

이 장에서 다룰 내용



1 DUAL 테이블과 SQL 함수 분류

2 숫자 함수

3 문자 처리 함수

4 날짜 함수

5 형 변환 함수

이 장에서 다룰 내용



- 6 NULL을 다른 값으로 변환하는 NVL 함수
- 7 선택을 위한 DECODE 함수
- 8 조건에 따라 서로 다른 처리가 가능한 CASE 함수

01. DUAL 테이블과 SQL 함수 분류



단일 행 함수 (Single-Row Functions)

: 함수가 행마다 적용됩니다.

어떤 테이블에 14개의 행이 있고 쿼리문에 조건이 없다면 14개의 행 모두 단일 행 함수가 적용되어 변환된 결과를 반환합니다.

```
SQL> select LOWER(ename)  
2   from emp;
```

01. DUAL 테이블과 SQL 함수 분류



```
SQL> select 20*40  
2 from emp;
```

-- emp의 14행에 대해 결과도 14개의 행으로 나타납니다.

20*40

```
-----  
800  
800  
800  
800  
800  
800  
800  
800  
800  
800  
800  
800  
800  
800  
800
```

1개의 행 결과만 나타낼려면?
하나의 행을 가진 테이블이 필요

14 rows selected.

01. DUAL 테이블과 SQL 함수 분류



- ❖ DUAL 테이블의 구조를 살펴보기 위해서 DESC 명령어를 사용합니다.

예 **DESC DUAL**

- ❖ DUAL 테이블은 DUMMY라는 단 하나의 컬럼으로 구성되어 있습니다.
- ❖ 이 컬럼에는 최대 길이는 1입니다.

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

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

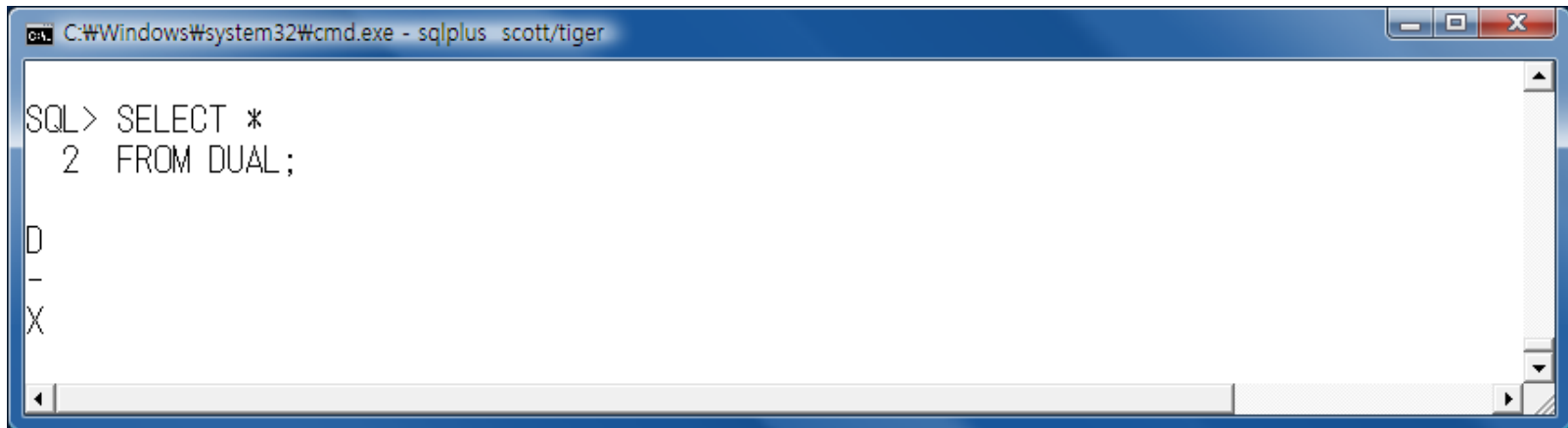
01. DUAL 테이블과 SQL 함수 분류



- ❖ DUMMY 컬럼엔 과연 어떤 값이 저장되어 있는 것일까요?

예

```
SELECT *  
FROM DUAL;
```



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

- ❖ DUAL 테이블은 DUMMY라는 단 하나의 컬럼에 X라는 단 하나의 로우만을 저장하고 있으나 이 값은 아무런 의미가 없습니다.
- ❖ 쿼리문의 수행 결과가 하나의 로우로 출력되도록 하기 위해서 단 **하나의 로우를 구성**하고 있을 뿐입니다.

02. 숫자 함수



구 분	설 명
ABS	절대값을 구한다.
COS	COSINE 값을 반환한다.
EXP	$e(2.71828183\cdots)$ 의 n 승을 반환한다.
FLOOR	n 이하의 가장 큰 정수를 반환합니다.
LOG	LOG값을 반환한다.
POWER	POWER(m, n) m 의 n 승을 반환한다.
SIGN	SIGN (n) $n < 0$ 이면 -1 , $n = 0$ 이면 0 , $n > 0$ 이면 1 을 반환한다.
SIN	SINE값을 반환한다.
TAN	TANGENT값을 반환한다.
ROUND	특정 자릿수에서 반올림한다.
TRUNC	특정 자릿수에서 잘라낸다. (버림)
MOD	입력 받은 수를 나눈 나머지 값을 반환한다.

2.1 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;
```

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

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

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


2.2 n이하의 가장 큰 정수를 반환하는 FLOOR 함수



- ❖ 형식) FLOOR(n)
- ❖ 양수인 경우 FLOOR 함수는 소수점 아래를 버립니다. 34.5678를 FLOOR 함수에 적용하면 34가 구해집니다.

예

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

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

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

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

2.2 n이하의 가장 큰 정수를 반환하는 FLOOR 함수



❖ 형식) FLOOR(n)

❖ 음수인 경우



예

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

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

```
-34.5678  FLOOR(-34.5678)
-----
-34.5678                      -35
```

2.3 특정 자릿수에서 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;
```

```
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
```

2.3 특정 자릿수에서 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;
---	---

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

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

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

2.3 특정 자릿수에서 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.

예

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

- ❖ ROUND(34.5678, -1)의 결과는 300이 됩니다.

```
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
```

2.3 특정 자릿수에서 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.

