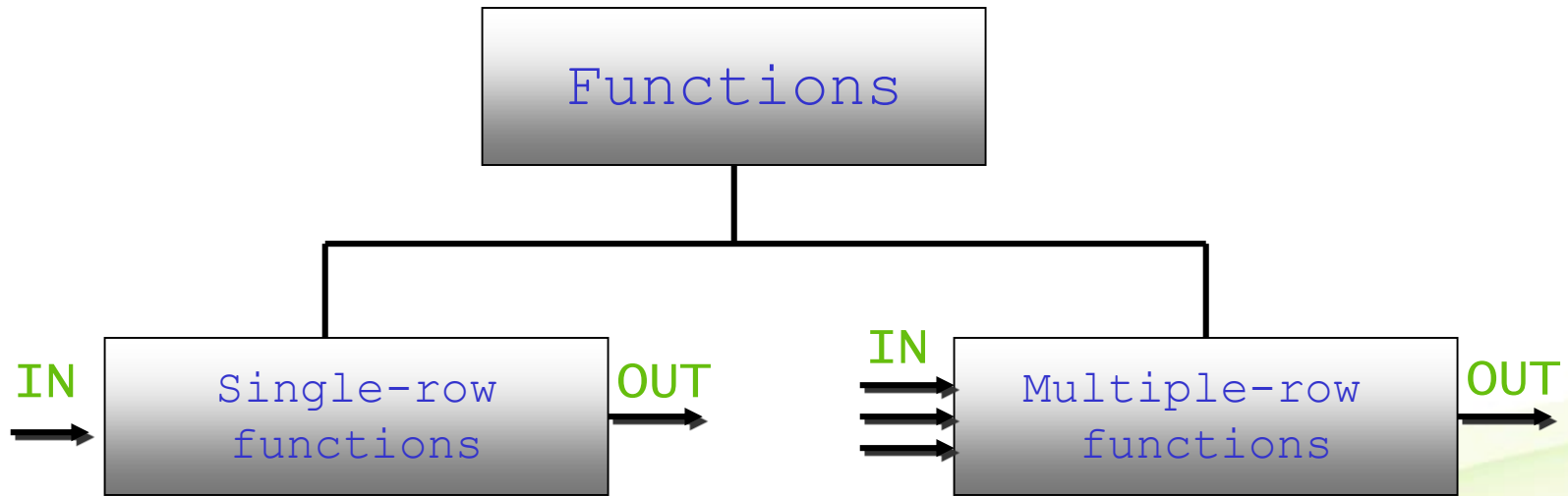


# 단일행 함수

강사 : 강병준

# 단일 행 함수의 두 가지 유형



- 오직 단일 행에서만 적용가능하고 행별로 하나의 결과를 리턴
- 문자, 숫자, 날짜, 형 변환 함수

- 여러 개의 행을 조작하여 행의 그룹당 하나의 결과를 리턴.

# 단일행 함수의 특징

1. 데이터 값을 조작하는데 사용
2. 하나 이상의 인수 (argument) 들을 받고 하나의 결과를 리턴.
3. 단일 행에서만 적용가능하며 행별로 하나의 결과를 리턴.
4. 데이터형 수정과 중첩이 가능.
5. `Select`, `where`, `order by`절에서 사용.
6. 단일 행 함수의 인수는 사용자가 제공하는 상수, 가변적인 값, 열 이름, 표현식 중의 하나가 될 수 있다.

# 단일 행 함수

1. 단일 행 함수는 하나의 행에 대해서만 적용 가능하고 행 별로 하나의 결과를 RETURN합니다.
2. `function_name (column | expression [ ,arg1,arg2, . . . . ])`
  - 1) `function_name` 함수 명
  - 2) `column` 데이터베이스의 Column Name
  - 3) `expression` 어떤 문자 스트링이거나 계산된 표현식
  - 4) `arg1,arg2` 함수에 의해 사용될 수 있는 인수
3. 단일 행 함수가 이용되는 곳
  - 1) 데이터에 대해 계산을 수행할 경우
  - 2) 각각의 데이터 항목을 변경할 경우
  - 3) 출력할 날짜 형식을 변경할 경우
  - 4) Column Data Type을 변경할 경우
4. 단일 행 함수의 종류
  - 1) 문자형 함수 : 문자를 입력 받고 문자와 숫자 값 모두를 RETURN할 수 있습니다.
  - 2) 숫자형 함수 : 숫자를 입력 받고 숫자를 RETURN합니다.
  - 3) 날짜형 함수 : 날짜형에 대해 수행하고 숫자를 RETURN하는 MONTHS\_BETWEEN 함수를 제외하고 모두 날짜 데이터형의 값을 RETURN합니다.
  - 4) 변환형 함수 : 어떤 데이터형의 값을 다른 데이터형으로 변환합니다.
  - 5) 일반적인 함수 : NVL, DECODE

# 문자 함수

## 문자 변환 함수

1. 문자함수 : 대소문자 변환 함수, 문자조작 함수
2. 대소문자 변환함수의 종류
  - LOWER : 알파벳값을 소문자로 변환
  - UPPER : 알파벳값을 대문자로 변환
  - INITCAP : 알파벳 문자값을 각 단어의 첫 글자만 대문자로 변환

## 문자 조작 함수

1. CONCAT : 두개의 매개변수만 사용하여 그 값을 연결
2. SUBSTR : 지정된 길이만큼 문자열을 출력
3. LENGTH : 문자열의 길이를 숫자로 리턴
4. INSTR : 지시된 문자의 위치를 숫자 값으로 리턴
5. LPAD : 문자값을 우측부터 채운다. ↔ RPAD
6. REPLACE : 문자열의 특정 문자를 다른 문자로 변환
7. TRIM : 특정 문자 제거 및 공백 제거

# 문자 함수

- LOWER함수: 대소문자가 혼합되어 있거나 대문자인 문자열을 소문자로 변환
- Syntax            LOWER( column | expression)
- 사용예            LOWER('MANAGER') → manager

EMP 테이블에서 ename이 scott인 데이터의 empno, ename, job(소문자로), deptno를 출력

```
SELECT empno,ename,LOWER(job),deptno
FROM emp
WHERE LOWER(ename) = 'scott';
```

	EMPNO	ENAME	LOWER(JOB)	DEPTNO
1	7788	SCOTT	analyst	20

# 문자 함수

1. UPPER 함수: 대문자가 혼합되어 있거나 소문자인 문자열을 대문자로 변환
2. Syntax            UPPER( column | expression)
3. 사용예            UPPER('manager') → MANAGER

EMP 테이블에서 ename이 scott인 데이터의 empno, ename, job, deptno를 출력

```
SELECT empno,ename,job,deptno  
FROM emp  
WHERE ename = UPPER('scott');
```

	EMPNO	ENAME	LOWER(JOB)	DEPTNO
1	7788	SCOTT	analyst	20

# 문자 함수

1. INITCAP 함수: 각 단어의 첫 번째 문자를 대문자로 나머지 문자는 소문자로 변경
2. Syntax            INITCAP( column | expression)
3. 사 용 예            INITCAP('ORACLE SERVER') → Oracle Server

DEPT 테이블에서 deptno, dname(첫 글자만 대문자로 변환), loc(첫 글자만 대문자로 변환)를 출력하여라.

```
SELECT deptno,INITCAP(dname),INITCAP(loc)
FROM dept;
```

	DEPTNO	INITCAP(DNAME)	INITCAP(LOC)
1	10	Accounting	New York
2	20	Research	Dallas
3	30	Sales	Chicago



# 문자 함수

## CONCAT 함수

두 개의 문자열을 합성합니다. CONCAT는 두개의 매개변수만 사용 가능합니다

Syntax CONCAT( column1 | expression1, column2 | expression2)

사 용 예 CONCA T('ORACLE' , 'SERVER') → ORACLESERVER

```
SQL> col e_name format a15
```

```
SQL> col e_empno format a15
```

```
SQL> col e_job format a15
```

```
SQL> SELECT empno,ename,job,CONCAT(empno,ename) e_name,  
2  CONCAT(ename,empno) e_empno,  
3  CONCAT(ename,job) e_job  
4  FROM emp  
5  WHERE deptno = 10;
```

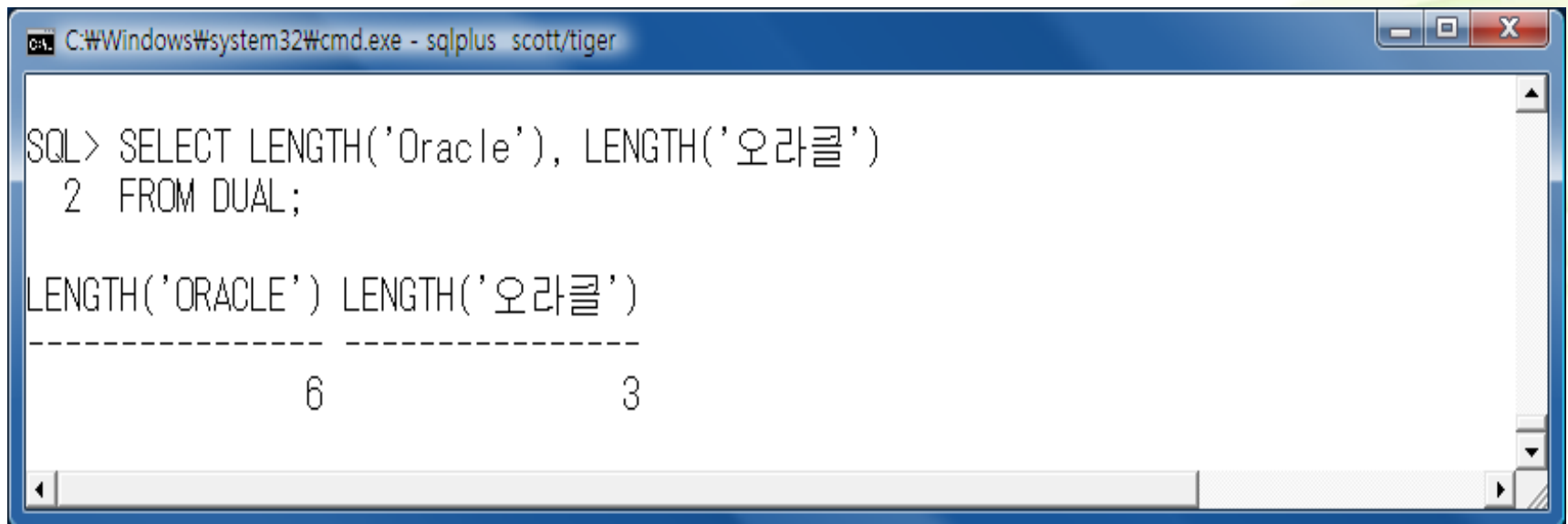
EMPNO	ENAME	JOB	E_NAME	E_EMPNO	E_JOB
7839	KING	PRESIDENT	7839KING	KING7839	KINGPRESIDENT
7782	CLARK	MANAGER	7782CLARK	CLARK7782	CLARKMANAGER
7934	MILLER	CLERK	7934MILLER	MILLER7934	MILLERCLERK

# 문자 길이를 구하는 LENGTH

- LENGTH 함수는 컬럼에 저장된 데이터 값이 몇 개의 문자로 구성되어 있는지 길이를 알려주는 함수입니다.
- 한글도 1글자로 처리합니다.
- 영문자와 한글의 길이를 구해봅시다.

예

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



```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT LENGTH('Oracle'), LENGTH('오라클')  
2  FROM DUAL;  
  
LENGTH('ORACLE') LENGTH('오라클')  
-----  
6                3
```

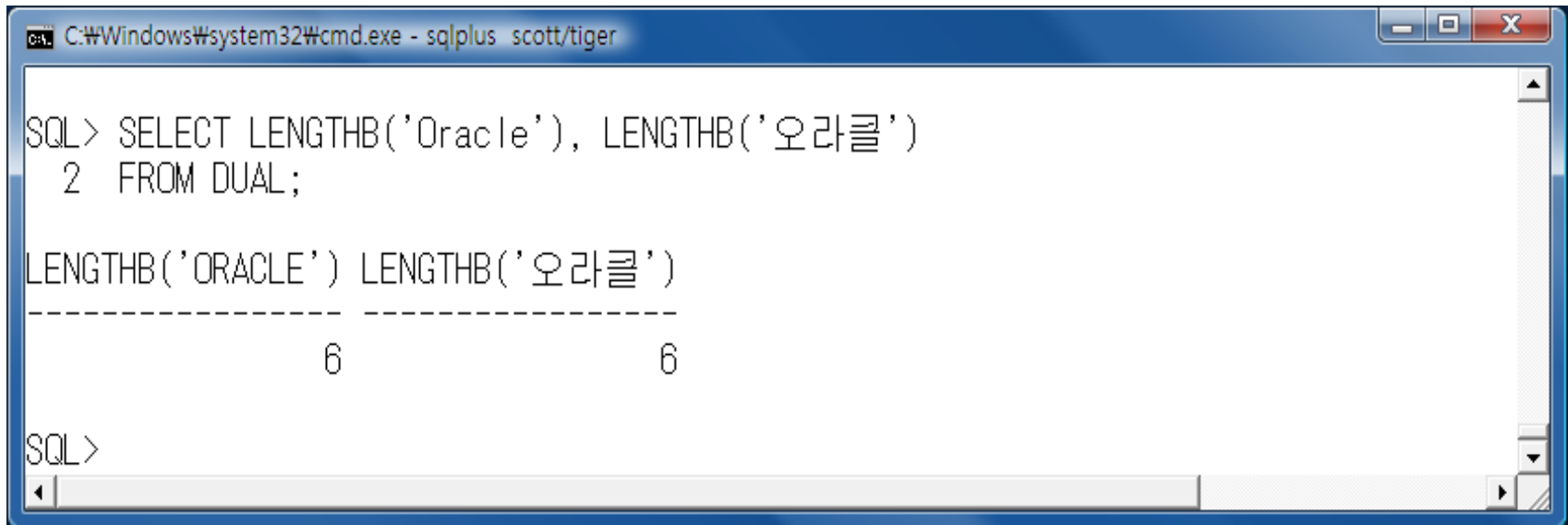
# 바이트 수를 알려주는 LENGTHB 함수

- LENGTHB 함수는 바이트 수를 알려주는 함수입니다.

예

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

- 인코딩 방식에 따라 다른 결과를 가져옵니다.



