

04장. 다양한 함수

이번 장에서는 오라클에서 데이터 처리를 위해서 제공하는 위한 다양한 함수(숫자 처리를 위한 함수, 문자 처리를 위한 함수, 날짜 처리를 위한 함수, 데이터 변환을 위한 함수, 어떤 데이터타입에도 사용가능한 일반 함수)에 대해 살펴봅니다.

학습 내용

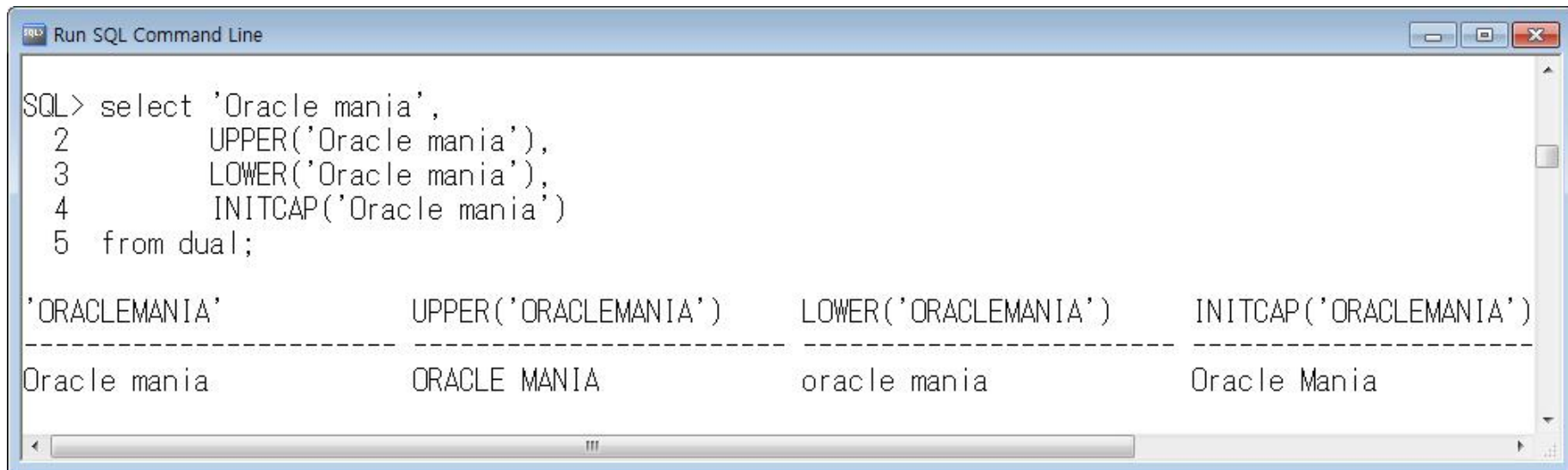
- ❖ 문자 함수
- ❖ 숫자 함수
- ❖ 날짜 함수
- ❖ 변환 함수
- ❖ 일반 함수

학습목표

- ❖ 문자형의 값을 조작하여 변환된 문자 값을 반환하는 문자 함수를 학습합니다.
- ❖ 숫자형 데이터를 조작하여 변환된 숫자 값을 반환하는 숫자 함수를 학습합니다.
- ❖ DATE(날짜)형을 사용하여 결과 값으로 날짜 또는 기간을 얻는 날짜 함수를 학습합니다.
- ❖ 자료형을 변환시키고자 할 때 사용하는 함수를 학습합니다.
- ❖ NULL을 다른 값으로 변환하는 NVL 함수를 학습합니다.
- ❖ 조건에 맞는 문장을 수행하는 DECODE와 CASE에 대해서 학습합니다.

01. 문자 처리 함수(대소문자 변환함수)

구 분	설 명
LOWER	소문자로 변환한다.
UPPER	대문자로 변환한다.
INITCAP	첫 글자만 대문자로 나머지 글자는 소문자로 변환한다.



```
SQL> select 'Oracle mania',
2          UPPER('Oracle mania'),
3          LOWER('Oracle mania'),
4          INITCAP('Oracle mania')
5  from dual;
```

'ORACLEMANIA'	UPPER('ORACLEMANIA')	LOWER('ORACLEMANIA')	INITCAP('ORACLEMANIA')
Oracle mania	ORACLE MANIA	oracle mania	Oracle Mania

01. 문자 처리 함수(문자 길이를 반환하는 함수)

구 분

설 명

LENGTH

문자의 길이를 반환한다.(한글 1Byte)

LENGTHB

문자의 길이를 반환한다.(한글 2Byte)

```
Run SQL Command Line

SQL> select length('OracleMania'), length('오라클매니아')
2 from dual;

LENGTH('ORACLEMANIA') LENGTH('오라클매니아')
-----
11                      6
```

```
Run SQL Command Line

SQL> select lengthb('OracleMania'), lengthb('오라클매니아')
2 from dual;

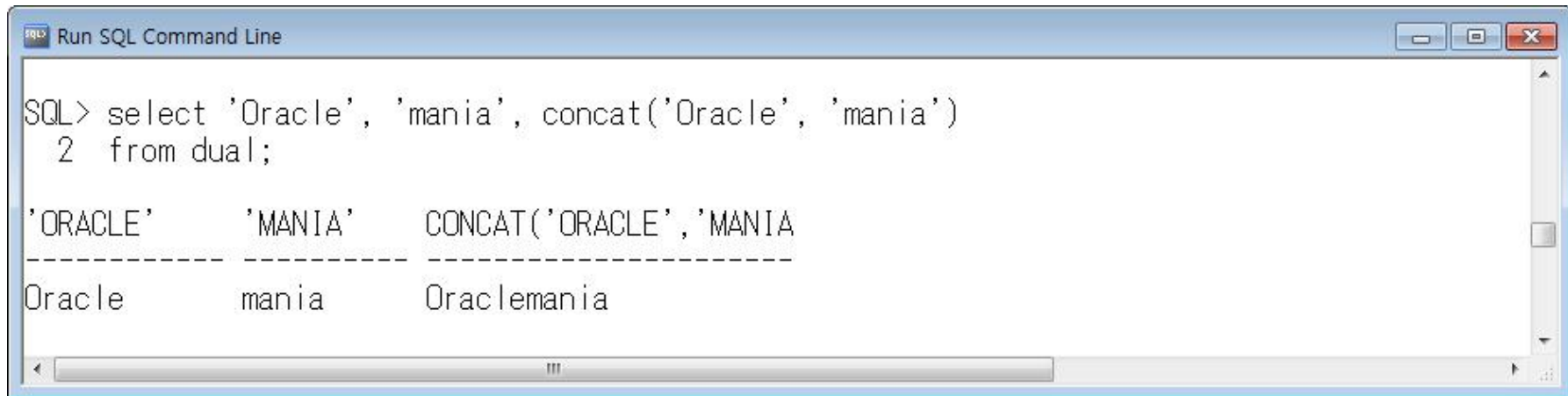
LENGTHB('ORACLEMANIA') LENGTHB('오라클매니아')
-----
11                      18
```

01. 문자 처리 함수(문자 조작 함수)

구 분	설 명
CONCAT	문자의 값을 연결한다.
SUBSTR	문자를 잘라 추출한다.(한글 1Byte)
SUBSTRB	문자를 잘라 추출한다.(한글 2Byte)
INSTR	특정 문자의 위치 값을 반환한다.(한글 1Byte)
INSTRB	특정 문자의 위치 값을 반환한다.(한글 2Byte)
LPAD, RPAD	입력 받은 문자열과 기호를 정렬하여 특정 길이의 문자열로 반환한다.
TRIM	잘라내고 남은 문자를 표시한다.

CONCAT

❖ 문자열을 결합하는 함수



The screenshot shows a window titled "Run SQL Command Line". Inside, the following SQL query is entered and executed:

```
SQL> select 'Oracle', 'mania', concat('Oracle', 'mania')  
2 from dual;
```

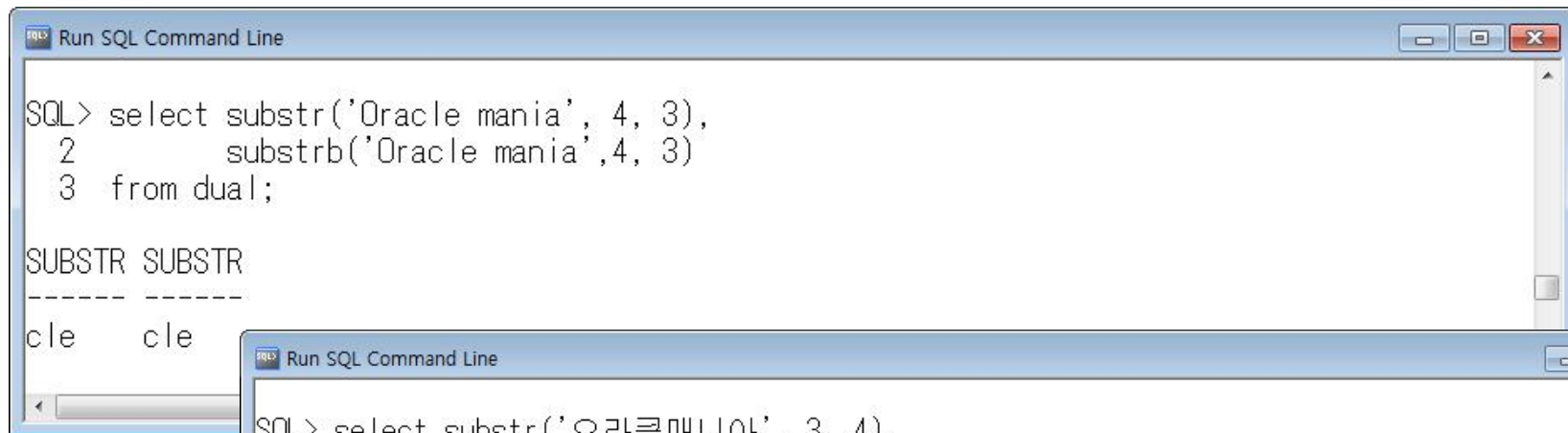
The output is displayed as a table with three columns: 'ORACLE', 'MANIA', and CONCAT('ORACLE','MANIA'). The first row shows the values 'Oracle', 'mania', and 'Oraclemania'.

