

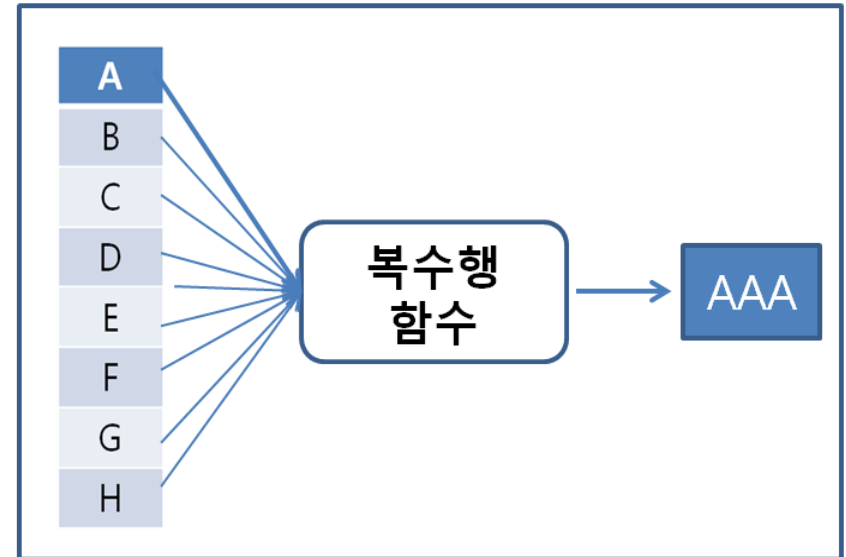
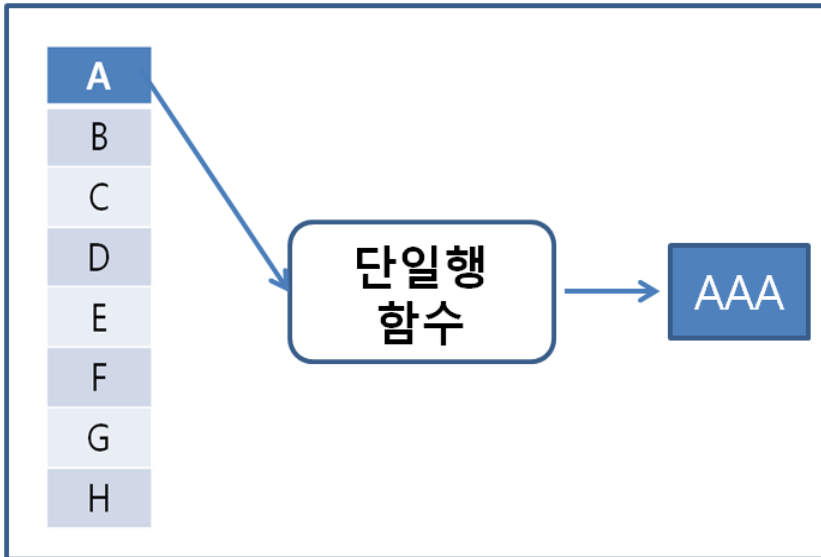
다양한 예제로 쉽게 배우는

오라클 SQL 과 PL/SQL

2장. SQL 단일 행 함수를 배웁니다

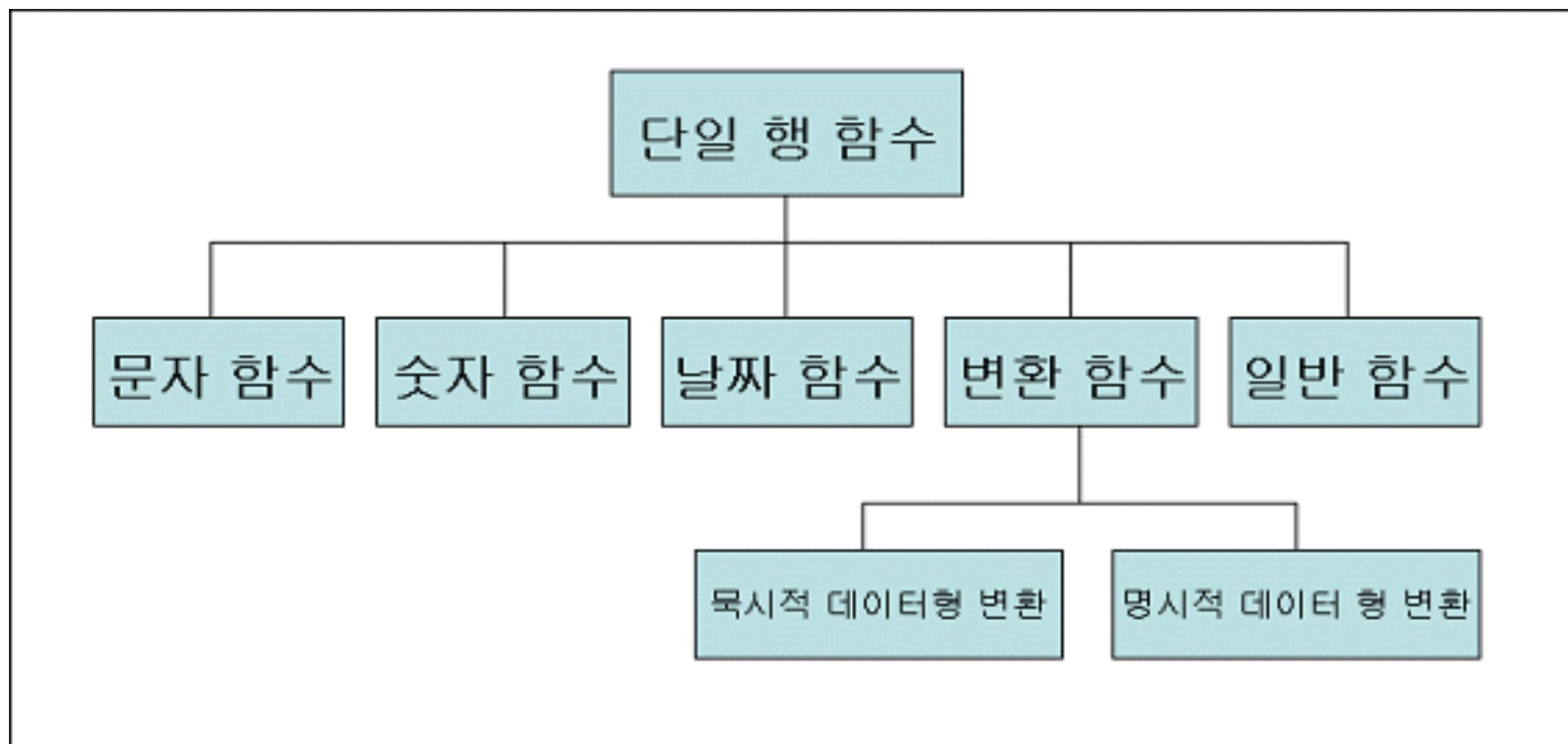
2. SQL 단일행 함수

- SQL 함수



2. SQL 단일행 함수

- SQL 단일 행 함수



2. SQL 단일행 함수

1. 문자함수 - 1

함 수 명	의 미	사 용 예
INITCAP	입력 값의 첫 글자만 대문자로 변환	INITCAP('abcd') -> Abcd
LOWER	입력 값을 전부 소문자로 변환	LOWER('ABCD') -> abcd
UPPER	입력 값을 전부 대문자로 변환	UPPER('abcd') -> ABCD
LENGTH	입력된 문자열의 길이 값을 출력	LENGTH('한글') -> 2
LENGTHB	입력된 문자열의 길이의 바이트 값을 출력	LENGTHB('한글') -> 4
CONCAT	두 문자열을 결합해서 출력. 연산자와 동일	CONCAT('A','B') -> AB
SUBSTR	주어진 문자에서 특정 문자만 추출	SUBSTR('ABC',1,2) -> AB
SUBSTRB	주어진 문자에서 특정 바이트만 추출	SUBSTRB('한글',1,2) -> 한
INSTR	주어진 문자에서 특정문자의 위치 추출	INSTR('A*B#','#') -> 4
INSTRB	주어진 문자에서 특정문자의 위치 바이트값 추출	INSTRB('한글로','로') -> 5

2. SQL 단일행 함수

1. 문자함수 - 2

함 수 명	의 미	사 용 예
LPAD	주어진 문자열에서 왼쪽으로 특정 문자를 채움	LPAD('love',6,'*') -> **love
RPAD	주어진 문자열에서 오른쪽으로 특정 문자를 채움	RPAD('love',6,'*') -> love**
LTRIM	주어진 문자열에서 왼쪽의 특정문자를 삭제함	LTRIM('*love','*') -> love
RTRIM	주어진 문자열에서 오른쪽의 특정문자를 삭제함	RTRIM('love*','*') -> love
REPLACE	주어진 문자열에서 A 를 B 로 치환함	REPLACE('AB','A','E') -> EB
REGEXP_REPLACE	주어진 문자열에서 특정패턴을 찾아 치환함	아래 예 참조
REGEXP_INSTR	주어진 문자열에서 특정패턴의 시작 위치를 반환	아래 예 참조
REGEXP_SUBSTR	주어진 문자열에서 특정패턴을 찾아 반환함	아래 예 참조
REGEXP_LIKE	주어진 문자열에서 특정패턴을 찾아 반환함	아래 예 참조
REGEXP_COUNT	주어진 문자열에서 특정패턴의 횟수를 반환	아래 예 참조

2. SQL 단일행 함수

1) INITCAP 함수:

첫 글자만 대문자로 출력하고 나머지는 전부 소문자로 출력하는 함수

문 법: INITCAP(문자열 또는 컬럼명)

-Student 테이블을 사용하여 1전공이 201 번인 학생들의 ID 를 첫 글자만 대문자로 출력하세요

```
oracle@localhost:~  
SCOTT>SELECT INITCAP(ID)  
2 FROM STUDENT  
3 WHERE deptno1 = 201 ;  
  
INITCAP(ID)  
-----  
Gunmandu  
Shyun1  
Nonnon1  
Munho  
Silverwt  
Youngmin  
  
6 rows selected.  
SCOTT>
```

```
oracle@localhost:~  
SCOTT>  
SCOTT>SELECT INITCAP('Pretty girl')  
2 FROM DUAL ;  
  
INITCAP('PR  
-----  
Pretty Girl  
SCOTT>
```

문자열 중간에 공백이 있는 경우에는 단어의 시작부분이 대문자로 바뀌는 것을 볼 수 있습니다.

2. SQL 단일행 함수

2) LOWER 함수:

입력되는 값을 전부 소문자로 변경하여 출력합니다.

문 법: LOWER(문자열 또는 컬럼명)

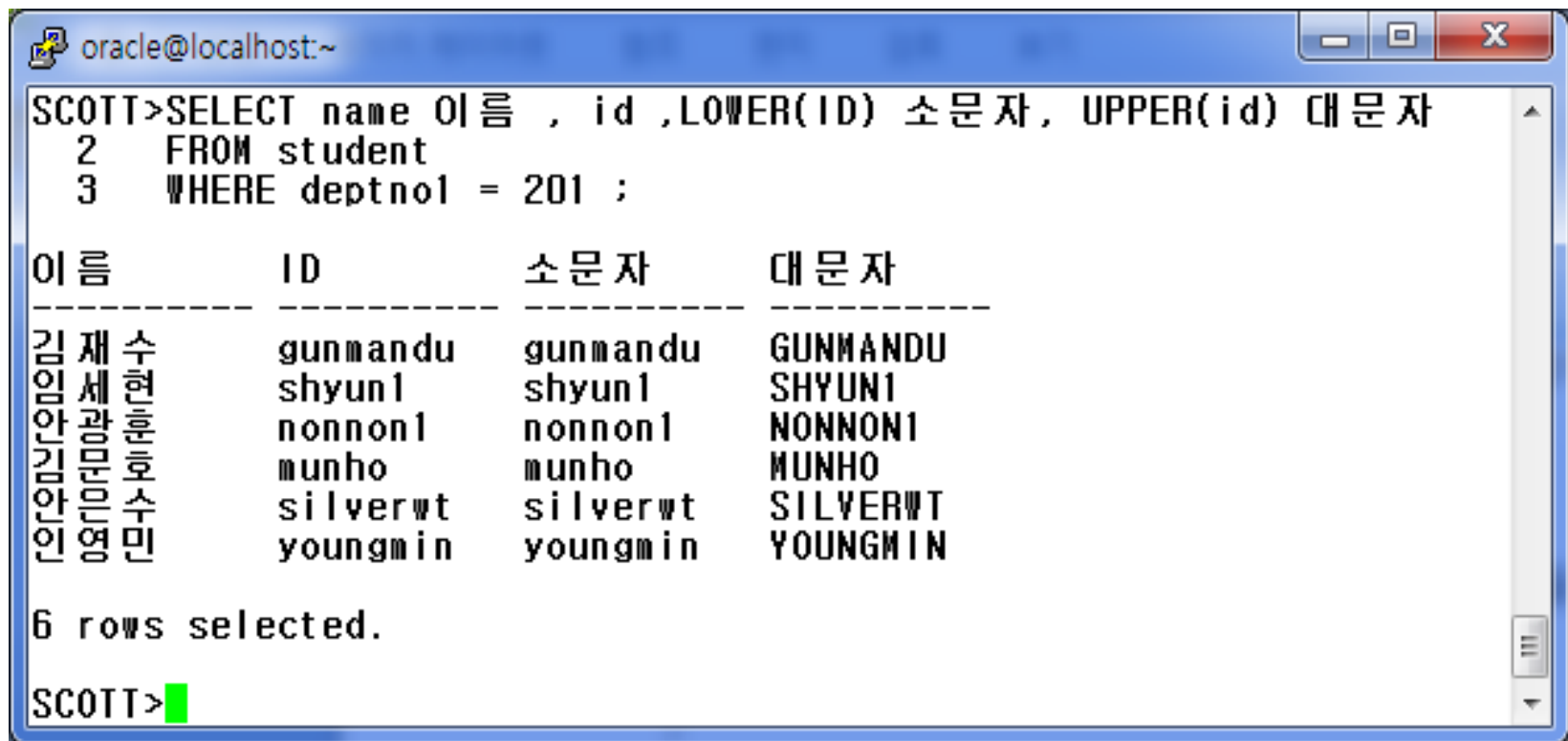
3) UPPER 함수:

입력되는 값을 전부 대문자로 변경하여 출력합니다.

