

데이터베이스

관계형 데이터베이스

04. SQL 주요 함수

first
coding

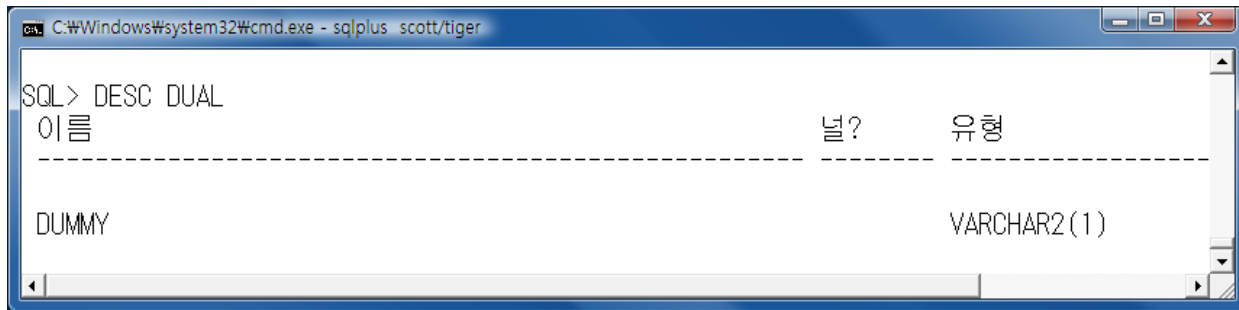
01 DUAL 테이블과 SQL 함수 분류

- 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

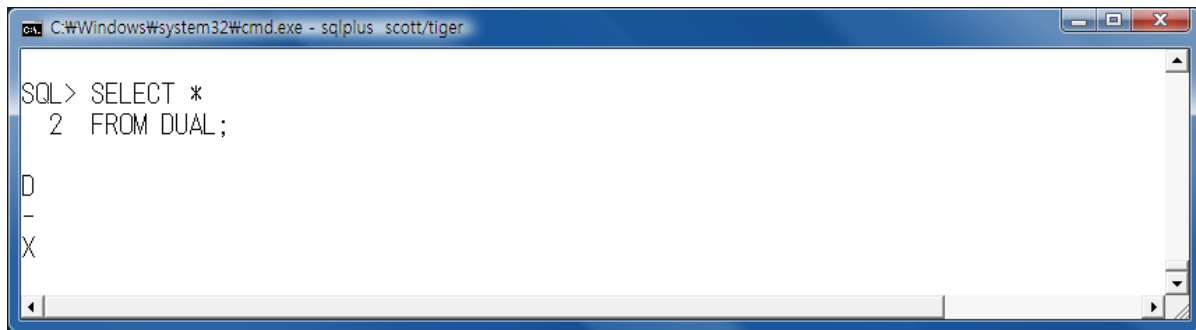
SQL> DESC DUAL
  이름                                널?       유형
-----
DUMMY                                YES        VARCHAR2(1)
```

01 DUAL 테이블과 SQL 함수 분류

- DUMMY 컬럼엔 과연 어떤 값이 저장되어 있는 것일까요?

예

```
SELECT *  
FROM DUAL;
```



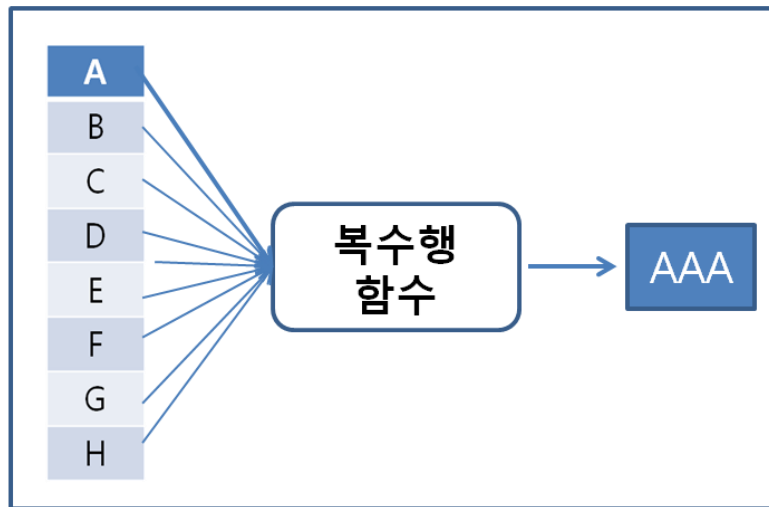
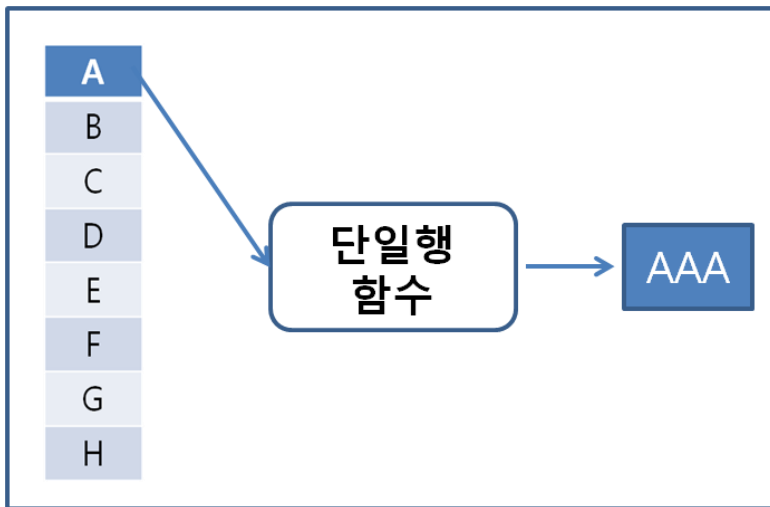
The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The prompt is "SQL>". The user has entered the query "SELECT * FROM DUAL;". The output displayed is a single row with a single column containing the value "X".

