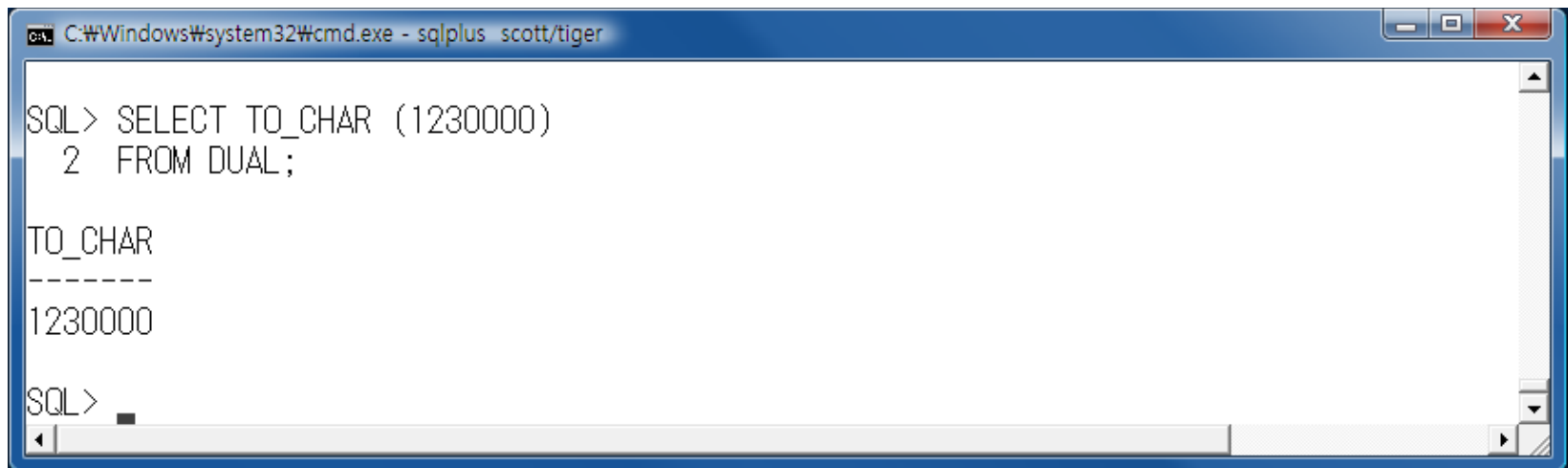
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD, HH24:MI:SS')  
      2 FROM DUAL;  
  
TO_CHAR(SYSDATE, 'YYY  
-----  
2009/01/17, 17:07:13  
  
SQL>
```

TO_CHAR

- ❖ 1230000이란 숫자를 문자 형태로 출력

예

```
SELECT TO_CHAR (1230000)
FROM DUAL;
```



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

SQL> SELECT TO_CHAR (1230000)
      2  FROM DUAL;

TO_CHAR
-----
1230000

SQL>
```

TO_CHAR

❖ 숫자 출력 형식을 나열한 표

구 분	설 명
0	자릿수를 나타내며 자릿수가 맞지 않을 경우 0으로 채움
9	자릿수를 나타내며 자릿수가 맞지 않아도 채우지 않음
L	각 지역별 통화 기호를 앞에 표시
.	소수점
,	천 단위 자리 구분