문 법: UPPER(문자열 또는 컬럼명)

2. SQL 단일행 함수

-Student 테이블에서 1 전공이 201 번인 학생들의 ID 를 이름과 함께 소문자 , 대문자로 출력하세요.



```
oracle@localhost:~  
SCOTT>SELECT name 이름 , id ,LOWER(id) 소문자, UPPER(id) 대문자  
2 FROM student  
3 WHERE deptno1 = 201 ;
```

이름	ID	소문자	대문자
김재수	gunmandu	gunmandu	GUNMANDU
임세현	shyun1	shyun1	SHYUN1
안광훈	nonnon1	nonnon1	NONNON1
김민호	munho	munho	MUNHO
안은수	silverwt	silverwt	SILVERWT
인영민	youngmin	youngmin	YOUNGMIN

```
6 rows selected.  
SCOTT>
```

2. SQL 단일행 함수

4) LENGTH / LENGTHB 함수:

입력된 문자열의 길이를 (바이트 수를) 계산해주는 함수입니다.

문 법: LENGTH(컬럼 또는 문자열) / LENGTHB(컬럼 또는 문자열)

- ① Student 테이블에서 ID 가 9글자 이상인 학생들의 이름과 ID 와 글자수를 출력하세요.

```
oracle@localhost:~
SCOTT>SELECT name 이름 , id , LENGTH(id) 글자수
2 FROM student
3 WHERE LENGTH(id) > 9 ;

이름          ID          글자수
-----
이문나      prettygirl          10

SCOTT>
```

2. SQL 단일행 함수

- ② Student 테이블에서 1 전공이 201 번인 학생들의 이름과 이름의 글자수, 이름의 바이트 수를 출력하세요

```
oracle@localhost:~
SCOTT>SELECT name 이름 , LENGTH(name) ,LENGTHB(name)
2   FROM student
3   WHERE deptno1 = 201 ;
```

이름	LENGTH(NAME)	LENGTHB(NAME)
김재수	3	6
임세현	3	6
안광훈	3	6
김민호	3	6
안민수	3	6
이영민	3	6

```
6 rows selected.
SCOTT>
```

2. SQL 단일행 함수

5) CONCAT 함수 (|| 연산자와 동일한 기능입니다)

문 법: CONCAT('문자열1' , '문자열2')

- Professor 테이블에서 교수들의 이름과 직급을 출력하세요.

```
oracle@localhost:~
SCOTT>SELECT CONCAT(name,position)
      2   FROM professor
      3   WHERE deptno=101 ;

CONCAT(NAME,POSITION)
-----
조인형정교수
박승곤조교수
송도권전임강사

SCOTT>
```

2. SQL 단일행 함수

6) SUBSTR 함수

문자열에서 특정길이의 문자를 추출할 때 사용하는 함수입니다

문 법: SUBSTR('문자열' 또는 컬럼명 , 1 , 4)

```
oracle@localhost:~
SCOTT>SELECT SUBSTR('ABCDE',2,3)
2 FROM dual;

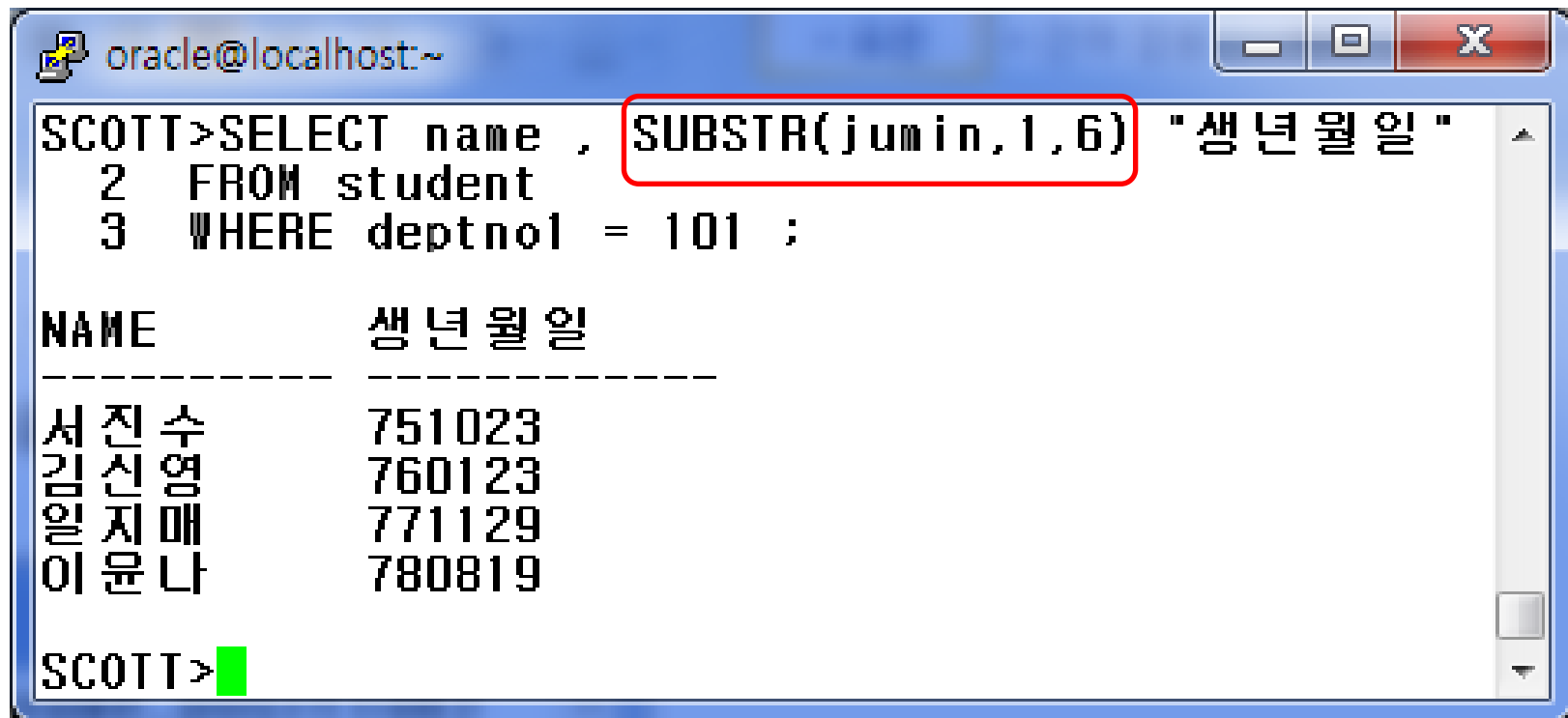
SUB
---
BCD
```

```
oracle@localhost:~
SCOTT>SELECT SUBSTR('ABCDE',-2,3)
2 FROM dual ;

SU
--
DE
```

2. SQL 단일행 함수

- ① Student 테이블에서 jumin 컬럼을 사용해서 1 전공이 101번인 학생들의 이름과 생년월일을 출력하세요.

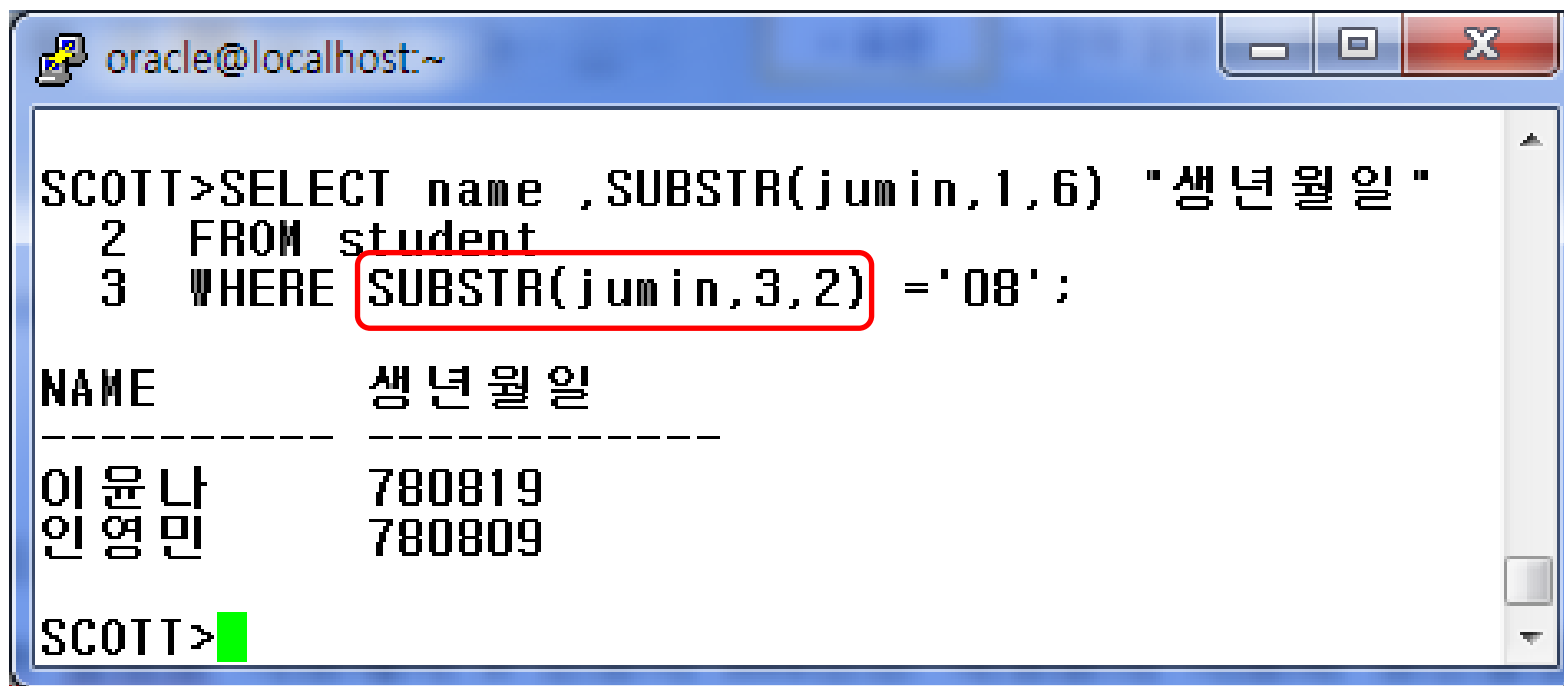


```
oracle@localhost:~  
SCOTT>SELECT name , SUBSTR(jumin,1,6) "생년월일"  
2 FROM student  
3 WHERE deptno1 = 101 ;  
  
NAME                생년월일  
-----  
서진수              751023  
김신영              760123  
일지매              771129  
이윤나              780819  
  
SCOTT>
```

The screenshot shows a terminal window with the Oracle SQL*Plus prompt. The user has entered a query to select the name and the first 6 characters of the jumin column from the student table where deptno1 is 101. The results are displayed in a table with two columns: NAME and 생년월일. The data rows are: 서진수 (751023), 김신영 (760123), 일지매 (771129), and 이윤나 (780819). The prompt SCOTT> is followed by a green cursor.

2. SQL 단일행 함수

- ② Student 테이블에서 jumin 컬럼을 사용해서 태어난 달이 8월인 사람의 이름과 생년월일을 출력하세요.



```
oracle@localhost:~  
SCOTT>SELECT name ,SUBSTR(jumin,1,6) "생년월일"  
2 FROM student  
3 WHERE SUBSTR(jumin,3,2) ='08';  
  
NAME          생년월일  
-----  
이문나        780819  
인영민        780809  
  
SCOTT>
```

The screenshot shows a terminal window with the Oracle SQL*Plus prompt. The user has entered a query to select names and birth dates from the 'student' table where the birth month is August. The query is: `SELECT name ,SUBSTR(jumin,1,6) "생년월일" FROM student WHERE SUBSTR(jumin,3,2) ='08';`. The results show two rows: '이문나' with birth date '780819' and '인영민' with birth date '780809'. The 'SUBSTR(jumin,3,2)' part of the WHERE clause is highlighted with a red box.