예

```
SELECT 35.678, ROUND(35.678, -1)
FROM DUAL;
```

- ❖ ROUND(35.678, -1)의 결과는 40이 됩니다.

```
SCOTT>SELECT 35.678, ROUND(35.678, -1)
2    FROM DUAL;

35.678 ROUND(35.678,-1)
-----
35.678                40
```

2.4 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;
```

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

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

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

- TRUNC 함수의 두 번째 인자 값이 20이면 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.

2.5 나머지 구하는 MOD 함수

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



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
```


<탄탄히 다지기>

1. 사번이 홀수인 사람들을 검색해 보십시오.

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

03. 문자 처리 함수



구 분	설 명
LOWER	소문자로 변환한다.
UPPER	대문자로 변환한다.
INITCAP	첫 글자만 대문자로 나머지 글자는 소문자로 변환한다.
CONCAT	문자의 값을 연결한다.
SUBSTR	문자를 잘라 추출한다.
SUBSTRB	문자를 잘라 추출한다.
LENGTH	문자의 길이를 반환한다
LENGTHB	문자의 길이를 반환한다. (바이트로 반환)

03. 문자 처리 함수



구 분	설 명
LPAD, RPAD	입력 받은 문자열과 기호를 정렬하여 특정 길이의 문자열로 반환한다.
TRIM	잘라내고 남은 문자를 표시한다.
CONVERT	CHAR SET을 변환한다.
ASCII	ASCII 코드 값으로 변환한다.
CHR	ASCII 코드 값을 문자로 변환한다.
REPLACE	문자열에서 특정 문자를 변경한다.

3.1 대문자로 변환하는 UPPER 함수

- ❖ UPPER 함수는 입력한 문자값을 대문자로 변환하는 함수입니다.
- ❖ 문자가 변환되는 결과를 명료하게 살펴보기 위해서 특정 테이블에 대한 컬럼에 대해서 함수를 적용하는 것은 나중에 하기로 하고 우선 문자 상수에 대해서 적용해보도록 합시다.



예

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

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT 'Welcome to Oracle', UPPER('Welcome to Oracle')  
2  FROM DUAL;  
  
'WELCOMETOORACLE' UPPER('WELCOMETOO  
-----  
Welcome to Oracle WELCOME TO ORACLE
```

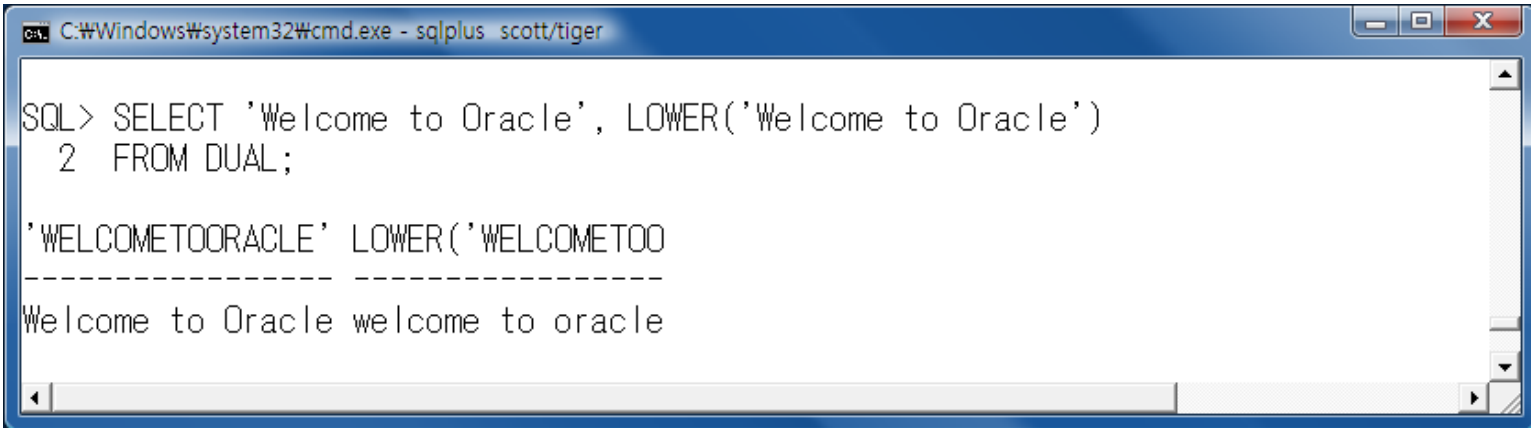
3.2 소문자로 변환하는 LOWER 함수

❖ LOWER 함수는 문자열을 모두 소문자로 변경합니다.



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('WELCOMETOO  
-----  
Welcome to Oracle welcome to oracle
```

3.3 이니셜만 대문자로 변환하는 INITCAP 함수

❖ INITCAP 함수는 문자열의 이니셜만 대문자로 변경합니다.



예

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

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

SQL> SELECT 'WELCOME TO ORACLE', INITCAP('WELCOME TO ORACLE')
2  FROM DUAL;