```
SQL> SELECT *  
2  FROM DUAL;  
  
D  
-  
X
```

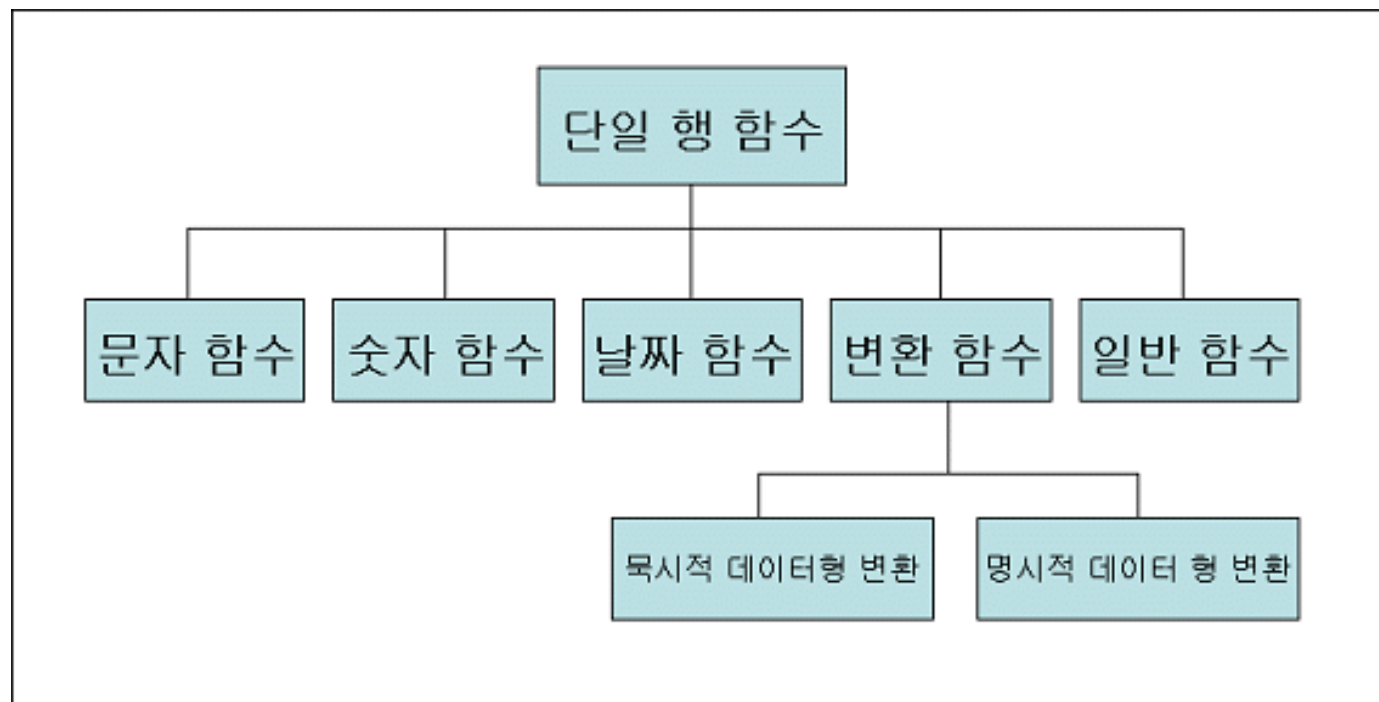
- DUAL 테이블은 DUMMY라는 단 하나의 컬럼에 X라는 단 하나의 로우만을 저장하고 있으나 이 값은 아무런 의미가 없습니다.
- 쿼리문의 수행 결과가 하나의 로우로 출력되도록 하기 위해서 단 하나의 로우를 구성하고 있을 뿐입니다.

01 DUAL 테이블과 SQL 함수 분류

- SQL 함수



- SQL 단일 행 함수

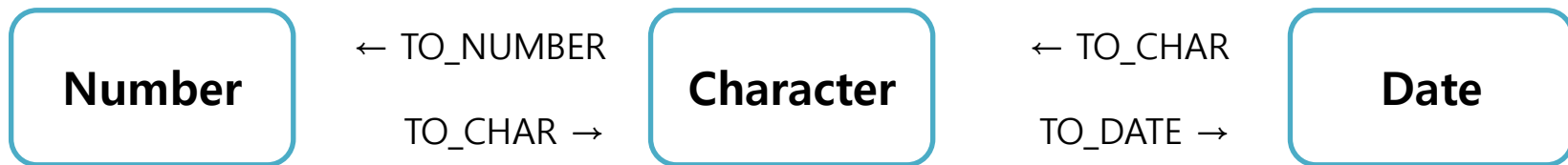


구 분	설 명
ABS	절대값을 구한다.
COS	COSINE 값을 반환한다.
EXP	e(2.71828183...)의 n승을 반환한다.
FLOOR	소수점 아래를 잘라낸다.(버림)
LOG	LOG값을 반환한다.
POWER	POWER(m, n) m의 n승을 반환한다.
SIGN	SIGN (n) $n < 0$ 이면 -1, $n = 0$ 이면 0, $n > 0$ 이면 1을 반환한다.
SIN	SINE값을 반환한다.
TAN	TANGENT값을 반환한다.
ROUND	특정 자릿수에서 반올림한다.
TRUNC	특정 자릿수에서 잘라낸다. (버림)
MOD	입력 받은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

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. (한글 1Byte)
SUBSTRB	문자를 잘라 추출한다. (한글 2Byte)
LENGTH	문자의 길이를 반환한다.(한글 1Byte)
LENGTHB	문자의 길이를 반환한다.(한글 2Byte)
LPAD, RPAD	입력 받은 문자열과 기호를 정렬하여 특정 길이의 문자열로 반환한다.
TRIM	잘라내고 남은 문자를 표시한다.
CONVERT	CHAR SET을 변환한다.
CHR	ASCII 코드 값으로 변환한다.
ASCII	ASCII 코드 값을 문자로 변환한다.
REPLACE	문자열에서 특정 문자를 변경한다.