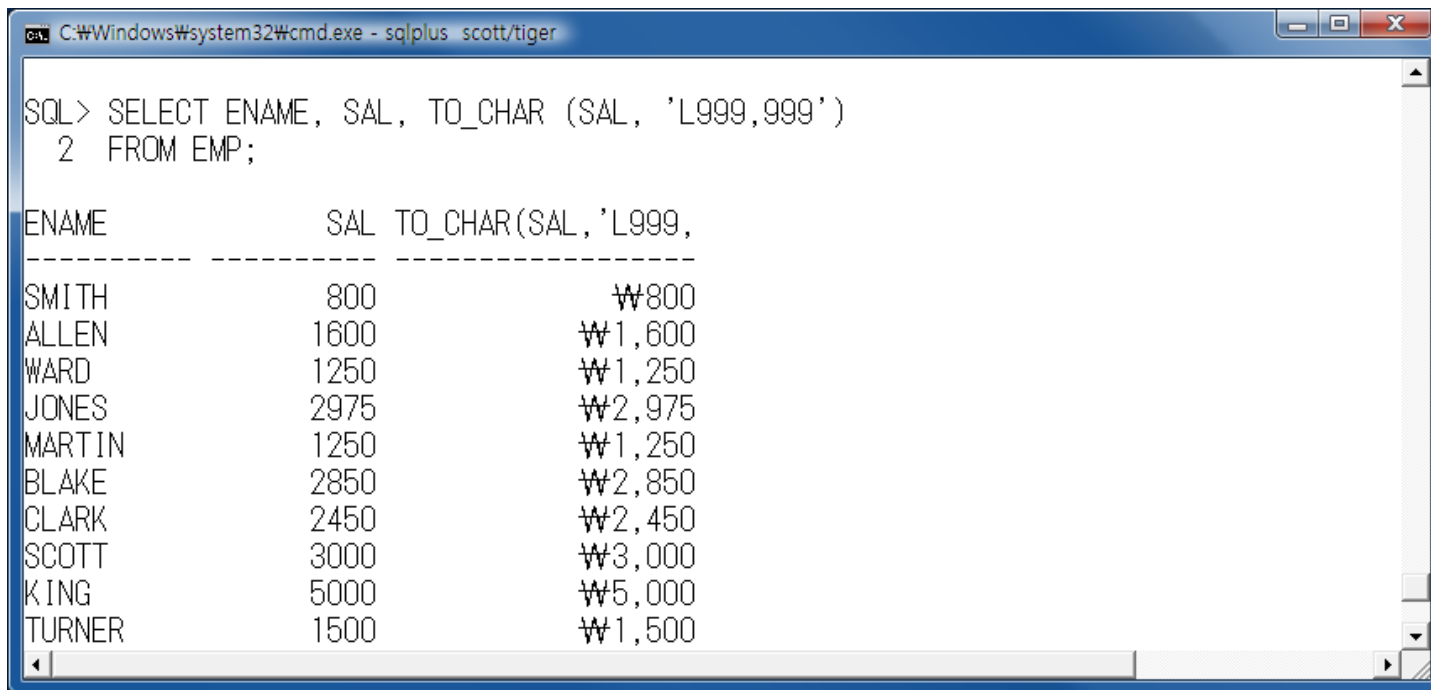


TO_CHAR

- ❖ 각 지역별 통화 기호를 앞에 붙이고 천 단위마다 콤마를 붙여서 출력(예 : ₩1,230,000) 하려면 표를 참조하여 다음과 같이 표현

예

```
SELECT ENAME, SAL, TO_CHAR (SAL, 'L999,999')  
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 query: `SQL> SELECT ENAME, SAL, TO_CHAR (SAL, 'L999,999')` followed by a blank line and `2 FROM EMP;`. The output displays a table with three columns: ENAME, SAL, and TO_CHAR(SAL, 'L999,999'). The salaries are formatted with the Korean won symbol (₩) and commas as thousands separators.

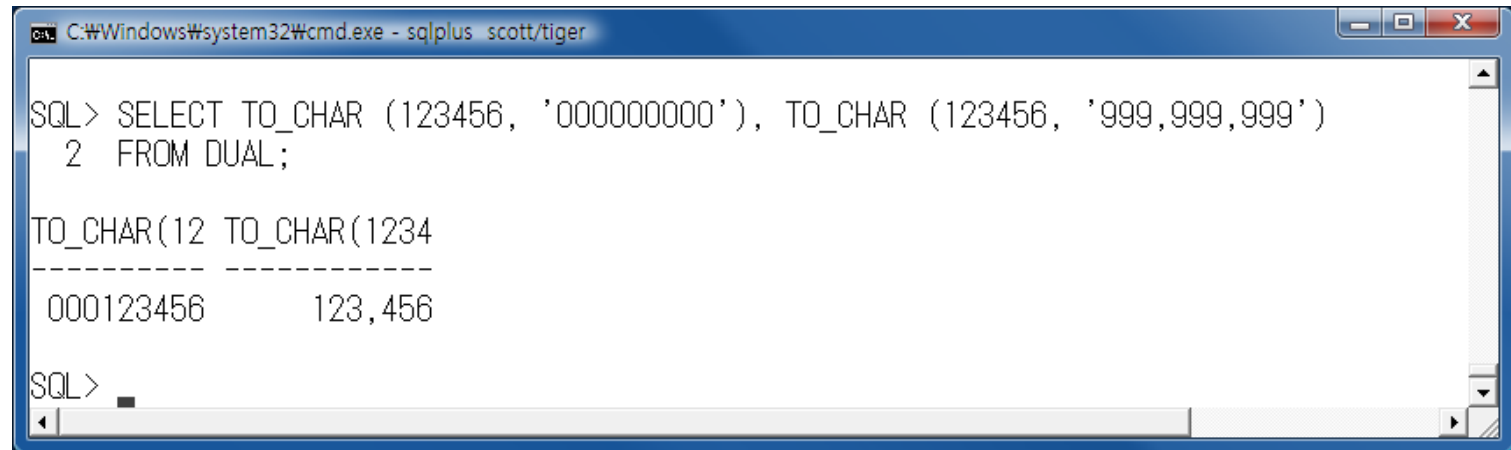
ENAME	SAL	TO_CHAR(SAL, 'L999,999')
SMITH	800	₩800
ALLEN	1600	₩1,600
WARD	1250	₩1,250
JONES	2975	₩2,975
MARTIN	1250	₩1,250
BLAKE	2850	₩2,850
CLARK	2450	₩2,450
SCOTT	3000	₩3,000
KING	5000	₩5,000
TURNER	1500	₩1,500

TO_CHAR

- ❖ 9는 자릿수를 나타내며 자릿수가 맞지 않으면 채우지 않지만 0은 자릿수를 나타내며 자릿수가 맞지 않을 경우 0으로 채움

예

```
SELECT TO_CHAR (123456, '000000000'),  
       TO_CHAR (123456, '999,999,999')  
FROM DUAL;
```



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT TO_CHAR (123456, '000000000'), TO_CHAR (123456, '999,999,999')  
2 FROM DUAL;  
  
TO_CHAR(123456, '000000000') TO_CHAR(123456, '999,999,999')  
-----  
000123456                123,456  
  
SQL>
```

TO_DATE

- ❖ 날짜 형은 세기, 년도, 월, 일. 시간, 분. 초와 같이 날짜와 시간에 대한 정보를 저장
- ❖ 오라클에서 기본 날짜 형식은 'YY/MM/DD' 형식으로 '년/월/일' 예를 들면 '06/03/08' 식으로 표현
- ❖ 만일 년도를 4자리로 출력하려면 'YYYY/MM/DD' 형식으로 지정
- ❖ TO_DATE 함수는 문자열을 날짜 형으로 변환