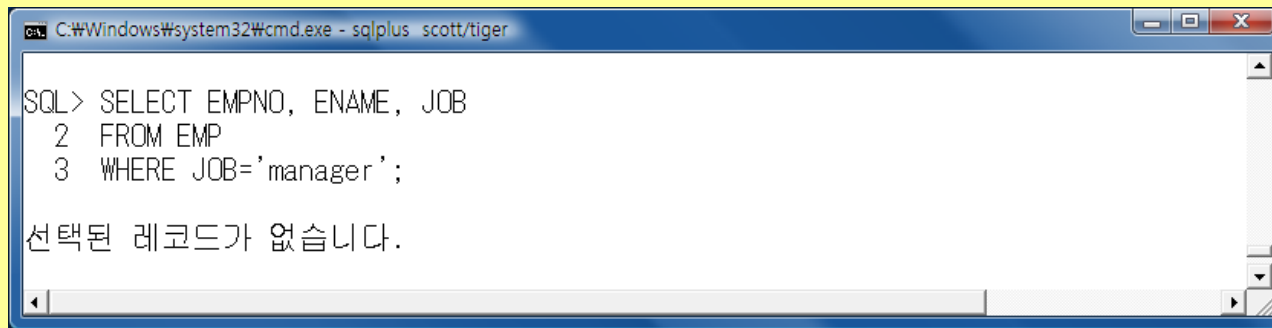
'WELCOMETOORACLE' INITCAP('WELCOMET
-----
WELCOME TO ORACLE Welcome To Oracle
```

탄탄히 다지기

2. 다음과 같이 쿼리문을 구성하면 과연 직급이 'manager'인 사원을 검색할까요?

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:

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

Below the query, the output message "선택된 레코드가 없습니다." (No records selected) is displayed.

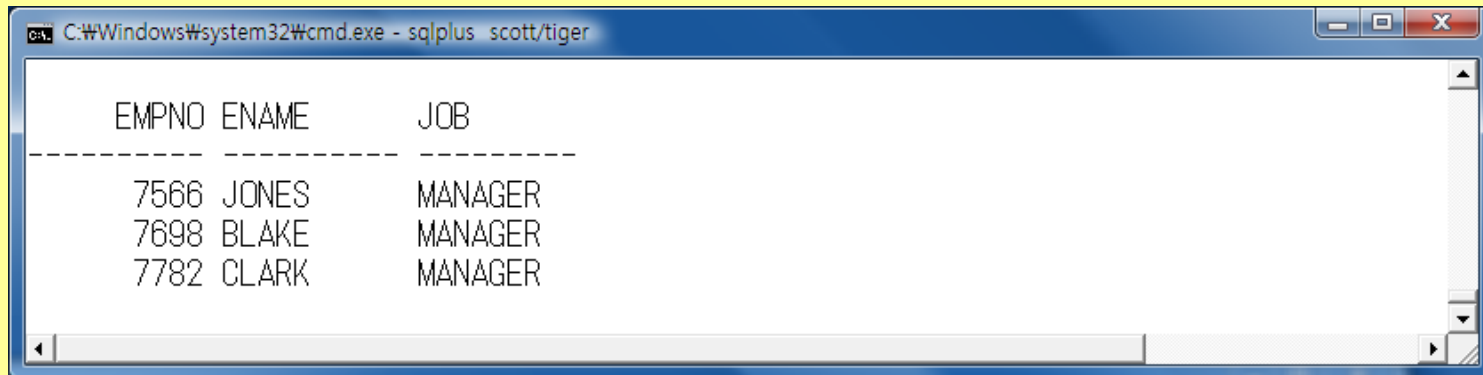
결론적으로 위와 같이 쿼리문을 기술하면 검색에 실패합니다. 왜냐하면 직급 칼럼에 저장된 데이터는 모두 대문자 형태이기 때문입니다. 굳이 소문자('manager')로 데이터를 검색해야 한다면 LOWER 함수를 사용해서 원하는 결과를 얻을 수 있습니다.

탄탄히 다지기

굳이 소문자('manager')로 데이터를 검색해야 한다면 LOWER 함수를 사용해서 원하는 결과를 얻을 수 있습니다. JOB 컬럼에 LOWER 함수를 적용하여 JOB 컬럼에 저장된 값을 소문자로 변경한 후에 검색에 참여하도록 합니다.

예

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



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

3.4 문자 길이를 구하는 LENGTH



- ❖ LENGTH 함수는 컬럼에 저장된 데이터 값이 몇 개의 문자로 구성되었는지 길이를 알려주는 함수입니다. 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
```

3.5 바이트 수를 알려주는 LENGTHB 함수



- ❖ 이번에 살펴볼 LENGTHB 함수는 바이트 수를 알려주는 함수입니다. LENGTH 함수와 어떤 차이가 있는지 살펴봅시다.

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

```
관리자: 명령 프롬프트 - sqlplus scott/tiger

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

LENGTHB('ORACLE') LENGTHB('오라클')
-----
               6               9
```

- ❖ 한글 1자는 3바이트를 차지합니다. 그렇기 때문에 수행 결과를 보면 한글 3자로 구성된 '오라클'의 LENGTHB 함수의 결과는 9가 됩니다.

3.6 문자열 일부만 추출하는 SUBSTR 함수



- ❖ SUBSTR 과 SUBSTRB 함수는 대상 문자열이나 컬럼의 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

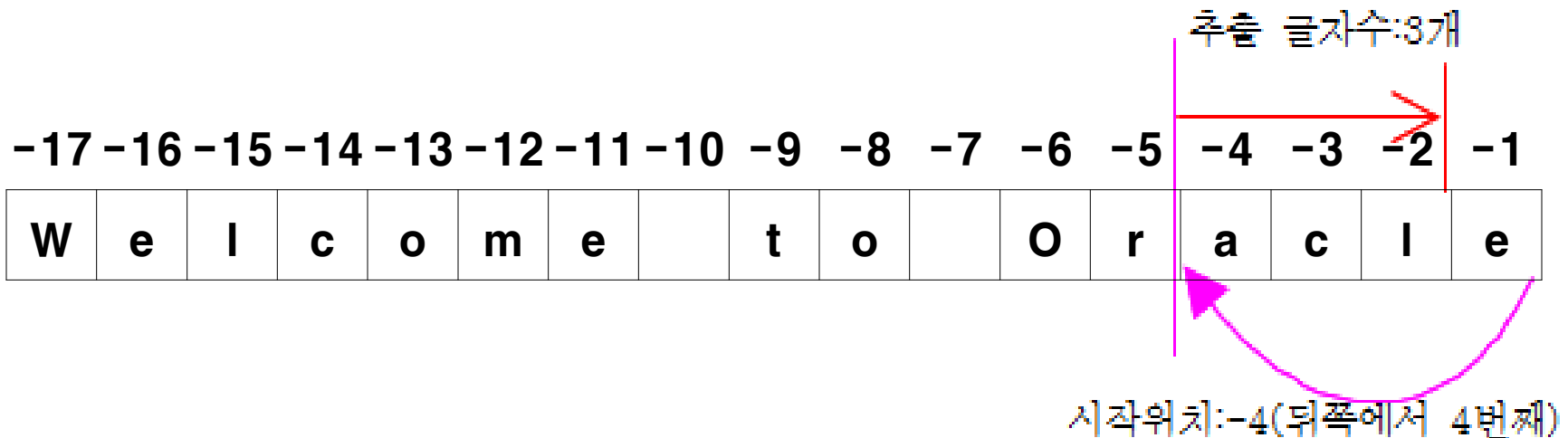
3.6 문자열 일부만 추출하는 SUBSTR 함수



- ❖ 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;
```



3.6 문자열 일부만 추출하는 SUBSTR 함수



- ❖ 직원들의 입사년도만 출력하려면 어떻게 해야 할까요?
- ❖ 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
```

3.6 문자열 일부만 추출하는 SUBSTR 함수



❖ 9월에 입사한 사원을 출력해 보세요.