구 분	설 명
SYSDATE	시스템 저장된 현재 날짜를 반환한다.
MONTHS_BETWEEN	두 날짜 사이가 몇 개월인지를 반환한다.
ADD_MONTHS	특정 날짜에 개월 수를 더한다.
NEXT_DAY	특정 날짜에서 최초로 도래하는 인자로 받은 요일의 날짜를 반환한다.
LAST_DAY	해당 달의 마지막 날짜를 반환한다.
ROUND	인자로 받은 날짜를 특정 기준으로 반올림한다.
TRUNC	인자로 받은 날짜를 특정 기준으로 버린다.

- 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_DATE가 있습니다.



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

05 1. 문자형으로 변환하는 TO_CHAR 함수

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

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

- 다음은 날짜 출력 형식의 종류를 나열한 표입니다.

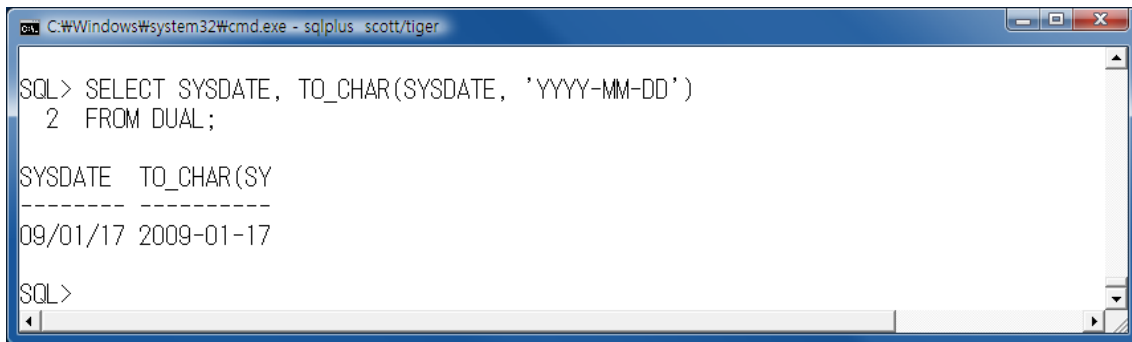
종류	의미
YYYY	년도 표현(4자리)
YY	년도 표현(2자리)
MM	월을 숫자로 표현
MON	월을 알파벳으로 표현
DAY	요일 표현
DY	요일을 약어로 표현

05 1. 문자형으로 변환하는 TO_CHAR 함수

- 다음은 현재 날짜를 기본 형식과 다른 형태로 출력해 봅시다.

예

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



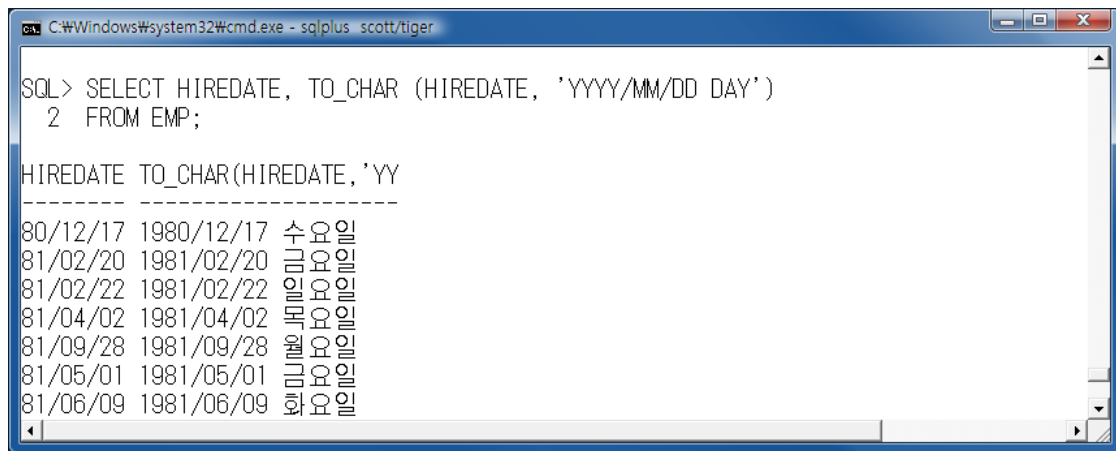
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT SYSDATE, TO_CHAR(SYSDATE, 'YYYY-MM-DD')  
2 FROM DUAL;  
  
SYSDATE    TO_CHAR(SY  
-----  
09/01/17   2009-01-17  
  
SQL>
```