2. SQL 단일행 함수

7) SUBSTRB 함수

이 함수는 SUBSTR 함수와 문법은 동일하며 차이점은 추출할 자리수가 아니라 추출 할 바이트 수를 지정함

```
oracle@localhost:~
SCOTT>SELECT SUBSTR(name,1,2)
2 FROM student
3 WHERE deptno1=101 ;

SUBS
-----
서진신
김일지
이문
```

```
oracle@localhost:~
SCOTT>SELECT SUBSTRB(name,1,2)
2 FROM student
3 WHERE deptno1=101 ;

SU
--
서
김
일
이

SCOTT>
```


2. SQL 단일행 함수

8) INSTR 함수

주어진 문자열이나 컬럼에서 특정 글자의 위치를 찾아주는 함수

문 법: INSTR('문자열' 또는 컬럼, 찾는 글자, 시작위치, 몇 번째인지(기본값은 1))

```
oracle@localhost:~  
SCOTT>SELECT INSTR('A*B*C*', '*', 1, 2)  
2 FROM dual;  
  
INSTR('A*B*C*', '*', 1, 2)  
-----  
4  
  
SCOTT>
```

```
oracle@localhost:~  
SCOTT>SELECT INSTR('A*B*C*', '*', 3, 2)  
2 FROM dual;  
  
INSTR('A*B*C*', '*', 3, 2)  
-----  
6  
  
SCOTT>
```

2. SQL 단일행 함수

시작위치를 - (마이너스) 로 줄 경우

```
oracle@localhost:~
SCOTT>SELECT INSTR('A*B*C*','*',-4,1)
2 FROM dual;

INSTR('A*B*C*','*',-4,1)
-----
2
```

```
oracle@localhost:~
SCOTT>SELECT INSTR('A*B*C*','*',-4,2)
2 FROM dual;

INSTR('A*B*C*','*',-4,2)
-----
0
```

```
oracle@localhost:~
SCOTT>
SCOTT>SELECT INSTR('A*B*C*','*',-2,2)
2 FROM dual;

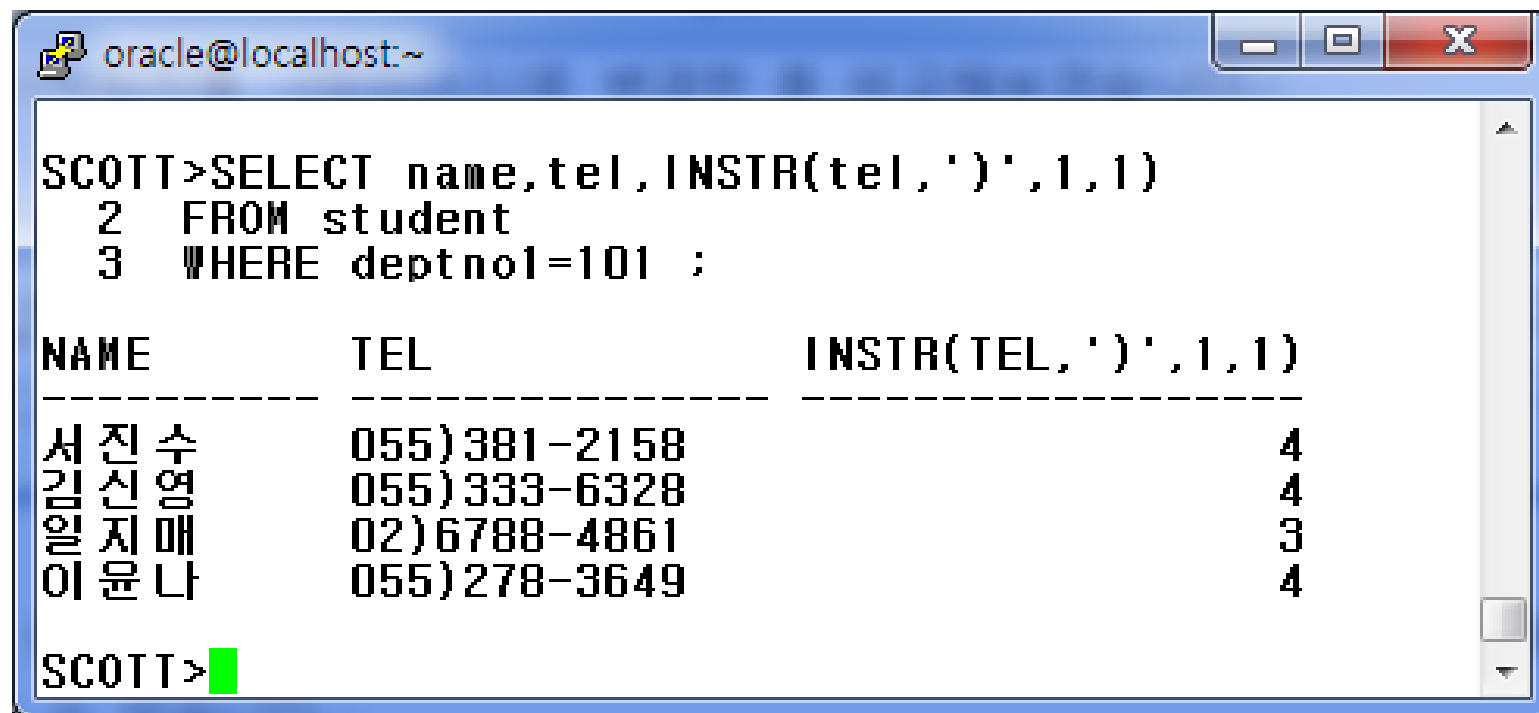
INSTR('A*B*C*','*',-2,2)
-----
2

SCOTT>
```

자리번호	1	2	3	4	5	6
문자열	A	*	B	*	C	*
2번 예제			시작위치	-3	-2	-1
3번 예제		2번째 * 표		1번째 * 표	시작위치	-1

2. SQL 단일행 함수

-Student 테이블의 tel 컬럼을 사용하여 학생의 이름과 전화번호, ')' 가 나오는 위치를 출력하세요.



```
oracle@localhost:~  
SCOTT>SELECT name,tel,INSTR(tel,')',1,1)  
2 FROM student  
3 WHERE deptno=101 ;  
  
NAME                TEL                INSTR(TEL,')',1,1)  
-----  
서진수              055)381-2158              4  
김신영              055)333-6328              4  
일지매              02)6788-4861              3  
이윤나              055)278-3649              4  
  
SCOTT>
```

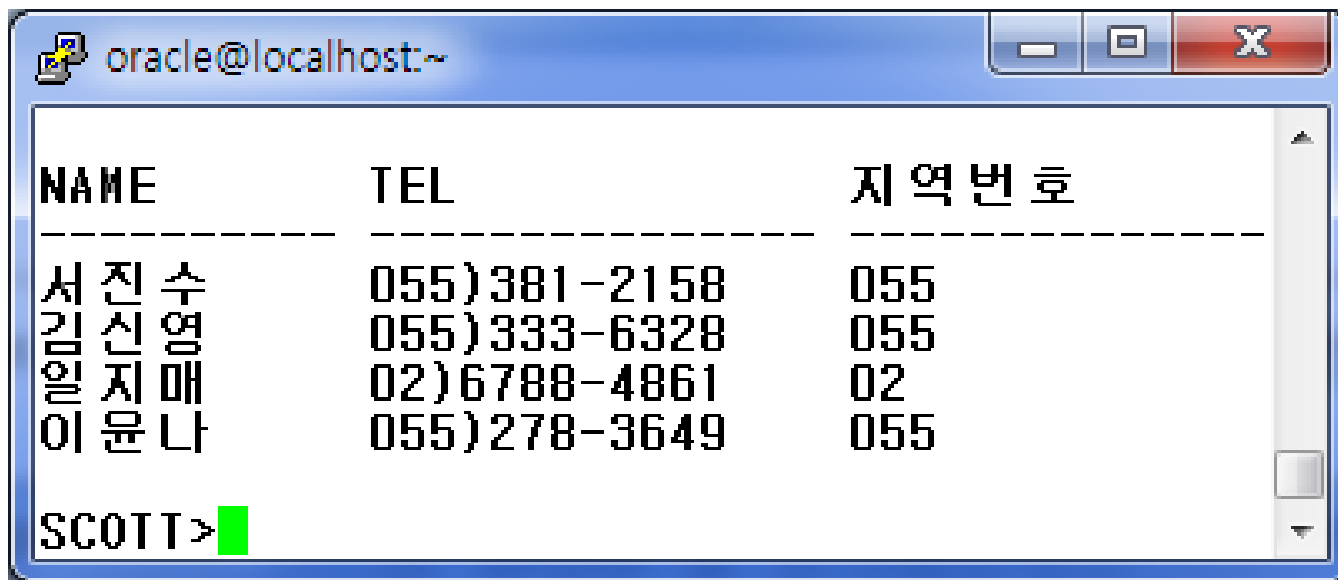
NAME	TEL	INSTR(TEL,')',1,1)
서진수	055)381-2158	4
김신영	055)333-6328	4
일지매	02)6788-4861	3
이윤나	055)278-3649	4

2. SQL 단일행 함수

문자 함수 퀴즈 1)

Student 테이블을 참조해서 아래 화면과 같이 1 전공이(deptno1 컬럼) 101번
인 학생의 이름과 전화번호와 지역번호를 출력하세요. 단 지역번호는 숫자만
나와야 합니다.

결과 화면)



The screenshot shows a terminal window titled 'oracle@localhost:~'. It displays the output of a SQL query. The output is a table with three columns: NAME, TEL, and 지역번호. The data is as follows:

NAME	TEL	지역번호
서진수	055)381-2158	055
김신열	055)333-6328	055
일지매	02)6788-4861	02
이윤나	055)278-3649	055

Below the table, the prompt 'SCOTT>' is visible with a green cursor.

2. SQL 단일행 함수

9) LPAD 함수

문 법: LPAD('문자열' 또는 컬럼명, 자리수 , '채울문자')

- Student 테이블에서 1 전공이 101 번인 학과 학생들의 id 를 총 10자리로 출력하되 왼쪽 빈 자리는 '\$' 기호로 채우세요.

```
oracle@localhost:~
SCOTT>SELECT id , LPAD(id,10,'$') "LPAD"
2 FROM student
3 WHERE deptno1=101 ;
```

ID	LPAD
75true	\$\$\$\$75true
bingo	\$\$\$\$bingo
onejmae	\$onejmae
prettygirl	prettygirl

```
SCOTT>
```

2. SQL 단일행 함수

문자함수 퀴즈 2)

Dept2 테이블을 사용하여
dname을 오른쪽의 결과가
나오도록 쿼리를 작성하세요.

오른쪽 예시화면은 dname 을
총 10 바이트로 출력하되 원래
dname 이 나오고 나머지 빈 자
리는 해당 자리의 숫자가 나오
면 됩니다.

즉 사장실은 이름이 총 6 바이
트이므로 숫자가 1234 까지 나
오는 것입니다.

결과 화면)

```

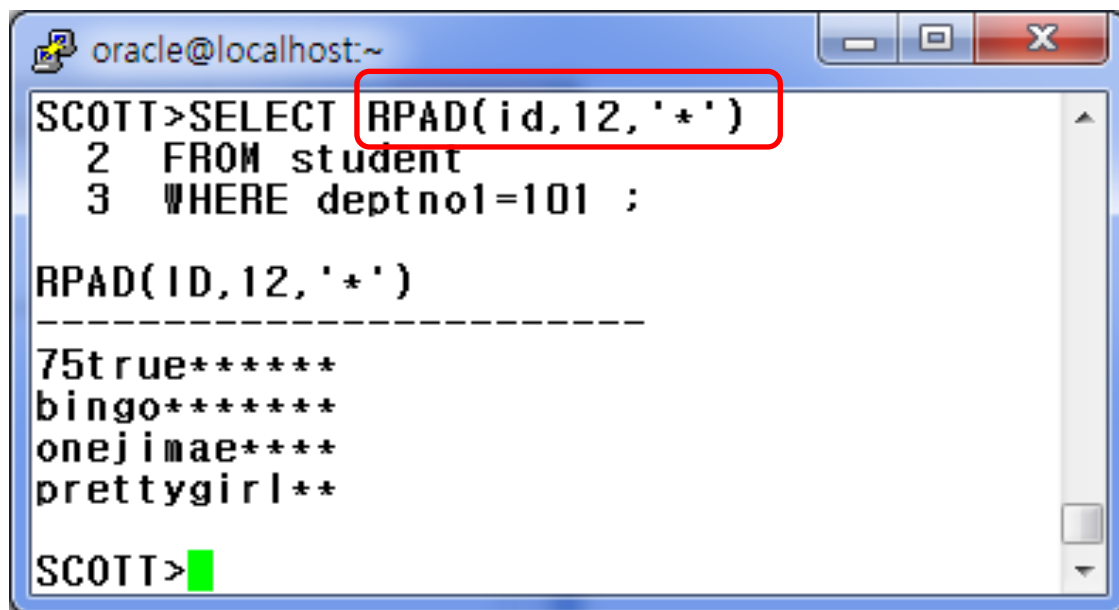
oracle@localhost:~
LPAD 연습
-----
1234사장실
1234경영지원부
1234재무관리팀
1234총무팀부
1234기술부
123H/℄지원
123S/℄지원
1234영업부
1234영업기획팀
1234영업1팀
    
```

2. SQL 단일행 함수

11) RPAD 함수

문 법: RPAD('문자열' 또는 컬럼명, 자리수 , '채울문자')

-Student 테이블에서 아래와 같이 id 를 12자리로 출력하되
오른쪽 빈자리에는 '*' 로 채우세요.



```

oracle@localhost:~
SCOTT>SELECT RPAD(id,12,'*')
2 FROM student
3 WHERE deptno1=101 ;

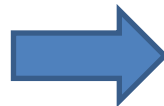
RPAD(ID,12,'*')
-----
75true*****
bingo*****
onejima*****
prettygirl**

SCOTT>
  
```

2. SQL 단일행 함수

문자 함수 퀴즈 3)

Dept2 테이블을 사용하여 **dtype**을 오른쪽의 결과가 나오도록 쿼리를 작성하세요.



결과 화면)

RPAD연습

```

-----
사장실7890
경영지원부
재무관리팀
총무팀7890
기술부7890
H/W지원890
S/W지원890
영업부7890
영업기획팀
영업1팀890
영업2팀890
영업3팀890
영업4팀890
    
```

13 rows selected.

2. SQL 단일행 함수

11) LTRIM 함수

LPAD , RPAD 함수는 채우는 함수이고 LTRIM , RTRIM 함수는 제거하는 함수임

문 법: LTRIM('문자열' 또는 컬럼명 , '제거할 문자')

-dept 2 테이블에서
dname 을 출력하되
왼쪽에 '영' 이란 글
자를 모두 제거하고
출력하세요



```

oracle@localhost:~
SCOTT>SELECT LTRIM(dname,'영')
2 FROM dept2;

LTRIM(DNAME,'영')
-----
사장실
경영지
재영관
총무원
기술부
H/원
S/원
영부
영기
영1팀
영2팀
영3팀
영4팀

13 rows selected.
SCOTT>
    
```

모두 '영' 이
라는 글자가
없어졌습니
다.