예

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

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT *  
2 FROM EMP  
3 WHERE SUBSTR(HIREDATE, 4, 2)='09';  
  
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  
  
SQL>
```

암시적 형변환 : 문자형=> 숫자

```
SCOTT>SELECT *  
2 FROM EMP  
3 WHERE SUBSTR(HIREDATE,1,2)=87;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7788	SCOTT	ANALYST	7566	87/07/13	3000		20
7876	ADAMS	CLERK	7788	87/07/13	1100		20

```
SCOTT>SELECT *  
2 FROM EMP  
3 WHERE ENAME=87;
```

WHERE ENAME=87

*

ERROR at line 3:

ORA-01722: invalid number

=>ENAME을 숫자로 변경해서 87과 비교하고자 함.

암시적 형변환 : 문자형=> 숫자

```
SCOTT>SELECT 1 + '2' FROM DUAL;
```

1+'2'

3

〈탄탄히 다지기〉

3. 87년도에 입사한 직원을 알아내기 위해 SUBSTR 함수를 이용하여 HIREDATE 컬럼에서 첫 글자부터 2개를 추출하여 그 값이 87인지를 체크하는 방법으로도 구해 보도록 하세요.
4. 다음은 이름이 E로 끝나는 사원을 검색해 보도록 합시다. SUBSTR 함수를 이용하여 ENAME 컬럼의 마지막 문자 한개만 추출해서 이름이 E로 끝나는 사원을 검색해 보도록 하시오.
<힌트> 시작 위치를 -1로 주고 추출할 문자 개수를 1로 주면 됩니다.

3.7 특정 문자의 위치를 구하는 INSTR 함수



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

예

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

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
5  
SQL> SELECT INSTR('WELCOME TO ORACLE', 'O')  
2 FROM DUAL;  
  
INSTR('WELCOMETOORACLE', 'O')  
-----  
5  
  
SQL>
```

3.7 특정 문자의 위치를 구하는 INSTR 함수



- ❖ INSTR 함수의 기본 형식은 다음과 같습니다.

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 반환합니다.

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>
```

<탄탄히 다지기>

5. 이름의 세 번째 자리가 R인 사원을 검색하기 위해서 와일드카드 _ 와 LIKE 연산자를 사용하여 다음과 같이 표현할 수 있습니다.

```
SQL> Select empno, ename   from emp  
      where ename like '___R%';
```

결과화면)

EMPNO	ENAME

7521	WARD
7654	MARTIN
7844	TURNER
7902	FORD

위 쿼리문을 지금 배운 INSTR함수와 SUBSTR함수를 사용하여 작성해 보십시오.

3.8 특정 기호로 채우는 LPAD/RPAD 함수



- ❖ **LPAD(LEFT PADDING)** 함수는 컬럼이나 대상 문자열을 명시된 자릿수에서 오른쪽에 나타내고, 남은 왼쪽 자리를 특정 기호로 채웁니다.

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>
```

3.8 특정 기호로 채우는 LPAD/RPAD 함수



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

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>
```

3.9 왼쪽에서 공백 문자를 삭제하는 LTRIM 함수

❖ LTRIM 함수는 문자열의 왼쪽(앞)의 공백 문자들을 삭제합니다.



예

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

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

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

LTRIM('OR
-----
Oracle

SQL>
```

3.10 오른쪽에서 공백 문자를 삭제하는 RTRIM 함수

❖ RTRIM 함수 역시 다음과 같이 기술하면 공백 문자를 잘라냅니다.



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>
```


3.11 특정 문자를 잘라내는 TRIM 함수



❖ 다음과 같이 기술하면 공백 문자를 잘라냅니다.

예 **SELECT TRIM(' Oracle ')**
FROM DUAL;

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

SQL> SELECT TRIM(' Oracle ')
2 FROM DUAL;

TRIM('
-----
Oracle

SQL>
```

SQL> SELECT LENGTH(TRIM(' Oracle '))
2 FROM DUAL;

LENGTH(TRIM('ORACLE'))

3.12 문자 바꾸는 REPLACE 함수



- ❖ 첫 번째 인자인 문자열이나 컬럼에서 두 번째 인자 문자1를 세 번째 인자 문자2로 바꾸어 출력합니다.

형식 REPLACE(문자열 또는 컬럼명 , 문자1, 문자2)

예

```
select 'abcde' as "문자열",  
       replace('abcde' , 'ab' , '**') as "replace"  
from dual
```

문자열	replace
abcde	**cde

3.12 문자 바꾸는 REPLACE 함수



이름	REPLACE
SMITH	**ITH
ALLEN	**LEN
WARD	**RD
JONES	**NES
MARTIN	**RTIN
BLAKE	**AKE
CLARK	**ARK
SCOTT	**OTT
KING	**NG
TURNER	**RNER
ADAMS	**AMS
JAMES	**MES
FORD	**RD
MILLER	**LLER

3.12 문자 바꾸는 REPLACE 함수



```
SELECT ENAME AS "이름",  
REPLACE(ENAME, SUBSTR(ENAME,1,2), '**') AS "REPLACE"  
FROM EMP;
```

3.12 문자 바꾸는 REPLACE 함수



```
1 select ename as "이름", replace(ename, substr(ename,1,2),'**') as "replace"  
2* from emp
```

이름	replace
SMITH	**ITH
ALLEN	**LEN
WARD	**RD
JONES	**NES
MARTIN	**RTIN
BLAKE	**AKE
CLARK	**ARK
SCOTT	**OTT
KING	**NG
TURNER	**RNER
ADAMS	**AMS
JAMES	**MES
FORD	**RD
MILLER	**LLER

14 rows selected.

04. 날짜 함수



구 분	설 명
SYSDATE	시스템 저장된 현재 날짜를 반환한다.

4.1 현재 날짜를 반환하는 SYSDATE 함수



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

예