형식	TO_DATE('문자', 'format')
----	-------------------------

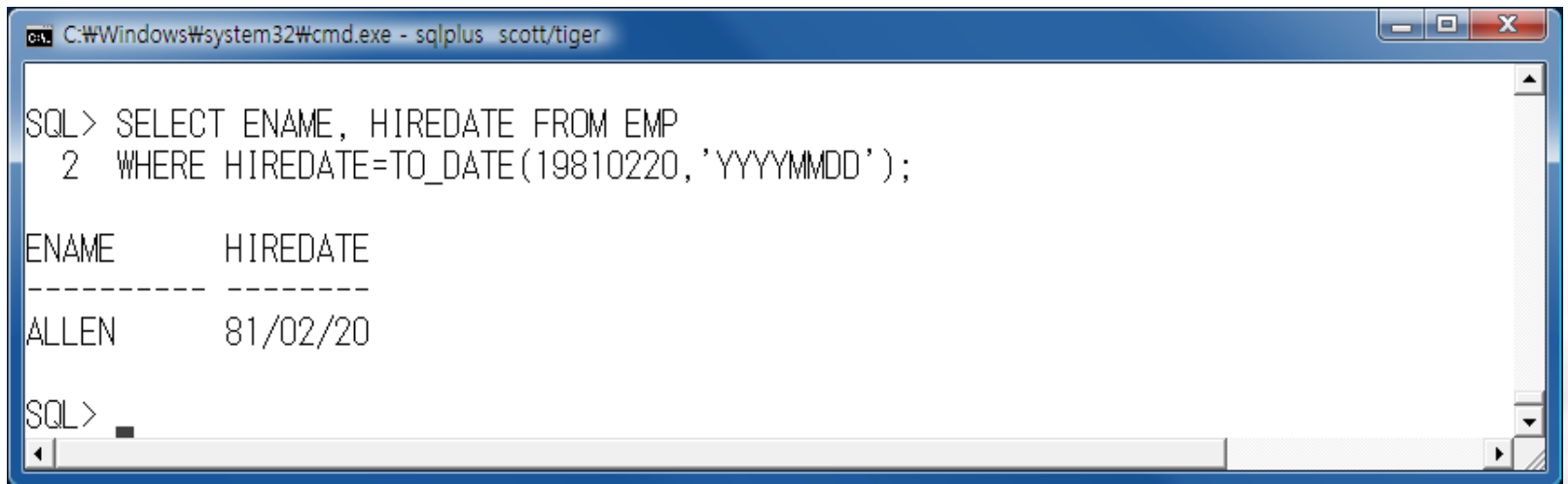


TO_DATE

- ❖ 숫자 형태인 19810220 을 TO_DATE 함수를 사용해서 날짜형으로 변환
- ❖ 두 번째 인자는 출력할 형식을 지정

예

```
SELECT ENAME, HIREDATE FROM EMP  
WHERE HIREDATE=TO_DATE(19810220,'YYYYMMDD')
```



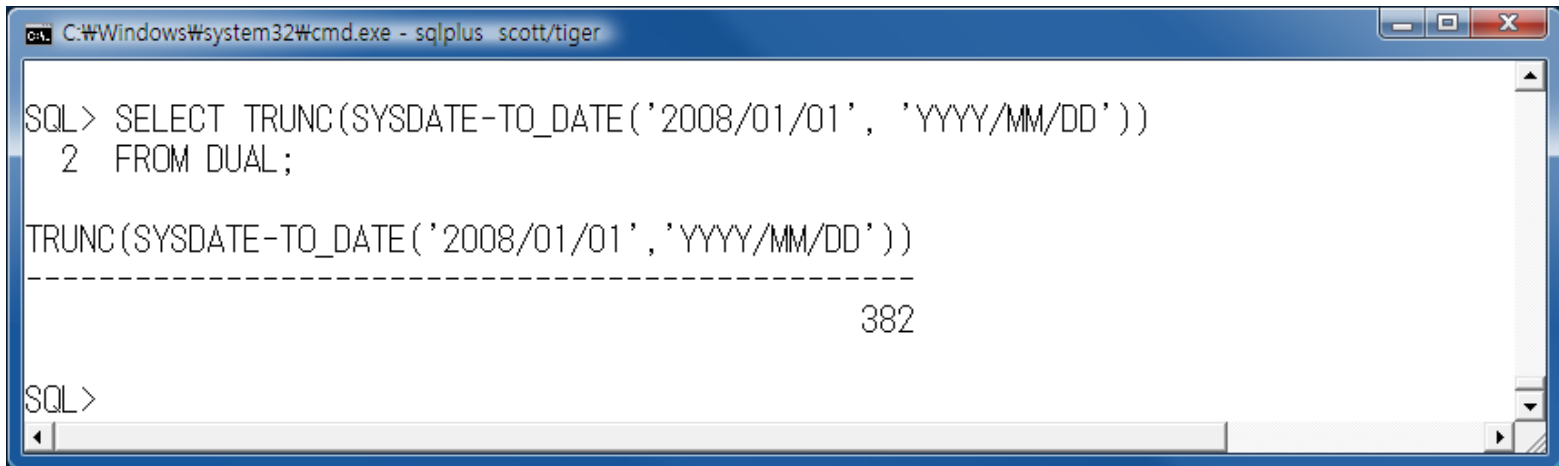
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT ENAME, HIREDATE FROM EMP  
2  WHERE HIREDATE=TO_DATE(19810220,'YYYYMMDD');  
  