'ORACLE'	'MANIA'	CONCAT('ORACLE','MANIA')
Oracle	mania	Oraclemania

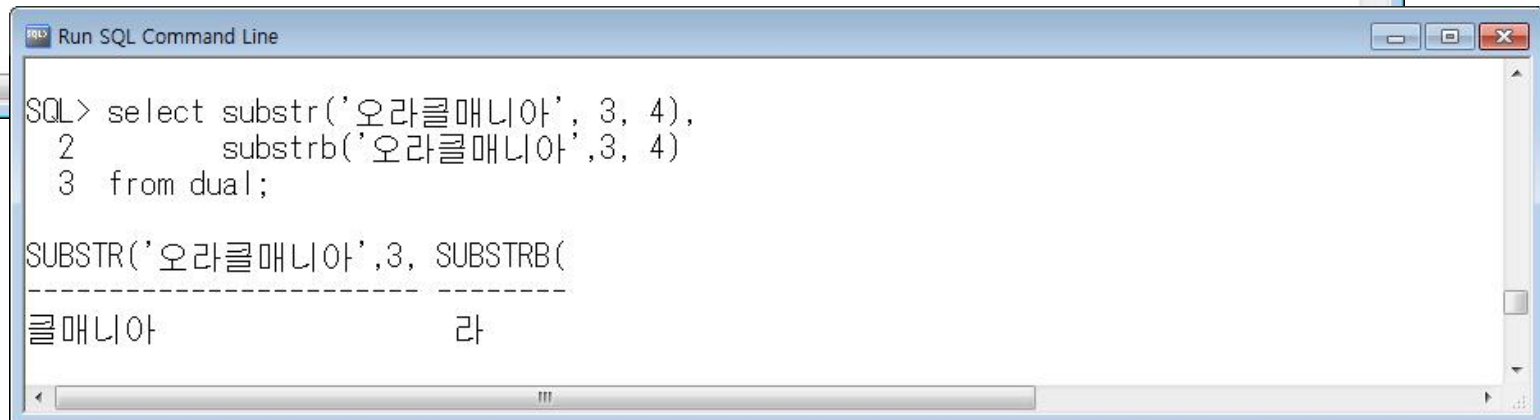
SUBSTR

❖ 기존 문자열에서 일부

형식 SUBSTR(**대상**, **시작위치**, **추출할 개수**)



```
SQL> select substr('Oracle mania', 4, 3),  
2          substrb('Oracle mania',4, 3)  
3  from dual;  
  
SUBSTR SUBSTR  
-----  
cle    cle
```

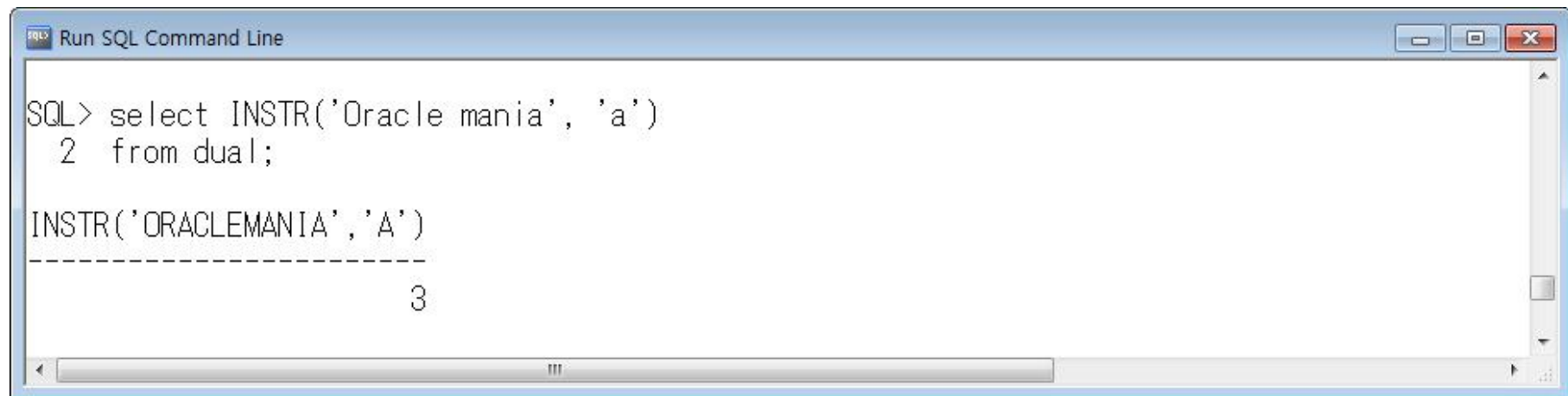


```
SQL> select substr('오라클매니아', 3, 4),  
2          substrb('오라클매니아',3, 4)  
3  from dual;  
  
SUBSTR('오라클매니아',3, SUBSTRB(  
-----  
클매니아          라
```


INSTR

❖ 문자열 내에 해당 문자가 어느 위치에 존재하는지를 알려 줌

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

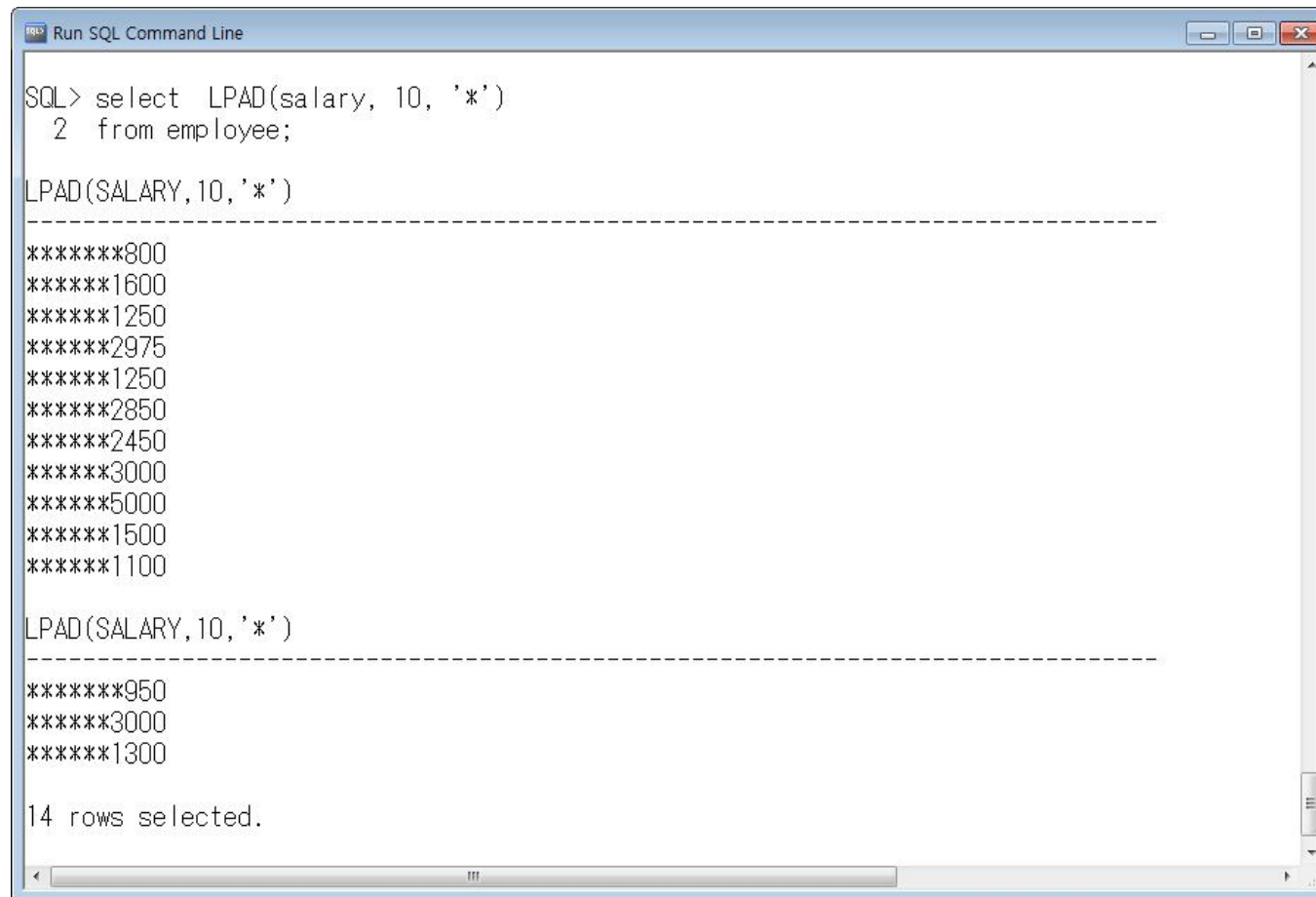


```
SQL> select INSTR('Oracle mania', 'a')
2  from dual;

INSTR('ORACLEMANIA','A')
-----
                        3
```

LPAD

- ❖ 칼럼이나 대상 문자열을 명시된 자릿수에서 오른쪽에 나타내고, 남은 왼쪽 자리를 특정 기호로 채움



```
Run SQL Command Line
SQL> select LPAD(salary, 10, '*')
      2 from employee;

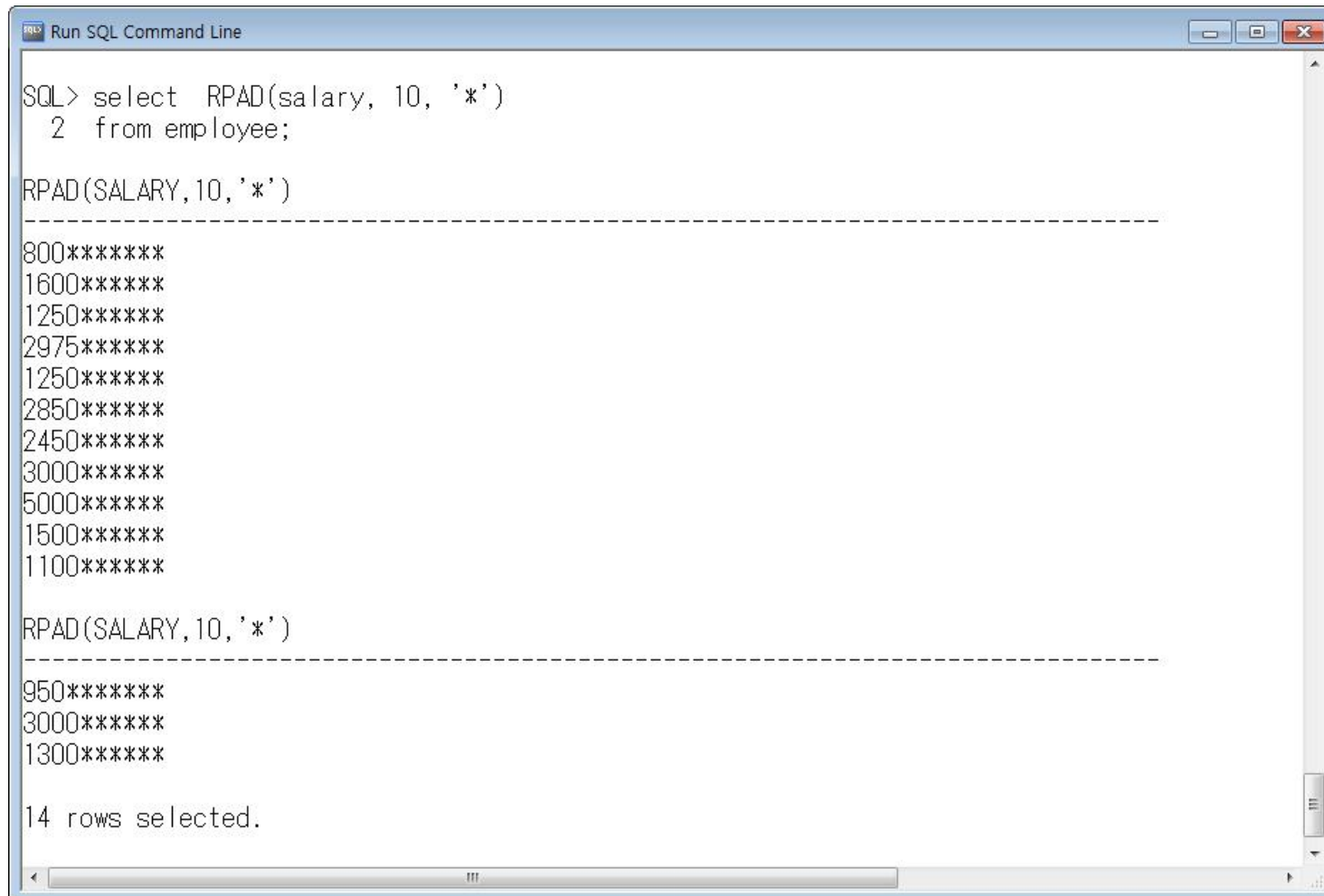
LPAD(SALARY,10,'*')
-----
*****800
*****1600
*****1250
*****2975
*****1250
*****2850
*****2450
*****3000
*****5000
*****1500
*****1100

LPAD(SALARY,10,'*')
-----
*****950
*****3000
*****1300

14 rows selected.
```