```
SELECT SYSDATE  
FROM DUAL;
```

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

4.2 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.

예	<code>SELECT SYSDATE-1 어제, SYSDATE 오늘, SYSDATE+1 내일 FROM DUAL;</code>
---	---

어제	오늘	내일
-----	-----	-----
24/01/11	24/01/12	24/01/13

〈탄탄히 다지기〉

6. 각 사원들의 현재까지의 근무 일수를 구해봅시다.

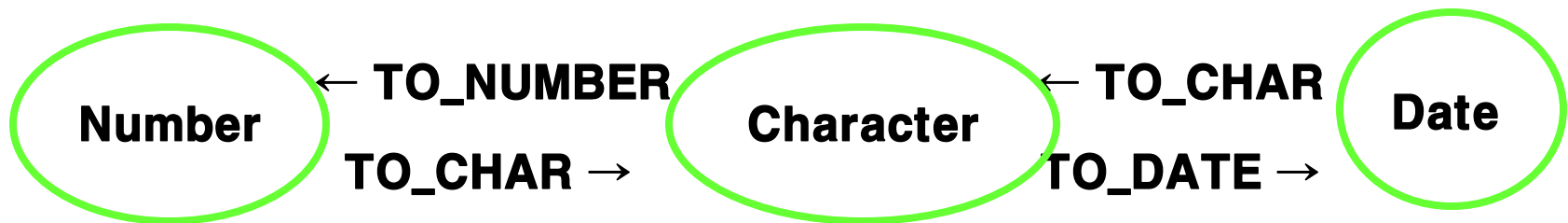
ENAME	근무일수
SMITH	15731일
ALLEN	15666일
WARD	15664일
JONES	15625일
MARTIN	15446일
BLAKE	15596일
CLARK	15557일
SCOTT	13332일
KING	15396일
TURNER	15466일
ADAMS	13332일
JAMES	15380일
FORD	15380일
MILLER	15329일

14 rows selected.

05. 형 변환 함수



- ❖ 오라클을 사용하다 보면 숫자, 문자, 날짜의 데이터 형을 다른 데이터형으로 변환해야 하는 경우가 생깁니다.
- ❖ 이럴 때 사용하는 함수가 형 변환 함수입니다. 형 변환 함수로는 TO_NUMBER, TO_CHAR, TO_DATE 가 있습니다.



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

5.1 문자형으로 변환하는 TO_CHAR 함수 날짜형을 문자형으로 변환하기

- ❖ DATE 형태의 데이터를 지정한 양식에 의해 VARCHAR2 형의 문자로 변환합니다.



형식 TO_CHAR (날짜 데이터, '출력형식')

- ❖ 다음은 날짜 출력 형식의 종류를 나열한 표입니다.

종류	의미
YYYY	연도 표현(4자리)
YY	연도 표현(2자리)
YEAR	단어로 된 연도(예)NINETEEN EIGHTY)
MM	월을 숫자로 표현
MONTH	단어로 된 달(예) DECEMBER)
MON	월을 표현(예)JAN)
DAY	요일 표현(예)SUNDAY, 일요일)
DY	요일을 약어로 표현(예)SUN, 일)

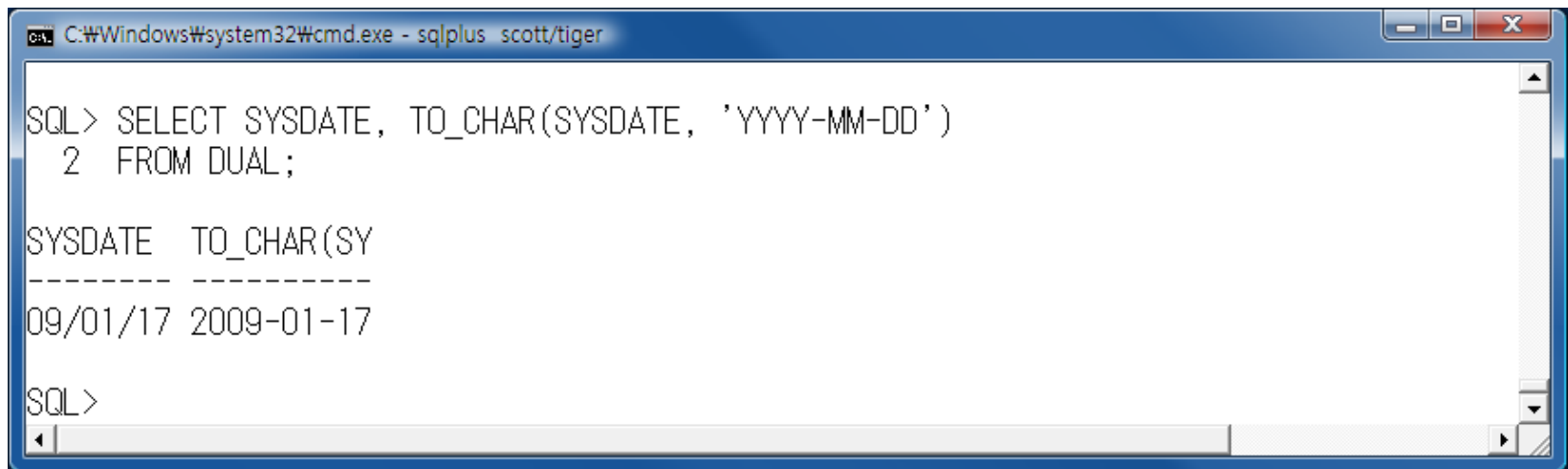
5.1 문자형으로 변환하는 TO_CHAR 함수 날짜형을 문자형으로 변환하기

❖ 다음은 현재 날짜를 기본 형식과 다른 형태로 출력해 봅시다.



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>
```

4.1 현재 날짜를 반환하는 SYSDATE 함수



- ❖ SYSDATE 함수는 시스템에 저장된 현재 날짜를 반환하는 함수입니다. 아래와 같이 출력되도록 작성해 보세요

예 2024년 01월 12일 금요일

```
select to_char(sysdate, 'YYYY') || '년',  
       to_char(sysdate, 'MM') || '월',  
       to_char(sysdate, 'DD') || '일',  
       to_char(sysdate, 'DAY')  
from dual;
```

```
select  
  to_char(sysdate, 'YYYY"년 "MM"월 "DD"일" DAY')  
from dual;
```

5.1 문자형으로 변환하는 TO_CHAR 함수 날짜형을 문자형으로 변환하기

❖ **사원들의 입사일을 출력하되 요일까지 함께 출력해 봅시다.**



예

```
SELECT HIREDATE, TO_CHAR (HIREDATE, 'YYYY/MM/DD DAY')  
FROM EMP;
```

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT HIREDATE, TO_CHAR (HIREDATE, 'YYYY/MM/DD DAY')  
2 FROM EMP;  
  
HIREDATE TO_CHAR(HIREDATE, 'YY  
-----  
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
```

5.1 문자형으로 변환하는 TO_CHAR 함수 날짜형을 문자형으로 변환하기

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



예