ENAME      HIREDATE  
-----  
ALLEN      81/02/20  
  
SQL>
```

TO_DATE

- ❖ 올해 며칠이 지났는지 현재 날짜에서 2008/01/01을 뺀 결과를 출력

예

```
SELECT TRUNC(SYSDATE-TO_DATE('2008/01/01', 'YYYY/MM/DD'))  
FROM DUAL;
```



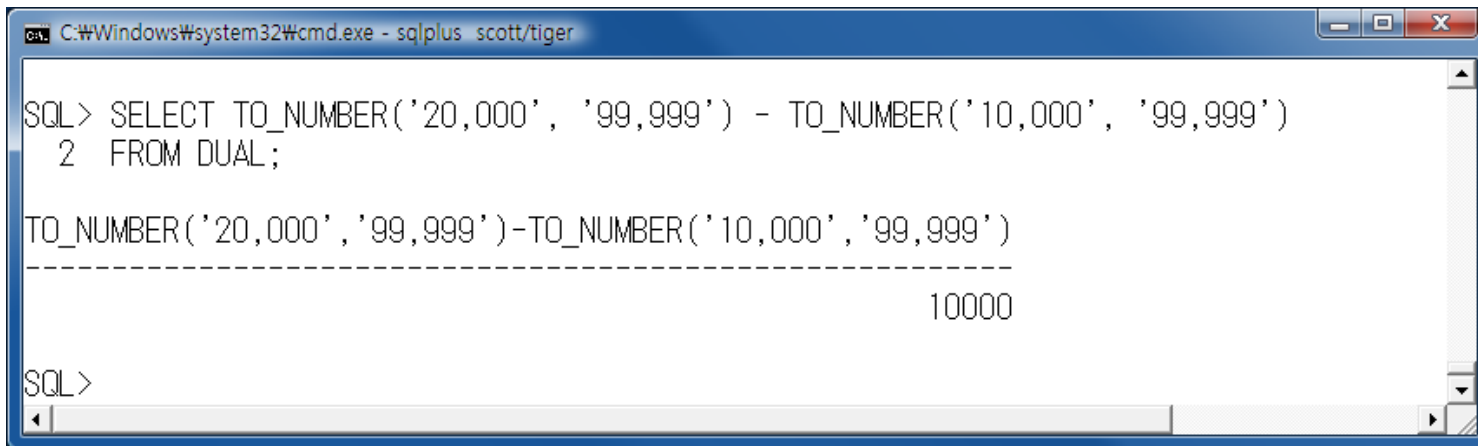
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT TRUNC(SYSDATE-TO_DATE('2008/01/01', 'YYYY/MM/DD'))  
2  FROM DUAL;  
  
TRUNC(SYSDATE-TO_DATE('2008/01/01', 'YYYY/MM/DD'))  
-----  
382  
  
SQL>
```

TO_NUMBER

- ❖ TO_NUMBER 함수는 특정 데이터를 숫자형으로 변환해 주는 함수
- ❖ 다음과 같이 '20,000'을 '10,000'의 차이를 알아보기 위해서 산술 연산을 하려면 문자형을 숫자형으로 변환한 후에 수행

예

```
SELECT TO_NUMBER('20,000', '99,999') - TO_NUMBER('10,000', '99,999')  
FROM DUAL;
```



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 TO_NUMBER('20,000', '99,999') - TO_NUMBER('10,000', '99,999')`. The prompt has moved to the next line, and the user has entered `2 FROM DUAL;`. The query has been executed, and the output is displayed: `TO_NUMBER('20,000', '99,999')-TO_NUMBER('10,000', '99,999')` followed by a dashed line and the result `10000`. The prompt is now `SQL>`.

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
SQL> SELECT TO_NUMBER('20,000', '99,999') - TO_NUMBER('10,000', '99,999')  
2 FROM DUAL;  
  
TO_NUMBER('20,000', '99,999')-TO_NUMBER('10,000', '99,999')  
-----  
10000  
  
SQL>
```

날짜 함수

구 분	설 명
SYSDATE	시스템 저장된 현재 날짜를 반환
MONTHS_BETWEEN	두 날짜 사이가 몇 개월인지를 반환
ADD_MONTHS	특정 날짜에 개월 수를 더함
NEXT_DAY	특정 날짜에서 최초로 도래하는 인자로 받은 요일의 날짜를 반환
LAST_DAY	해당 달의 마지막 날짜를 반환
ROUND	인자로 받은 날짜를 특정 기준으로 반올림
TRUNC	인자로 받은 날짜를 특정 기준으로 버림

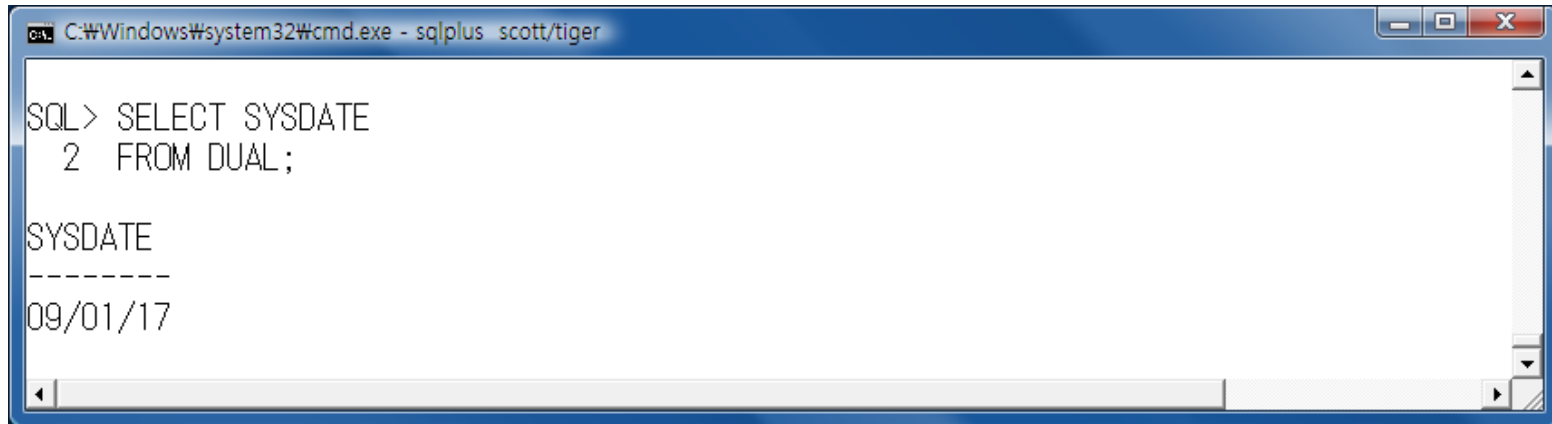


SYSDATE

- ❖ SYSDATE 함수는 시스템에 저장된 현재 날짜를 반환하는 함수

예