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

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

The output of the query is displayed as follows:

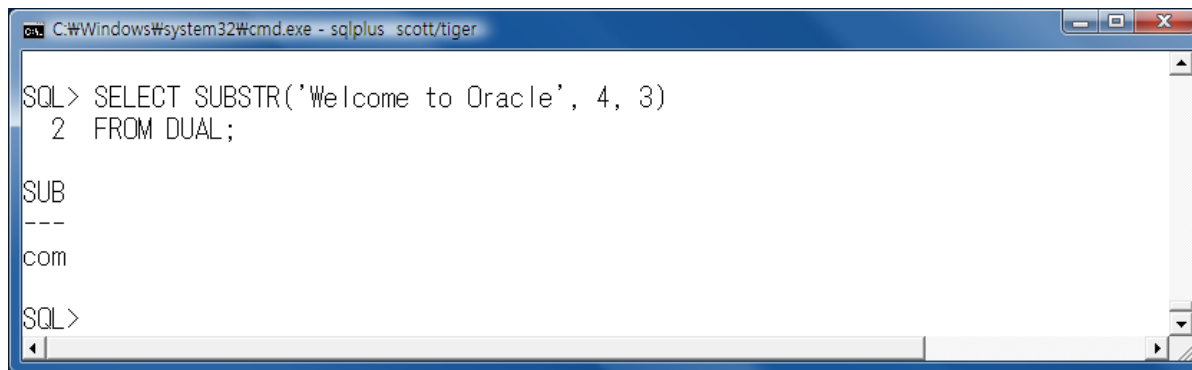
```
LENGTHB('ORACLE') LENGTHB('오라클')  
-----  
6                6
```

The window shows that both 'Oracle' and '오라클' have a byte length of 6 when using the LENGTHB function.

# 문자열 일부만 추출하는 SUBSTR 함수

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

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



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

SQL> SELECT SUBSTR('Welcome to Oracle', 4, 3)
2  FROM DUAL;

SUB
---
com

SQL>
```

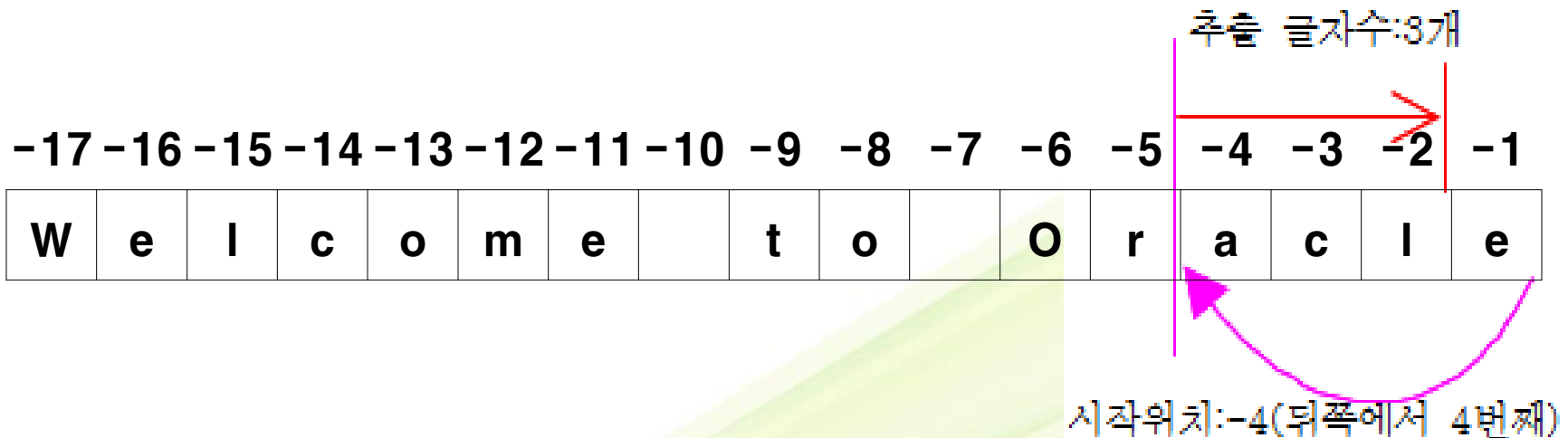
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
W	e	l	c	o	m	e		t	o		O	r	a	c	l	e

# 문자열 일부만 추출하는 SUBSTR 함수

- 시작위치 인자 값을 음수 값으로 줄 수 있는데 이때는 문자열의 앞쪽이 아닌 뒤 쪽에서부터 세어서 시작위치를 잡습니다.

예

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

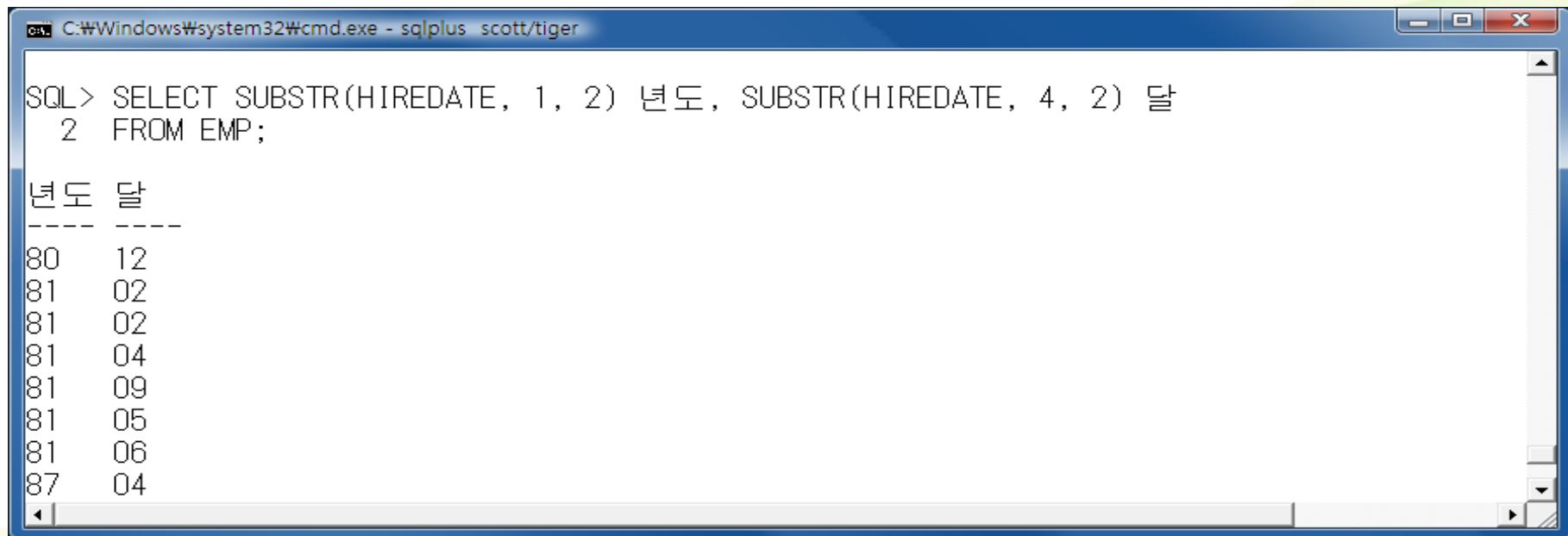


# 문자열 일부만 추출하는 SUBSTR 함수

- 직원들의 입사년도만 출력하려면 SUBSTR 함수를 이용해서 입사일을 저장하고 있는 HIREDATE 컬럼에서 첫 글자부터 2개를 추출하면 됩니다.
- 입사한 달만 출력하려면 HIREDATE 컬럼에서 네번째 글자부터 2개를 추출하면 됩니다.

예

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



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

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

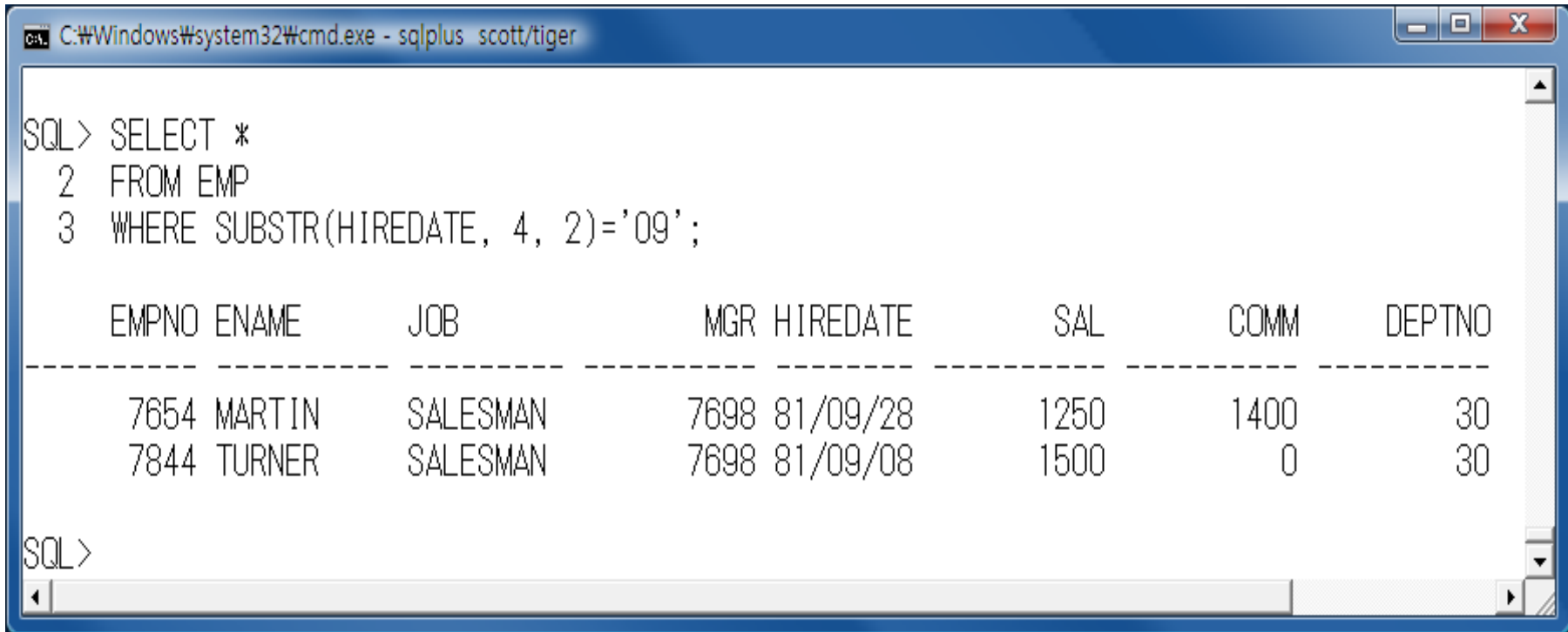
년도	달
80	12
81	02
81	02
81	04
81	09
81	05
81	06
87	04

# 문자열 일부만 추출하는 SUBSTR 함수

- 12월에 입사한 사원을 출력해보시오.

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 left.

# 연습문제

- EMP 테이블에서 Hiredate의 년도가 1987인 사원의 모든 정보를 출력하시오  
(단 substr 함수를 이용해서 where절을 만드시오)
- EMP 테이블에서 ename이 E로 끝나는 사원의 모든 정보를 출력하시오.  
(단 substr 함수를 이용해서 where절을 만드시오)

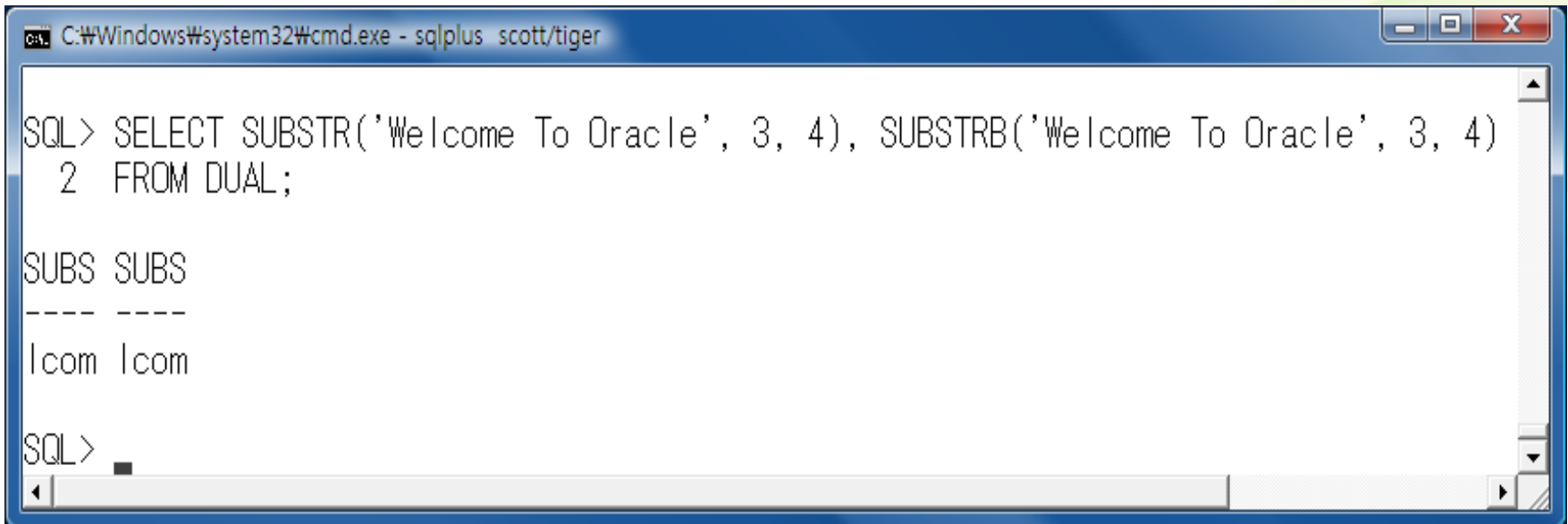


## 바이트 수를 기준으로 문자열 일부만 추출하는 SUBSTRB 함수

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

예

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



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The window shows an SQL query being executed in the SQL\*Plus environment. The query is:   
SQL> SELECT SUBSTR('Welcome To Oracle', 3, 4), SUBSTRB('Welcome To Oracle', 3, 4)  
2 FROM DUAL;  
The output shows two columns, both containing the string "lcom". The first column is labeled "SUBS" and the second is labeled "SUBS". The output is formatted with dashed lines under the column headers. The prompt "SQL>" is visible at the bottom left of the window.

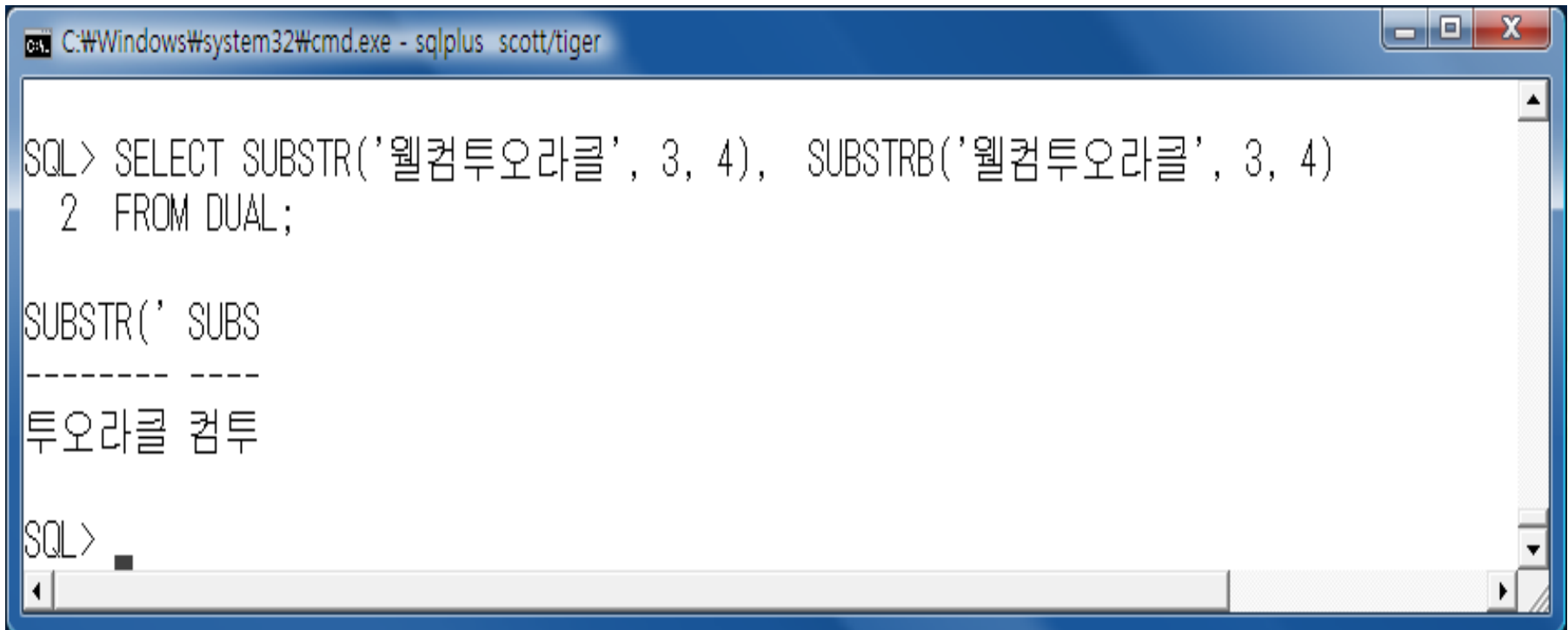
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT SUBSTR('Welcome To Oracle', 3, 4), SUBSTRB('Welcome To Oracle', 3, 4)  
2 FROM DUAL;  
  
SUBS SUBS  
---- ----  
lcom lcom  
  
SQL>
```