RPAD

- ❖ 대상 문자열을 명시된 자릿수에서 왼쪽에 나타내고, 남은 오른쪽 자리를 특정 기호로 채움



```
SQL> select  RPAD(salary, 10, '*')
          2  from employee;

RPAD(SALARY,10,'*')
-----
800*****
1600*****
1250*****
2975*****
1250*****
2850*****
2450*****
3000*****
5000*****
1500*****
1100*****

RPAD(SALARY,10,'*')
-----
950*****
3000*****
1300*****

14 rows selected.
```

TRIM

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

```
SQL> select 'Oracle mania',
2          TRIM('O' FROM 'Oracle mania')
3  from dual;

'ORACLEMANIA'          TRIM('O' FROM 'ORACLEMAN
-----
Oracle mania          racle mania
```

```
SQL> select TRIM(' Oracle mania ')
2  from dual;

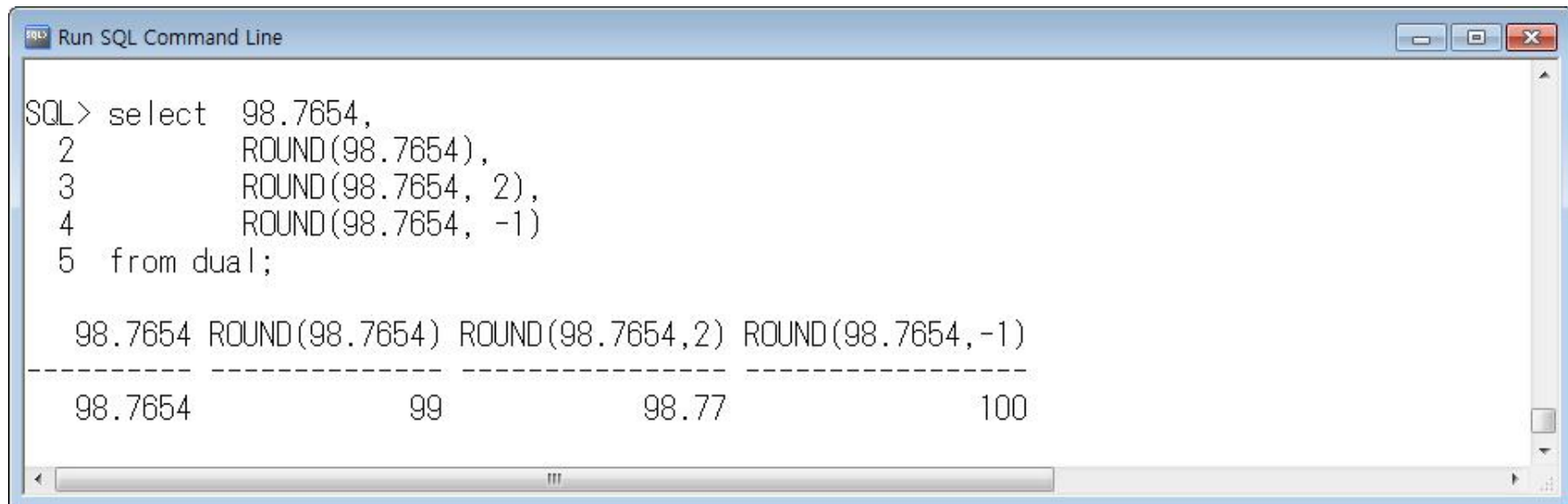
TRIM('ORACLEMANIA')
-----
Oracle mania
```

02. 숫자 함수

구 분	설 명
ROUND	특정 자릿수에서 반올림한다.
TRUNC	특정 자릿수에서 잘라낸다. (버림)
MOD	입력 받은 수를 나눈 나머지 값을 반환한다.

ROUND

❖ 반올림하기 위한 함수



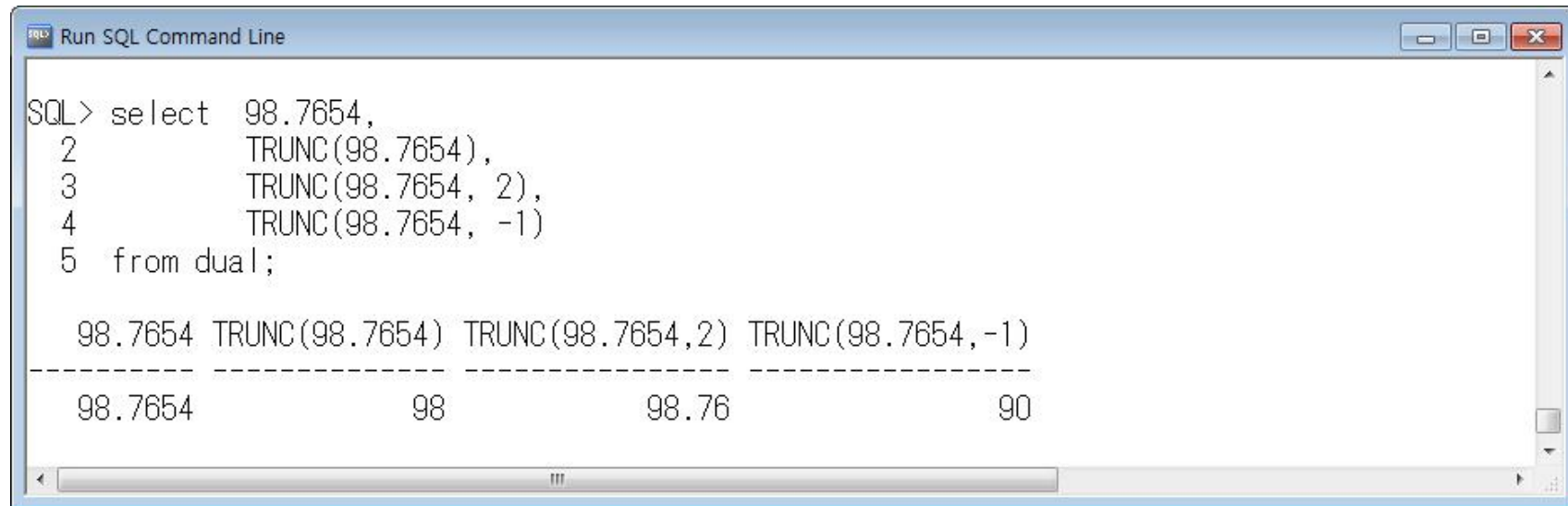
The screenshot shows a SQL Command Line window with the following SQL query and its results:

```
SQL> select 98.7654,  
2          ROUND(98.7654),  
3          ROUND(98.7654, 2),  
4          ROUND(98.7654, -1)  
5  from dual;
```

98.7654	ROUND(98.7654)	ROUND(98.7654,2)	ROUND(98.7654,-1)
98.7654	99	98.77	100

TRUNC

❖ 소수점 n째 자리까지 남기고 나머지는 버림



The screenshot shows a window titled "Run SQL Command Line" with a SQL prompt. The command entered is a SELECT statement with five columns: the original value 98.7654, and its truncations at 0, 2, and -1 decimal places. The output shows the original value and the three truncated results, with a dashed line separating the header from the data row.

```
SQL> select 98.7654,  
2          TRUNC(98.7654),  
3          TRUNC(98.7654, 2),  
4          TRUNC(98.7654, -1)  
5  from dual;
```

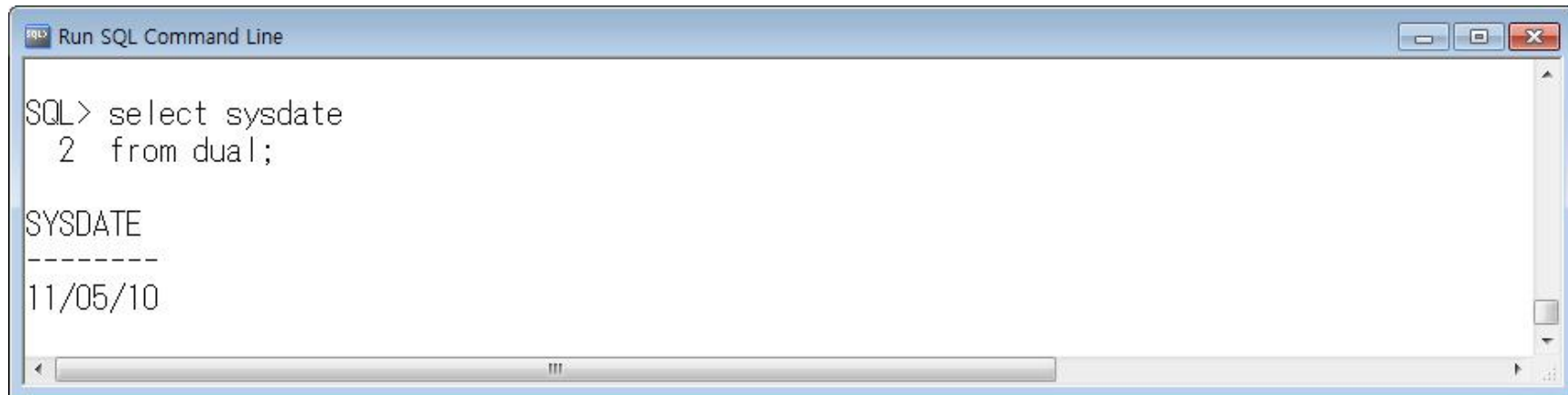
98.7654	TRUNC(98.7654)	TRUNC(98.7654,2)	TRUNC(98.7654,-1)
98.7654	98	98.76	90

03. 날짜 함수

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.