```
SELECT SYSDATE  
FROM DUAL;
```



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 command "SELECT SYSDATE" on the first line and "2 FROM DUAL;" on the second line. The output shows "SYSDATE" followed by a dashed line and the date "09/01/17".

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT SYSDATE  
2 FROM DUAL;  
  
SYSDATE  
-----  
09/01/17
```


날짜 연산

- ❖ 날짜 형 데이터에 숫자를 더하면(날짜+숫자) 그 날짜로부터 그 기간만큼 지난 날짜를 계산
- ❖ 날짜 형 데이터에 숫자를 빼면(날짜-숫자) 그 날짜로부터 그 기간만큼 이전 날짜를 계산
- ❖ 1은 하루에 해당

예

```
SELECT SYSDATE-1 어제, SYSDATE 오늘, SYSDATE+1 내일  
FROM DUAL;
```

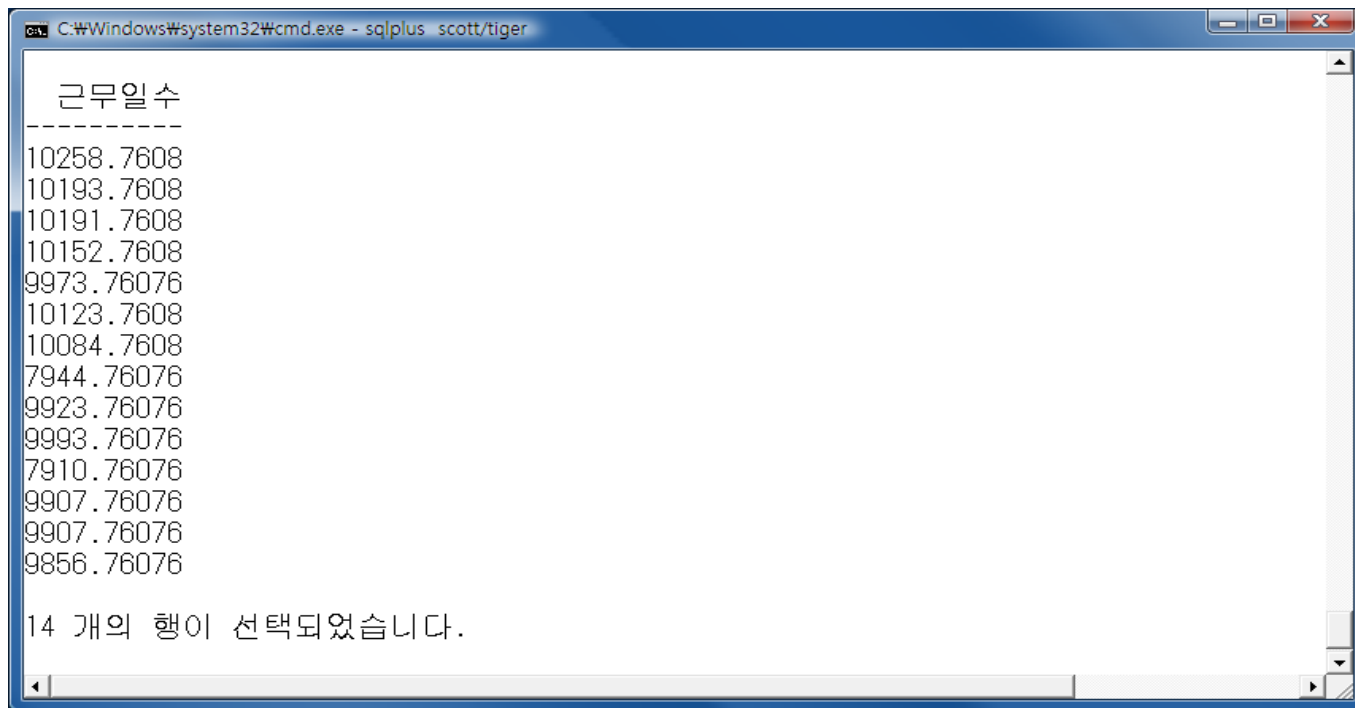
C:\WINDOWS\system32\cmd.exe - sqlplus scott/tiger

```
SQL> SELECT SYSDATE-1 어제, SYSDATE 오늘, SYSDATE+1 내일  
2 FROM DUAL;
```

어제	오늘	내일
09/01/12	09/01/13	09/01/14

연습문제

- ❖ 날짜에서 날짜를 빼는 연산도 사용할 수 있는데 많이 사용되는 연산
- ❖ emp 테이블에서 각 사원들의 현재까지의 근무 일수를 조회하기
- ❖ 입사일은 hiredate 컬럼에 저장되어 있음



The screenshot shows a SQL*Plus window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The query result is displayed as follows:

```
근무일수
-----
10258.7608
10193.7608
10191.7608
10152.7608
9973.76076
10123.7608
10084.7608
7944.76076
9923.76076
9993.76076
7910.76076
9907.76076
9907.76076
9856.76076
```

14 개의 행이 선택되었습니다.