05 1. 문자형으로 변환하는 TO_CHAR 함수

- 직원들의 입사일을 출력하되 요일까지 함께 출력해 보시다.

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 following SQL query:

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

The output of the query is displayed below the prompt:

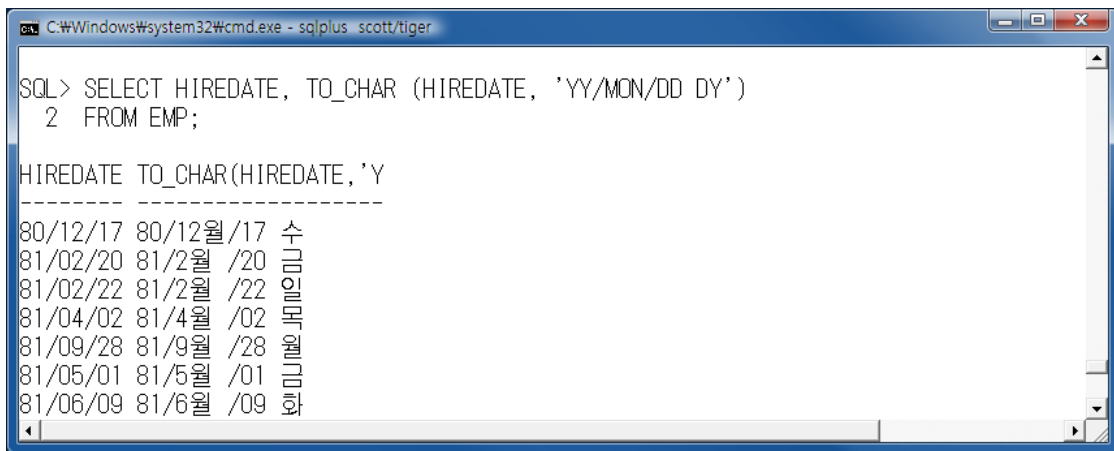
```
HIREDATE TO_CHAR(HIREDATE, 'YY  
-----  
80/12/17 1980/12/17 수요일  
81/02/20 1981/02/20 금요일  
81/02/22 1981/02/22 일요일  
81/04/02 1981/04/02 목요일  
81/09/28 1981/09/28 월요일  
81/05/01 1981/05/01 금요일  
81/06/09 1981/06/09 화요일
```