2. SQL 단일행 함수

12) RTRIM 함수

문 법: RTRIM('문자열' 또는 컬럼명, '제거할 문자')

- Dept2 테이블에서 dname 을 출력하되 오른쪽 끝에 '부' 라는 글자는 제거하고 출력하세요.

```

oracle@localhost:~
SCOTT>SELECT RTRIM(dname, '부')
2 FROM dept2 ;

RTRIM(DNAME, '부')
-----
사장실
경영지원
재무관리팀
총무팀
기술
H/팀지원
S/팀지원
영업
영업기획팀
영업1팀
영업2팀
영업3팀
영업4팀

13 rows selected.

SCOTT>
  
```

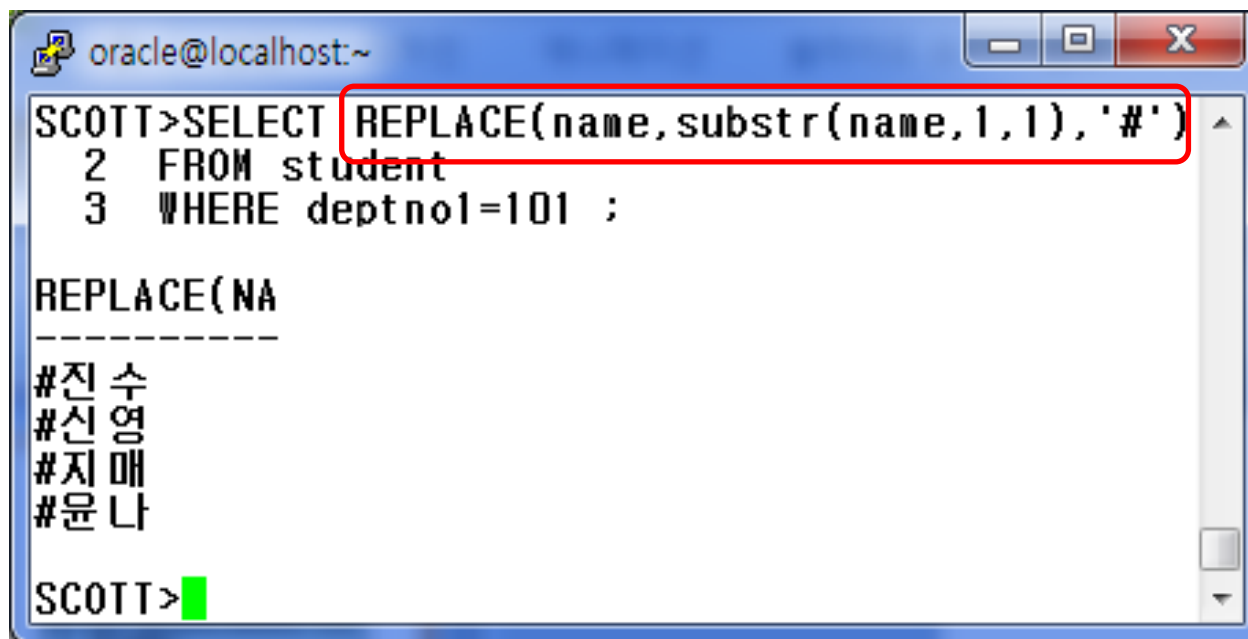
모두 '부' 라는 글자가 없어졌습니다.

2. SQL 단일행 함수

13) REPLACE 함수

문 법: REPLACE('문자열' 또는 컬럼명 , '문자1' , '문자2')

-Student 테이블에서 학생들의 이름을 출력하되 성 부분은 '#' 으로 표시되게 출력하세요.



```

oracle@localhost:~
SCOTT>SELECT REPLACE(name,substr(name,1,1),'#')
2 FROM student
3 WHERE deptno1=101 ;

REPLACE(NA
-----
#진 수
#신 영
#지 매
#윤 나

SCOTT>
  
```

2. SQL 단일행 함수

문자 함수 퀴즈 4)

-Student 테이블에서 아래와 같이 1 전공(deptno1)이 101 번인 학생들의 이름을 출력하되 가운데 글자만 '#' 으로 표시되게 출력하세요.



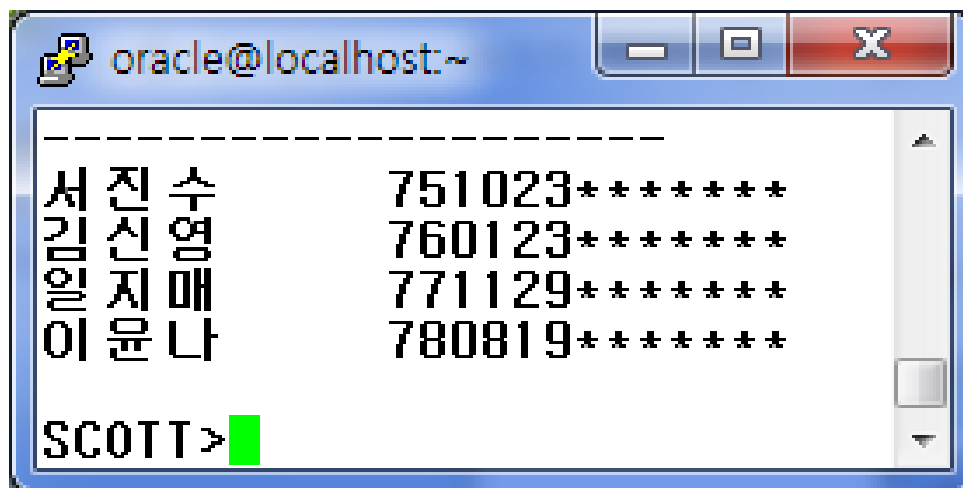
```
REPLACE
-----
서 #수
김 #영
일 #매
이 #나

SCOTT >
```

2. SQL 단일행 함수

문자 함수 퀴즈 5)

-Student 테이블에서 아래와 같이 1 전공(deptno1)이 101 번인 학생들의 이름과 주민등록번호를 출력하되 주민등록번호의 뒤 7자리는 '*' 로 표시되게 출력하세요.



```

-----
서진수      751023*****
김신영      760123*****
일지매      771129*****
이윤나      780819*****

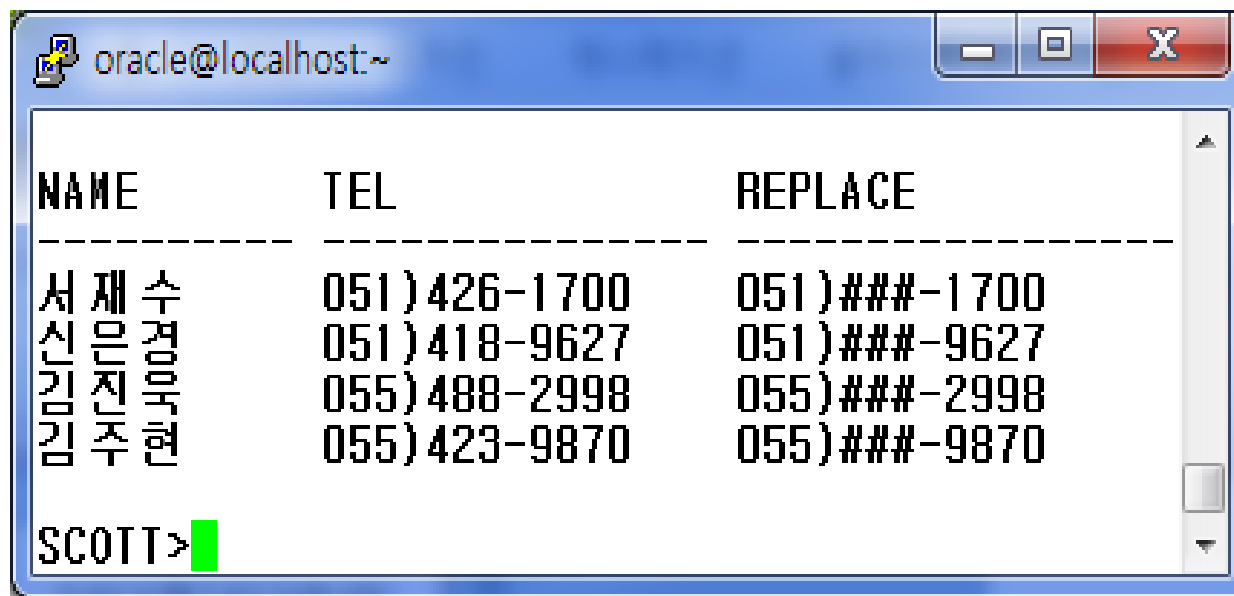
SCOTT>

```

2. SQL 단일행 함수

문자 함수 퀴즈 6)

- Student 테이블에서 아래 그림과 같이 1 전공이 102 번인 학생들의 이름과 전화번호, 전화번호에서 국번 부분만 '#' 처리하여 출력하세요. 단 모든 국번은 3자리로 간주합니다.



NAME	TEL	REPLACE
서재수	051)426-1700	051)###-1700
신은경	051)418-9627	051)###-9627
김진욱	055)488-2998	055)###-2998
김주현	055)423-9870	055)###-9870

SCOTT>

2. SQL 단일행 함수

2. SQL 문장에서 정규식 사용하기
(10g 부터 추가됨)

2. SQL 단일행 함수

- 정규식이란 ?

사용 기호	의 미	사 용 예
^	Pattern으로 시작하는 line 출력	'^pattern'
\$	Pattern으로 끝나는 line 출력	'pattern\$'
.	P로 시작하여 n으로 끝나는 line (. → 1 character)	'p n'
*	모든 이라는 뜻. 글자수가 0 일수도 있음.	'[a-z]*'
[]	Pattern에 해당하는 한 문자	'[Pp]attern'
[^]	Pattern에 해당하지 않는 한 문자	'[^a-m]attern'

2. SQL 단일행 함수

14) REGEXP_LIKE 함수

- 사용 예제 1 : 특정 문자나 숫자를 포함하는 결과 출력하기

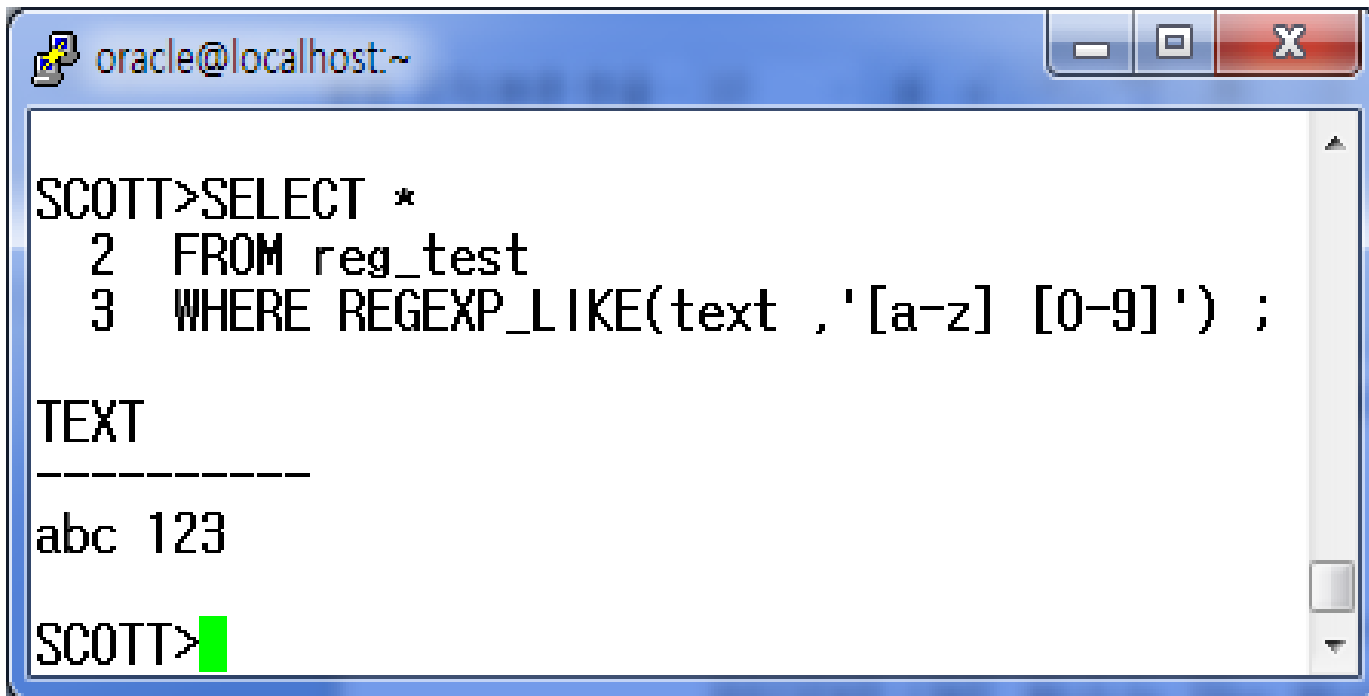
```
oracle@localhost:~
SCOTT>SELECT *
      2 FROM reg_test
      3 WHERE REGEXP_LIKE(text , '[a-z][0-9]') ;

TEXT
-----
a1b2c3
aabbcc123

SCOTT>
```

2. SQL 단일행 함수

- 사용 예제 2: 공백을 한 칸 포함하는 경우 찾기



```

oracle@localhost:~
SCOTT>SELECT *
      2  FROM reg_test
      3  WHERE REGEXP_LIKE(text , '[a-z] [0-9]') ;

TEXT
-----
abc 123

SCOTT>
  