## 바이트 수를 기준으로 문자열 일부만 추출하는 SUBSTRB 함수

- 한글 1자는 2바이트 이상을 차지하기 때문에 SUBSTR 함수와 SUBSTRB 함수의 결과가 달라집니다.

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". The user has entered the following SQL query:

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

The output of the query is displayed below the prompt:

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

The output shows that the SUBSTR function returns 'SUBS' (characters 3-6) and the SUBSTRB function returns '투오라클 컴투' (characters 3-10, which is 8 bytes). This demonstrates that SUBSTRB correctly handles multi-byte characters by counting in bytes.

# 문자 함수

## INSTR 함수..

명명된 문자의 위치를 숫자 값으로 RETURN 한다...

Syntax..	INSTR( column   expression, m[,n] )..
사 용 예..	INSTR('MILLER', 'L', 1, 2) → 4..

EMP 테이블에서 이름 중 'L'자의 위치를 출력하여라...

```
SQL> SELECT ename, INSTR(ename, 'L') e_null, INSTR(ename, 'L', 1, 1) e_11, ..
2      INSTR(ename, 'L', 1, 2) e_12, INSTR(ename, 'L', 4, 1) e_41, ..
3      INSTR(ename, 'L', 4, 2) e_42.
4 FROM emp.
5 ORDER BY ename;..
```

ENAME	E_NULL	E_11	E_12	E_41	E_42..
ADAMS	0	0	0	0	0..
ALLEN	2	2	3	0	0..
BLAKE	2	2	0	0	0..
CLARK	2	2	0	0	0..
FORD	0	0	0	0	0..
JAMES	0	0	0	0	0..
JONES	0	0	0	0	0..
KING	0	0	0	0	0..
MARTIN	0	0	0	0	0..
MILLER	3	3	4	4	0..

# 문자 함수

## LPAD 함수

문자값을 우측부터 채웁니다..

Syntax..	LPAD(column   expression, n, 'string')..
사 용 예..	LPAD('MILLER', 10, '*') → *****MILLER..

아래 두 문장의 결과를 보고 차이점을 설명하여라..

```
SQL> SELECT ename,LPAD(ename,15,'*'),sal,LPAD(sal,10,'*')..
2 FROM emp..
3 WHERE deptno = 10;..
```

ENAME	LPAD(ENAME,15,'*')	SAL	LPAD(SAL,10,'*')
KING	*****KING	5000	*****5000..
CLARK	*****CLARK	2450	*****2450..
MILLER	*****MILLER	1300	*****1300..

```
SQL> SELECT deptno,dname,LPAD(dname,20,'*')..
2 FROM dept;..
```

DEPTNO	DNAME	LPAD(DNAME,20,'*')
10	ACCOUNTING	*****ACCOUNTING..
20	RESEARCH	*****RESEARCH..
30	SALES	*****SALES..
40	OPERATIONS	*****OPERATIONS..

# 문자 함수

## RPAD 함수

문자값을 좌측부터 채웁니다...

Syntax	RPAD(column   expression, n, 'string').
사 용 예	RPAD('MILLER', 10, '*') → MILLER*****.

아래 두 문장의 결과를 보고 차이점을 설명하여라.

```
SQL> SELECT ename,RPAD(ename,15,'*'),sal,RPAD(sal,10,'*')
2 FROM emp.
3 WHERE deptno = 10;
```

ENAME	RPAD(ENAME,15,'*')	SAL	RPAD(SAL,10,'*')
KING	KING*****	5000	5000*****
CLARK	CLARK*****	2450	2450*****
MILLER	MILLER*****	1300	1300*****

# 문자 함수

## LTRIM 함수

왼쪽 문자를 지우는 함수 입니다..

Syntax..	LTRIM(column1   expression1, column1   expression1)..	..
사 용 예..	LTRIM('MILLER', 'M') → ILLER..	..

EMP 테이블에서 10 번 부서에 대하여 담당 업무 중 좌측에 'A'를 삭제하고 급여 중 좌측의 1 을 삭제하여 출력하여라. ..

```
SQL> SELECT ename, job, LTRIM(job, 'A'), sal, LTRIM(sal, 1)..  
2 FROM emp;..
```

ENAME	JOB	LTRIM(JOB	SAL	LTRIM(SAL, 1)..
KING	PRESIDENT	PRESIDENT	5000	5000..
BLAKE	MANAGER	MANAGER	2850	2850..
CLARK	MANAGER	MANAGER	2450	2450..
JONES	MANAGER	MANAGER	2975	2975..
MARTIN	SALESMAN	SALESMAN	1250	250..
ALLEN	SALESMAN	SALESMAN	1600	600..
TURNER	SALESMAN	SALESMAN	1500	500..
JAMES	CLERK	CLERK	950	950..
WARD	SALESMAN	SALESMAN	1250	250..
FORD	ANALYST	NALYST	3000	3000..
SMITH	CLERK	CLERK	800	800..
SCOTT	ANALYST	NALYST	3000	3000..
ADAMS	CLERK	CLERK	1100	00..
MILLER	CLERK	CLERK	1300	300..

# 문자 함수

## RTRIM 함수

오른쪽 문자를 지우는 함수 입니다.

Syntax	RTRIM(column1   expression1, column2   expression2)
사 용 예	RTRIM( 'MILLER' , 'R' ) → MILLE

EMP 테이블에서 10번 부서에 대하여 담당 업무 중 우측에 'T'를 삭제하고 급여 중

우측의 0을 삭제하여 출력하여라. .

```
SQL> SELECT ename, job, RTRIM(job, 'T'), sal, RTRIM(sal, 0)
2 FROM emp
3 WHERE deptno = 10;
```

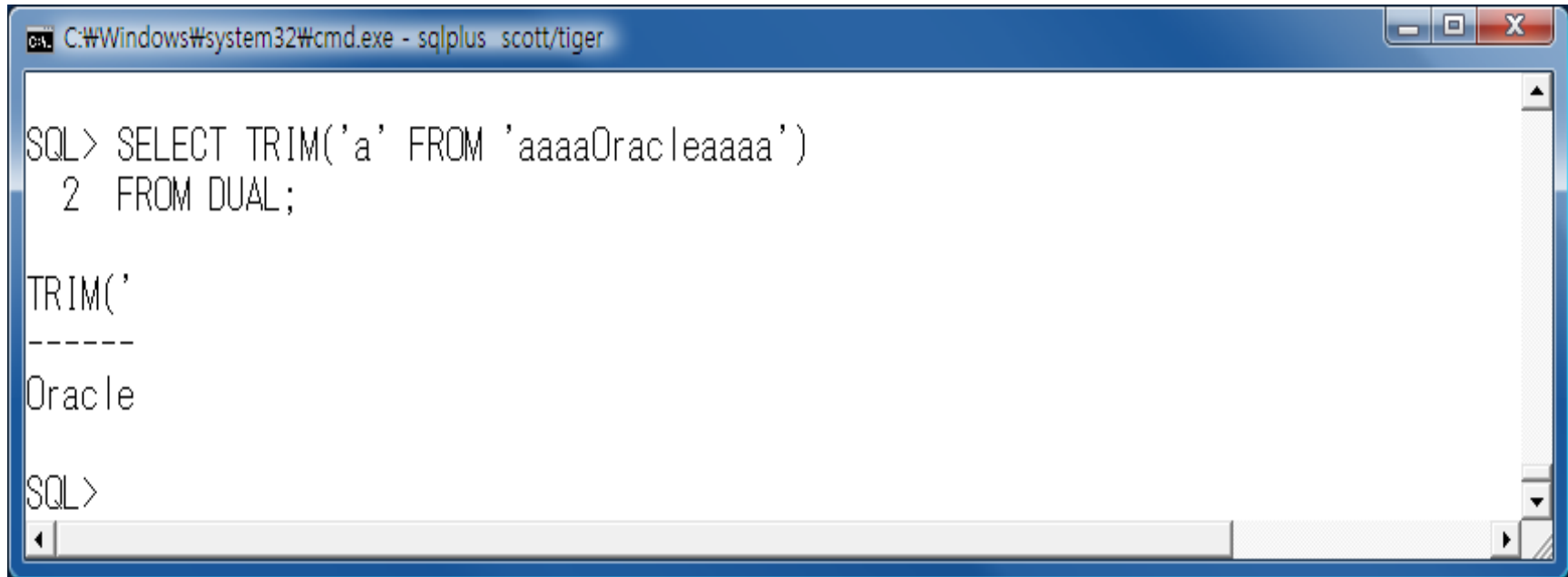
ENAME	JOB	RTRIM(JOB	SAL	RTRIM(SAL, 0)
KING	PRESIDENT	PRESIDEN	5000	5
CLARK	MANAGER	MANAGER	2450	245
MILLER	CLERK	CLERK	1300	13

# 특정 문자를 잘라내는 TRIM 함수

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

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('a'  
-----  
Oracle  
  
SQL>
```

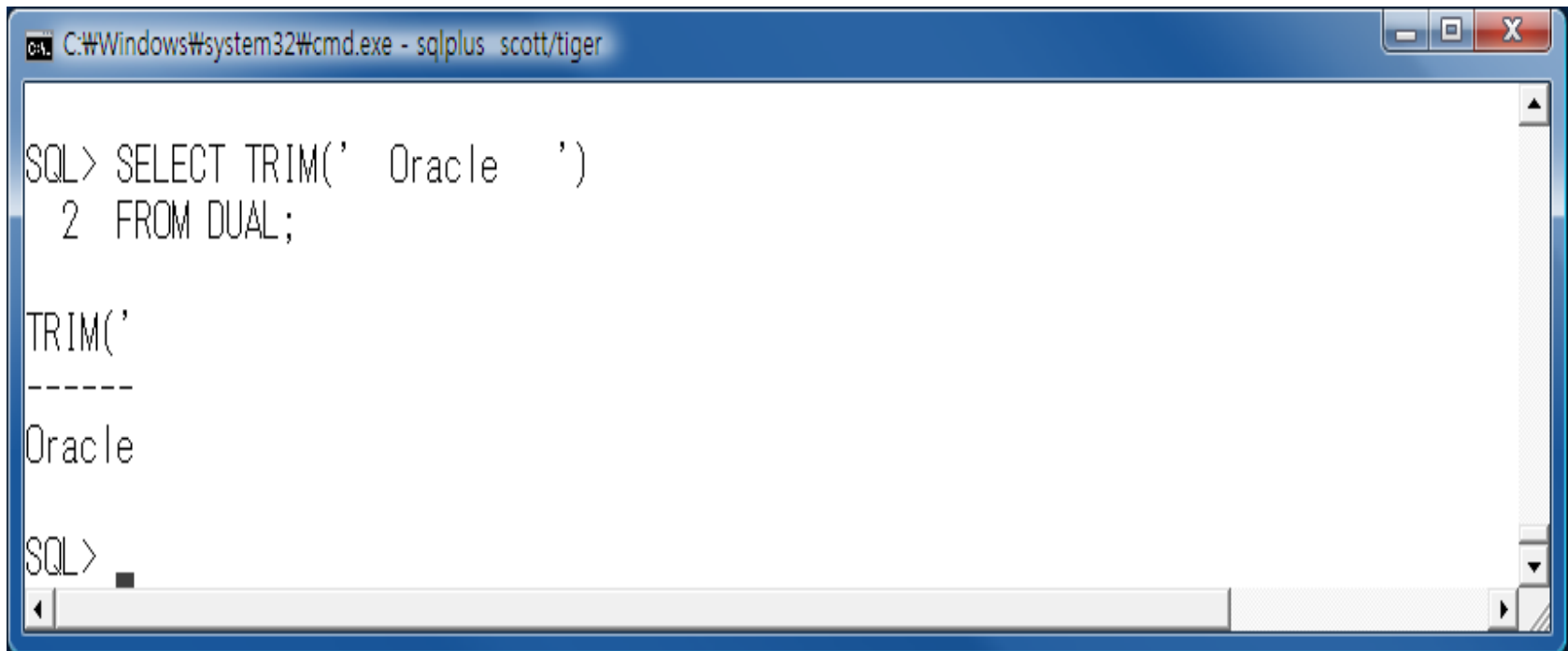


# 특정 문자를 잘라내는 TRIM 함수

- 다음과 같이 기술하면 시작하는 부분과 끝나는 부분의 공백 문자를 잘라냅니다.

예

```
SELECT TRIM(' 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 TRIM(' Oracle ')" followed by a new line and "2 FROM DUAL;". The output shows the result of the TRIM function: "Oracle". The prompt is now "SQL>".