05 1. 문자형으로 변환하는 TO_CHAR 함수

- 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;
```



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

The output of the query is displayed in a table format:

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 일
81/04/02	81/4월 /02 화
81/09/28	81/9월 /28 목
81/05/01	81/5월 /01 금
81/06/09	81/6월 /09 화

05 1. 문자형으로 변환하는 TO_CHAR 함수

- 다음은 시간 출력 형식의 종류를 나열한 표입니다.

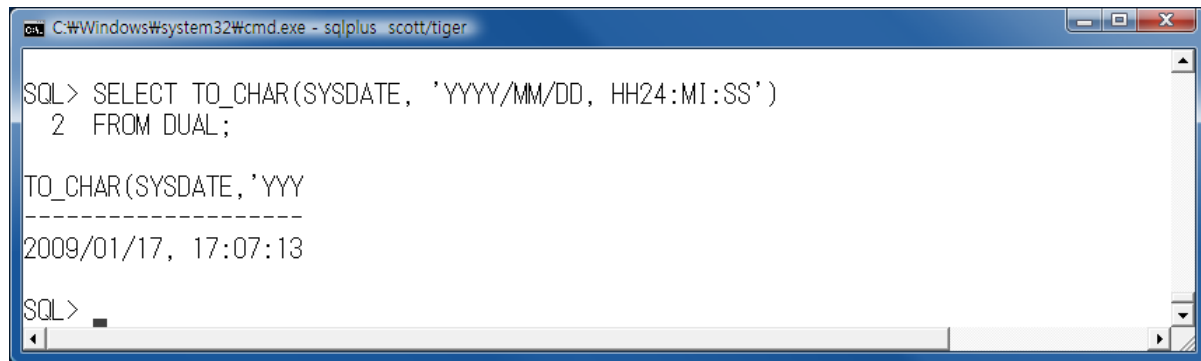
종류	의미
AM 또는 PM	오전(AM), 오후(PM) 시각 표시
A.M 또는 P.M	오전(A.M), 오후(P.M) 시각 표시
HH또는 HH12	시간(1~12)
HH24	24시간으로 표현(0~23)
MI	분 표현
SS	초 표현

05 1. 문자형으로 변환하는 TO_CHAR 함수

- 다음은 현재 날짜와 시간을 출력하는 예제입니다.

예

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



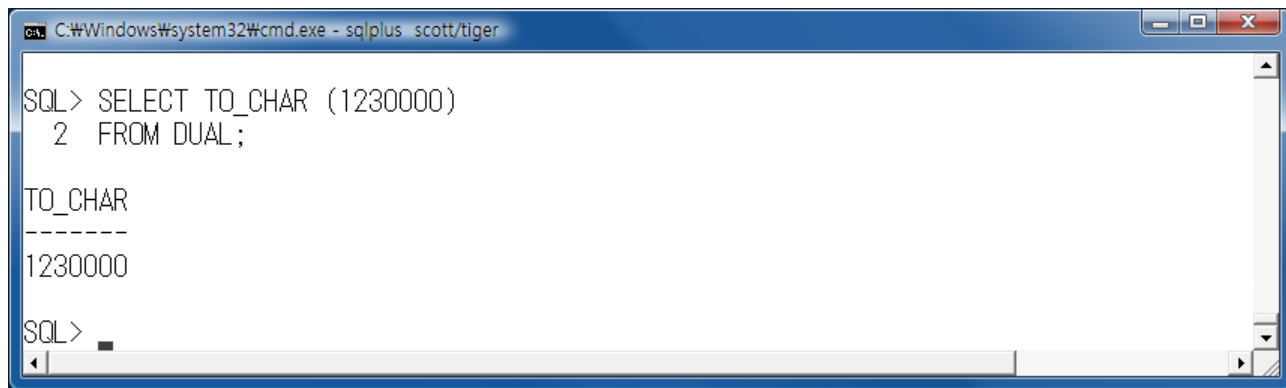
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT TO_CHAR(SYSDATE, 'YYYY/MM/DD, HH24:MI:SS')  
2 FROM DUAL;  
  
TO_CHAR(SYSDATE, 'YYY  
-----  
2009/01/17, 17:07:13  
  
SQL>
```