```

2. SQL 단일행 함수

- 사용 예제 3 : 공백이 여러 개일 경우 찾기

```
oracle@localhost:~
SCOTT>SELECT *
  2 FROM reg_test
  3 WHERE REGEXP_LIKE(text , '[a-z]?[0-9]') ;

TEXT
-----
ABC123
ABC 123
abc 123
a1b2c3
aabbcc123

SCOTT>
```

```
oracle@localhost:~
SCOTT>SELECT *
  2 FROM reg_test
  3 WHERE REGEXP_LIKE(text , '[a-z]*[0-9]') ;

TEXT
-----
ABC123
ABC 123
abc 123
a1b2c3
aabbcc123

SCOTT>
```

2. SQL 단일행 함수

- 사용 예제 4: 연속적인 글자 수 지정하기

```
oracle@localhost:~
SCOTT>SELECT *
2 FROM reg_test
3 WHERE REGEXP_LIKE(text , '[A-Z]{3}') ;

TEXT
-----
ABC123
ABC 123

SCOTT>
```

대문자가 연속적으로 3글자 오는 경우

```
oracle@localhost:~
SCOTT>SELECT *
2 FROM reg_test
3 WHERE REGEXP_LIKE(text , '[a-z]{3}') ;

TEXT
-----
abc 123
aabbcc123

SCOTT>
```

소문자가 연속적으로 3글자 오는 경우

2. SQL 단일행 함수

- 사용 예제 4: 연속적인 글자 수 지정하기

```
oracle@localhost:~
SCOTT>SELECT *
2 FROM reg_test
3 WHERE REGEXP_LIKE(text , '[0-9]{3}' ) ;

TEXT
-----
ABC123
ABC 123
abc 123
aabbcc123

SCOTT>
```

숫자가 연속적으로 3글자 오는 경우

```
oracle@localhost:~
SCOTT>SELECT *
2 FROM reg_test
3 WHERE REGEXP_LIKE(text , '[A-Z][0-9]{3}' ) ;

TEXT
-----
ABC123

SCOTT>
```

대문자와 숫자를 함께 검색하는데 대문자가 먼저 연속적으로 3글자 오는 경우

2. SQL 단일행 함수

- 사용 예제 4: 연속적인 글자 수 지정하기

```
oracle@localhost:~
SCOTT>SELECT *
  2 FROM reg_test
  3 WHERE REGEXP_LIKE(text , '[0-9][A-Z]{3}') ;

no rows selected

SCOTT>
```

소문자와 숫자를 함께 검색하는데 숫자가 먼저 연속적으로 3글자 오는 경우

```
oracle@localhost:~
SCOTT>SELECT *
  2 FROM reg_test
  3 WHERE REGEXP_LIKE(text , '[A-Z]{2}') ;

TEXT
-----
ABC123
ABC 123

SCOTT>
```

대문자가 연속으로 2글자 오는 경우

2. SQL 단일행 함수

- 사용 예제 5: 시작되는 문자와 끝나는 문자 지정하기

```
oracle@localhost:~
SCOTT>SELECT *
2 FROM reg_test
3 WHERE REGEXP_LIKE(text, '^[0-9]');

TEXT
-----
123123
123abc

SCOTT>
```

숫자로 시작하는 행만 출력하기

```
oracle@localhost:~
SCOTT>SELECT *
2 FROM reg_test
3 WHERE REGEXP_LIKE(text, '^[a-z]||^[0-9]') ;

TEXT
-----
abc 123
a1b2c3
aabbcc123
123123
123abc

SCOTT>
```

숫자나 소문자로 시작하는 행만
출력하기

2. SQL 단일행 함수

```
oracle@localhost:~
SCOTT>SELECT *
  2 FROM reg_test
  3 WHERE REGEXP_LIKE(text, '[a-z]$') ;

TEXT
-----
123abc

SCOTT>
```

소문자로 끝나는 행을 출력

```
oracle@localhost:~
SCOTT>SELECT *
  2 FROM reg_test
  3 WHERE REGEXP_LIKE(text, '^[^0-9]') ;

TEXT
-----
ABC123
ABC 123
abc 123
a1b2c3
aabbcc123
?/!@#$*&
W~*()..

7 rows selected.

SCOTT>
```

숫자로 시작하지 않는 행을 출력

2. SQL 단일행 함수

```
oracle@localhost:~  
SCOTT>SELECT *  
2 FROM reg_test  
3 WHERE REGEXP_LIKE(text, '^[^a-z]') ;  
  
TEXT  
-----  
ABC123  
ABC 123  
?/!@#$*&  
#~*().,  
123123  
123abc  
  
6 rows selected.  
SCOTT>
```

소문자로 시작하지 않는 행을 출력

2. SQL 단일행 함수

```
oracle@localhost:~  
SCOTT>SELECT *  
2 FROM reg_test  
3 WHERE REGEXP_LIKE(text , 'A|1') ;  
  
TEXT  
-----  
ABC123  
ABC 123  
abc 123  
a1b2c3  
aabbcc123  
123123  
123abc  
  
7 rows selected.  
  
SCOTT>
```

대문자 A 나 숫자 1 을 포함하고 있는
행을 출력

2. SQL 단일행 함수

```
oracle@localhost:~
SCOTT>SELECT *
      2 FROM reg_test
      3 WHERE REGEXP_LIKE(text , 'a|1') ;

TEXT
-----
ABC123
ABC 123
abc 123
a1b2c3
aabbcc123
123123
123abc

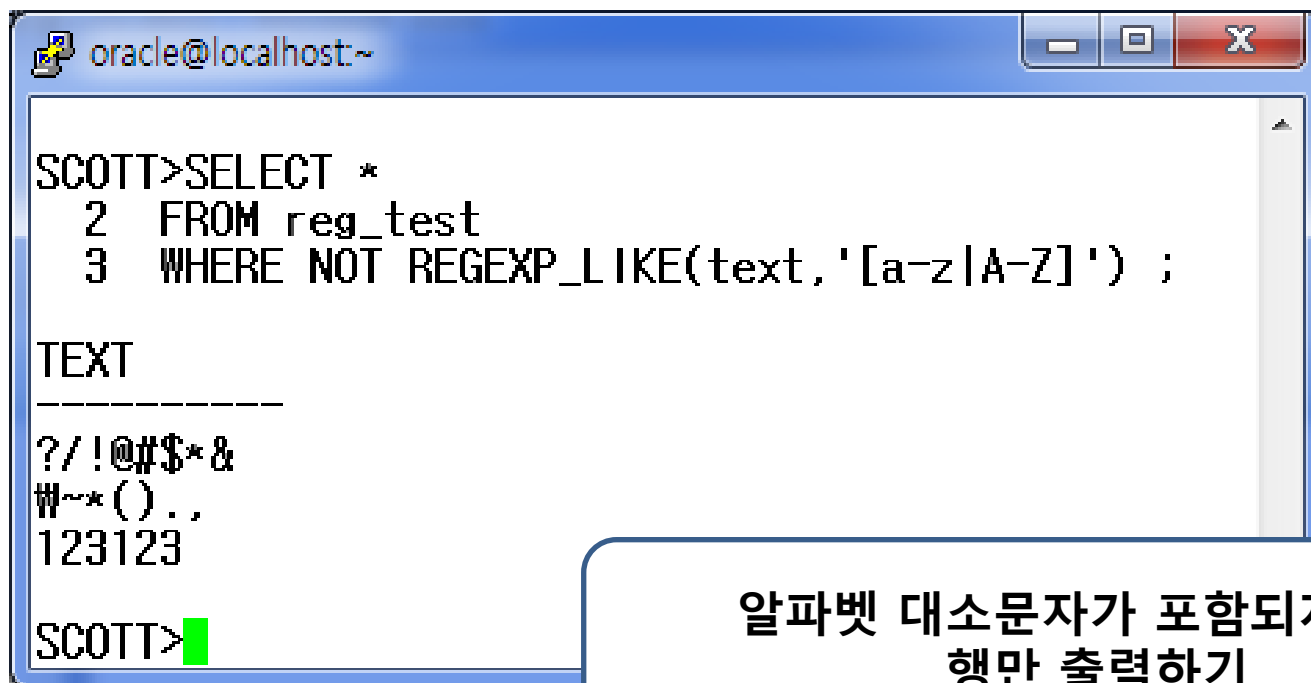
7 rows selected.

SCOTT>
```

소문자 a 나 숫자 1을 포함하는
행을 출력

2. SQL 단일행 함수

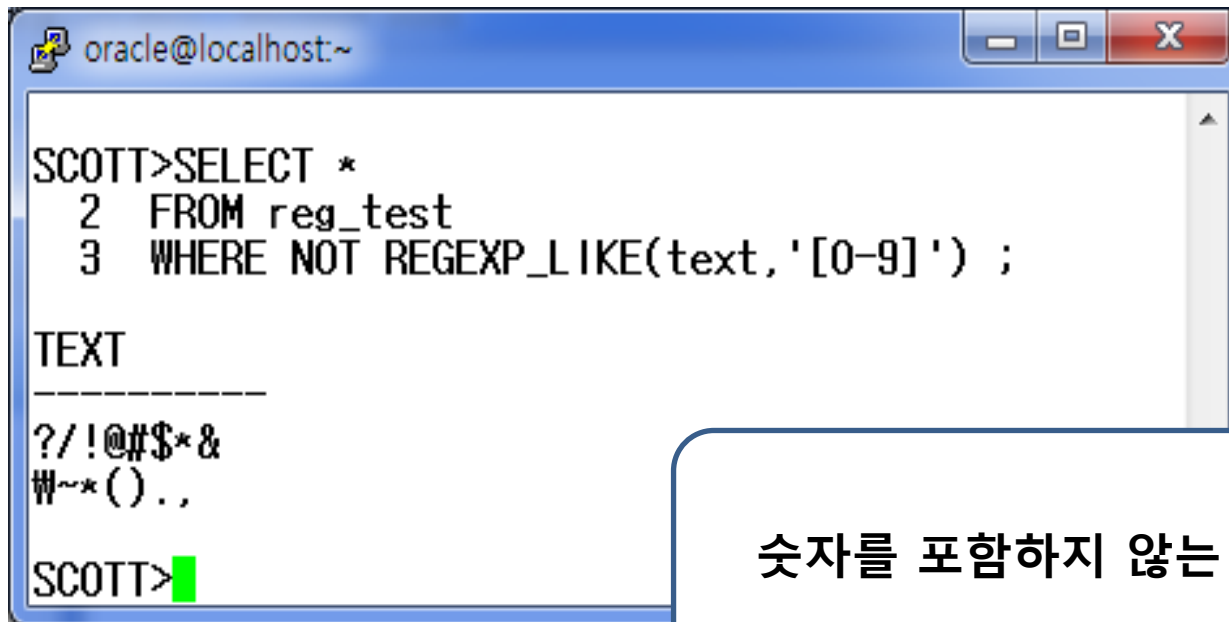
- 특정 조건이 아닌 (NOT) 경우 출력하기



```
oracle@localhost:~  
SCOTT>SELECT *  
2 FROM reg_test  
3 WHERE NOT REGEXP_LIKE(text, '[a-zA-Z]') ;  
  
TEXT  
-----  
?/!@#$*&  
#~*().,  
123123  
SCOTT>
```

알파벳 대소문자가 포함되지 않는
행만 출력하기

2. SQL 단일행 함수

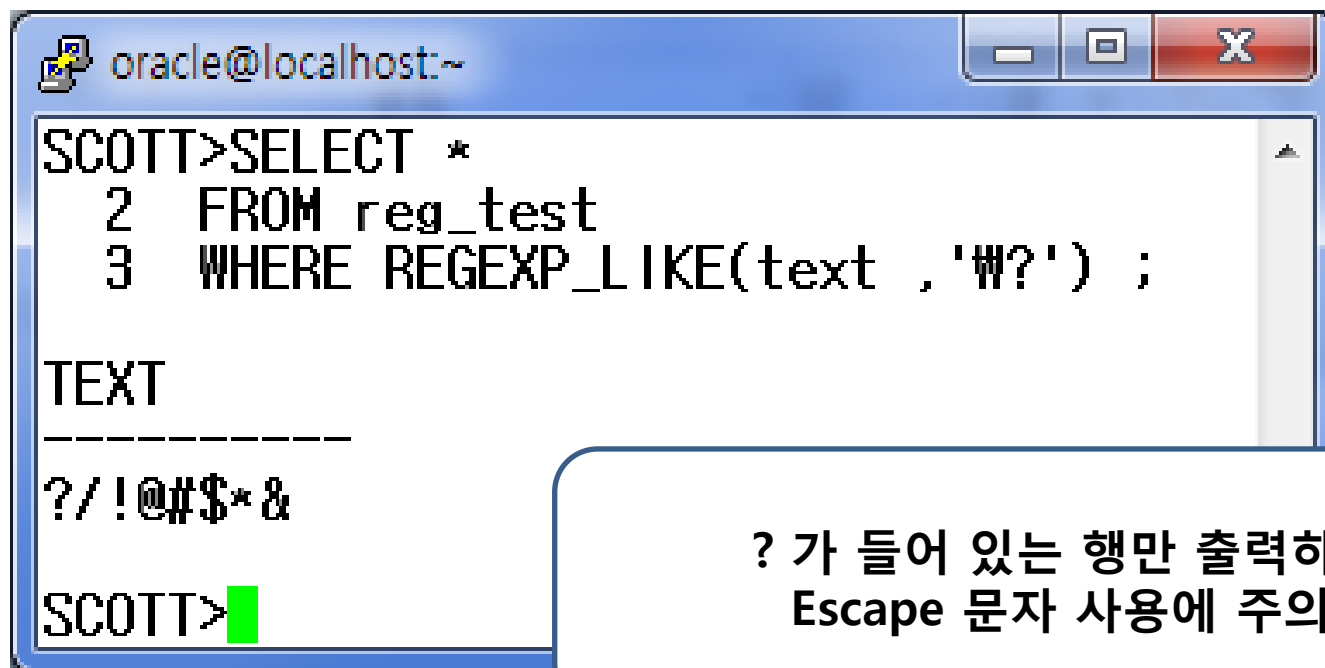


```
oracle@localhost:~  
SCOTT>SELECT *  
2  FROM reg_test  
3  WHERE NOT REGEXP_LIKE(text, '[0-9]') ;  
  
TEXT  
-----  
?/!@#$*&  
#~*().,  
  
SCOTT>
```

숫자를 포함하지 않는 행만 출력하기

2. SQL 단일행 함수

- 특수 문자 찾기



```
oracle@localhost:~  
SCOTT>SELECT *  
2 FROM reg_test  
3 WHERE REGEXP_LIKE(text , '₩?') ;  
  
TEXT  
-----  
?/!@#*$*&  
SCOTT>
```

? 가 들어 있는 행만 출력하기
Escape 문자 사용에 주의!