ROUND

- ❖ ROUND 함수는 포맷 모델을 지정하면 숫자 이외에 날짜에 대해서도 반올림을 할 수 있음

형식 ROUND (*date, format*)

포맷 모델	단위
CC, SCC	4자리 연도의 끝 두 글자를 기준으로 반올림
SYYY, YYYY, YEAR SYEAR, YYY, YY, Y	년(7월 1일부터 반올림)
DDD, D, J	일을 기준
HH, HH12, HH24	시를 기준
Q	한 분기의 두 번째 달의 16일을 기준으로 반올림
MONTH, MON, MM, RM	월(16일을 기준으로 반올림)
DAY,DY,D	한 주가 시작되는 날짜
MI	분을 기준

ROUND

- ❖ ROUND 함수의 포맷 모델로 MONTH를 지정하였기에 특정 날짜(DATE)를 달(MONTH)을 기준으로 반올림한 날짜를 리턴
- ❖ 일을 기준으로 16일보다 적으면 이번 달 1일을 크면 다음달 1일을 리턴하며 6월 9일을 반올림하면 6월 1일이 되고 11월 17일은 반올림하여 12월 1일이 리턴되며 또한 1월 23일을 반올림하면 2월 1일을 리턴
- ❖ 다음은 입사일을 달을 기준으로 반올림

예

```
SELECT HIREDATE, ROUND (HIREDATE, 'MONTH')  
FROM EMP;
```



TRUNC

- ❖ TRUNC 함수 역시 숫자를 잘라내는 것뿐만 아니라 날짜를 잘라낼 수 있음
- ❖ ROUND 함수와 마찬가지로 포맷 형식을 주어 다양한 기준으로 날짜를 잘라낼 수 있음
- ❖ 특정 날짜(DATE)를 달(MONTH)을 기준으로 버림한 날짜를 구하기 위해서는 다음과 같이 표현

형식	TRUNC (<i>date</i> , <i>format</i>)
----	---------------------------------------

- ❖ 예는 입사일을 달을 기준으로 절삭한 예제

예	SELECT HIREDATE, TRUNC(HIREDATE, 'MONTH') FROM EMP;
---	--



NVL

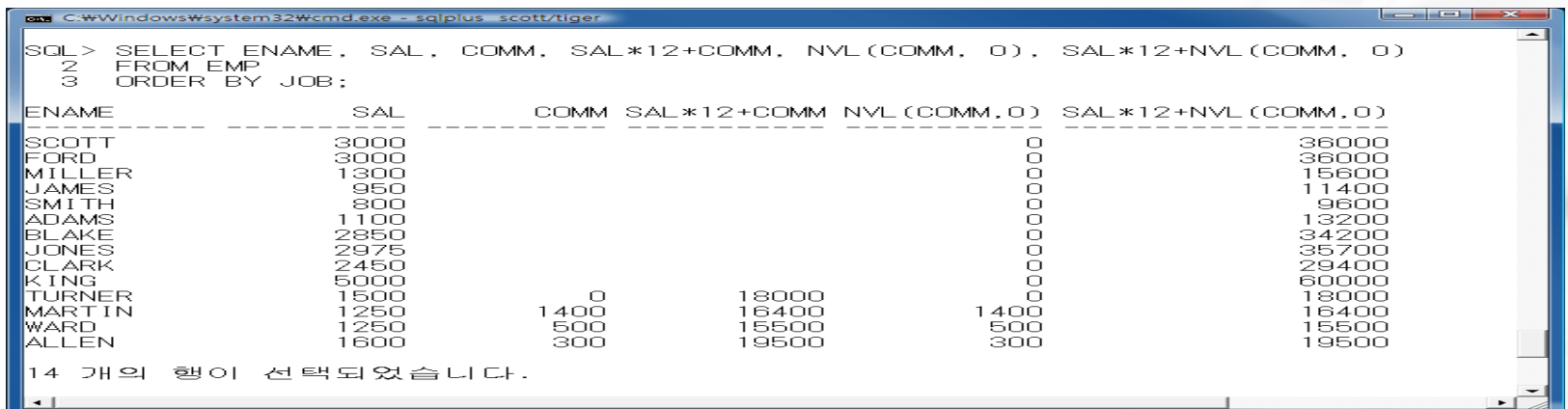
- ❖ NVL 함수는 NULL을 0 또는 다른 값으로 변환하기 위해서 사용하는 함수

형식 `NVL(data, NULL일 때의 처리)`

- ❖ COMM의 값이 NULL 일 때는 0으로 처리해서 $SAL * 12 + COMM$ 을 처리

예

```
SELECT ENAME, SAL, COMM, SAL*12+COMM,  
       NVL(COMM, 0), SAL*12+NVL(COMM, 0)  
FROM EMP  
ORDER BY JOB;
```



SQL> SELECT ENAME, SAL, COMM, SAL*12+COMM, NVL (COMM, 0), SAL*12+NVL (COMM, 0)
2 FROM EMP
3 ORDER BY JOB;

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

14 개의 행이 선택되었습니다.

NVL2

- ❖ NVL2 함수는 NULL일 때 와 NULL이 아닐 때 다른 결과를 만들기 위한 함수

형식

NVL2(data, data가 NULL일 아닐 때 처리, NULL일 때의 처리)