현재 날짜를 반환하는 SYSDATE 함수

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



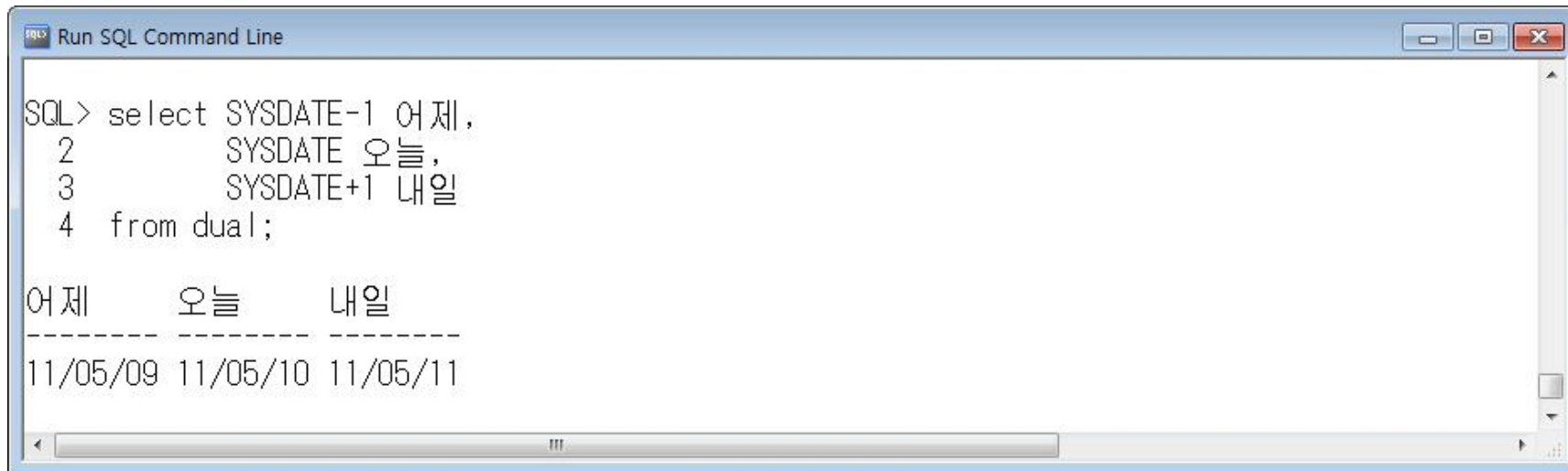
```
SQL> select sysdate
2  from dual;

SYSDATE
-----
11/05/10
```

The screenshot shows a window titled "Run SQL Command Line". Inside, the SQL command "select sysdate" is entered on the first line, and "2 from dual;" is entered on the second line. The output shows the column name "SYSDATE" followed by a dashed line separator and the date "11/05/10".

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.



The screenshot shows a window titled "Run SQL Command Line". Inside, the following SQL query is entered:

```
SQL> select SYSDATE-1 어제,  
2          SYSDATE 오늘,  
3          SYSDATE+1 내일  
4  from dual;
```

The output of the query is displayed as a table with three columns: 어제, 오늘, and 내일. The dates shown are 11/05/09, 11/05/10, and 11/05/11 respectively.

어제	오늘	내일
11/05/09	11/05/10	11/05/11

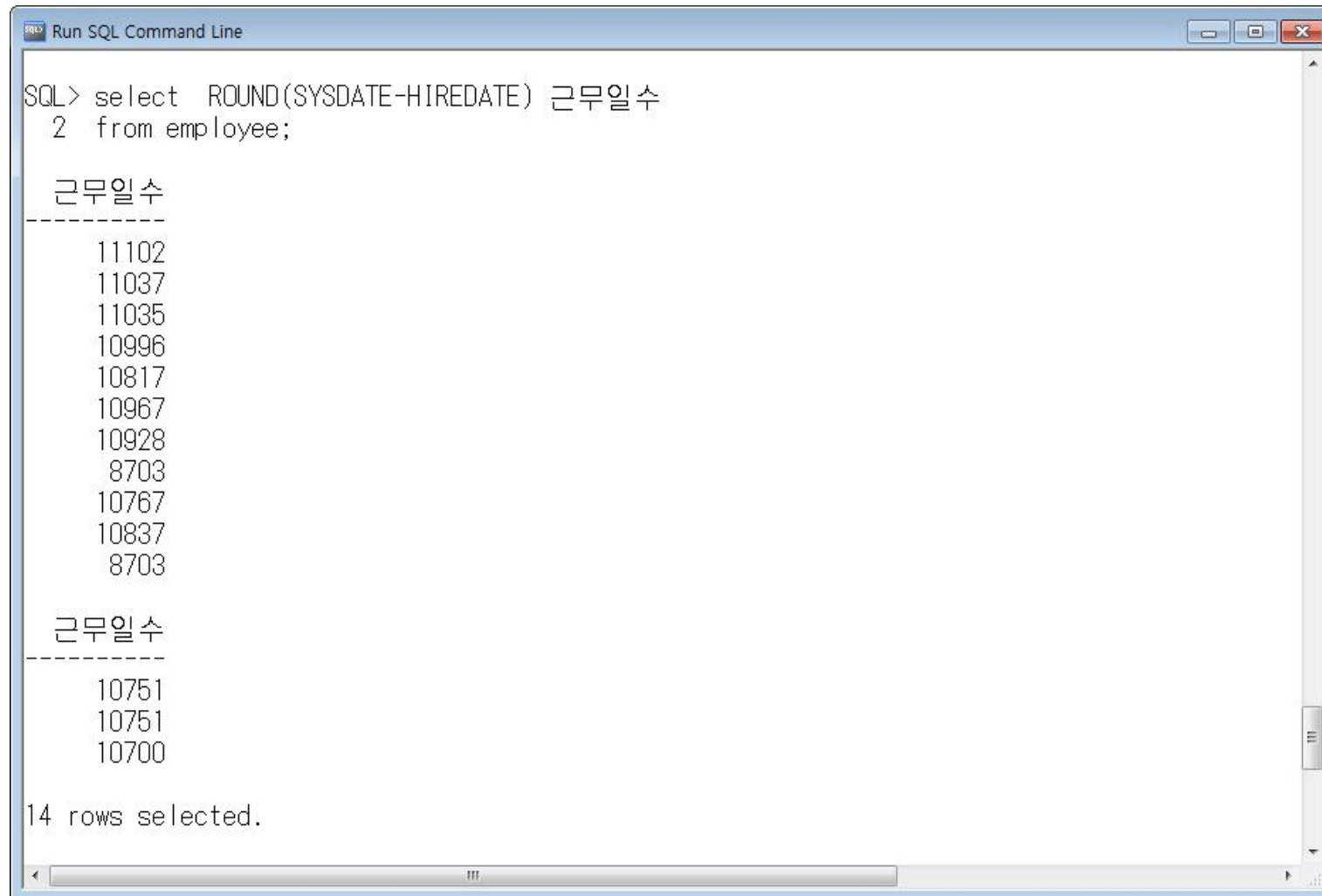
ROUND

❖ 날짜에 대한 반올림을 할 수 있습니다.

형식 ROUND (*date, format*)

포맷 모델	단위
CC, SCC	4자리 연도의 끝 두 글자를 기준으로 반올림
SYYY, YYYYY, 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

ROUND



Run SQL Command Line

```
SQL> select  ROUND(SYSDATE-HIREDATE) 근무일수
2  from employee;
```

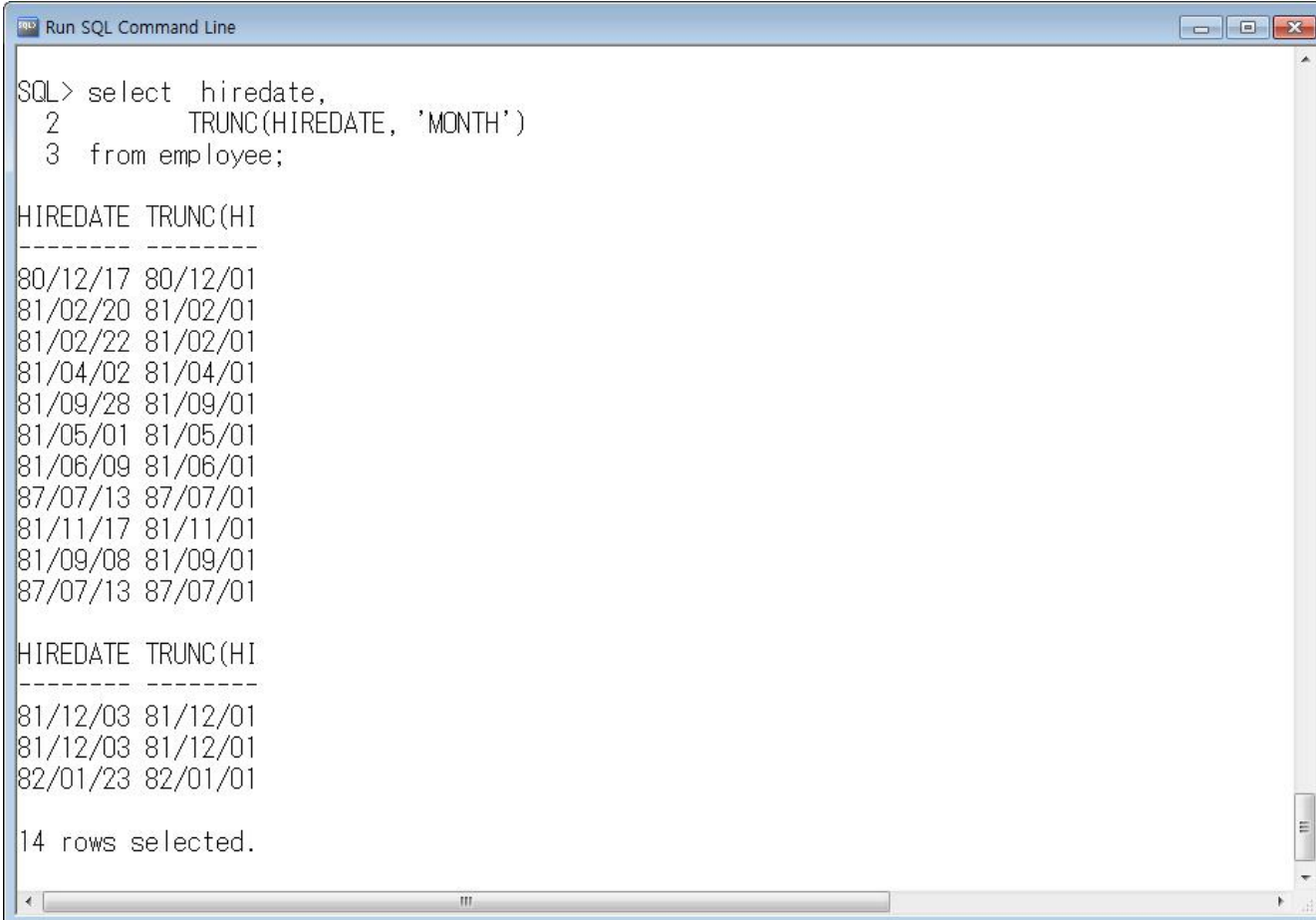
근무일수
11102
11037
11035
10996
10817
10967
10928
8703
10767
10837
8703
10751
10751
10700

14 rows selected.