2. SQL 단일행 함수

```
oracle@localhost:~
SCOTT>SELECT *
      2 FROM reg_test
      3 WHERE NOT REGEXP_LIKE(text, 'W?') ;

TEXT
-----
ABC123
ABC 123
abc 123
a1b2c3
aabbcc123
W~*()..
123123
123abc

8 rows selected.

SCOTT>
```

? 가 들어가 있지 않는 모든 행을 출력

2. SQL 단일행 함수

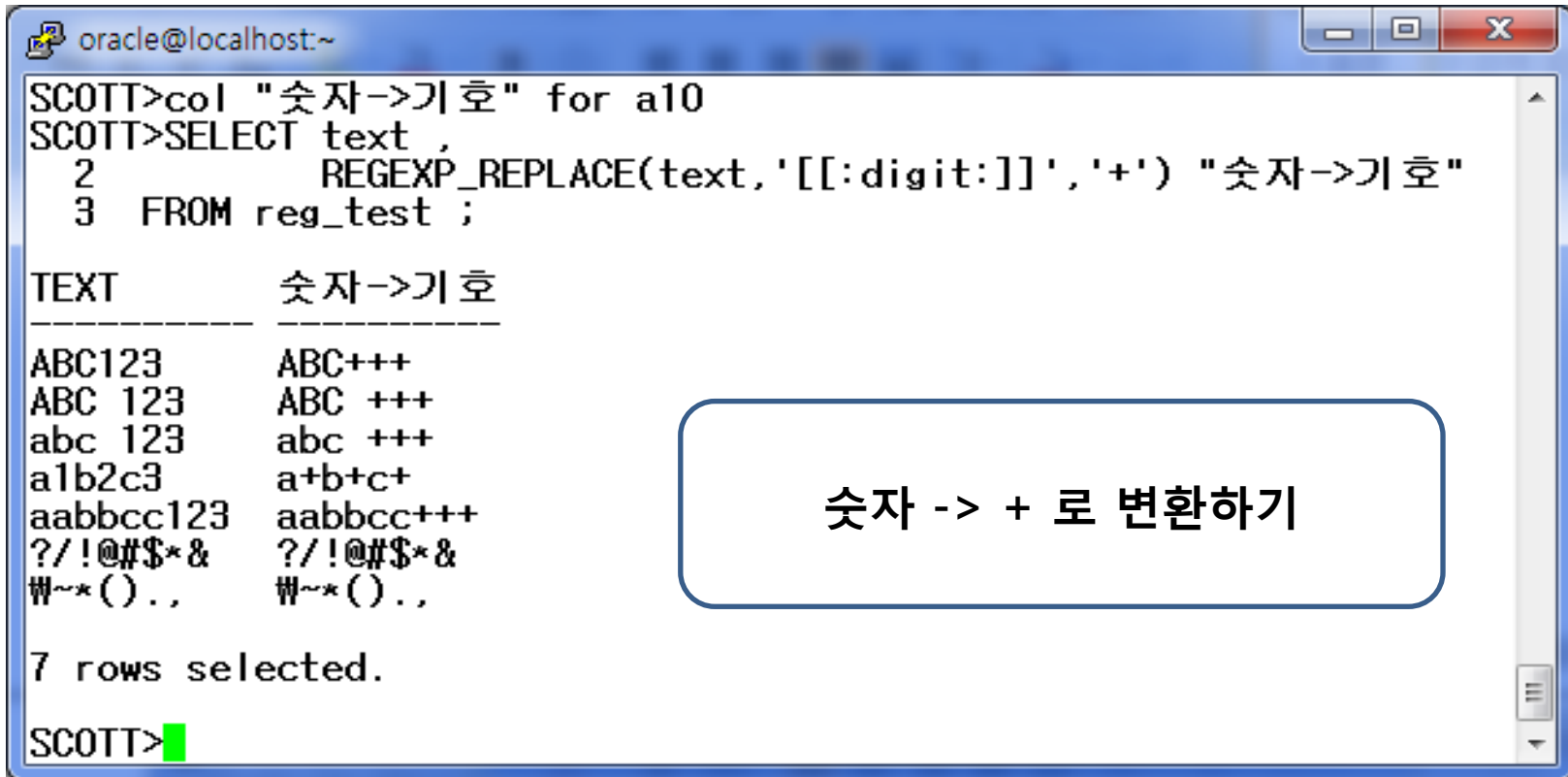
15) REGEXP_REPLACE 함수

```
- 문 법: REGEXP_REPLACE (source_char, pattern  
                        [, replace_string  
                        [, position  
                        [, occurrence  
                        [, match_param]]])  
)
```

- 첫 번째 인수인 Source 는 원본 데이터를 의미.
- 두 번째 인수인 pattern 은 찾고자 하는 패턴을 의미.
- 세 번째 인수인 replace_string 은 변환 하고자 하는 형태.
- 네 번째 인수인 position 은 검색 시작위치를 지정.
- 다섯 번째 인수인 occurrence 는 패턴과 일치가 발생하는 횟수를 의미.
- 여섯 번째 match_parameter 는 기본값으로 검색되는 옵션을 바꿀 수 있음.

2. SQL 단일행 함수

- 사용 예제 1 : 모든 숫자를 특수 기호로 변경하기

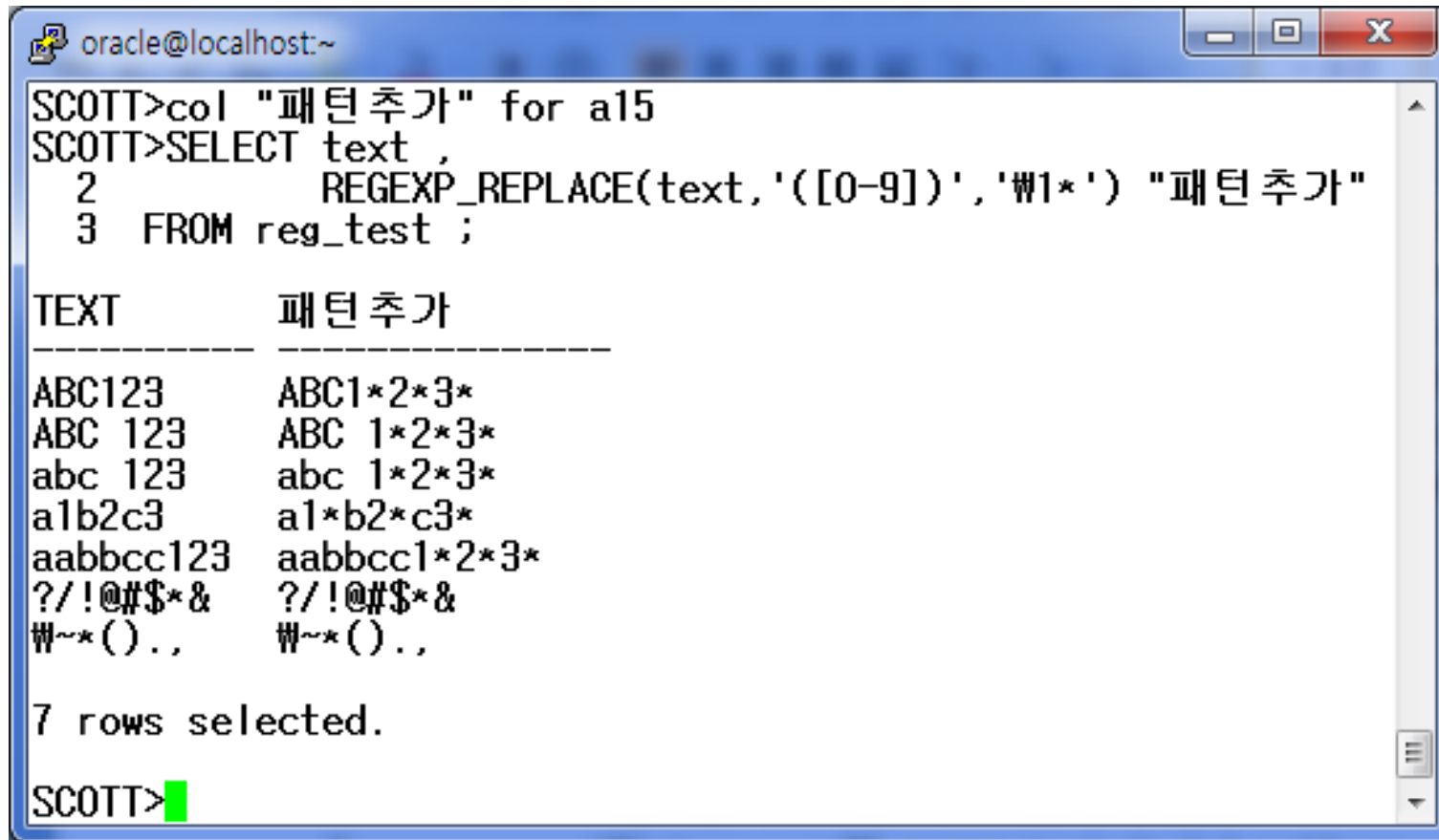


```
oracle@localhost:~  
SCOTT>col "숫자->기호" for a10  
SCOTT>SELECT text ,  
2          REGEXP_REPLACE(text, '[:digit:]', '+') "숫자->기호"  
3 FROM reg_test ;  
  
TEXT          숫자->기호  
-----  
ABC123        ABC+++  
ABC 123       ABC +++  
abc 123       abc +++  
a1b2c3        a+b+c+  
aabbcc123     aabbcc+++  
?/!@#$$*&    ?/!@#$$*&  
₩~*()..       ₩~*()..  
  
7 rows selected.  
  
SCOTT>
```

숫자 -> + 로 변환하기

2. SQL 단일행 함수

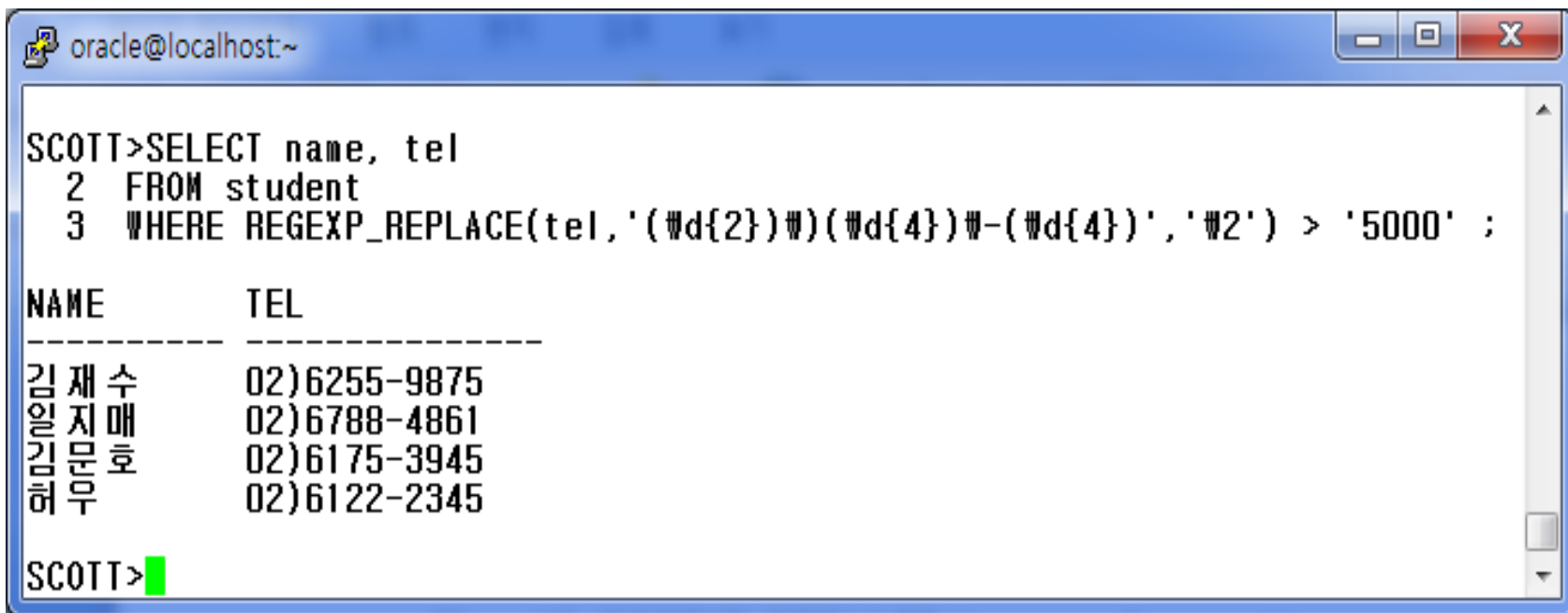
-사용 예제 2: 특정 패턴을 찾아서 패턴을 추가하기



```
oracle@localhost:~  
SCOTT>col "패턴 추가" for a15  
SCOTT>SELECT text ,  
2          REGEXP_REPLACE(text, '([0-9])', '₩1*') "패턴 추가"  
3 FROM reg_test ;  
  
TEXT          패턴 추가  
-----  
ABC123        ABC1*2*3*  
ABC 123       ABC 1*2*3*  
abc 123       abc 1*2*3*  
a1b2c3        a1*b2*c3*  
aabbcc123     aabbcc1*2*3*  
?/?!@#$*&    ?/?!@#$*&  
₩~*()..       ₩~*()..  
  
7 rows selected.  
  
SCOTT>
```

2. SQL 단일행 함수

- 사용 예제 3: Student 테이블에서 지역번호가 2자리이고 전화국번이 4 자리인 전화번호를 가진 학생의 이름과 전화번호를 출력하세요.