예

```
SELECT ENAME, SAL, COMM, SAL*12+COMM,  
NVL2(COMM,SAL*12+COMM, SAL*12)  
FROM EMP  
ORDER BY JOB;
```

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

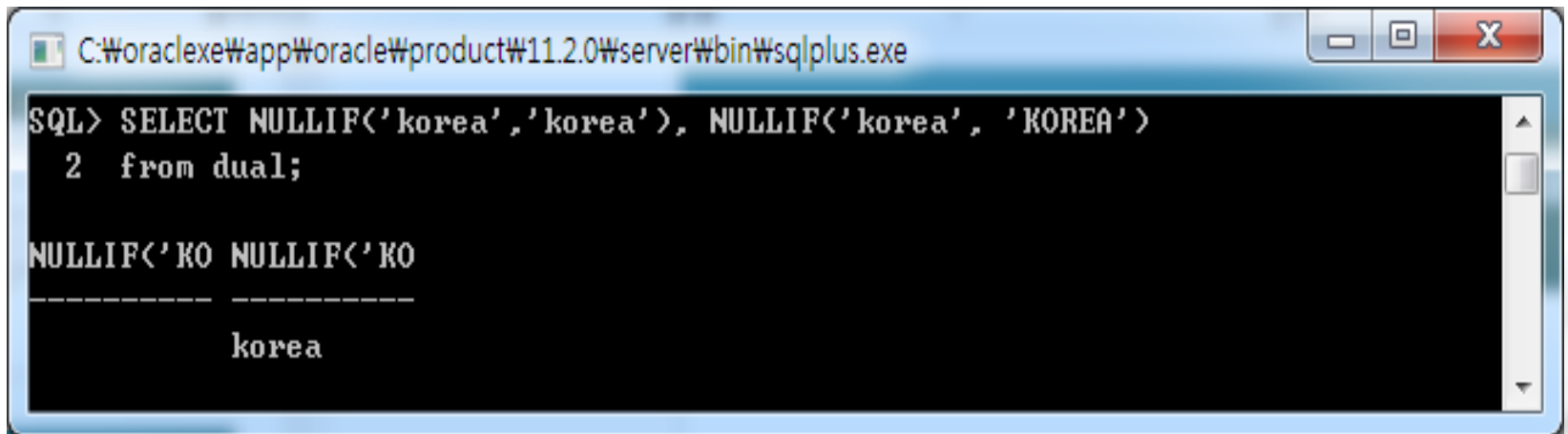
14 개의 행이 선택되었습니다.

NULLIF

형식 NULLIF(*data1*, *data2*)

예

```
SELECT NULLIF('korea','korea'), NULLIF('korea','KOREA')  
from dual;
```



The screenshot shows a Windows command prompt window titled "C:\Oracle\app\Oracle\product\11.2.0\server\bin\sqlplus.exe". The prompt is "SQL>". The user has entered the following SQL statement:

```
SQL> SELECT NULLIF('korea','korea'), NULLIF('korea','KOREA')  
2  from dual;
```

The output of the query is displayed below the prompt:

```
NULLIF('KO NULLIF('KO  
-----  
korea
```

The output shows two columns. The first column contains the value 'korea' (truncated to 10 characters as 'NULLIF('KO'). The second column contains the value 'KOREA' (truncated to 10 characters as 'NULLIF('KO'). The output is formatted with a header line and a separator line.

COALESCE

형식

COALESCE(*data1, data2, ..., datan*)

예

COMM이 NULL이 아니면 COMM을 그렇지 않으면 SAL을 SAL이 NULL이면 0으로 출력
SELECT ENAME, SAL, COMM, COALESCE(COMM, SAL, 0)
FROM EMP;



DECODE

- ❖ DECODE 함수는 프로그램 언어에서 많이 사용되는 switch case 문과 같은 기능
- ❖ 여러 가지 경우에 대해서 선택
- ❖ DECODE 함수의 기본 형식

형식

```
DECODE (표현식, 조건1, 결과1,  
        조건2, 결과2,  
        조건3, 결과3,  
        기본결과n  
)
```



DECODE

- ❖ 부서번호에 해당되는 부서명을 구하는 예제를 CASE 함수를 사용하여 작성

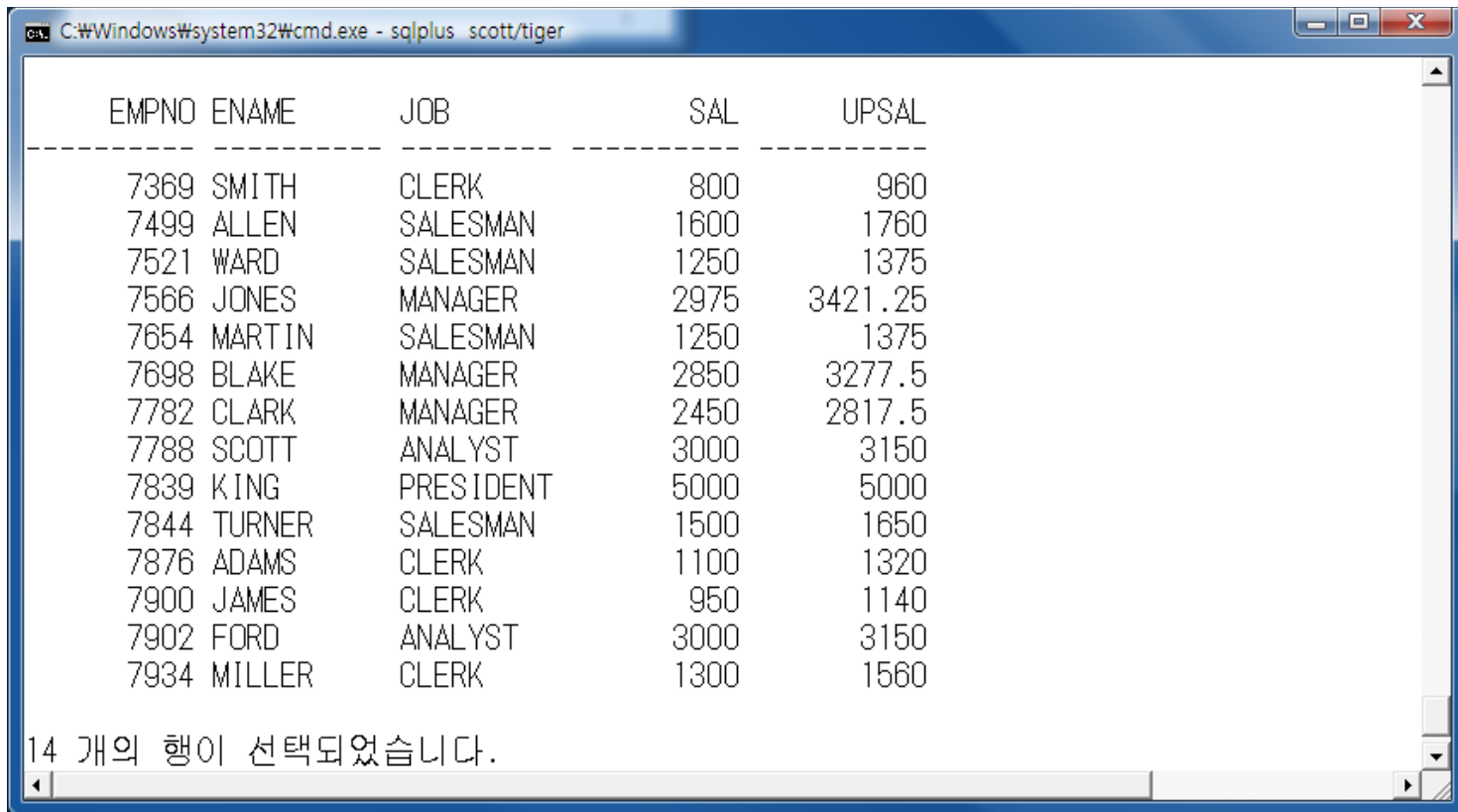
형식

```
SELECT ENAME, DEPTNO,  
       DECODE(DEPTNO, 10, 'ACCOUNTING',  
               20, 'RESEARCH',  
               30, 'SALES',  
               40, 'OPERATIONS' )  
       AS DNAME  
FROM EMP;
```