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

# 문자 함수

## TRANSLATE 함수

특정 문자열을 대체하는 함수 입니다. 즉 str1 을 str2 문자로 대체하는 함수이다...

Syntax..	TRANSLATE(column1   expression1, 'string1', 'string2')..
사 용 예..	TRANSLATE('MILLER', 'L', '*') → MI**ER..

EMP 테이블에서 성명을 소문자로 바꾸어 출력하여라...

```
SQL> var u_lower varchar2(10)..
SQL> var n_h varchar2(10)..
SQL> col u_lower format a10..
SQL> col n_h format a10..
SQL> ..
SQL> SELECT empno,ename,TRANSLATE(ename, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ',..
2   'abcdefghijklmnopqrstuvwxyz') u_lower,..
3   sal,TRANSLATE(sal, '0123456789',..
4   '영일이삼사오육칠팔구') n_h..
5 FROM emp..
6 WHERE deptno = 10;..
```

EMPNO	ENAME	U_LOWER	SAL	N_H..
7839	KING	king	5000	오영영영..
7782	CLARK	clark	2450	미사오영..
7934	MILLER	miller	1300	일삼영영..

# 문자 함수

## REPLACE 함수

특정 문자열을 대신하는 함수 입니다.

Syntax	REPLACE(column1   expression1, 'string1' , 'string2' )
사 용 예	REPLACE( 'JACK and JUE' , 'J' , 'BL' ) → BLACK and BLUE

EMP 테이블에서 JOB 에 'A'를 '\$'로 바꾸어 출력하여라...

```
SQL> SELECT ename, job, REPLACE(job, 'A', '$'), sal..  
2 FROM emp;..
```

ENAME	JOB	REPLACE(J	SAL..
KING	PRESIDENT	PRESIDENT	5000..
BLAKE	MANAGER	M\$N\$GER	2850..
CLARK	MANAGER	M\$N\$GER	2450..
JONES	MANAGER	M\$N\$GER	2975..
MARTIN	SALESMAN	S\$LESM\$N	1250..

```
. . . . .  
14 rows selected..
```

# 숫자 함수

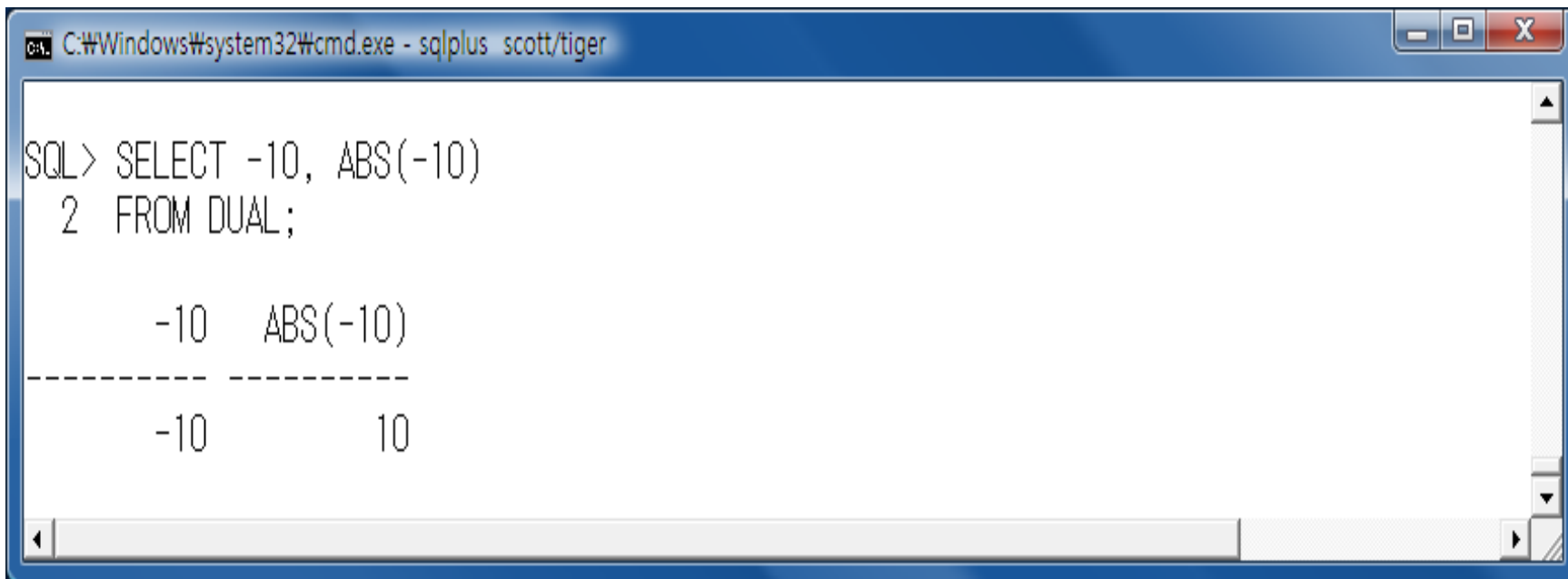
1. ABS (n) : ABS함수는 절대값을 계산
2. CEIL (n) : CEIL함수는 주어진 값보다는 크지만 가장 근접하는 최소값을 리턴
3. EXP (n) : EXP함수는 주어진 값의 e의 승수를 나타냄  
(e = 2.171828183...)
4. FLOOR (n) : FLOOR함수는 주어진 값보다 작거나 같은 최대 정수 값을 구함
5. LN (n) : LN함수는 주어진 값의 자연로그 값을 반환
6. MOD (m, n) : MOD함수는 m을 n으로 나누어 남은 값을 반환. n이 0일 경우 m을 반환
7. POWER (m, n) : POWER함수는 m의 n승 값을 계산
8. ROUND : 명시된 소수점으로 반올림 값.  
ROUND (column/expression, n)  
열, 표현식의 값을 소수점 n자리까지 반올림
9. n이 생략 또는 0이면 소수점이 없어진다.
10. TRUNC : 명시된 소수점으로 절삭 값.  
TRUNC (column/expression, n)  
열, 표현식의 값을 소수점 n자리까지 절삭

# 절대값 구하는 ABS 함수

- ABS 함수는 절대값을 구합니다. 절대값은 방향은 없고 크기만 있는 것으로서 주어진 데이터가 음수일 경우 양수로 표현합니다.
- -10 에 대한 절대값을 구해 봅시다.

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".

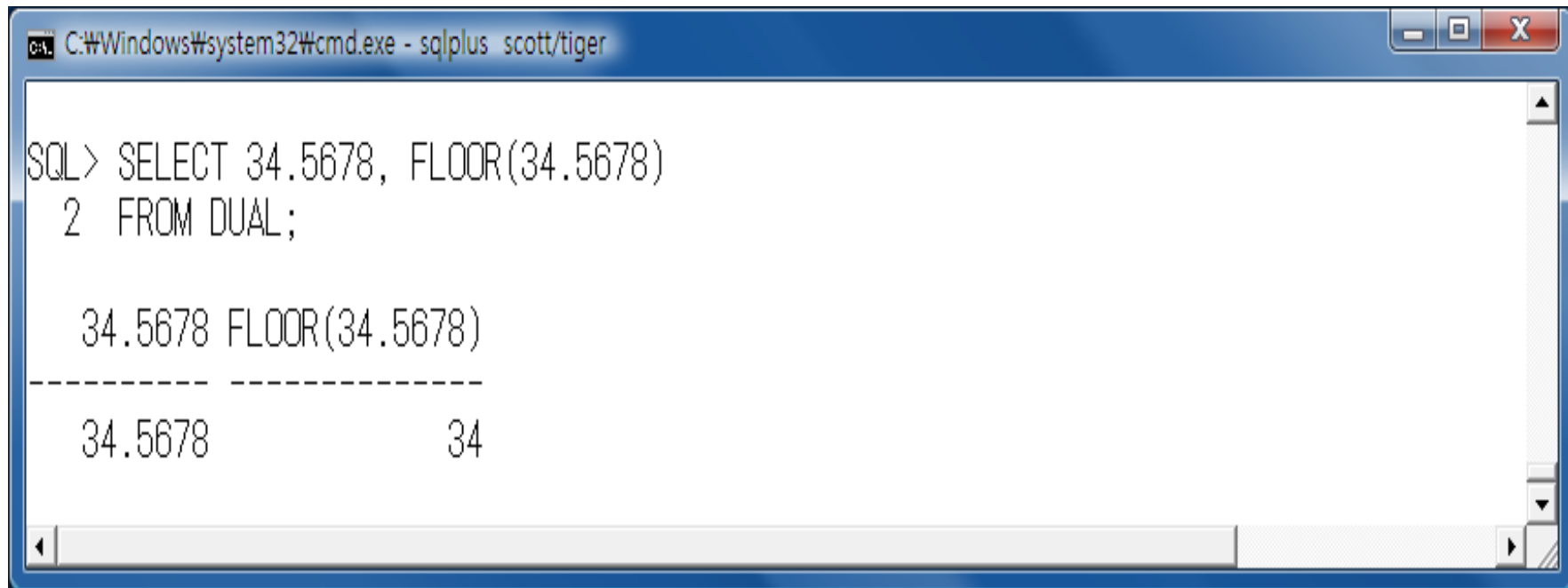
```
SQL> SELECT -10, ABS(-10)  
2  FROM DUAL;  
  
-10  ABS(-10)  
-----  
-10      10
```

# 소수점 아래를 버리는 FLOOR 함수

- FLOOR 함수는 소수점 아래를 버립니다. 34.5678를 FLOOR 함수에 적용하면 34가 구해집니다.
- 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 SQL query: "SELECT 34.5678, FLOOR(34.5678) FROM DUAL;". The output is displayed as follows:

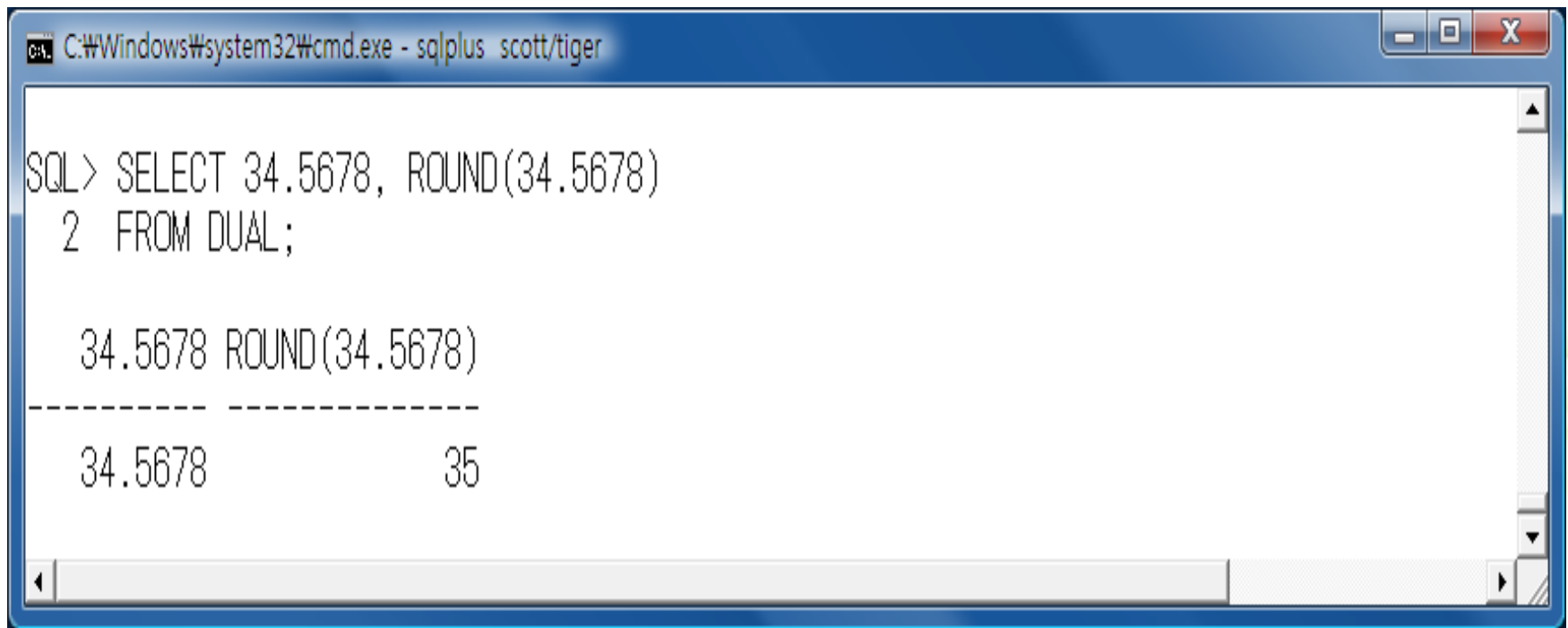
```
34.5678 FLOOR(34.5678)
-----
34.5678          34
```

# 반올림하는 ROUND 함수

- 34.5678를 반올림하면 35입니다. 이와 같이 반올림한 결과를 구하기 위한 함수로 오라클에서는 ROUND가 제공됩니다.