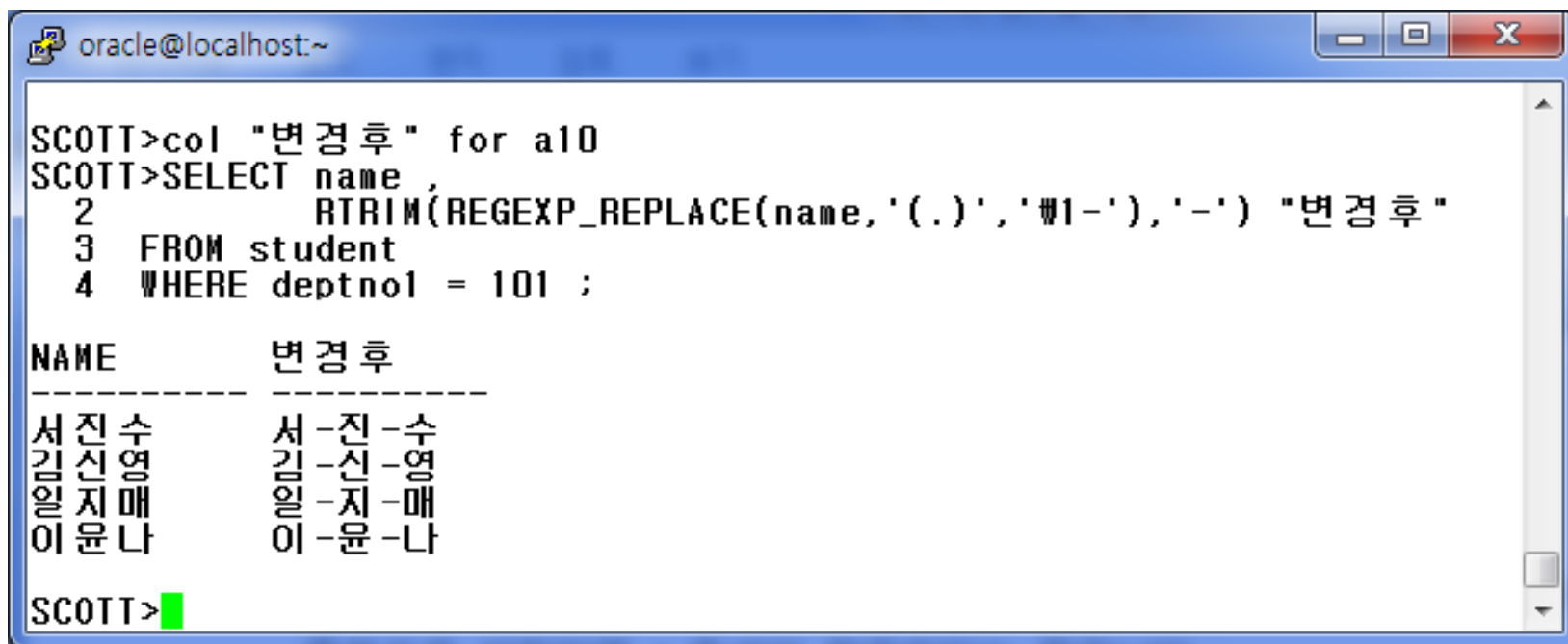


```
oracle@localhost:~  
SCOTT>SELECT name, tel  
2 FROM student  
3 WHERE REGEXP_REPLACE(tel, '(\d{2})\)(\d{4})\-(\d{4})', '\2') > '5000' ;  
  
NAME          TEL  
-----  
김 재 수      02)6255-9875  
일 지 매      02)6788-4861  
김 문 호      02)6175-3945  
허 우         02)6122-2345  
  
SCOTT>
```

2. SQL 단일행 함수

- 사용 예제 4 :

Student 테이블에서 제 1 전공이 101 번인 학생의 이름과 변경 후 모양이 아래의 예제화면과 같이 되도록 출력하세요.



```
oracle@localhost:~  
SCOTT>col "변경 후" for a10  
SCOTT>SELECT name ,  
2          RTRIM(REGEXP_REPLACE(name, '(.)', '##1-'), '-') "변경 후"  
3 FROM student  
4 WHERE deptno1 = 101 ;
```

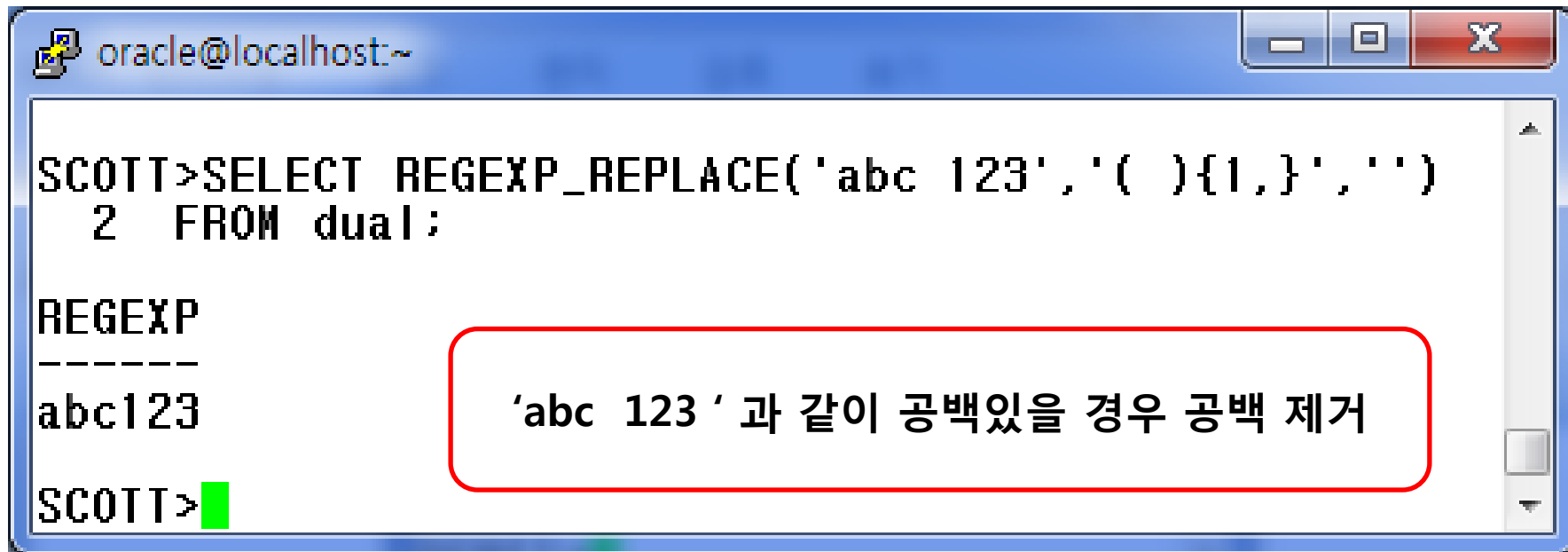
NAME	변경 후
서진수	서-진-수
김신영	김-신-영
일지매	일-지-매
이윤나	이-윤-나

```
SCOTT>
```

2. SQL 단일행 함수

- 사용 예제 5:

사용자에게 입력 받은 문자 가운데 공백이 여러 개 들어 있을 경우 그 공백을 제거시키는 방법



```

oracle@localhost:~
SCOTT>SELECT REGEXP_REPLACE('abc 123','( ){1,}','')
2 FROM dual;

REGEXP
-----
abc123

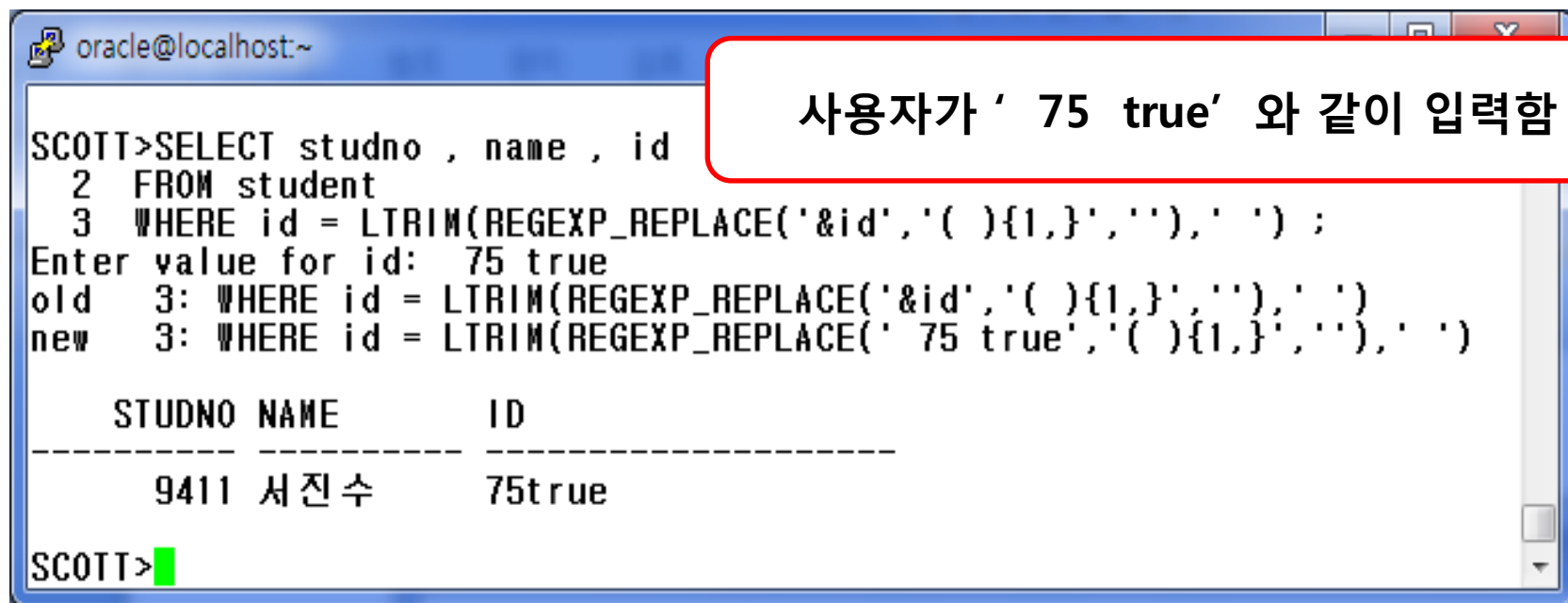
SCOTT>
    
```

'abc 123 ' 과 같이 공백있을 경우 공백 제거

2. SQL 단일행 함수

- 사용 예제 6:

사용자가 회원검색을 할 때 공백 문자를 가장 먼저 입력하고 아이디 중간에도 공백이 있을 때 모든 공백을 제거해야 할 경우

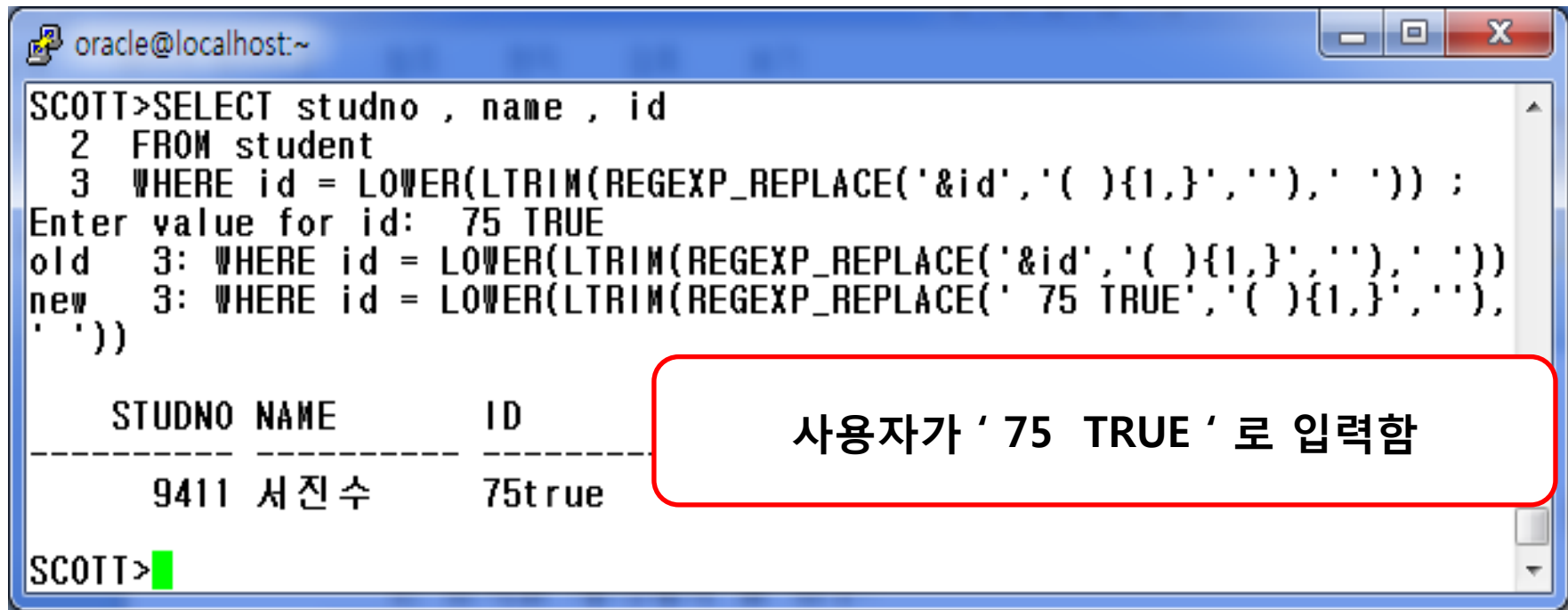


```
oracle@localhost:~  
SCOTT>SELECT studno , name , id  
2 FROM student  
3 WHERE id = LTRIM(REGEXP_REPLACE('&id','( ){1,}',''),' ');  
Enter value for id: 75 true  
old 3: WHERE id = LTRIM(REGEXP_REPLACE('&id','( ){1,}',''),' ');  
new 3: WHERE id = LTRIM(REGEXP_REPLACE(' 75 true','( ){1,}',''),' ');  
  
STUDNO NAME ID  
-----  
9411 서진수 75true  
  
SCOTT>
```