```
SELECT HIREDATE, TO_CHAR (HIREDATE, 'YY/MON/DD DY')  
FROM EMP;
```

```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT HIREDATE, TO_CHAR (HIREDATE, 'YY/MON/DD DY')  
2 FROM EMP;  
  
HIREDATE TO_CHAR(HIREDATE, 'Y  
-----  
80/12/17 80/12월 /17 수  
81/02/20 81/2월 /20 금  
81/02/22 81/2월 /22 일  
81/04/02 81/4월 /02 목  
81/09/28 81/9월 /28 월  
81/05/01 81/5월 /01 금  
81/06/09 81/6월 /09 화
```

5.1 문자형으로 변환하는 TO_CHAR 함수

날짜형을 문자형으로 변환하기



❖ 다음은 시간 출력 형식의 종류를 나열한 표입니다.

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

5.1 문자형으로 변환하는 TO_CHAR 함수 날짜형을 문자형으로 변환하기

❖ 다음은 현재 날짜와 시간을 출력하는 예제입니다.



예

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

```
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, 'YYYY/MM/DD, HH24:MI:SS')  
-----  
2009/01/17, 17:07:13  
  
SQL>
```

5.1 문자형으로 변환하는 TO_CHAR 함수

날짜형을 문자형으로 변환하기 - 응용예제

❖ 다음 처럼 현재 날짜와 시간을 출력하는 문을 작성해 보세요



예 2022년11월16일 수요일 오후 05:01:02

```
select
  to_char(sysdate, 'YYYY"년"MM"월"DD"일
    " DAY PM HH:MI:SS')
from dual;
```

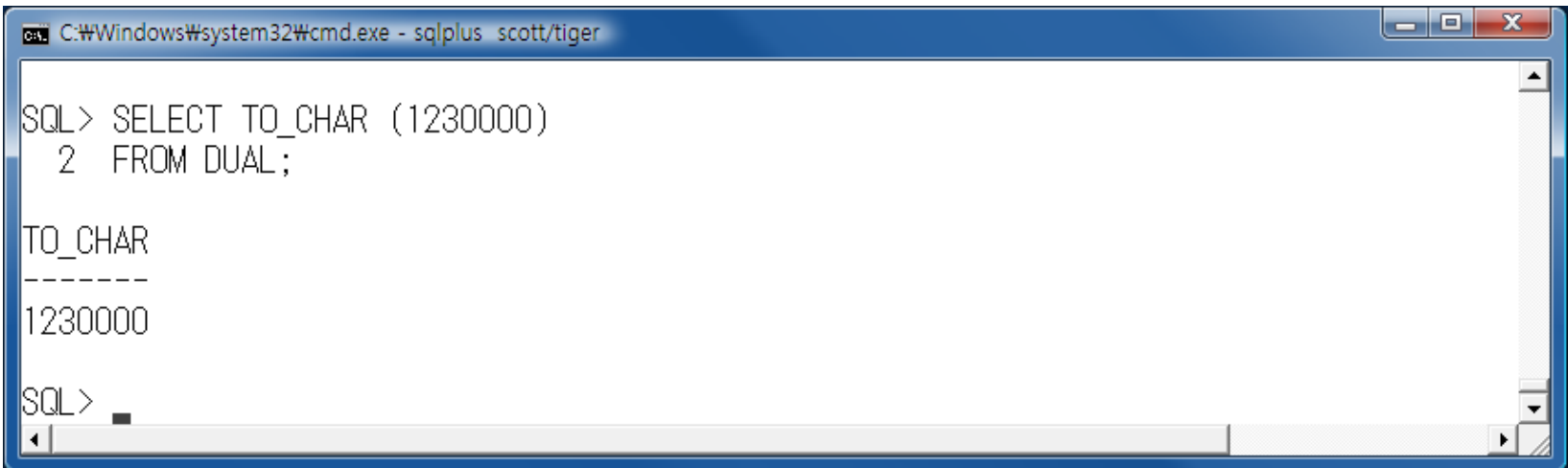
5.1 문자형으로 변환하는 TO_CHAR 함수 숫자형을 문자형으로 변환하기

❖ 1230000이란 숫자를 문자 형태로 출력하는 예입니다.



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>
```

5.1 문자형으로 변환하는 TO_CHAR 함수 숫자형을 문자형으로 변환하기

❖ 다음 표는 숫자 출력 형식을 나열한 표입니다.



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

5.1 문자형으로 변환하는 TO_CHAR 함수 숫자형을 문자형으로 변환하기

- ❖ 각 지역별 통화 기호를 앞에 붙이고 천 단위마다 콤마를 붙여서 출력(예 : ₩1,230,000) 하려면 어떻게 해야 할까요? 위 표를 참조하여 다음과 같이 표현해야 합니다.



예

```
SELECT ENAME, SAL, TO_CHAR (SAL, 'L999,999')  
FROM EMP;
```

SQL> SELECT ENAME, SAL, TO_CHAR (SAL, 'L999,999')
2 FROM EMP;

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

5.1 문자형으로 변환하는 TO_CHAR 함수 숫자형을 문자형으로 변환하기

- ❖ 9는 자릿수를 나타내며 자릿수가 맞지 않으면 채우지 않습니다. 하지만 0은 자릿수를 나타내며 자릿수가 맞지 않을 경우 0으로 채웁니다.



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(12  TO_CHAR(1234
-----
000123456      123,456

SQL>
```

5.2 날짜형으로 변환하는 TO_DATE 함수



- ❖ 날짜 형으로 변환하기에 앞서 날짜 형에 대해서 살펴보도록 합니다. 날짜 형은 세기, 년도, 월, 일. 시간, 분. 초와 같이 날짜와 시간에 대한 정보를 저장합니다.
- ❖ 오라클에서 기본 날짜 형식은 'YY/MM/DD' 형식으로 '년/월/일' 예를 들면 '06/03/08' 식으로 나타냅니다.
- ❖ 만일 년도를 4자리로 출력하려면 'YYYY/MM/DD' 형식으로 지정합니다.
- ❖ TO_DATE 함수는 문자열 또는 숫자를 날짜 형으로 변환합니다.

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

5.2 날짜형으로 변환하는 TO_DATE 함수

- ❖ 숫자형태인 19810220 을 TO_DATE 함수를 사용해서 날짜형으로 변환합니다. 이때 두 번째 인자는 출력할 형식을 지정해 주어야 합니다.



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>
```


5.2 날짜형으로 변환하는 TO_DATE 함수

- ❖ 현재 날짜에서 '2023/12/26' 을 뺀 결과를 출력하는 예제로 경과일을 구해 봅시다.