예

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

34.5678	ROUND(34.5678)
34.5678	35

# 반올림하는 ROUND 함수

- ROUND 함수는 지정한 자릿수에서 반올림을 할 수 있도록 자릿수를 지정할 수 있습니다.

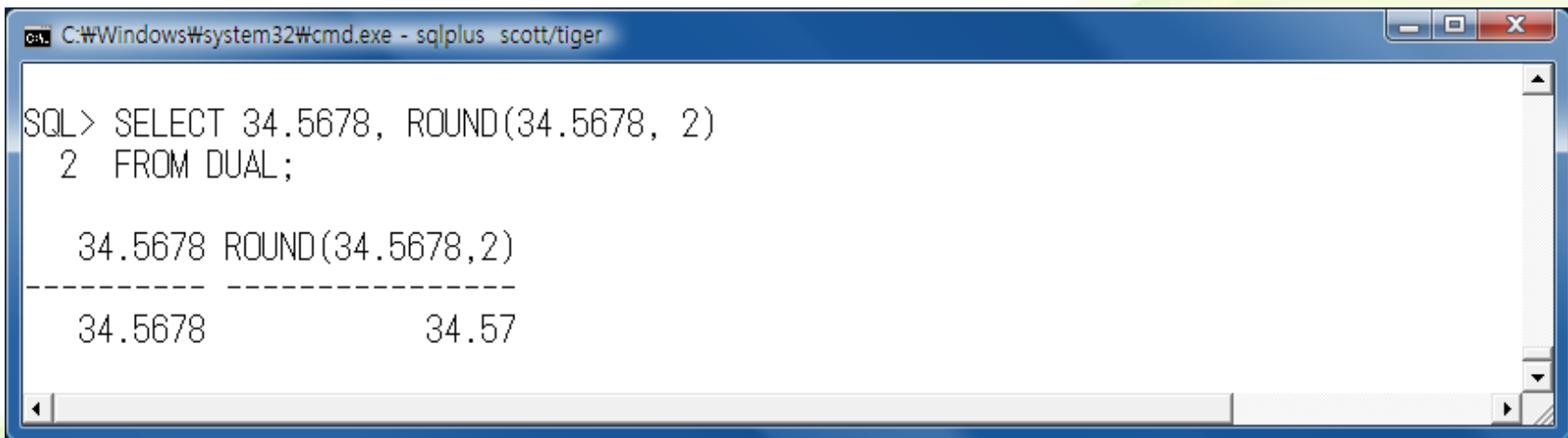
형식

ROUND(대상, 자릿수)

- ROUND 함수의 두 번째 인자 값이 2이면 소수점 이하 세 번째 자리에서 반올림하여 소수점 이하 두 번째 자리까지 표시합니다. 즉, ROUND(34.5678, 2)의 결과는 34.57이 됩니다.

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 query "SELECT 34.5678, ROUND(34.5678, 2) FROM DUAL;". The output shows two columns: "34.5678" and "ROUND(34.5678,2)". A dashed line separates the header from the data row, which shows the values "34.5678" and "34.57".

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

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

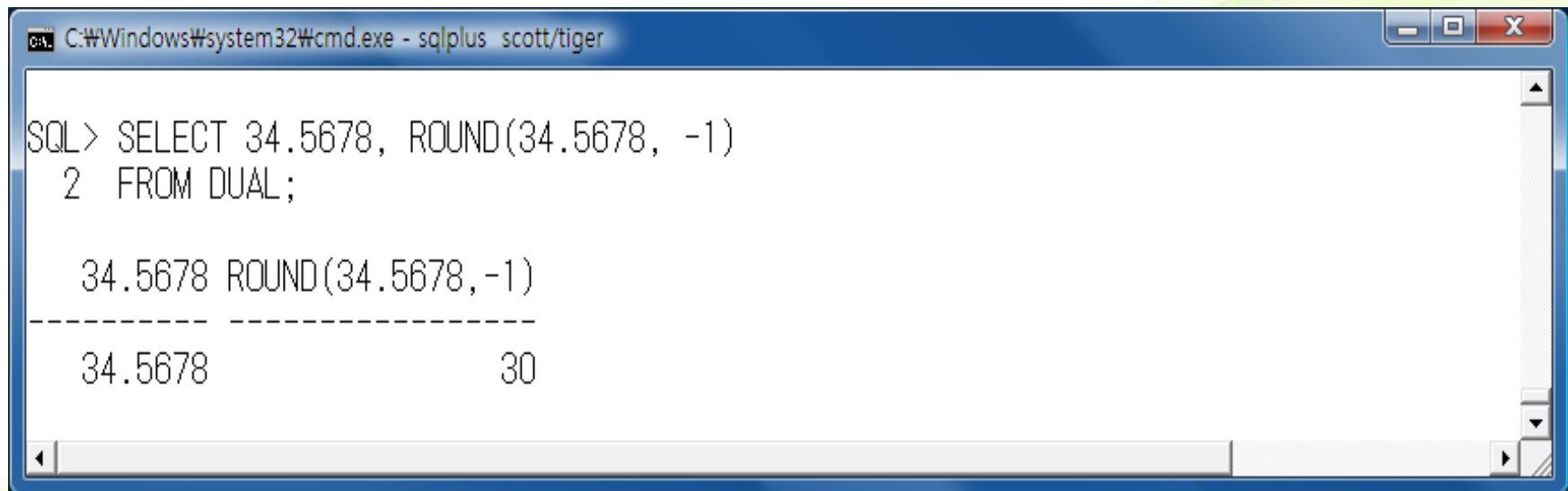


# 반올림하는 ROUND 함수

- 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 일어나게 됩니다.
- ROUND(34.5678, -1)의 결과는 30이 됩니다.

예

```
SELECT 34.5678, ROUND(34.5678, -1)
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, -1) FROM DUAL;". The output shows two columns: "34.5678" and "ROUND(34.5678,-1)". A dashed line separates the header from the data row, which shows the values "34.5678" and "30".

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

SQL> SELECT 34.5678, ROUND(34.5678, -1)
      2  FROM DUAL;

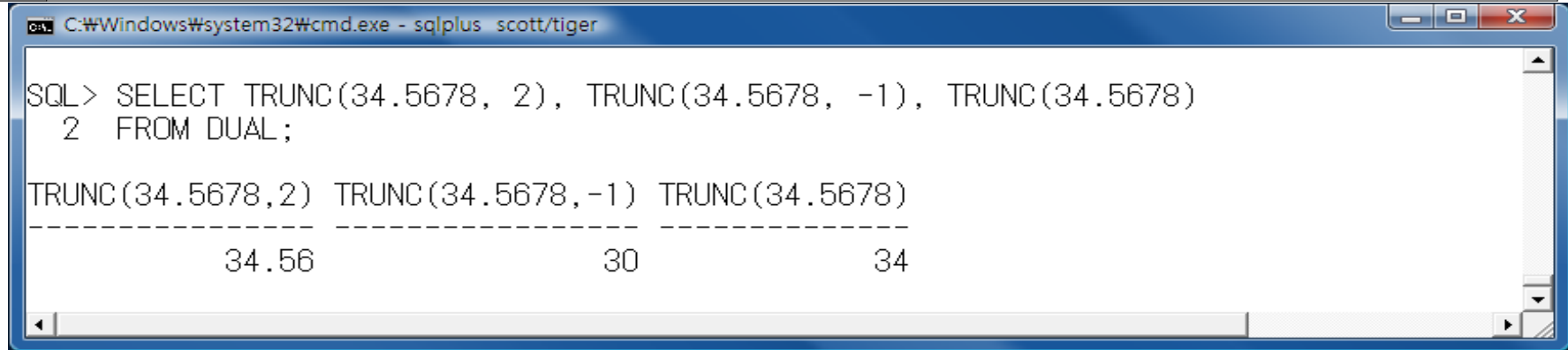
      34.5678 ROUND(34.5678,-1)
-----
      34.5678              30
```

# 특정 자릿수에서 잘라내는 TRUNC 함수

- TRUNC 함수는 지정한 자리 수 이하를 버린 결과를 구해주는 함수입니다.

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 SQL query: "SELECT TRUNC(34.5678, 2), TRUNC(34.5678, -1), TRUNC(34.5678) FROM DUAL;". The output is displayed as follows:

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 표시합니다.
- 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 수행합니다.

# 숫자 함수

## MOD 함수

숫자의 나머지를 구하는 함수입니다.

Syntax	MOD(column1   expression1 , n)
사 용 예	MOD(10, 3) → 1

EMP 테이블에서 급여를 30으로 나눈 나머지를 구하여 출력하여라

SQL> **SELECT sal, MOD(sal,30)**

2 **FROM emp**

3 **WHERE deptno = 10;**

SAL	MOD(SAL,30)
-----	-----
5000	20
2450	20
1300	10

# 숫자 함수

## POWER 함수

거듭제곱을 구하는 함수 입니다...

Syntax	POWER(column1   expression1 , n)
사 용 예	POWER(2, 3) → 8

## SQRT 함수

제곱근을 구하는 함수 입니다...

Syntax	SQRT( column1   expression1 )
사 용 예	SQRT(4) → 2

## SIGN 함수

주어진 숫자가 양수인지 음수인지 또는 0 인지를 구하는 함수 입니다...

Syntax	SIGN( column1   expression1 )
사 용 예	SIGN(100) → 1

# 숫자 함수

## CHR 함수

ASCII Code 값에 해당하는 문자를 구하는 함수 입니다...

Syntax	CHR( column1   expression1 )
사 용 예	CHR(65) → A

EMP 테이블에서 20 번 부서 중 이름과 담당 업무를 연결하여 출력하여라. 단 담당 업무를 한 줄 아래로 출력하여라.

```
SQL> SELECT empno,ename,job,ename || CHR(10) || job FROM emp
3 WHERE deptno = 20;
```

EMPNO	ENAME	JOB	ENAME    CHR(10)    JOB
7566	JONES	MANAGER	JONES MANAGER
7902	FORD	ANALYST	FORD ANALYST
7369	SMITH	CLERK	SMITH CLERK
7788	SCOTT	ANALYST	SCOTT ANALYST

# 날짜 함수

- ORACLE의 날짜 형식
  - 세기, 년, 월, 일, 시간, 분, 초의 내부 숫자 형식
  - 기본 날짜 형식 : 'DD-MON-YY'
  - 날짜 범위 : January 1, 4712 B.C와 December 31, 9999 A.D. 사이
  - SYSDATE : 현재의 날짜와 시간을 리턴
  - DUAL 테이블
    - `SYS에 의해 소유되며 모든 사용자가 액세스 가능
    - `DUMMY라는 하나의 열과 x값을 가지는 하나의 행
    - `오직 한번만 값을 리턴 하고자 할 때 유용

# 날짜 함수

- 데이터베이스는 날짜를 숫자로 저장하므로 산술 연산 적용.

작업	결과	설 명
날짜 + 숫자	날짜	일 수를 날짜에 더한다.
날짜 - 숫자	날짜	날짜에서 일 수를 뺀다.
날짜 - 날짜	일 수	어떤 날짜에서 다른 날짜를 뺀다.
날짜 + 숫자/24	날짜	시간을 24로 나누어서 날짜에 더한다.

# 날짜 함수

- 날짜 함수
  - `MONTHS_BETWEEN (date1, date2)`
    - Date1과 date2사이의 월 수를 리턴
  - `ADD_MONTHS (date, n)`: 월 수 n을 date에 더한다.
  - `NEXT_DAY (date, 'char')`
    - date다음의 명시된 요일 ('char')의 날짜를 찾는다.
  - `LAST_DAY (date)`
    - date를 포함하는 월의 마지막 날을 찾는다.
  - `ROUND (date [, 'fmt' ] )`
    - fmt에 명시된 단위에 대해 반올림한 date를 리턴
  - `TRUNC (date [, 'fmt' ] )`
    - fmt에 명시된 단위에 대해 절삭한 date를 리턴