TRUNC

❖ 날짜를 잘라낼 수 있습니다.

형식 **TRUNC** (*date, format*)



```
SQL> select hiredate,  
2          TRUNC(HIREDATE, 'MONTH')  
3  from employee;
```

HIREDATE	TRUNC(HI
80/12/17	80/12/01
81/02/20	81/02/01
81/02/22	81/02/01
81/04/02	81/04/01
81/09/28	81/09/01
81/05/01	81/05/01
81/06/09	81/06/01
87/07/13	87/07/01
81/11/17	81/11/01
81/09/08	81/09/01
87/07/13	87/07/01

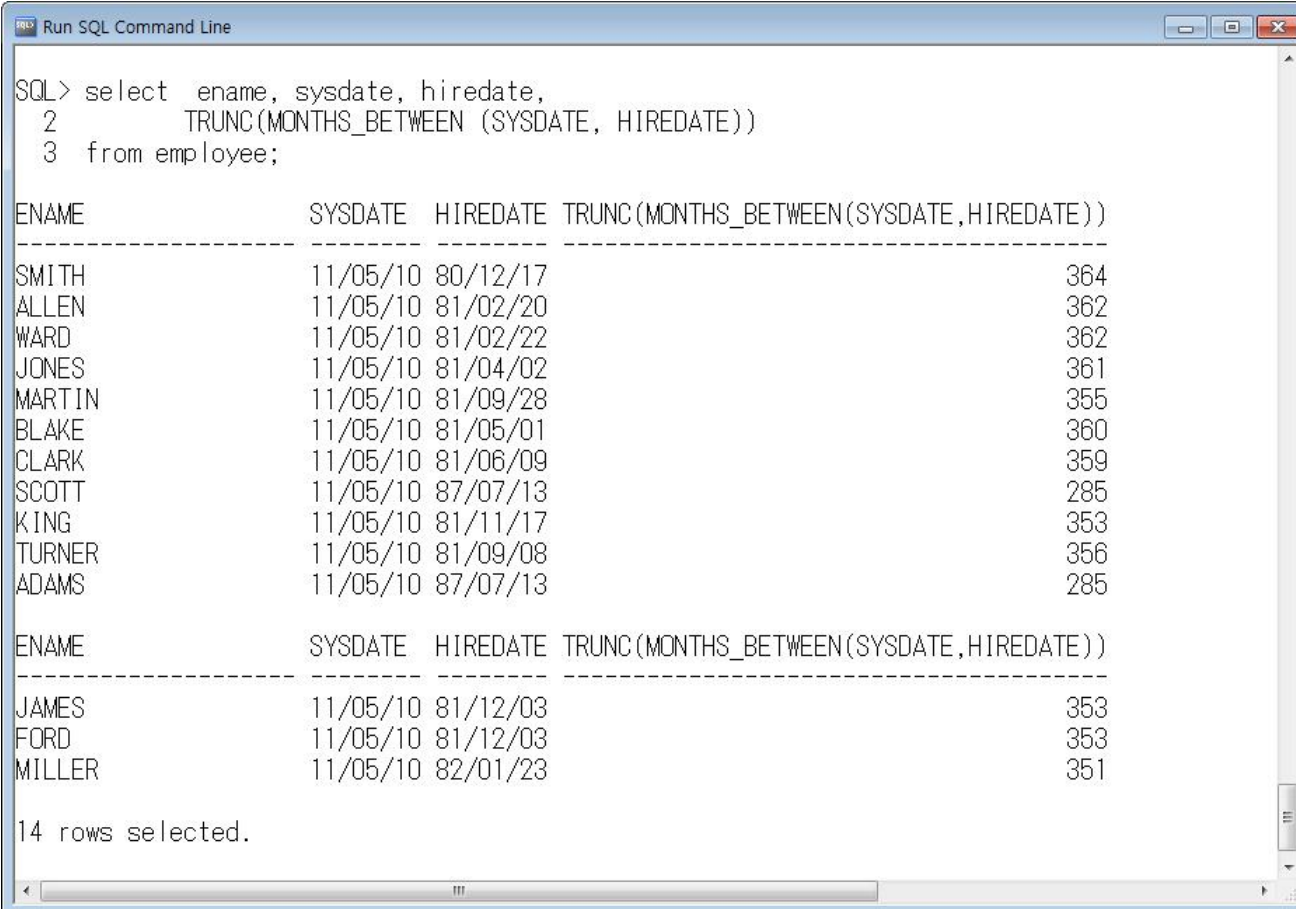
HIREDATE	TRUNC(HI
81/12/03	81/12/01
81/12/03	81/12/01
82/01/23	82/01/01

14 rows selected.

MONTHS_BETWEEN

- ❖ MONTHS_BETWEEN 함수는 날짜와 날짜 사이의 개월 수를 구하는 함수입니다.

형식 MONTHS_BETWEEN (*date1*, *date2*)



```
SQL> select  ename, sysdate, hiredate,
2            TRUNC(MONTHS_BETWEEN (SYSDATE, HIREDATE))
3            from employee;
```

ENAME	SYSDATE	HIREDATE	TRUNC(MONTHS_BETWEEN(SYSDATE,HIREDATE))
SMITH	11/05/10	80/12/17	364
ALLEN	11/05/10	81/02/20	362
WARD	11/05/10	81/02/22	362
JONES	11/05/10	81/04/02	361
MARTIN	11/05/10	81/09/28	355
BLAKE	11/05/10	81/05/01	360
CLARK	11/05/10	81/06/09	359
SCOTT	11/05/10	87/07/13	285
KING	11/05/10	81/11/17	353
TURNER	11/05/10	81/09/08	356
ADAMS	11/05/10	87/07/13	285

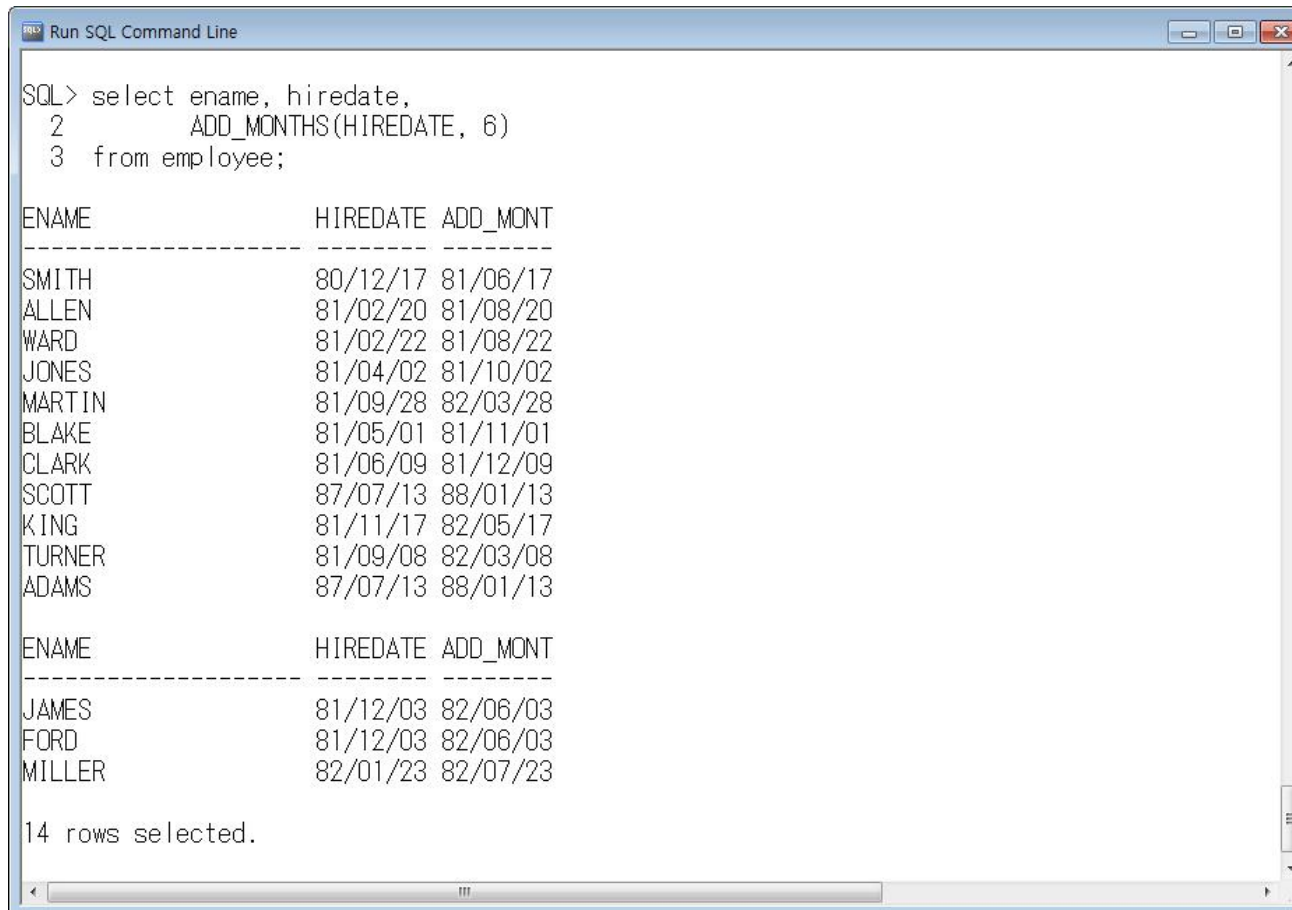
ENAME	SYSDATE	HIREDATE	TRUNC(MONTHS_BETWEEN(SYSDATE,HIREDATE))
JAMES	11/05/10	81/12/03	353
FORD	11/05/10	81/12/03	353
MILLER	11/05/10	82/01/23	351

14 rows selected.