연습문제

- ❖ emp 테이블에서 empno, ename, job, sal, sal의 값을 JOB이 'ANALYST'인 사원은 5%, 'SALESMAN'인 사원은 10%, 'MANAGER'인 사원은 15%, 'CLERK'인 사원은 20%를 인상한 값을 그 이외의 job을 가진 사원은 sal을 upsal 이라는 헤딩으로 출력



EMPNO	ENAME	JOB	SAL	UPSAL
7369	SMITH	CLERK	800	960
7499	ALLEN	SALESMAN	1600	1760
7521	WARD	SALESMAN	1250	1375
7566	JONES	MANAGER	2975	3421.25
7654	MARTIN	SALESMAN	1250	1375
7698	BLAKE	MANAGER	2850	3277.5
7782	CLARK	MANAGER	2450	2817.5
7788	SCOTT	ANALYST	3000	3150
7839	KING	PRESIDENT	5000	5000
7844	TURNER	SALESMAN	1500	1650
7876	ADAMS	CLERK	1100	1320
7900	JAMES	CLERK	950	1140
7902	FORD	ANALYST	3000	3150
7934	MILLER	CLERK	1300	1560

14 개의 행이 선택되었습니다.

CASE

- ❖ CASE 함수 역시 여러 가지 경우에 대해서 하나를 선택하는 함수
- ❖ DECODE 함수와 차이점이 있다면 DECODE 함수는 조건이 일치(= 비교 연산자)하는 경우에 대해서만 적용되는 반면, CASE 함수는 다양한 비교 연산자를 이용하여 조건을 제시할 수 있으므로 범위를 지정할 수도 있음
- ❖ CASE 함수는 프로그램 언어의 if, else if, else 와 유사한 구조를 가짐

형식

```
CASE 표현식 WHEN 조건1 THEN 결과1  
              WHEN 조건2 THEN 결과2  
              WHEN 조건3 THEN 결과3  
              ELSE 결과n  
END
```



CASE

- ❖ 부서번호에 해당되는 부서명을 구하는 예제를 CASE 함수를 사용하여 작성

예

```
SELECT ENAME, DEPTNO,  
       CASE WHEN DEPTNO=10 THEN 'ACCOUNTING'  
            WHEN DEPTNO=20 THEN 'RESEARCH'  
            WHEN DEPTNO=30 THEN 'SALES'  
            WHEN DEPTNO=40 THEN 'OPERATIONS'  
       END AS DNAME  
FROM EMP;
```



연습문제

1. 현재 날짜를 출력하고 열의 레이블은 Current Date로 출력하는 SELECT 문장을 기술
2. EMP 테이블에서 hiredate가 9월인 사원의 모든 정보를 조회(SUBSTR 함수 이용)
3. EMP 테이블에서 이름(ename), 입사일(hiredate), 입사일로부터 5개월 후 돌아오는 날짜를 구하여 출력하는 SELECT 문장을 기술
4. EMP 테이블에서 이름(ename), 입사일(hiredate), 입사일로부터 현재까지의 지나온 개월 수, 급여(sal), 입사일부터 현재까지의 급여의 총계를 출력하는 SELECT 문장을 기술

5. EMP 테이블에서 다음의 결과가 출력되도록 작성하시오.

Sal의 값이 3배가 되도록 출력

Dream Salary

KING earns \$5,000.00 monthly but wants \$15,000.00

BLAKE earns \$2,850.00 monthly but wants \$8,550.00

CLARK earns \$2,450.00 monthly but wants \$7,350.00

.....

14 rows SELECTed

연습문제

6. EMP 테이블에서 모든 사원의 이름(ename)과 급여(sal)(15자리로 출력하고 좌측의 빈 곳은 “*”로 대치)를 출력하는 SELECT 문장을 기술
7. EMP 테이블에서 모든 사원의 정보를 이름(ename), 업무(job), 입사일(hiredate), 입사한 요일을 출력하는 SELECT 문장을 기술
8. EMP 테이블에서 이름(ename)의 길이가 6자 이상인 사원의 정보를 이름(ename), 이름의 글자 수, 업무(job)를 출력하는 SELECT 문장을 기술
9. EMP 테이블에서 모든 사원의 정보를 이름(ename), 업무(job), 급여(sal), 보너스(comm), 급여+보너스를 출력하는 SELECT 문장을 기술(단 comm의 값은 null 인 경우 0으로 처리)