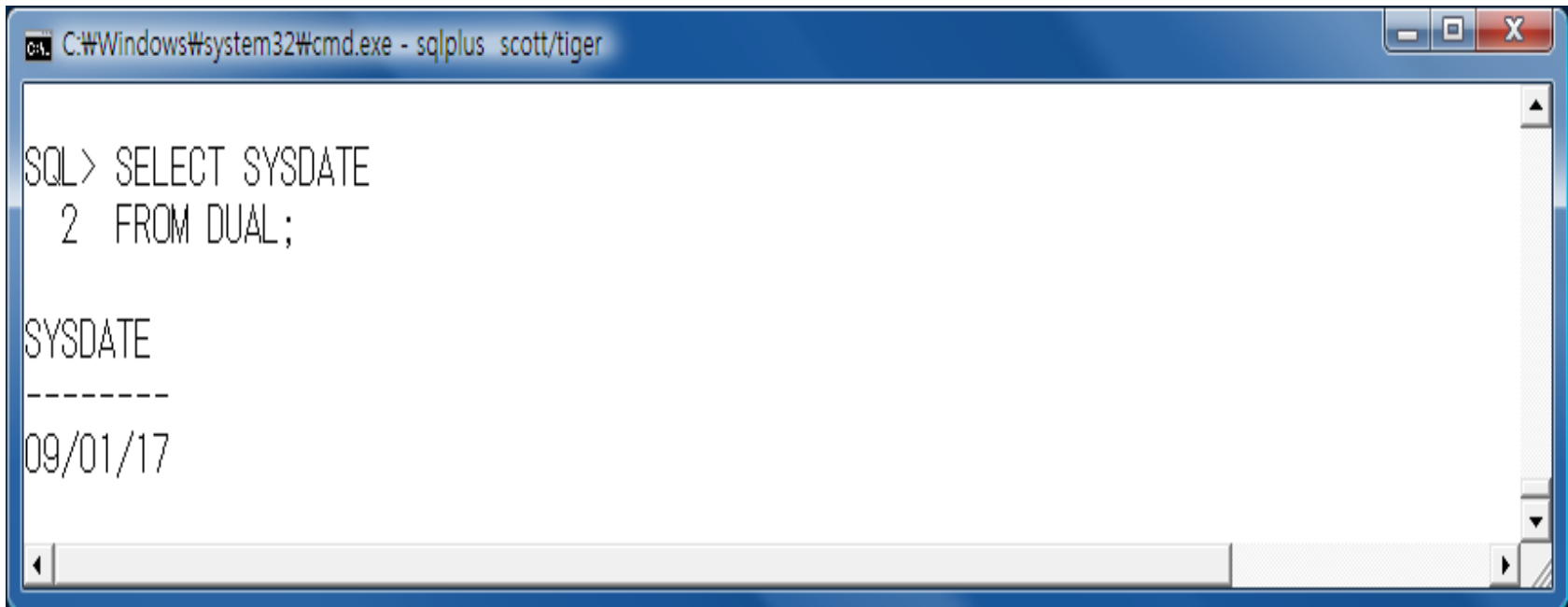


# 현재 날짜를 반환하는 SYSDATE 함수

- SYSDATE 함수는 시스템에 저장된 현재 날짜를 반환하는 함수입니다. 다음은 시스템에서 현재 날짜를 얻어 와서 출력하는 예제

예

```
SELECT SYSDATE  
FROM DUAL;
```



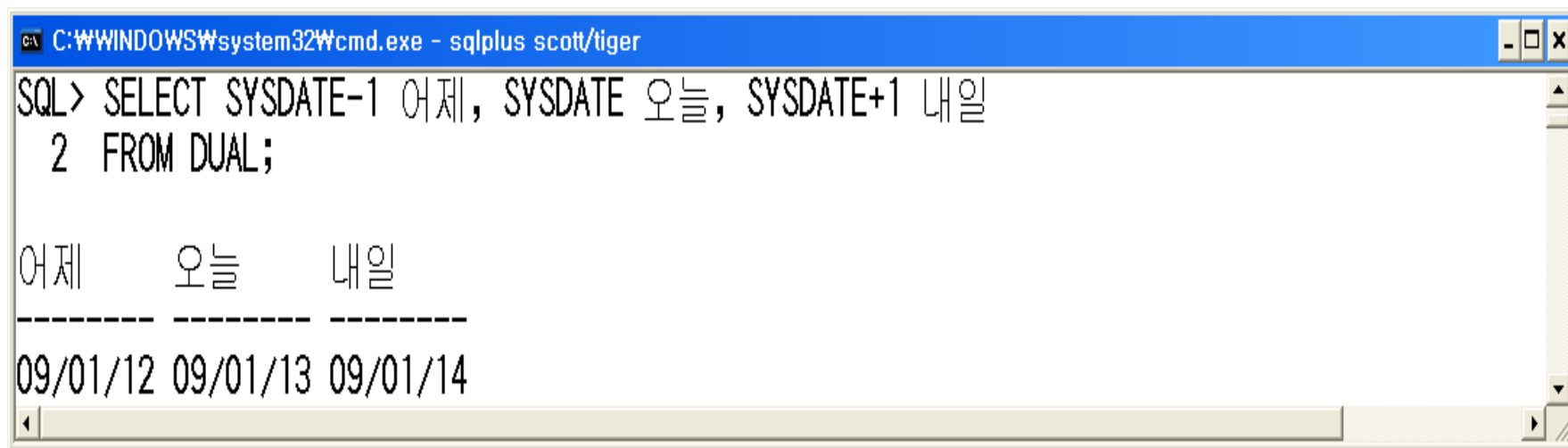
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT SYSDATE  
2  FROM DUAL;  
  
SYSDATE  
-----  
09/01/17
```

# 날짜 연산

- 날짜 형 데이터에 숫자를 더하면(날짜+숫자) 그 날짜로부터 그 기간만큼 지난 날짜를 계산합니다. 날짜 형 데이터에 숫자를 빼면(날짜-숫자) 그 날짜로부터 그 기간만큼 이전 날짜를 구합니다. (1은 하루에 해당합니다.)

예

```
SELECT SYSDATE-1 어제, SYSDATE 오늘, SYSDATE+1 내일  
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 SYSDATE-1 어제, SYSDATE 오늘, SYSDATE+1 내일  
2 FROM DUAL;". The output is displayed as a table with three columns: "어제", "오늘", and "내일". The dates shown are "09/01/12", "09/01/13", and "09/01/14" respectively.

```
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
```

# 날짜 함수

EMP 테이블에서 현재까지 근무일 수가 몇 주 /몇 일 인가를 출력하여라. 단 근무 일수가 많은 사람 순으로 출력하여라.

```
SQL> SELECT ename,hiredate,sysdate,sysdate - hiredate "Total Days",  
2 TRUNC((sysdate - hiredate) / 7, 0) Weeks,  
3 ROUND(MOD((sysdate - hiredate), 7), 0) DAYS,  
4 FROM emp,  
5 ORDER BY sysdate - hiredate DESC;
```

ENAME	HIREDATE	SYSDATE	Total Days	WEEKS	DAYS
SMITH	17-DEC-80	01-MAR-99	6648.5677	949	6
ALLEN	20-FEB-81	01-MAR-99	6583.5677	940	4
WARD	22-FEB-81	01-MAR-99	6581.5677	940	2
JONES	02-APR-81	01-MAR-99	6542.5677	934	5
BLAKE	01-MAY-81	01-MAR-99	6513.5677	930	4
CLARK	09-JUN-81	01-MAR-99	6474.5677	924	7
TURNER	08-SEP-81	01-MAR-99	6383.5677	911	7
MARTIN	28-SEP-81	01-MAR-99	6363.5677	909	1
KING	17-NOV-81	01-MAR-99	6313.5677	901	7
JAMES	03-DEC-81	01-MAR-99	6297.5677	899	5
FORD	03-DEC-81	01-MAR-99	6297.5677	899	5

.....  
14 rows selected..

# 날짜 함수

EMP 테이블에서 10번 부서 중 현재까지의 근무 월수를 계산하여 출력하여라.

```
SQL> SELECT      ename,hiredate,SYSDATE,MONTHS_BETWEEN(SYSDATE,hiredate)
m_between,
2 TRUNC(MONTHS_BETWEEN(SYSDATE,hiredate),0) t_between,
3 FROM emp,
4 WHERE deptno = 10,
5 ORDER BY MONTHS_BETWEEN(SYSDATE,hiredate) DESC;
```

ENAME	HIREDATE	SYSDATE	M_BETWEEN	T_BETWEEN
CLARK	09-JUN-81	10-FEB-99	212.04812	212
KING	17-NOV-81	10-FEB-99	206.79005	206
MILLER	23-JAN-82	10-FEB-99	204.5965	204

# 날짜 함수

## ADD\_MONTHS 함수

- 1) 날짜에 월을 더합니다(ADD\_MONTHS(hiredate,10)).
- 2) 날짜에 월을 뺍니다(ADD\_MONTHS(hiredate,-10)).
- 3) 결과의 날짜형입니다...

Syntax	ADD_MONTHS(date1, n)
사 용 예	ADD_MONTHS(hiredate,5) → 23-JUN-82

EMP 테이블에서 10 번 부서 중 입사 일자로부터 5 개월이 지난 후 날짜를 계산하여 출력하여라...

SQL> SELECT ename, hiredate, ADD_MONTHS(hiredate,5) a_month,		
2 FROM emp,		
3 WHERE deptno = 10,		
4 ORDER BY hiredate DESC;		
ENAME	HIREDATE	A_MONTH
-----	-----	-----
MILLER	23-JAN-82	23-JUN-82
KING	17-NOV-81	17-APR-82
CLARK	09-JUN-81	09-NOV-81

# 날짜 함수

## NEXT\_DAY 함수..

- 1) 명시된 요일의 돌아오는 날짜를 계산 합니다..
- 2) 요일이 아니라 숫자도 기술 가능(SUNDAY:1, MONDAY:2, . . . . .).
- 3) NLS\_LANG 이 KOREAN\_KOREA.KO16KSC5601 로 되어 있으면 한글도 사용 가능(일요일, 월요일, 화요일, . . . . .).

Syntax..	NEXT_DAY(date1, 'string'   n )..
사 용 예..	NEXT_DAY(hiredate, 'FRIDAY') → 29-JAN-82.. NEXT_DAY(hiredate, '금요일') → 29-JAN-82..

EMP 테이블에서 10 번 부서 중 입사 일자로부터 돌아오는 금요일을 계산하여 출력

☞ 하더라..

```
SQL> SELECT ename, hiredate, NEXT_DAY(hiredate, 'FRIDAY') n_day,..  
2      NEXT_DAY(hiredate, 6) n_6, NEXT_DAY(hiredate, 7) n_7..  
3 FROM emp..  
4 WHERE deptno = 10..  
5 ORDER BY hiredate DESC;..
```

ENAME	HIREDATE	N_DAY	N_6	N_7..
MILLER	23-JAN-82	29-JAN-82	29-JAN-82	30-JAN-82..
KING	17-NOV-81	20-NOV-81	20-NOV-81	21-NOV-81..
CLARK	09-JUN-81	12-JUN-81	12-JUN-81	13-JUN-81..

# 날짜 함수

## LAST\_DAY 함수

- 1) 월의 마지막 날짜를 계산.
- 2) 윤년, 평년은 자동 계산.

Syntax	LAST_DAY(date)
사 용 예	LAST_DAY(hiredate) → 30-NOV-81

EMP 테이블에서 입사한 달의 근무 일수를 계산하여 출력하여라. 단 토요일과 일

+ 요일도 근무 일수에 포함한다...

```
SQL> SELECT empno,ename,hiredate, LAST_DAY(hiredate) l_last,
2 LAST_DAY(hiredate) - hiredate l_day
3 FROM emp
4 ORDER BY LAST_DAY(hiredate) - hiredate DESC;
```

EMPNO	ENAME	HIREDATE	L_LAST	L_DAY
7698	BLAKE	01-MAY-81	31-MAY-81	30
7566	JONES	02-APR-81	30-APR-81	28
7900	JAMES	03-DEC-81	31-DEC-81	28
7902	FORD	03-DEC-81	31-DEC-81	28
7844	TURNER	08-SEP-81	30-SEP-81	22
7788	SCOTT	09-DEC-82	31-DEC-82	22
7782	CLARK	09-JUN-81	30-JUN-81	21
7876	ADAMS	12-JAN-83	31-JAN-83	19
7369	SMITH	17-DEC-80	31-DEC-80	14
7839	KING	17-NOV-81	30-NOV-81	13

# 특정 기준으로 반올림하는 ROUND

- ROUND 함수는 포맷 모델을 지정하면 숫자 이외에 날짜에 대해서도 반올림을 할 수 있습니다.

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	한주가 시작되는 날짜
MI	분을 기준



# 특정 기준으로 반올림하는 ROUND

- ROUND 함수의 포맷 모델로 MONTH를 지정하였기에 특정 날짜 (DATE)를 달(MONTH)을 기준으로 반올림한 날짜를 구합니다.
- 일을 기준으로 16일보다 적으면 이번 달 1일을 크면 다음달 1일을 구합니다. 6월 9일 반올림하며 6월 1일이 되고 11월 17일은 반올림 하여 12월 1일이 됩니다. 또한 1월 23일을 반올림하면 2월 1일이 됩니다.
- 다음은 입사일을 달을 기준으로 반올림한 예제입니다.

예

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

# 날짜 함수

## TRUNC 함수

- 1) 명시된 형식으로 절삭 합니다...
- 2) 날짜를 가장 가까운 년도 또는 월로 절삭할 수 있습니다. ...
  - ① fmt 에 명시된 단위에 대해 절삭한 날짜를 계산.
  - ② fmt 가 생략되면 날짜를 가장 가까운 날짜로 절삭한다...

Syntax	TRUNC(date1 [,fmt] )
사 용 예	TRUNC('25-JUN-99','MONTH') → 01-JUN-99. TRUNC('25-JUN-98','YEAR') → 01-JAN-98.

EMP 테이블에서 10 번 부서 중 입사한 달의 ROUND 과 TRUNC 함수를 비교합니다...

```
SQL> SELECT ename,hiredate,ROUND(hiredate,'MONTH') m_round,
2 TRUNC(hiredate,'MONTH') m_trunc, ROUND(hiredate,'YEAR') y_round,
3 TRUNC(hiredate,'YEAR') y_trunc,
4 FROM emp,
5 WHERE deptno = 10,
6 ORDER BY hiredate DESC;
```

ENAME	HIREDATE	M_ROUND	M_TRUNC	Y_ROUND	Y_TRUNC
MILLER	23-JAN-82	01-FEB-82	01-JAN-82	01-JAN-82	01-JAN-82
KING	17-NOV-81	01-DEC-81	01-NOV-81	01-JAN-82	01-JAN-81
CLARK	09-JUN-81	01-JUN-81	01-JUN-81	01-JAN-81	01-JAN-81

# 변환 함수



# 형 변환 함수

- 암시적 데이터형 변환
  - 값 할당 시, **ORACLE**서버가 자동으로 변환
    - - VARCHAR2 or CHAR를 NUMBER로
    - - NVARCHAR2 or CHAR를 DATE로
    - - NUMBER를 VARCHAR2로
    - - DATE를 VARCHAR2로
  - 표현식 계산의 경우
    - - VARCHAR2 or CHAR를 NUMBER로
    - - VARCHAR2 or CHAR를 DATE로
  - 주의 사항
    - - CHAR가 NUMBER로의 변환은 오직 문자 string이 적절한 숫자일 때
    - - CHAR가 DATE로의 변환은 문자 string이 기본 형식 **[NLS\_DATE\_FORMAT]**를 가질 경우에만 성공

# 변환 함수

- 명시적 데이터형 변환

- `TO_CHAR`(*number* | *date* [, '*fmt*'] )