05 1. 문자형으로 변환하는 TO_CHAR 함수

- 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>
```


05 1. 문자형으로 변환하는 TO_CHAR 함수

- 다음 표는 숫자 출력 형식을 나열한 표입니다.

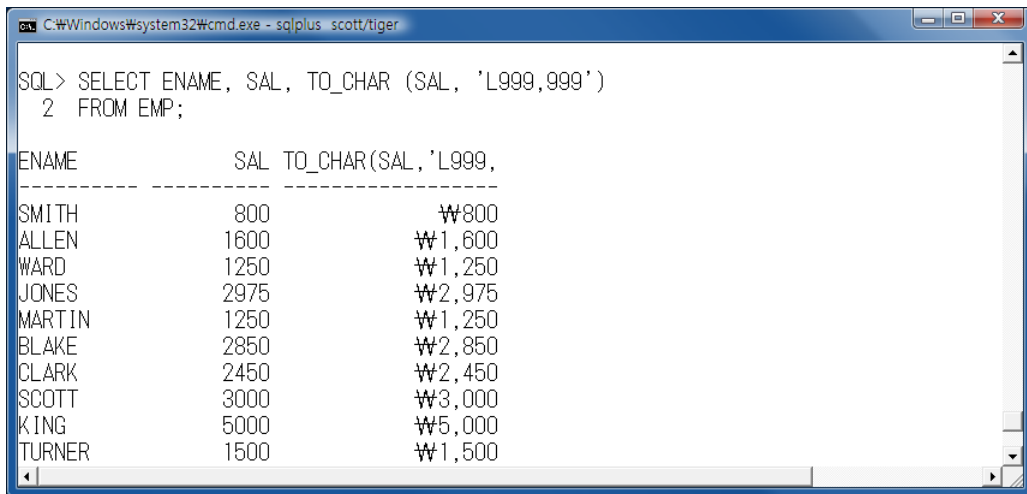
구 분	설 명
0	자릿수를 나타내며 자릿수가 맞지 않을 경우 0으로 채운다.
9	자릿수를 나타내며 자릿수가 맞지 않아도 채우지 않는다.
L	각 지역별 통화 기호를 앞에 표시한다.
.	소수점
,	천 단위 자리 구분

05 1. 문자형으로 변환하는 TO_CHAR 함수

- 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;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - sqlplus scott/tiger". The SQL command entered is "SQL> SELECT ENAME, SAL, TO_CHAR (SAL, 'L999,999') FROM EMP;". The output is a table with three columns: ENAME, SAL, and TO_CHAR(SAL, 'L999,999'). The data is as follows:

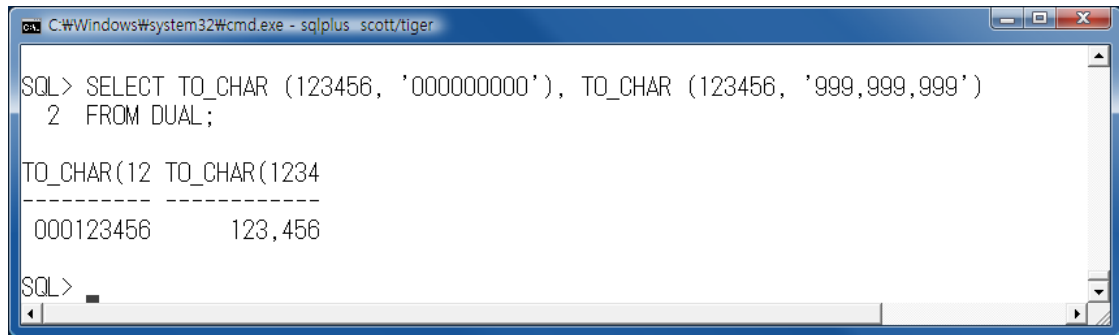
ENAME	SAL	TO_CHAR(SAL, 'L999,999')
SMITH	800	₩800
ALLEN	1600	₩1,600
WARD	1250	₩1,250
JONES	2975	₩2,975
MARTIN	1250	₩1,250
BLAKE	2850	₩2,850
CLARK	2450	₩2,450
SCOTT	3000	₩3,000
KING	5000	₩5,000
TURNER	1500	₩1,500

05 1. 문자형으로 변환하는 TO_CHAR 함수

- 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>
```

05 2. 날짜형으로 변환하는 TO_DATE 함수

- 날짜 형으로 변환하기에 앞서 날짜 형에 대해서 살펴보도록 합시다. 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 문자열을 날짜 형으로 변환합니다.