예

```
SELECT  
  TRUNC(SYSDATE-TO_DATE('2023/12/26', 'YYYY/MM/DD'))||'일'  
  as "경과일"  
FROM DUAL;
```

5.3 숫자형으로 변환하는 TO_NUMBER 함수



- ❖ TO_NUMBER 함수는 특정 데이터를 숫자형으로 변환해 주는 함수입니다.
- ❖ 다음과 같이 '20,000'을 '10,000'의 차이를 알아보기 위해서 빼기를 해 봅시다. 산술 연산을 하려면 문자형을 숫자형으로 변환한 후에 실행해야 합니다.

예

```
SELECT '20,000' - '10,000'  
FROM DUAL;
```

5.3 숫자형으로 변환하는 TO_NUMBER 함수



- ❖ TO_NUMBER 함수는 특정 데이터를 숫자형으로 변환해 주는 함수입니다.
- ❖ 다음과 같이 '20,000'을 '10,000'의 차이를 알아보기 위해서 빼기를 해 봅시다. 산술 연산을 하려면 문자형을 숫자형으로 변환한 후에 실행해야 합니다.

예

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

```
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>
```

06. NULL을 다른 값으로 변환하는 NVL 함수

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



예

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

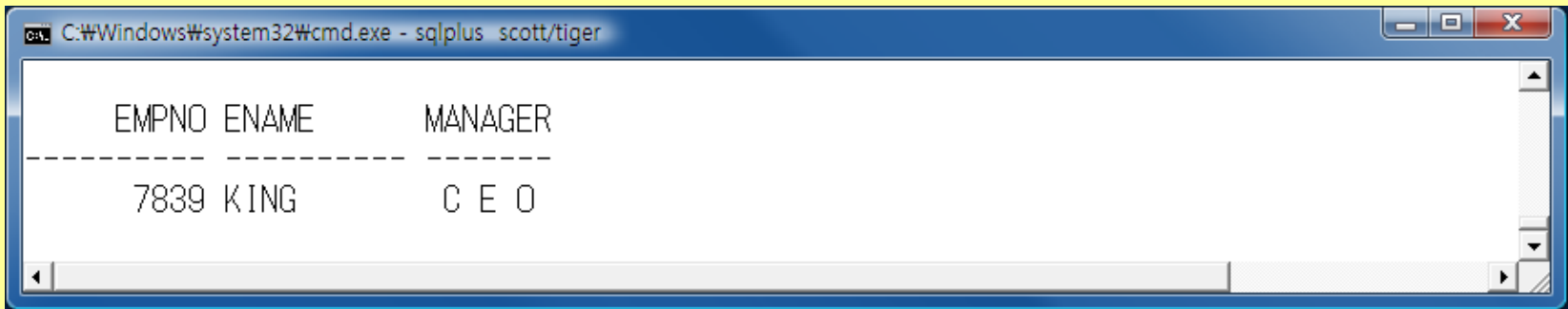
C:\Windows\system32\cmd.exe - sqlplus scott/tiger

```
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		36000	0	36000
FORD	3000		36000	0	36000
MILLER	1300		15600	0	15600
JAMES	950		11400	0	11400
SMITH	800		9600	0	9600
ADAMS	1100		13200	0	13200
BLAKE	2850		34200	0	34200
JONES	2975		35700	0	35700
CLARK	2450		29400	0	29400
KING	5000		60000	0	60000

<탄탄히 다지기>

7. 모든 사원은 자신의 상관(MANAGER)이 있다. 하지만 EMP 테이블에 유일하게 상관이 없는 로우가 있는데 그 사원의 MGR 칼럼 값이 NULL이다. 상관이 없는 사원만 출력하되 MGR 칼럼 값 NULL 대신 CEO로 출력해 봅시다 단, NVL()를 사용해 보세요.



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

EMPNO	ENAME	MANAGER
7839	KING	C E O

07. 선택을 위한 DECODE 함수



- ❖ DECODE 함수는 프로그램 언어에서 가장 많이 사용되는 switch case 문과 같은 기능을 갖습니다.
- ❖ 즉, 여러 가지 경우에 대해서 선택할 수 있도록 합니다.
- ❖ 다음은 DECODE 함수의 기본 형식입니다.

형식	DECODE (컬럼명, 조건1, 결과1, 조건2, 결과2, 조건3, 결과3, 기본결과n -- 생략 가능합니다.
) 컬럼명이 조건1과 같으면 결과1 을 선택하고 컬럼명이 조건2와 같으면 결과2 를 선택하고 컬럼명이 조건3과 같으면 결과3 을 선택하고 아니며 기본결과를 선택합니다.

07. 선택을 위한 DECODE 함수



예제