ADD_MONTHS

❖ 특정 개월 수를 더한 날짜를 구하는 함수

형식 **ADD_MONTHS** (*date*, *number*)



```
SQL> select ename, hiredate,
2          ADD_MONTHS(HIREDATE, 6)
3          from employee;
```

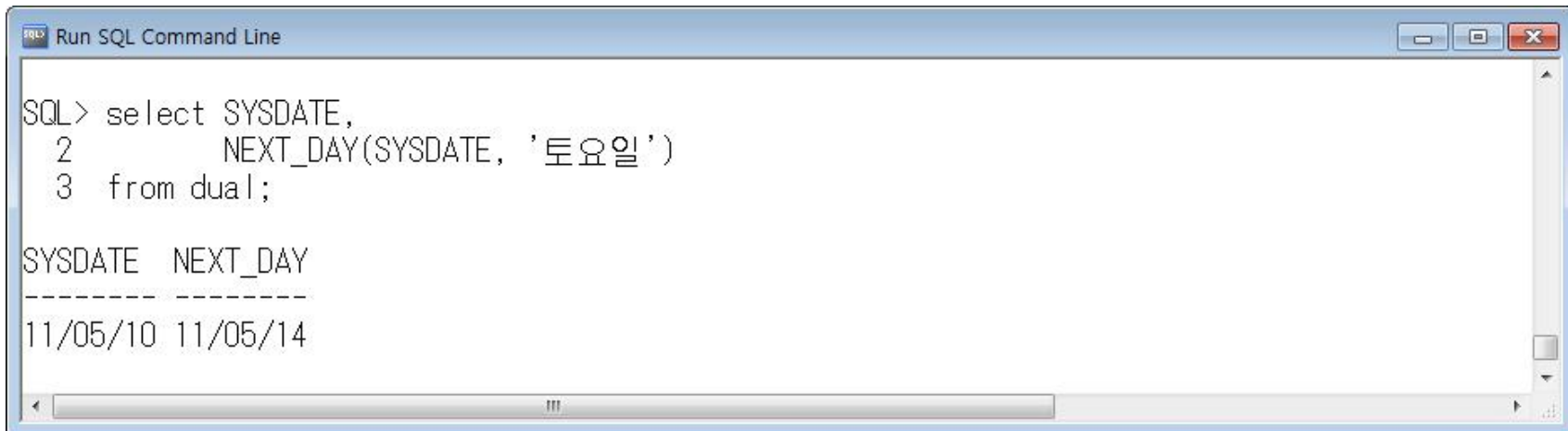
ENAME	HIREDATE	ADD_MONT
SMITH	80/12/17	81/06/17
ALLEN	81/02/20	81/08/20
WARD	81/02/22	81/08/22
JONES	81/04/02	81/10/02
MARTIN	81/09/28	82/03/28
BLAKE	81/05/01	81/11/01
CLARK	81/06/09	81/12/09
SCOTT	87/07/13	88/01/13
KING	81/11/17	82/05/17
TURNER	81/09/08	82/03/08
ADAMS	87/07/13	88/01/13
ENAME	HIREDATE	ADD_MONT
JAMES	81/12/03	82/06/03
FORD	81/12/03	82/06/03
MILLER	82/01/23	82/07/23

14 rows selected.

NEXT_DAY

- ❖ NEXT_DAY 함수는 해당 날짜를 기준으로 최초로 도래하는 요일에 해당되는 날짜를 반환하는 함수입니다.

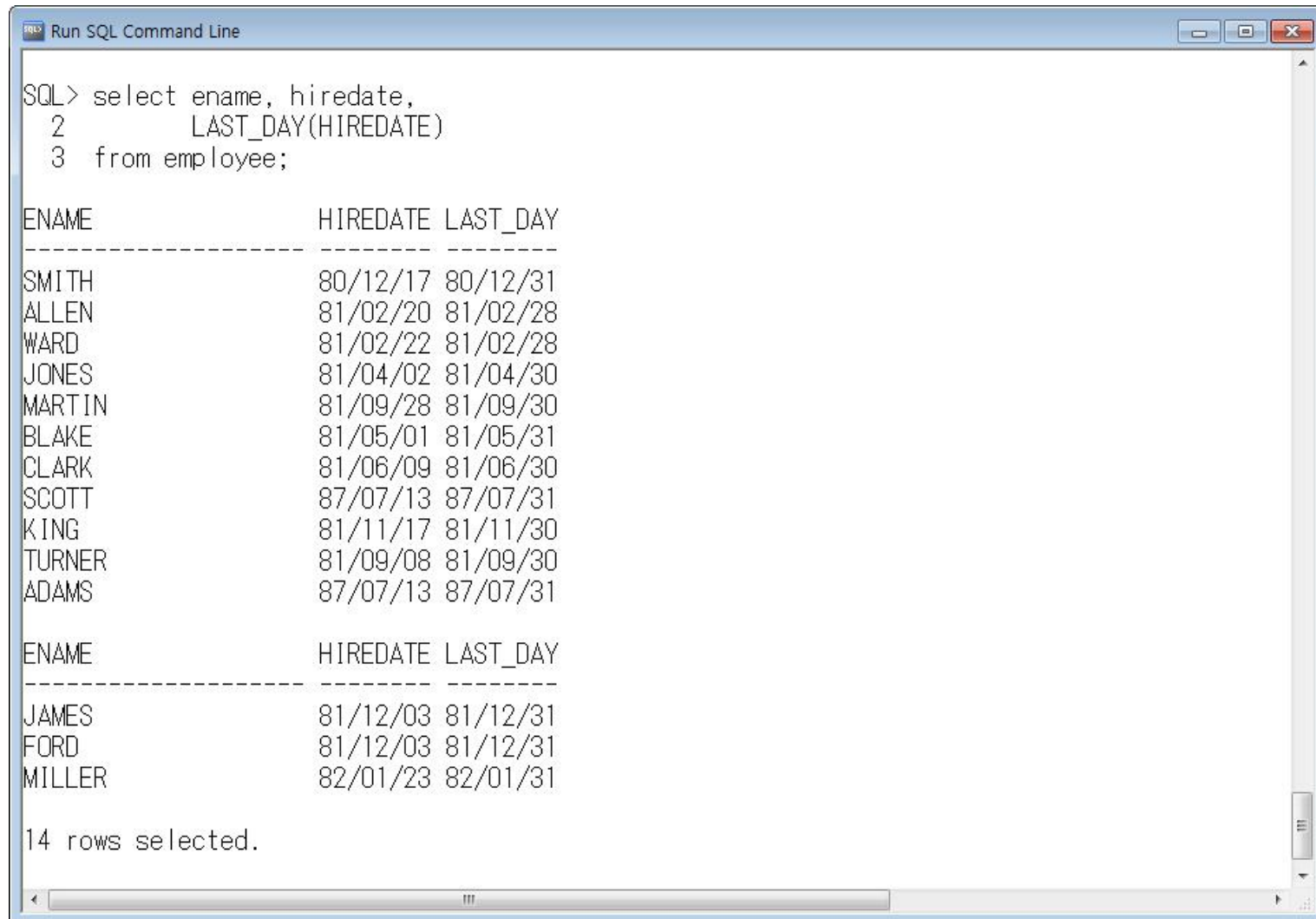
형식 NEXT_DAY (*date*, *요일*)



```
SQL> select SYSDATE,  
2         NEXT_DAY(SYSDATE, '토요일')  
3   from dual;  
  
SYSDATE  NEXT_DAY  
-----  
11/05/10 11/05/14
```


LAST_DAY

- ❖ **LAST_DAY** 함수는 해당 날짜가 속한 달의 마지막 날짜를 반환하는 함수입니다.



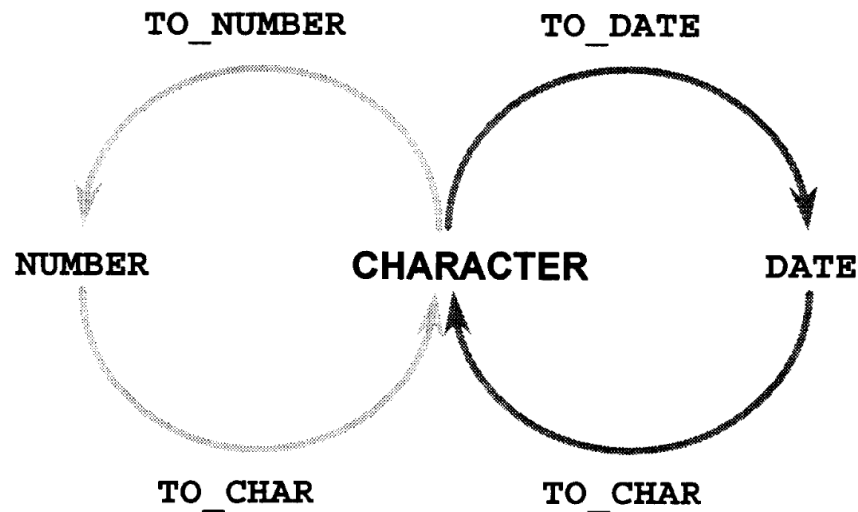
```
SQL> select ename, hiredate,
2          LAST_DAY(HIREDATE)
3  from employee;
```

ENAME	HIREDATE	LAST_DAY
SMITH	80/12/17	80/12/31
ALLEN	81/02/20	81/02/28
WARD	81/02/22	81/02/28
JONES	81/04/02	81/04/30
MARTIN	81/09/28	81/09/30
BLAKE	81/05/01	81/05/31
CLARK	81/06/09	81/06/30
SCOTT	87/07/13	87/07/31
KING	81/11/17	81/11/30
TURNER	81/09/08	81/09/30
ADAMS	87/07/13	87/07/31
JAMES	81/12/03	81/12/31
FORD	81/12/03	81/12/31
MILLER	82/01/23	82/01/31

14 rows selected.

04. 형 변환 함수

- ❖ 오라클에서 데이터형으로 변환해야 하는 경우에는 TO_NUMBER, TO_CHAR, TO_DATE를 사용



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

TO_CHAR

❖ 날짜나 숫자를 문자로 변환하기 위해서 사용

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

TO_CHAR

```
Run SQL Command Line

SQL> select ename, hiredate,
2         TO_CHAR(hiredate, 'YY-MM'),
3         TO_CHAR(hiredate, 'YYYY/MM/DD DAY')
4 from employee;
```

ENAME	HIREDATE	TO_CHAR(HI	TO_CHAR(HIREDATE, 'YYYY/MM/DDDAY')
SMITH	80/12/17	80-12	1980/12/17 수요일
ALLEN	81/02/20	81-02	1981/02/20 금요일
WARD	81/02/22	81-02	1981/02/22 일요일
JONES	81/04/02	81-04	1981/04/02 목요일
MARTIN	81/09/28	81-09	1981/09/28 월요일
BLAKE	81/05/01	81-05	1981/05/01 금요일
CLARK	81/06/09	81-06	1981/06/09 화요일
SCOTT	87/07/13	87-07	1987/07/13 월요일
KING	81/11/17	81-11	1981/11/17 화요일
TURNER	81/09/08	81-09	1981/09/08 화요일
ADAMS	87/07/13	87-07	1987/07/13 월요일