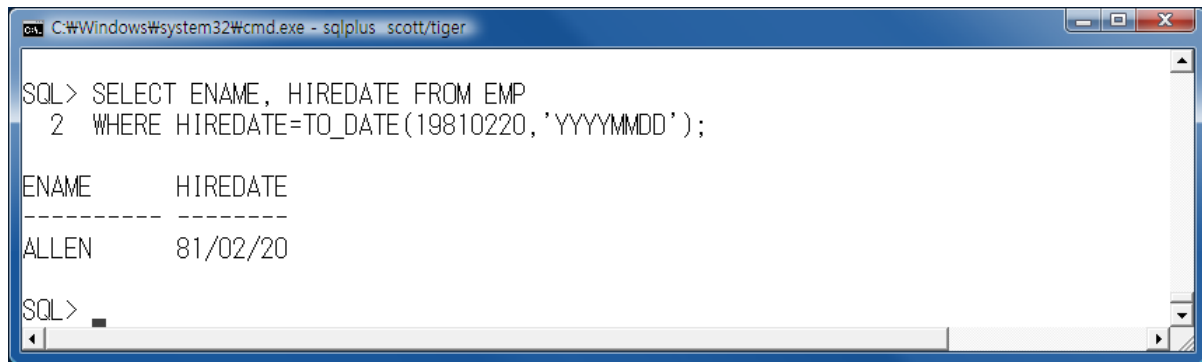
형식	TO_DATE('문자', 'format')
----	-------------------------

05 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')
```



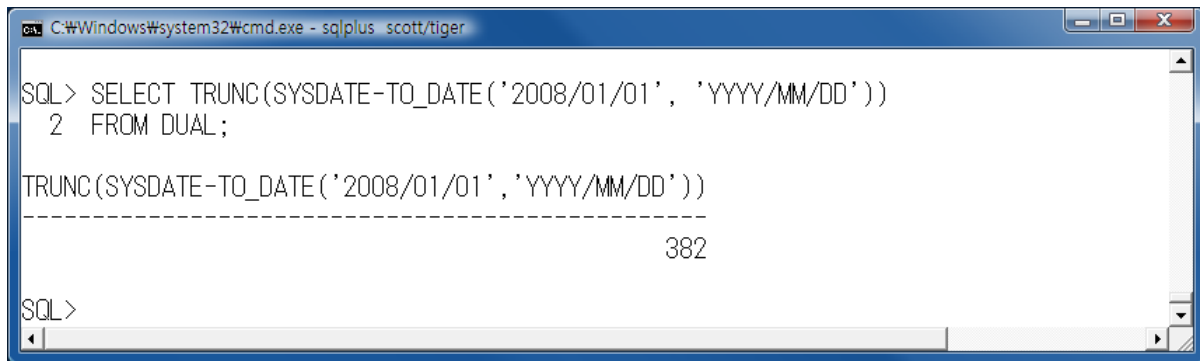
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT ENAME, HIREDATE FROM EMP  
2  WHERE HIREDATE=TO_DATE(19810220,'YYYYMMDD');  
  
ENAME      HIREDATE  
-----  
ALLEN      81/02/20  
  
SQL>
```

05 2. 날짜형으로 변환하는 TO_DATE 함수

- 다음은 올해 며칠이 지났는지 현재 날짜에서 2008/01/01을 뺀 결과를 출력하는 예제입니다.

예

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



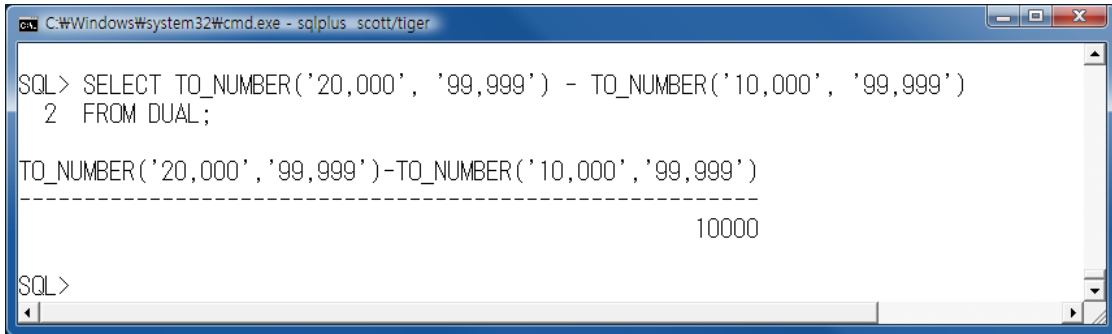
```
C:\Windows\system32\cmd.exe - sqlplus scott/tiger  
  
SQL> SELECT TRUNC(SYSDATE-TO_DATE('2008/01/01', 'YYYY/MM/DD'))  
2 FROM DUAL;  
  
TRUNC(SYSDATE-TO_DATE('2008/01/01', 'YYYY/MM/DD'))  
-----  
382  
  
SQL>
```