```
select deptno,  
       decode(deptno,  
              10, 'A',  
              20, 'B',  
              'DEFAULT')  
from emp  
order by deptno;
```

DEPTNO DECODE(DEPTNO,

DEPTNO	DECODE(DEPTNO,
10	A
10	A
10	A
20	B
20	B
20	B
20	B
20	B
30	DEFAULT
30	DEFAULT
30	DEFAULT
30	DEFAULT
30	DEFAULT
30	DEFAULT

07. 선택을 위한 DECODE 함수

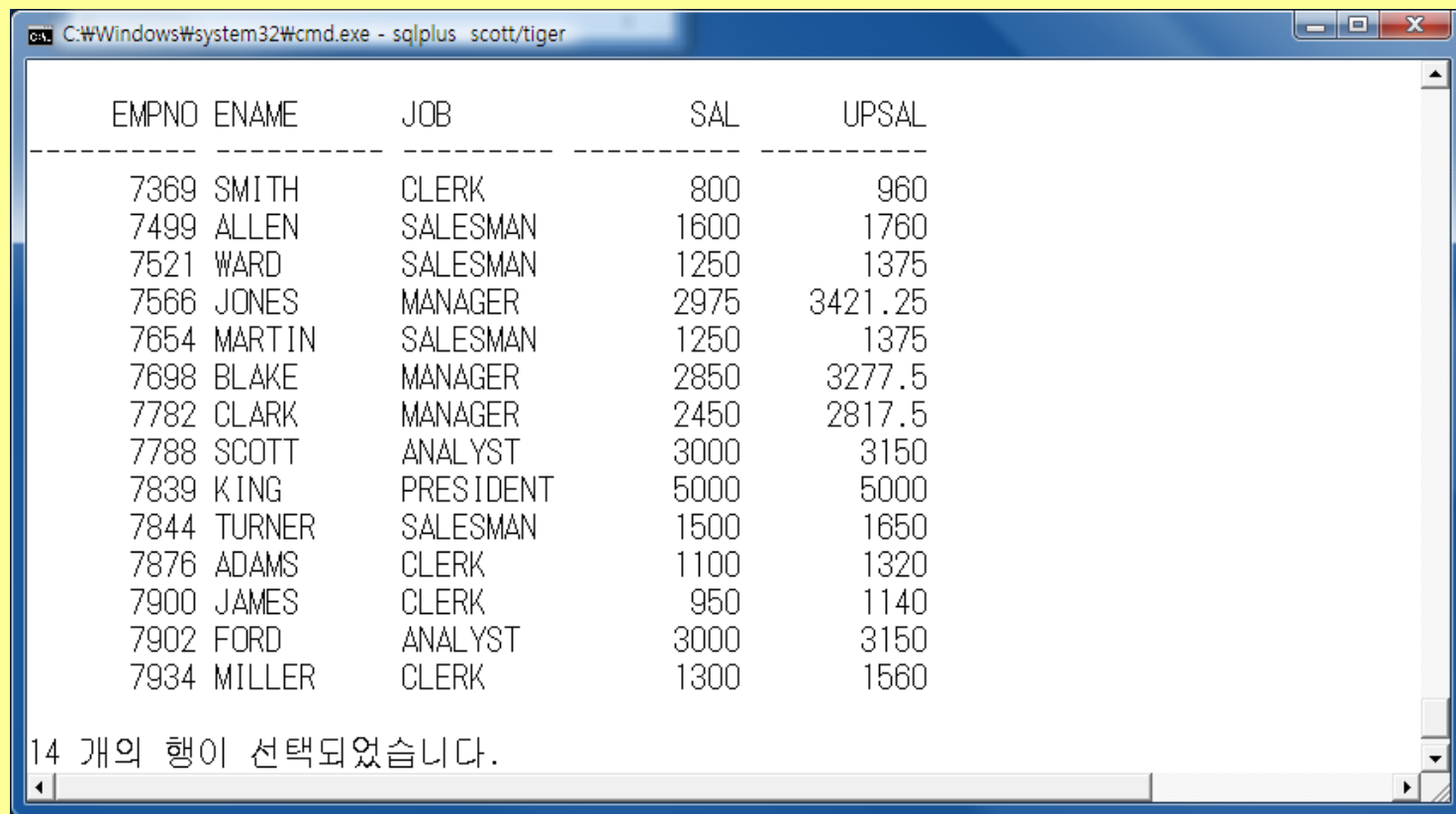


**SELECT ENAME, DEPTNO,
DECODE(DEPTNO,
10, 'ACCOUNTING',
20, 'RESEARCH',
30, 'SALES',
40, 'OPERATIONS')
AS DNAME
FROM EMP;**

ENAME	DEPTNO	DNAME
SMITH	20	RESEARCH
ALLEN	30	SALES
WARD	30	SALES
JONES	20	RESEARCH
MARTIN	30	SALES
BLAKE	30	SALES
CLARK	10	ACCOUNTING
SCOTT	20	RESEARCH
KING	10	ACCOUNTING
TURNER	30	SALES
ADAMS	20	RESEARCH
JAMES	30	SALES
FORD	20	RESEARCH
MILLER	10	ACCOUNTING

탄탄히 다지기

8. 직급에 따라 급여를 인상하도록 하자. 직급이 'ANALYST'인 사원은 5%, 'SALESMAN'인 사원은 10%, 'MANAGER'인 사원은 15%, 'CLERK'인 사원은 20%인 인상한다.



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.

08. 조건에 따라 서로 다른 처리가 가능한 CASE 함수



- ❖ CASE 함수 역시 여러 가지 경우에 대해서 하나를 선택하는 함수입니다.
 - ❖ 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 있습니다.
- ▼
- ❖ CASE 함수는 프로그램 언어의 if else if else 와 유사한 구조를 갖습니다.

형식	CASE	표현식	WHEN	조건1	THEN	결과1	
			WHEN	조건2	THEN	결과2	
			WHEN	조건3	THEN	결과3	
			ELSE	결과n			-- 생략가능
	END						

08. 조건에 따라 서로 다른 처리가 가능한 CASE 함수

- ❖ 부서번호에 해당되는 부서명을 구하는 예제를 이번에는 CASE 함수를 사용하여 작성해 봅시다. (= 조건인 경우 사용합니다.)



❖ 예

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

08. 조건에 따라 서로 다른 처리가 가능한 CASE 함수



- ❖ 부서번호에 해당되는 부서명을 구하는 예제를 이번에는 CASE 함수를 사용하여 작성해 봅시다. (비교 조건이 =인 경우)

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;
```

- ❖ (비교 조건이 =이 아닌 경우)

예

```
SELECT ENAME, DEPTNO, SAL,  
       CASE  
         WHEN SAL>=700 AND SAL<=1200 THEN 'LEVEL1'  
         WHEN SAL>=1201 AND SAL<=1400 THEN 'LEVEL2'  
         WHEN SAL>=1401 AND SAL<=2000 THEN 'LEVEL3'  
         WHEN SAL>=2001 AND SAL<=3000 THEN 'LEVEL4'  
         ELSE 'LEVEL5'  
       END AS GRADE  
FROM EMP;
```

탄탄히 다지기



9. 급여에 따라 다음과 같이 인상하도록 합니다.(CASE 사용)

급여가 1000미만이면 8%

1000~2000 5%

2001~3000 3%

이외 1% 인상

ENAME	DEPTNO	SAL	UPSAL
SMITH	20	800	864
ALLEN	30	1600	1680
WARD	30	1250	1312.5
JONES	20	2975	3064.25
MARTIN	30	1250	1312.5
BLAKE	30	2850	2935.5
CLARK	10	2450	2523.5
SCOTT	20	3000	3090
KING	10	5000	5050
TURNER	30	1500	1575
ADAMS	20	1100	1155
JAMES	30	950	1026
FORD	20	3000	3090
MILLER	10	1300	1365

14 rows selected.