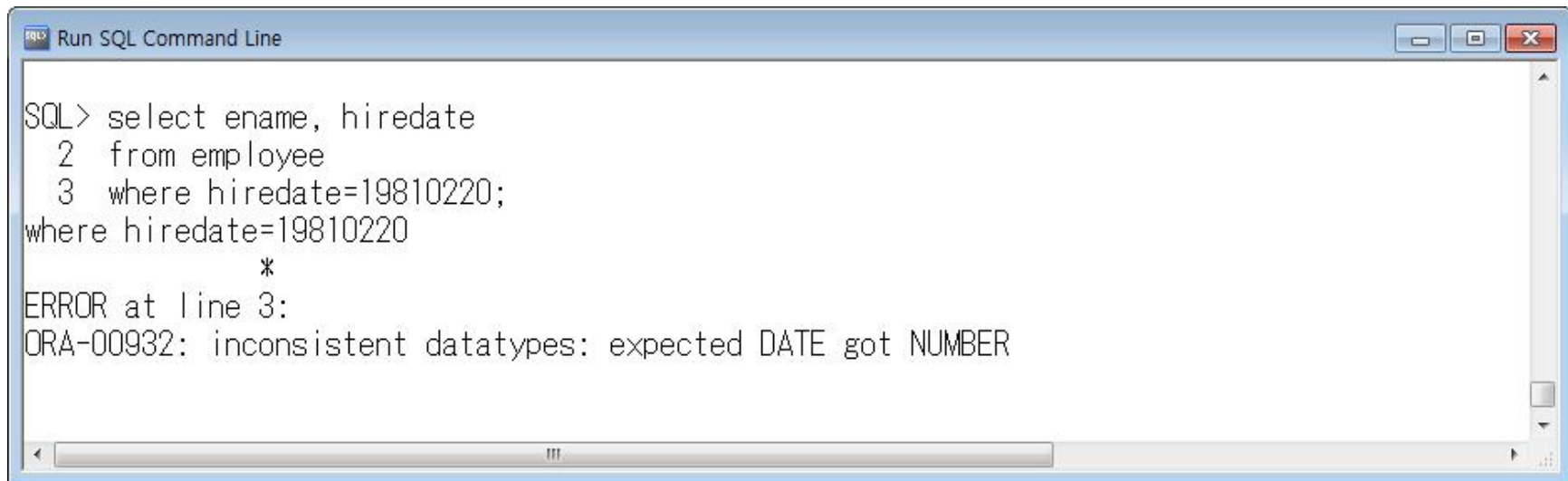
ENAME	HIREDATE	TO_CHAR(HI	TO_CHAR(HIREDATE, 'YYYY/MM/DDDAY')
JAMES	81/12/03	81-12	1981/12/03 목요일
FORD	81/12/03	81-12	1981/12/03 목요일
MILLER	82/01/23	82-01	1982/01/23 토요일

14 rows selected.

TO_DATE

❖ 문자열을 날짜 형으로 변환합니다.

형식 **TO_DATE('문자' , 'format')**

A screenshot of a 'Run SQL Command Line' window. The window has a title bar with 'Run SQL Command Line' and standard Windows window controls. The main area contains the following text:

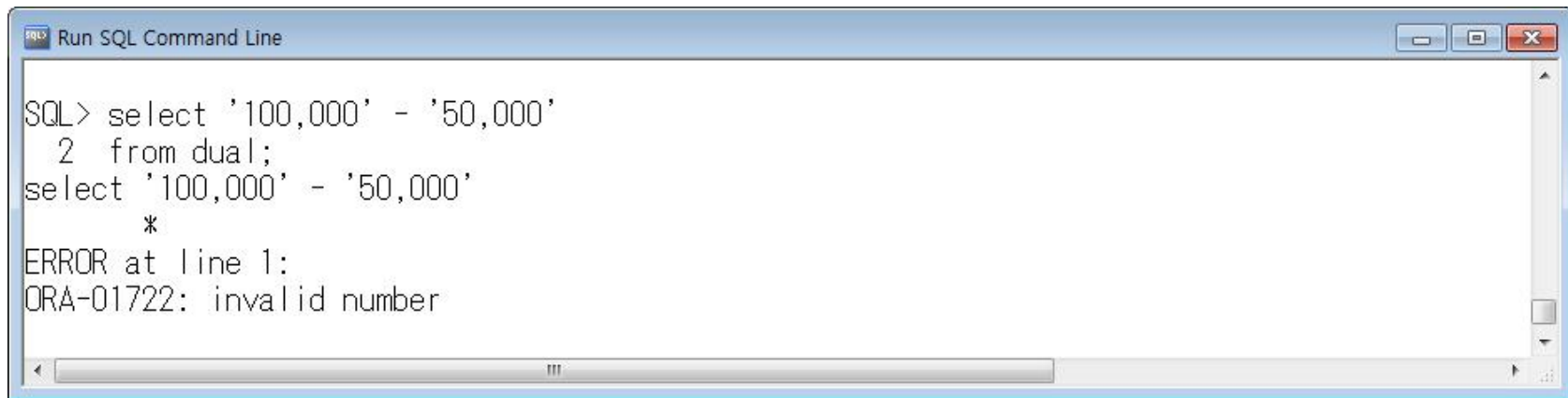
```
SQL> select ename, hiredate
2  from employee
3  where hiredate=19810220;
where hiredate=19810220
          *
```

Below the SQL text, an error message is displayed:

```
ERROR at line 3:
ORA-00932: inconsistent datatypes: expected DATE got NUMBER
```

TO_NUMBER

❖ 특정 데이터를 숫자형으로 변환해 주는 함수



The screenshot shows a window titled "Run SQL Command Line". Inside, the following SQL commands are entered:

```
SQL> select '100,000' - '50,000'  
      2  from dual;  
select '100,000' - '50,000'  
      *
```

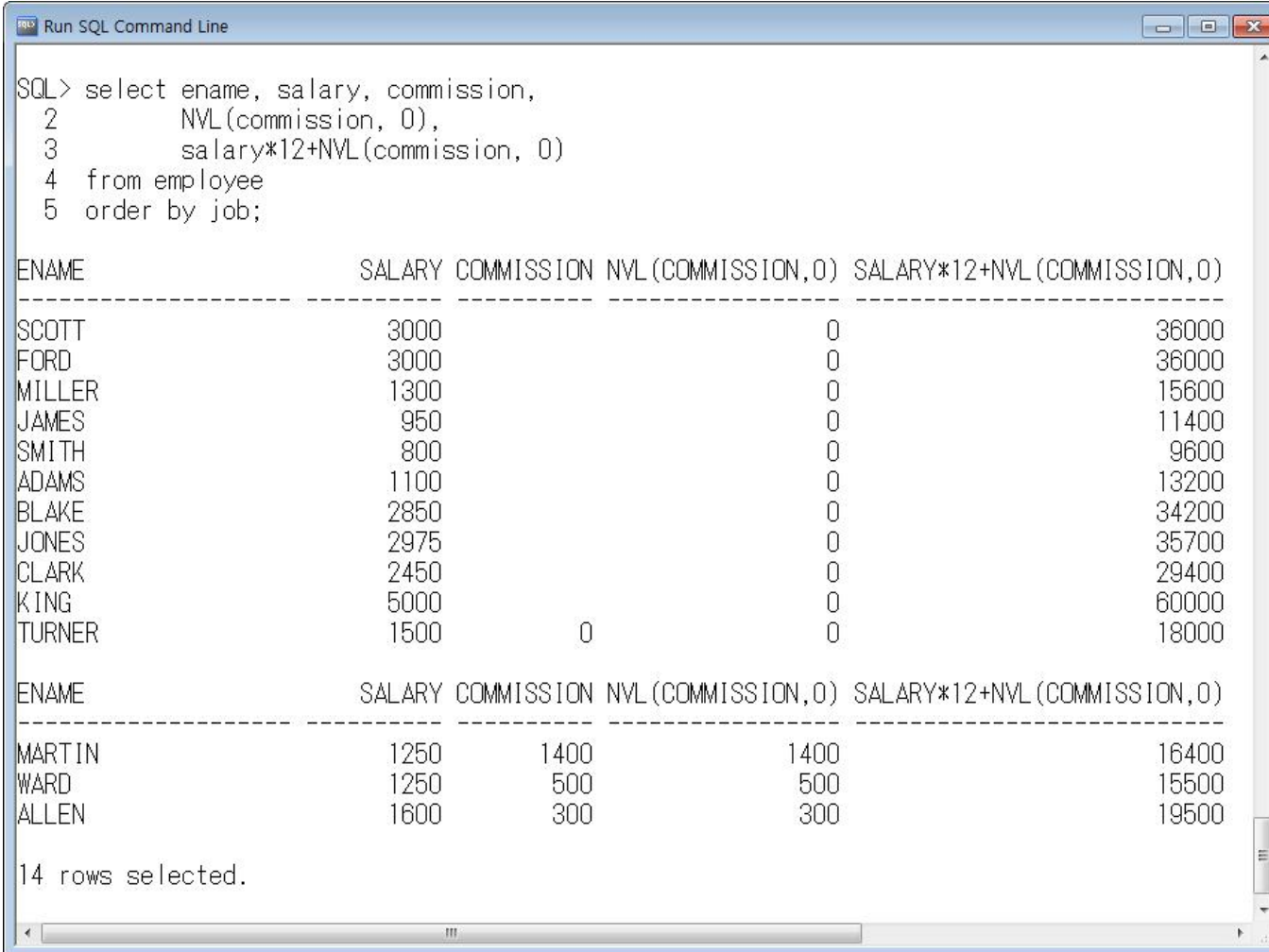
The output shows an error:

```
ERROR at line 1:  
ORA-01722: invalid number
```

05. 일반 함수 (NVL 함수)

❖ NULL을 0 또는 다른 값으로 변환하기 위해서 사용하는 함수

형식 NVL(expr1, expr2)



```
SQL> select ename, salary, commission,  
2          NVL(commission, 0),  
3          salary*12+NVL(commission, 0)  
4  from employee  
5  order by job;
```