05 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;
```



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 following SQL query:

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

The output of the query is displayed below the prompt:

```
TO_NUMBER('20,000', '99,999')-TO_NUMBER('10,000', '99,999')  
-----  
10000
```

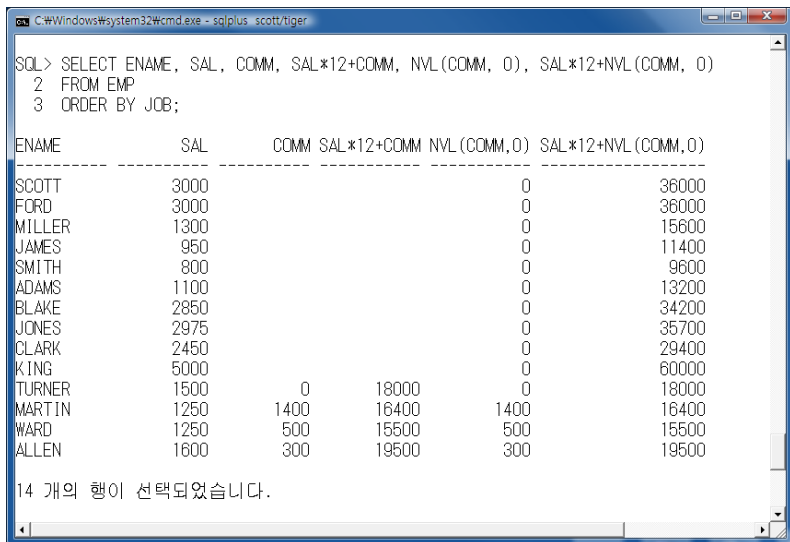
The prompt "SQL>" is visible again at the bottom of the window.

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;
```



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.

- 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.

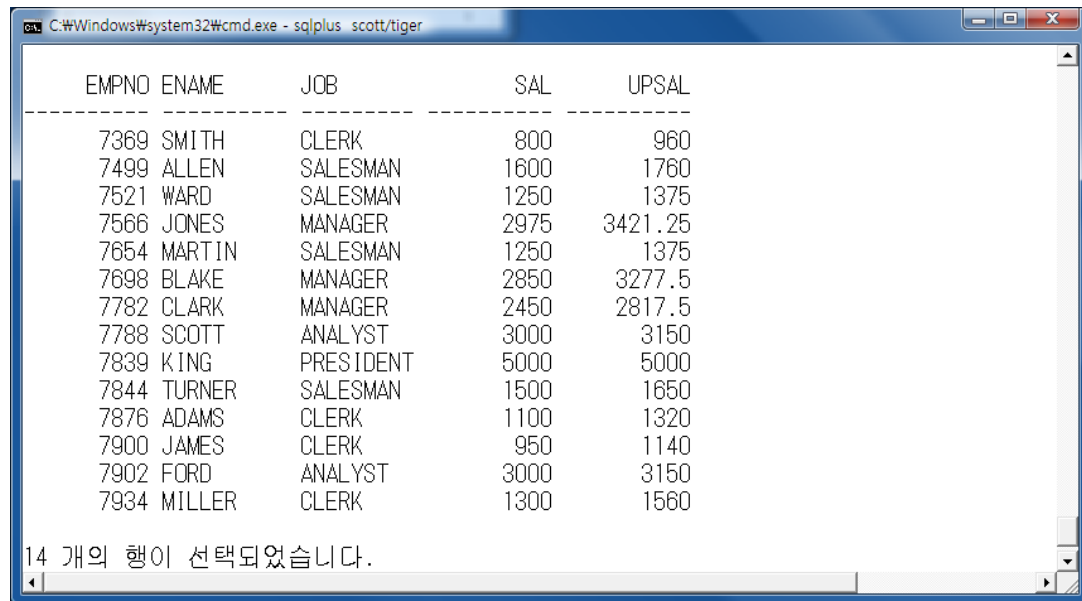
형식	DECODE (표현식, 조건1, 결과1, 조건2, 결과2, 조건3, 결과3, 기본결과n)
----	--

- 부서번호에 해당되는 부서명을 구하는 예제를 이번에는 CASE 함수를 사용하여 작성해 봅시다.

형식

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

- 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.



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

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	<pre>CASE 표현식 WHEN 조건1 THEN 결과1 WHEN 조건2 THEN 결과2 WHEN 조건3 THEN 결과3 ELSE 결과n END</pre>
----	--

08 조건에 따라 서로 다른 처리가 가능한 CASE 함수

- 부서번호에 해당되는 부서명을 구하는 예제를 이번에는 CASE 함수를 사용하여 작성해 봅시다.

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;
```