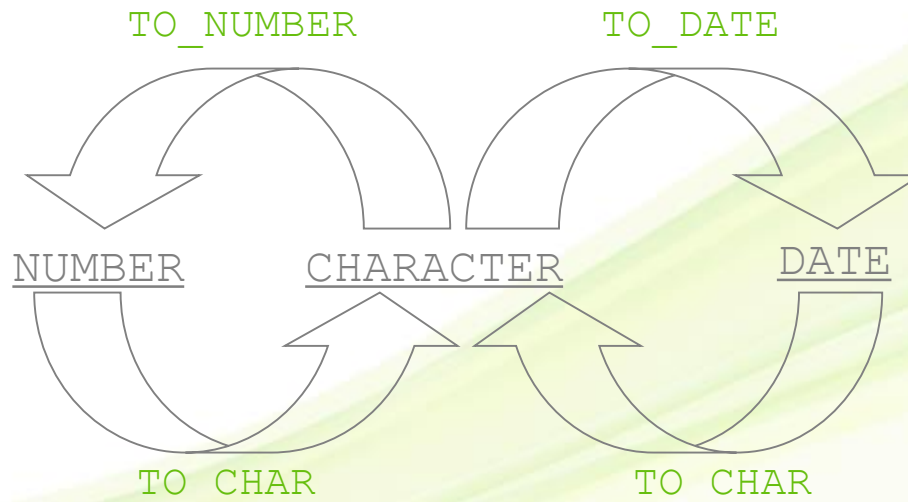
- 숫자나 문자값을 포맷 모델 *fmt*를 사용하여 `VARCHAR2`문자로 변환

- `TO_NUMBER`(*char*)

- 숫자를 포함하는 문자 스트링을 숫자로 변환

- `TO_DATE`(*char* [, '*fmt*'] )

- 날짜를 나타내는 문자로 명시된 *fmt*에 따라 날짜로 변환



# 날짜를 가진 TO\_CHAR 함수

```
TO_CHAR(date, 'fmt')  

```

- 포맷 모델:
  - 단일 인용부호로 둘러싸여 있어야 하고 대소문자를 구분
  - 어떤 타당한 날짜 형식 요소도 포함할 수 있다.
  - 추가된 공백을 제거하거나 앞부분의 0을 없애기 위한 fm요소를 가질 수 있다.
  - 데이터 값을 콤마로 구분

# 날짜 형식 요소

날짜 형식	설명	
YYYY	네자리 연도	2006년 = 2006
YY	두자리 연도	2006년 = 01
MM	달 수	12월 = 12
MON	세자리 영문 달 이름	12월 = DEC
DDD	한 해 단위에서의 날짜 수	12월 30일 = 364
DD	한 달 단위에서의 날짜 수	3월 23일 = 23
D	일 주일 단위에서의 날짜수	토요일 = 7
DY	세자리 영문 요일 이름	일요일 = SUN
HH (HH12)	12시간으로 표현되는 시간	오후 2시 = 2
HH24	24시간으로 표현되는 시간	오후 2시 = 14
MI	분	
SS	초	

# 변환 함수

## 시간 형식

- 시간 요소는 날짜의 시간 부분을 형식화(HH24:MI:SS AM → 15:34:32 PM)
- 문자열에 이중 인용 부호를 사용하여 문자열을 추가  
(DD “of” MONTH→10 of OCTOBER)
- 숫자 접미사는 숫자를 문자로 변환(ddsph → fourteenth)
- 시간 형식의 종류

요 소	설 명
AM or PM	정오 지시자
A.M or P.M	있 있는 정오 지시자
HH or HH12 or HH24	하루 중 시간(1-12, 0-23)
MI	분(0-59)
SS	초(0-59)
SSSSS	자정 이후의 초(0-86399)

## 기타 형식

요 소	설 명
/	사용 문자가 결과에 다시 나타난다
“of the”	인용 부호내의 문자가 결과에 출력



# 변환 함수

숫자에 영향을 주는 접미사

요 소	설 명
TH	서수(DDTH → 4TH)
SP	명시한 수(DDSP → FOUR)
SPTH or THSP	명시한 서수(DDSPTH → FOURTH)

EMP 테이블에서 10 번 부서 중 입사 일자를 '1 May 1981'와 '1998 년 1 월 1 일'의  
형태로 출력하여라.

```
SQL> var t_hiredate varchar2(30);
SQL> var t_kor varchar2(20);
SQL> col t_hiredate format a30;
SQL> col t_kor format a20;
SQL> SELECT ename,hiredate,TO_CHAR(hiredate, 'fmDD Month YYYY') t_hiredate,
2  TO_CHAR(hiredate, 'YYYY"년" MM"월" DD"일 "') t_kor,
3  FROM emp,
4  WHERE deptno = 10,
5  ORDER BY hiredate DESC;
```

ENAME	HIREDATE	T_HIREDATE	T_KOR
MILLER	23-JAN-82	23 January 1982	1982 년 01 월 23 일
KING	17-NOV-81	17 November 1981	1981 년 11 월 17 일
CLARK	09-JUN-81	9 June 1981	1981 년 06 월 09 일

# 변환 함수

## 숫자 형식을 변환

TO\_CHAR 함수를 사용하여 숫자 값을 문자로 출력하기 위해 사용한다...

Syntax	TO_CHAR( number, 'fmt' )	1
사 용 예	TO_CHAR(sal, '\$999,999') → \$3,000	1

## 숫자를 가진 TO\_CHAR 함수

- 1) 숫자 값을 문자로 변환할 때 즉 NUMBER 형을 VARCHAR2 로 전환할 때
- 2) 이 기법은 연결(Concatenation) 시에 유용

### Guidelines

- 1) 형식에 의해 제공되는 자릿수를 초과하는 숫자에 대해서는 "##"을 출력
- 2) 지정된 소수 값을 형식에서 제공하는 소수점 자리로 반올림 한다

# 변환 함수

## 숫자 형식 모델

요 소	설 명	예	결 과
9	9의 수는 출력 폭을 결정	999999	1234
0	무효의 0을 출력	099999	001234
\$	달러 기호	\$999999	\$1234
L	지역 화폐 기호	L999999	¥1234
.	명시한 위치에 소수점	999999.99	1234.00
,	명시한 위치에 콤마	999,999	1,234
M	우측에 마이너스 기호(음수 값)	999999M	1234-
PR	음수를 “( )”로 묶는다	999999PR	<1234>
EEEE	과학적인 부호 표기	99.999EEEE	1.234E+03
V	10을 n번 곱합니다	9999V99	123400
B	0을 0이 아닌 공백으로 출력	B9999.99	1234.00

# 변환 함수

EMP 테이블에서 부서 20중 급여 앞에 \$를 삽입하고 3자리마다 ,를 출력하여라

```
SQL> SELECT empno,ename,job,sal,TO_CHAR(sal,'$999,999')
2 FROM emp
3 WHERE deptno = 20
4 ORDER BY sal DESC;
```

EMPNO	ENAME	JOB	SAL	TO_CHAR(S
-----	-----	-----	-----	-----
7902	FORD	ANALYST	3000	\$3,000
7788	SCOTT	ANALYST	3000	\$3,000
7566	JONES	MANAGER	2975	\$2,975
7876	ADAMS	CLERK	1100	\$1,100
7369	SMITH	CLERK	800	\$800

# 변환 함수

## TO\_NUMBER 함수

숫자를 포함하는 문자열을 숫자로 변환 합니다...

Syntax	TO_NUMBER( char )
사 용 예	TO_NUMBER('1234') → 1234

## TO\_DATE 함수

날짜를 나타내는 문자열을 명시된 날짜로 변환 합니다...

Syntax	TO_DATE( char [, 'fmt' ] )
사 용 예	TO_DATE('19990220181030', 'YYYYMMDDHH24MISS') → 1999/02/20 18:10:30

February 22, 1981 에 입사한 사원의 정보를 이름, 업무, 입사일자를 출력하여라...

```
SQL> SELECT ename, job, TO_CHAR(hiredate, 'Month DD, YYYY') t_hire
2 FROM emp
3 WHERE hiredate = TO_DATE('February 22, 1981', 'Month DD, YYYY');
```

ENAME	JOB	T_HIRE
WARD	SALESMAN	February 22, 1981

# 기타 함수

## ■ CASE

CASE 함수는 SQL 문장 내에서 IF-THEN-ELSE와 같은 흐름제어문의 역할을 한다.  
CASE 함수의 문법은 다음과 같다.

```
CASE expr WHEN comparison_expr1 THEN return_expr1
      [WHEN comparison_expr2 THEN return_expr2
      WHEN comparison_exprn THEN return_exprn
      ELSE else_expr]
END
```

```
SQL> SELECT EMPNO, ENAME, SAL, JOB,
2      CASE JOB WHEN 'ANALYST' THEN SAL*1.1
3              WHEN 'CLERK' THEN SAL*1.2
4              WHEN 'MANAGER' THEN SAL*1.3
5              WHEN 'PRESIDENT' THEN SAL*1.4
6              WHEN 'SALESMAN' THEN SAL*1.5
7              ELSE SAL
8      END "급여"
9      FROM EMP;
```

EMPNO	ENAME	SAL	JOB	급여
7369	SMITH	800	CLERK	960
7499	ALLEN	1600	SALESMAN	2400

# 기타 함수

## DECODE 함수

☞ CASE 나 IF-THEN-ELSE-END IF 문장의 조건적 조회를 가능하게 함.

Syntax.	DECODE(col   expr,search1,result1[,search2,result2,...][,default]).
사 용 예.	DECODE(deptno, 10, sal*1.1, 20, sal*1.5, 30, sal*1.2, sal).

EMP 테이블에서 JOB 이 ANALYST 이면 급여 증가는 10%이고 JOB 이 CLERK 이면 급여 증가는 15%이고 JOB 이 MANAGER 이면 급여 증가는 20%입니다. 다른 업무에 대해서는 급여 증가가 없습니다. 직원번호, 이름, 업무, 급여, 증가된 급여를 출력하여라...

```
SQL> SELECT empno,ename,job,sal,DECODE(job,'ANALYST',sal*1.1,
2  'CLERK',sal*1.15,'MANAGER',sal*1.2,sal) d_sal
3  FROM emp
4  ORDER BY sal DESC;
```

EMPNO	ENAME	JOB	SAL	D_SAL
7839	KING	PRESIDENT	5000	5000
7902	FORD	ANALYST	3000	3300

.....

# 기타 함수

## 중첩 함수

- 1) 단일행 함수는 여러 LEVEL 에 걸쳐 중첩 가능..
- 2) 중첩 함수는 가장 하위 LEVEL 에서 상위 LEVEL 순으로 진행..

Syntax..	F3( F2( F1(col,arg1), arg2), arg3)..
사 용 예..	NVL(TO_CHAR(mgr), 'No Manager')..

다음의 결과를 분석하여 보아라...

```
SQL> col t_rpad format a20..
SQL> col r_r format a20..
SQL> SELECT deptno,dname,RPAD(dname,20,'*') t_rpad,..
       2  RPAD(RTRIM(dname),20,'*') r_r,loc..
       3  FROM dept;..
```

DEPTNO	DNAME	T_RPAD	R_R	LOC..
10	ACCOUNTING	ACCOUNTING	***** ACCOUNTING*****	NEW YORK..
20	RESEARCH	RESEARCH	***** RESEARCH*****	DALLAS..
30	SALES	SALES	***** SALES*****	CHICAGO..
40	OPERATIONS	OPERATIONS	***** OPERATIONS*****	BOSTON..



# 기타 함수

## EXTRACT

EXTRACT는 입력값인 날짜 데이터 타입에서 지정된 값을 추출하는 함수이다. EXTRACT의 문법은 다음과 같다.

```
SELECT EXTRACT ([YEAR] [MONTH] [DAY]  
FROM [datetime_value_expression]  
      [interval-value_expression]);
```

SYSDATE 함수의 결과값으로부터 년도를 추출하는 방법은 다음과 같다.

```
SQL> SELECT EXTRACT (YEAR FROM SYSDATE) FROM DUAL;  
  
EXTRACT(YEARFROMSYSDATE)  
-----  
2004
```

사원 테이블에서 10번 부서에 근무하는 직원의 입사월을 출력하면 다음과 같다.

```
SQL> SELECT ENAME, HIREDATE, EXTRACT (MONTH FROM HIREDATE)  
2 FROM EMP  
3 WHERE DEPTNO = 10;
```

ENAME	HIREDATE	EXTRACT(MONTHFROMHIREDATE)
CLARK	81/06/09	6
KING	81/11/17	11
MILLER	82/01/23	1

## ◆ 연습 문제 ◆

1. 현재 날짜를 출력하고 열 레이블은 Current Date로 출력하는 SELECT 문장을 기술하시오.
2. EMP 테이블에서 현재 급여에 15%가 증가된 급여를  
사원번호, 이름, 업무, 급여, 증가된 급여(New Salary), 증가액(Increase)를  
출력하는 SELECT 문장을 기술하시오.
3. EMP 테이블에서 이름, 입사일, 입사일로부터 6개월 후 돌아오는  
월요일 구하여 출력하는 SELECT 문장을 기술하시오.
4. EMP 테이블에서 이름, 입사일, 입사일로부터 현재까지의 월수, 급여,  
입사일부터 현재까지의 급여의 총계를 출력하는 SELECT 문장을  
기술하시오.
5. EMP 테이블에서 모든 사원의 이름과 급여(15자리로 출력 좌측의  
빈곳은 “\*”로 대치)를 출력하는 SELECT 문장을 기술하시오.
6. EMP 테이블에서 모든 사원의 정보를 이름, 업무, 입사일, 입사한  
요일을 출력하는 SELECT 문장을 기술하시오.
7. EMP 테이블에서 이름의 길이가 6자 이상인 사원의 정보를  
이름, 이름의 글자수, 업무를 출력하는 SELECT 문장을 기술하시오.
8. EMP 테이블에서 모든 사원의 정보를  
이름, 업무, 급여, 보너스, 급여+보너스를 출력하는 SELECT 문장을  
기술하시오.

# 연습문제

1. 사원 테이블의 사원명에서 2번째 문자부터 3개의 문자를 추출하시오.
2. 사원 테이블에서 입사일이 12월인 사원의 사번, 사원명, 입사일을 검색하시오.
3. 다음과 같은 결과를 검색할 수 있는 SQL 문장을 작성하시오

EMPNO	ENAME	<u>급여</u>
7369	SMITH	*****800
7499	ALLEN	*****1600
7521	WARD	*****1250
...		
7934	MILLER	*****1300

14 개의 행이 선택되었습니다.

4. 다음과 같은 결과를 검색할 수 있는 SQL 문장을 작성하시오

EMPNO	ENAME	<u>입사일</u>
7369	SMITH	1980-12-17
7499	ALLEN	1981-02-20
7521	WARD	1981-02-22
...		
7934	MILLER	1982-01-23

14 개의 행이 선택되었습니다.