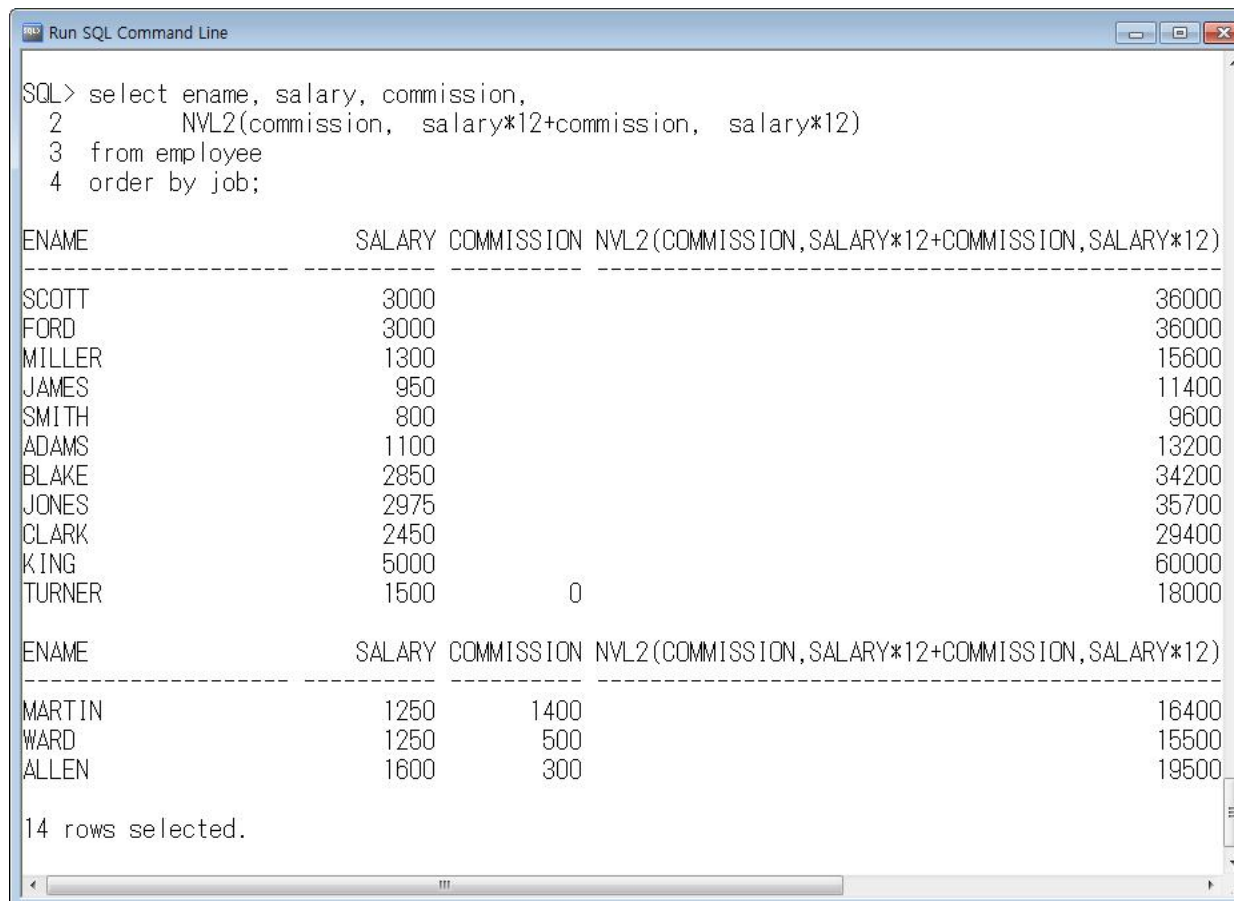
ENAME	SALARY	COMMISSION	NVL(COMMISSION,0)	SALARY*12+NVL(COMMISSION,0)
SCOTT	3000		0	36000
FORD	3000		0	36000
MILLER	1300		0	15600
JAMES	950		0	11400
SMITH	800		0	9600
ADAMS	1100		0	13200
BLAKE	2850		0	34200
JONES	2975		0	35700
CLARK	2450		0	29400
KING	5000		0	60000
TURNER	1500	0	0	18000
MARTIN	1250	1400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

14 rows selected.

NVL 2 함수

- ❖ expr1을 검사하여 그 결과가 널이면 expr2를 반환하고 널이 아니면 expr3을 반환

형식 NVL2(expr1, expr2, expr3)



```
SQL> select ename, salary, commission,
2         NVL2(commission, salary*12+commission, salary*12)
3   from employee
4  order by job;
```

ENAME	SALARY	COMMISSION	NVL2(COMMISSION,SALARY*12+COMMISSION,SALARY*12)
SCOTT	3000		36000
FORD	3000		36000
MILLER	1300		15600
JAMES	950		11400
SMITH	800		9600
ADAMS	1100		13200
BLAKE	2850		34200
JONES	2975		35700
CLARK	2450		29400
KING	5000		60000
TURNER	1500	0	18000

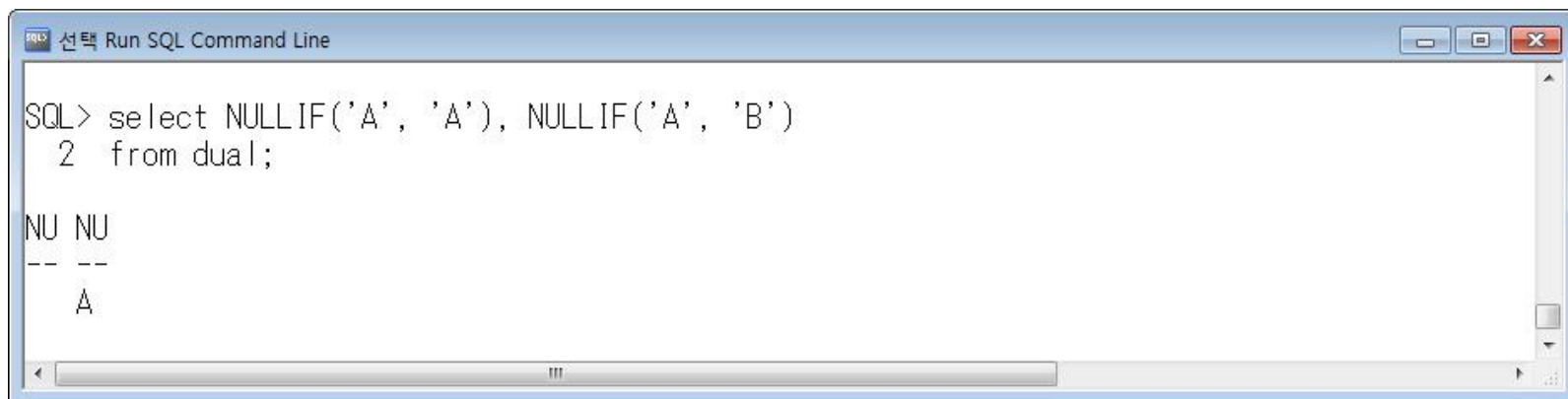
ENAME	SALARY	COMMISSION	NVL2(COMMISSION,SALARY*12+COMMISSION,SALARY*12)
MARTIN	1250	1400	16400
WARD	1250	500	15500
ALLEN	1600	300	19500

14 rows selected.

NULLIF 함수

- ❖ 두 표현식을 비교하여 동일한 경우에는 널을 반환하고 동일하지 않으면 첫 번째 표현식을 반환

형식 NVLIF(expr1, expr2)



```
선택 Run SQL Command Line

SQL> select NULLIF('A', 'A'), NULLIF('A', 'B')
       2  from dual;

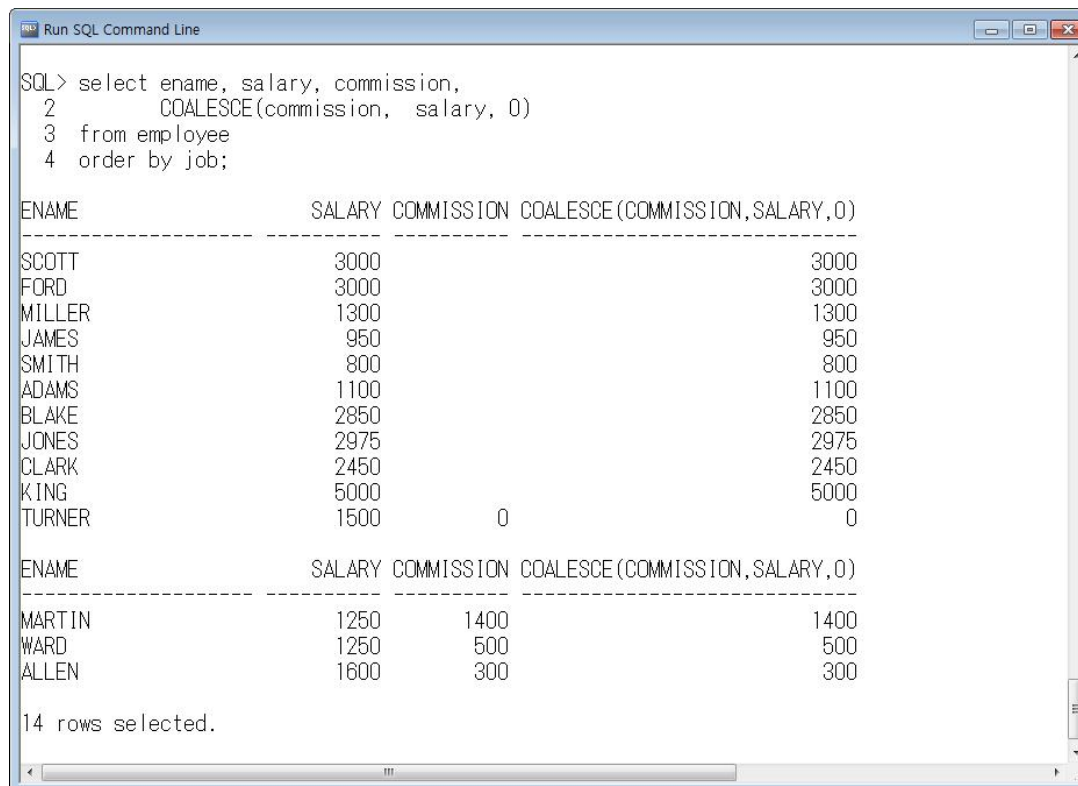
NU NU
-- --
  A
```

COALESCE

- ❖ 인수 중에서 NULL이 아닌 첫 번째 인수를 반환

형식 **COALESCE(expr-1, expr-2, ... ,expr-n)**

- ❖ expr-1이 NULL이 아니면 expr-1을 반환합니다. expr-1이 NULL이고 expr-2가 NULL이 아니면 expr-2를 반환합니다.



```
SQL> select ename, salary, commission,
2          COALESCE(commission, salary, 0)
3  from employee
4  order by job;
```

ENAME	SALARY	COMMISSION	COALESCE (COMMISSION, SALARY, 0)
SCOTT	3000		3000
FORD	3000		3000
MILLER	1300		1300
JAMES	950		950
SMITH	800		800
ADAMS	1100		1100
BLAKE	2850		2850
JONES	2975		2975
CLARK	2450		2450
KING	5000		5000
TURNER	1500	0	0
MARTIN	1250	1400	1400
WARD	1250	500	500
ALLEN	1600	300	300

14 rows selected.

DECODE

❖ 조건에 따라 다양한 선택이 가능

형식

**DECODE (표현식, 조건1, 결과1,
조건2, 결과2,
조건3, 결과3,
기본결과n**

)

```
Run SQL Command Line

SQL> select ename, dno,
2          DECODE(dno, 10, 'ACCOUNTING',
3                    20, 'RESEARCH',
4                    30, 'SALES',
5                    40, 'OPERATIONS',
6                    'DEFAULT') AS DNAME
7   from employee
8  order by dno;

ENAME          DNO DNAME
-----
CLARK           10 ACCOUNTING
KING            10 ACCOUNTING
MILLER          10 ACCOUNTING
JONES           20 RESEARCH
FORD            20 RESEARCH
ADAMS           20 RESEARCH
SMITH           20 RESEARCH
SCOTT           20 RESEARCH
WARD            30 SALES
TURNER          30 SALES
ALLEN           30 SALES

ENAME          DNO DNAME
-----
JAMES           30 SALES
BLAKE           30 SALES
MARTIN          30 SALES

14 rows selected.
```

CASE

❖ 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
END

```
Run SQL Command Line

SQL> select ename, dno,
2      CASE WHEN dno=10 THEN 'ACCOUNTING'
3            WHEN dno=20 THEN 'RESEARCH'
4            WHEN dno=30 THEN 'SALES'
5            WHEN dno=40 THEN 'OPERATIONS'
6            ELSE 'DEFAULT'
7      END AS DNAME
8 from employee
9 order by dno;
```

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

14 rows selected.