# 연습문제

5. 사원 테이블에서 급여에 따라 사번, 이름, 급여, 등급을 검색하는 SQL 문장을 작성 하시오. (Hint : CASE 함수 사용)

<u>급여</u>	<u>등급</u>
0~1000	E
1001~2000	D
2001~3000	C
3001~4000	B
4001~5000	A

EMPNO	ENAME	SAL	
7369	SMITH	800	E
7499	ALLEN	1600	D
7521	WARD	1250	D
7566	JONES	2975	C
...			
7934	MILLER	1300	D

14 개의 행이 선택되었습니다.

# 연습문제

6. 사원 테이블에서 급여에 따라 사번, 이름, 급여, 등급을 검색하는 SQL 문장을 작성 하시오 (case. decode)

급여	등급
0 - 999	E
1000 ~ 1999	D
2000 ~ 2999	C
3000 ~ 3999	B
4000 이상	A

7. EMP 테이블에서 다음의 결과가 출력되도록 작성하시오.

Sal의 값이 3배가 되도록 출력

Dream Salary

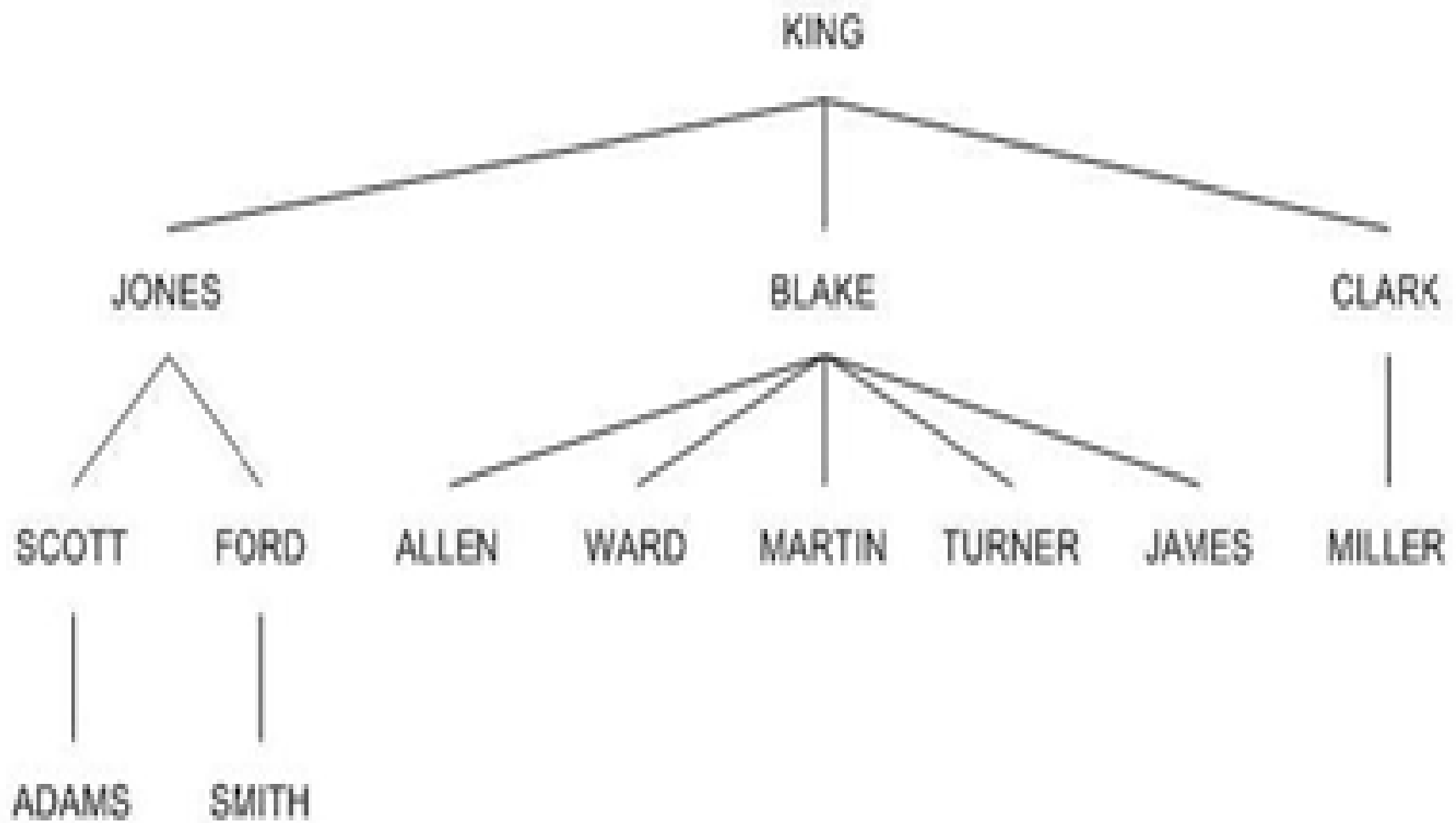
-----  
KING earns \$5,000.00 monthly but wants \$15,000.00

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

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

.....

# EMP TABLE 의 계층화



# emp table 계층화 쿼리

```
SQL> COLUMN org_chart FORMAT A15
SQL> SELECT LPAD(' ', 3 * LEVEL-3) ||ename AS org_chart,
2          LEVEL, empno, mgr, deptno
3 FROM      emp
4 START WITH mgr IS NULL
5 CONNECT BY PRIOR empno = mgr;
```

## – LEVEL

- : 계층적 질의문에서 검색된 결과에 대한 계층별로 레벨 표시

## – START WITH

- : 계층적인 출력 형식을 표현하기 위한 최상위 행

## – CONNECT BY prior

- : 계층 관계의 데이터를 지정하는 컬럼

# EMP TABLE 의 계층화

```
SQL> SELECT LPAD(ENAME, LENGTH(ENAME)+(LEVEL*2)-2, ' ') NAME  
2 FROM EMP  
3 START WITH ENAME = 'KING'  
4 CONNECT BY PRIOR EMPNO = MGR;
```

NAME

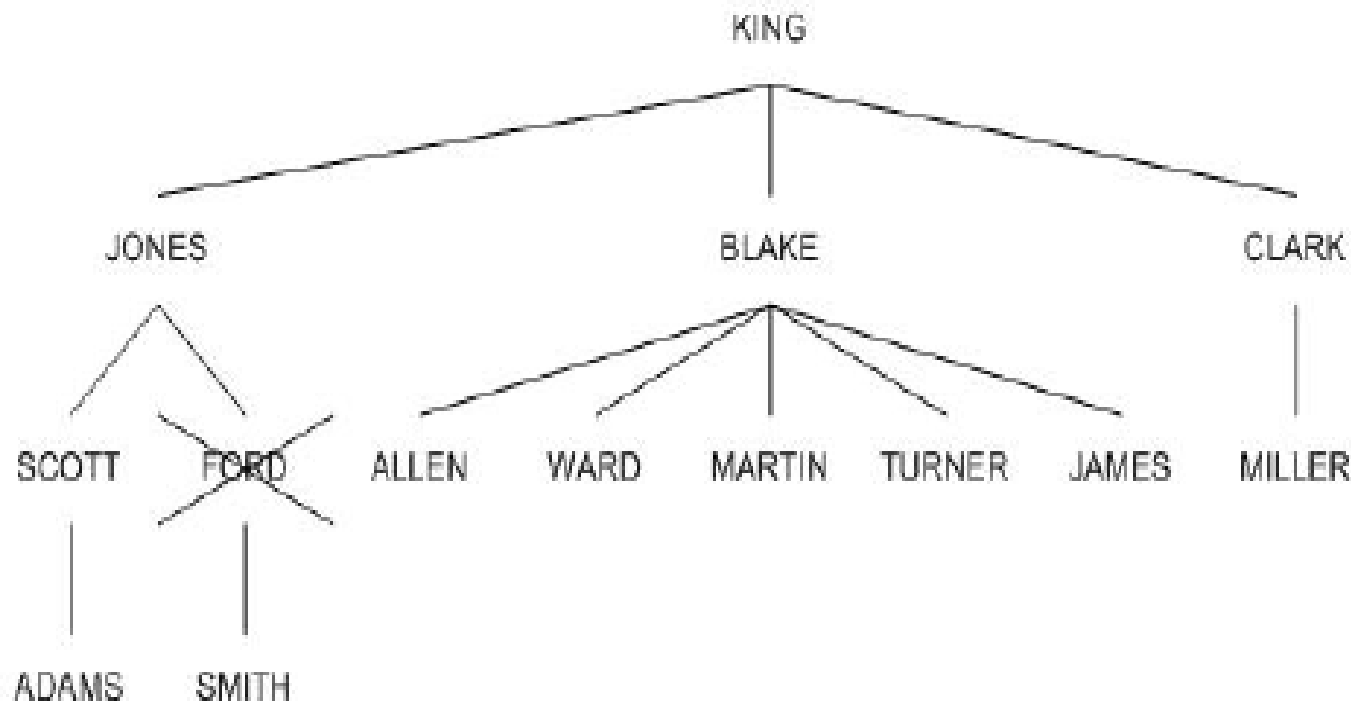
```
-----+-----  
KING  
  JONES  
    SCOTT  
      ADAMS  
        FORD  
          SMITH  
BLAKE  
  ALLEN  
    WARD  
      MARTIN  
        TURNER  
          JAMES  
CLARK  
  MILLER
```



# EMP TABLE 의 계층화

## 트리 구조의 노드(Node) 및 브랜치(Branch) 제거

'KING'으로부터 하향식 전개된 계층형 쿼리 결과에서 'FORD'를 제외하고 출력하려면 WHERE 절에서 제거할 노드를 제외 할 수 있는 조건을 기술해주면 된다.



# EMP TABLE 의 계층화

```
SQL> SELECT LPAD(ENAME, LENGTH(ENAME)+(LEVEL*2)-2, ' ') NAME  
2 FROM EMP  
3 WHERE ENAME != 'FORD'  
4 START WITH ENAME = 'KING'  
5 CONNECT BY PRIOR EMPNO = MGR;
```

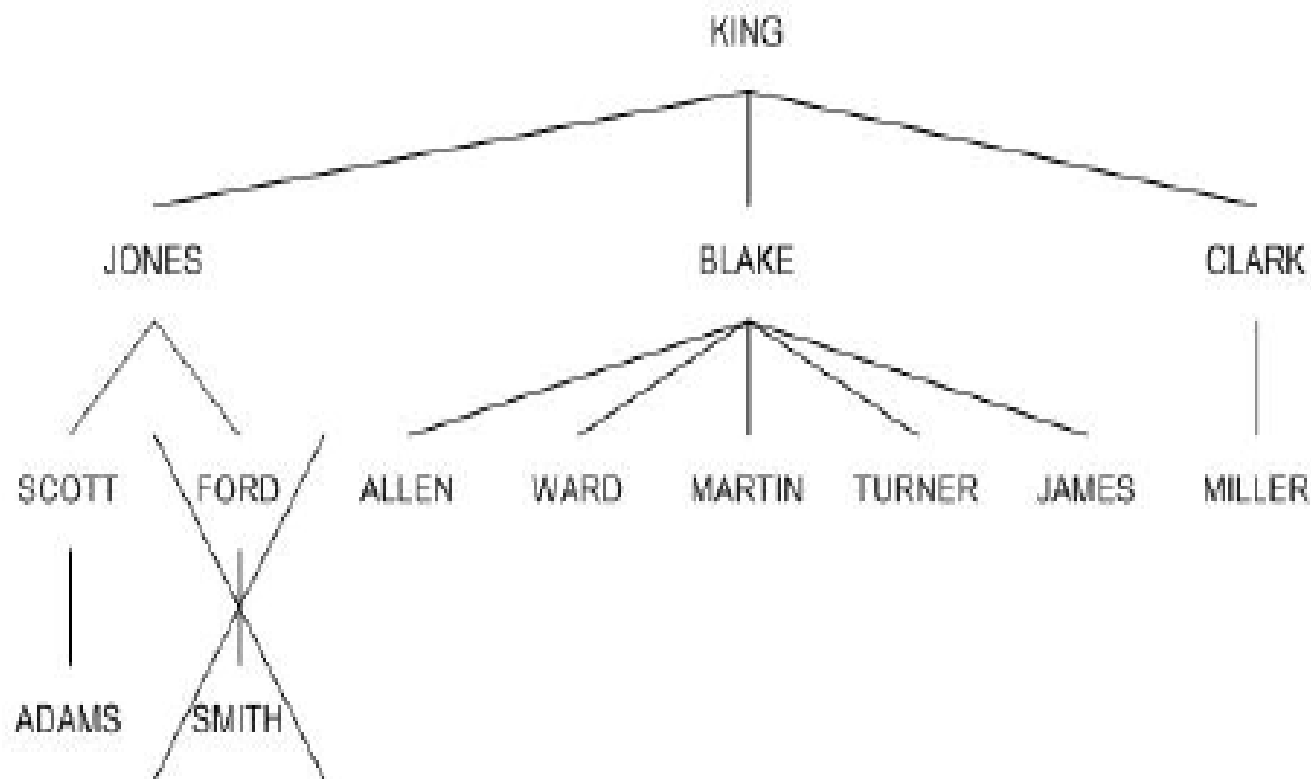
NAME

---

```
KING  
  JONES  
    SCOTT  
      ADAMS  
        SMITH  
BLAKE  
  ALLEN  
  WARD  
  MARTIN  
  TURNER  
  JAMES  
CLARK  
  MILLER
```

# EMP TABLE 의 계층화

또한, 아래 그림과 같이 트리 구조에서 'FORD'의 브랜치를 제거하고 계층형 쿼리를 진행하고자 하는 경우에는 CONNECT BY 절 뒤에 제거할 브랜치의 최고 관리자를 제외 할 수 있는 조건을 기술하면 된다.



# EMP TABLE 의 계층화

```
SQL> SELECT LPAD(ENAME, LENGTH(ENAME)+(LEVEL*2)-2, ' ') NAME  
2 FROM EMP  
3 START WITH ENAME = 'KING'  
4 CONNECT BY PRIOR EMPNO = MGR  
5 AND ENAME != 'FORD';
```

NAME

---

KING

    JONES

        SCOTT

            ADAMS

BLAKE

    ALLEN

    WARD

    MARTIN

    TURNER

    JAMES

CLARK

    MILLER