사용자가 ' 75 true' 와 같이 입력함

2. SQL 단일행 함수

- 대소문자 구분 없이 입력 받고 조회하는 경우

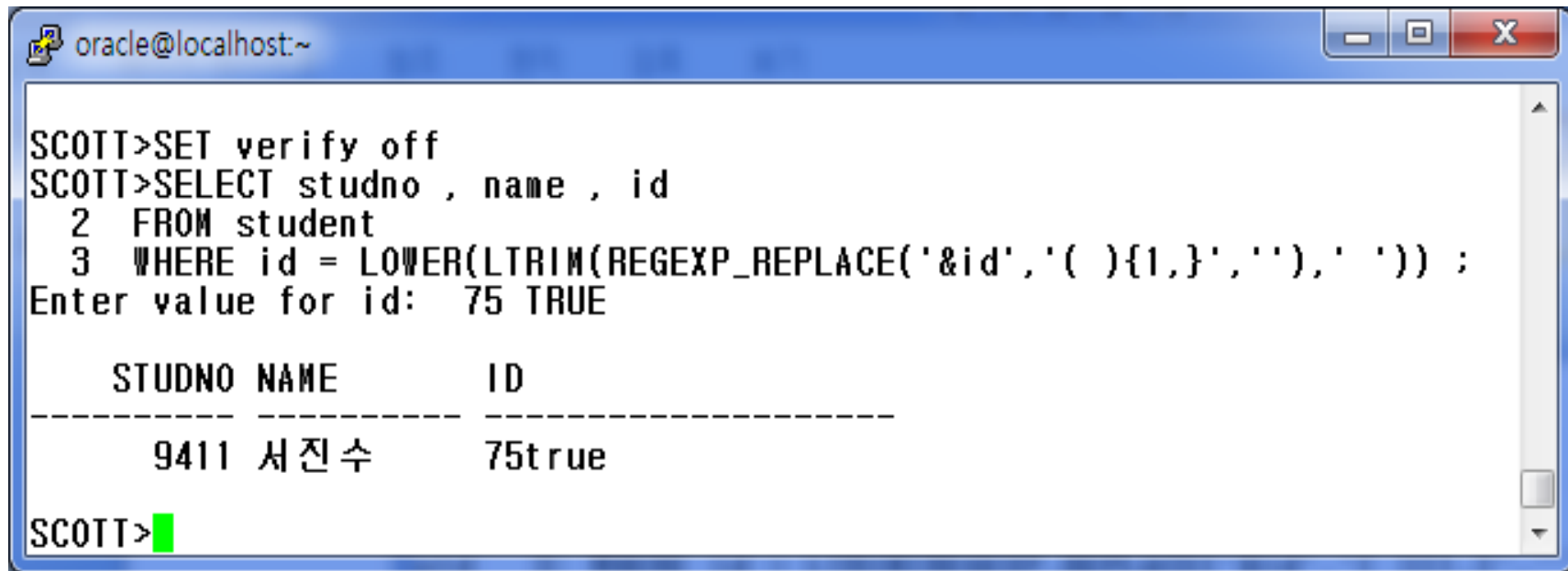


```
oracle@localhost:~  
SCOTT>SELECT studno , name , id  
2 FROM student  
3 WHERE id = LOWER(LTRIM(REGEXP_REPLACE('&id','( ){1,}',''),' ')) ;  
Enter value for id: 75 TRUE  
old 3: WHERE id = LOWER(LTRIM(REGEXP_REPLACE('&id','( ){1,}',''),' '))  
new 3: WHERE id = LOWER(LTRIM(REGEXP_REPLACE(' 75 TRUE','( ){1,}',''),  
' '))  
  
STUDNO NAME ID  
-----  
9411 서진수 75true  
  
SCOTT>
```

사용자가 ' 75 TRUE ' 로 입력함

2. SQL 단일행 함수

- SET verify off 사용하여 화면 정리하기

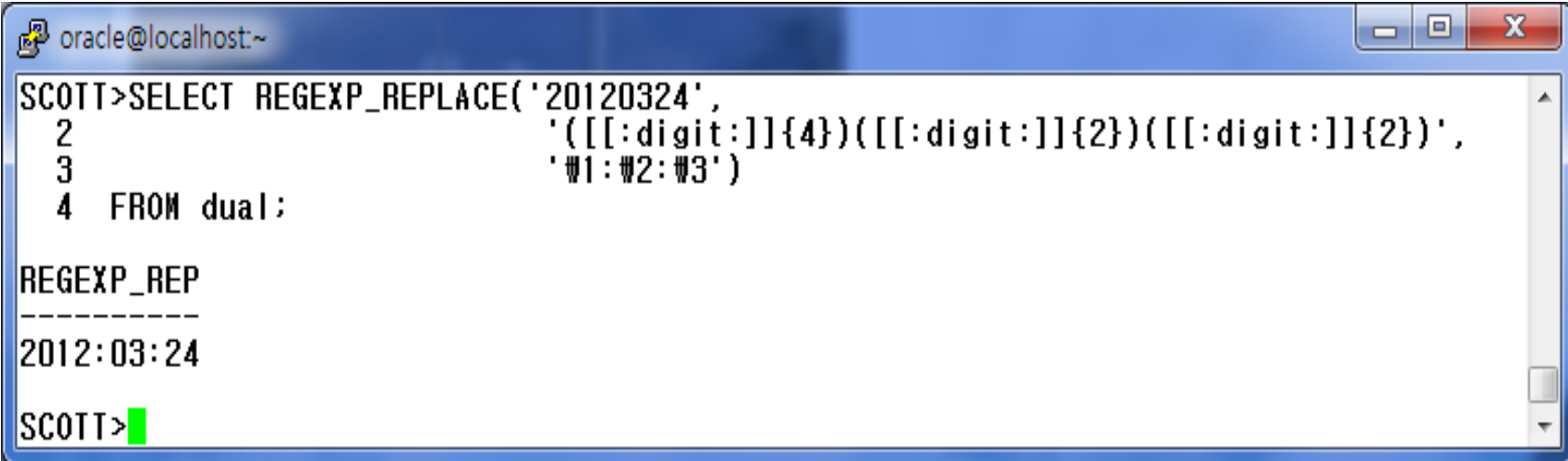


```
oracle@localhost:~  
SCOTT>SET verify off  
SCOTT>SELECT studno , name , id  
2 FROM student  
3 WHERE id = LOWER(LTRIM(REGEXP_REPLACE('&id','( ){1,}',''),' ')) ;  
Enter value for id: 75 TRUE  
  
STUDNO NAME ID  
-----  
9411 서진수 75true  
  
SCOTT>
```


2. SQL 단일행 함수

-사용 예제 6: 특정 문자열을 다른 형태로 바꿀 때

아래 화면은 '20120324' 형태로 이루어진 데이터를 '2012:03:24' 의 형태로 변형하는 예제입니다.



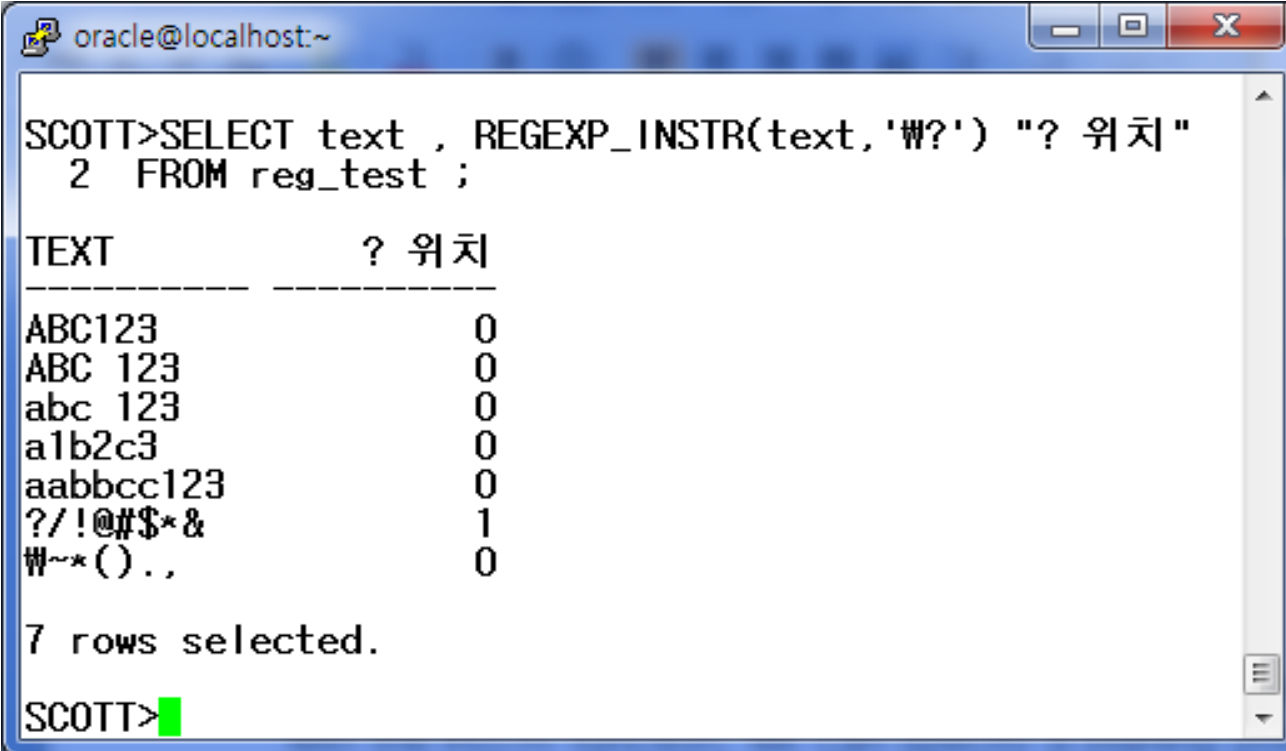
```
oracle@localhost:~  
SCOTT>SELECT REGEXP_REPLACE('20120324',  
2      '([[:digit:]]{4})([[:digit:]]{2})([[:digit:]]{2})',  
3      '#1:#2:#3')  
4  FROM dual;  
  
REGEXP_REP  
-----  
2012:03:24  
  
SCOTT>
```

2. SQL 단일행 함수

16) REGEXP_INSTR 함수

-사용 예제 1: 특정 문자의 위치를 찾는 방법

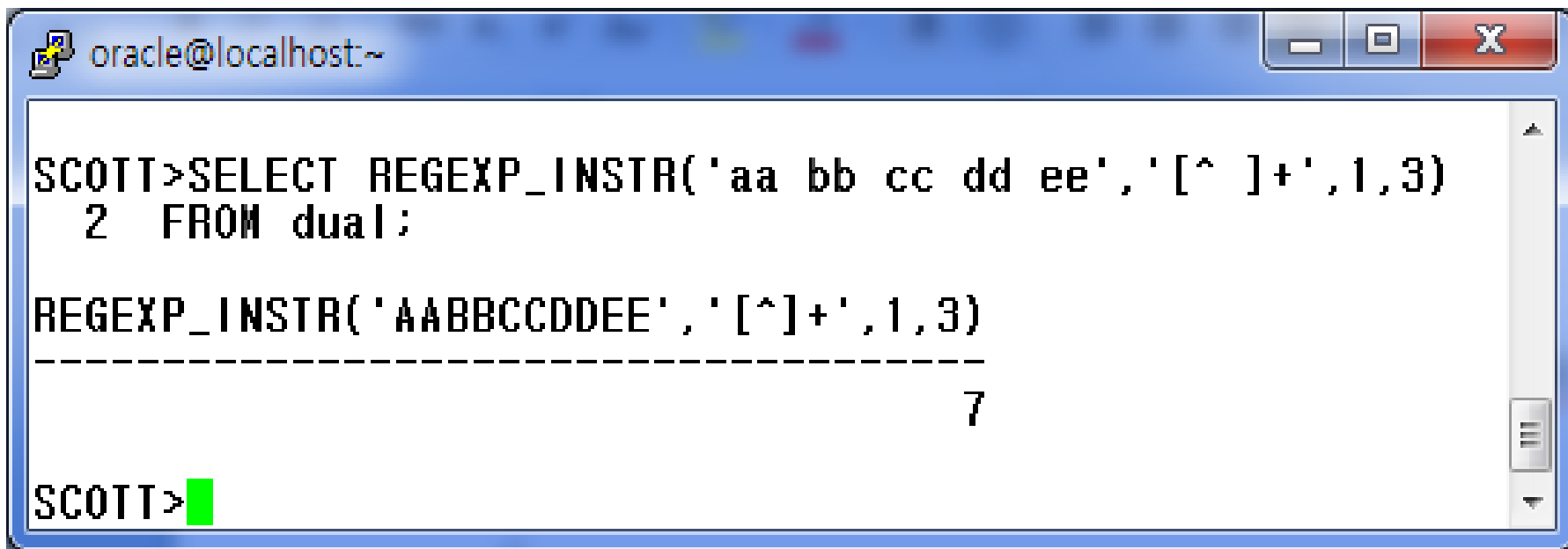
아래 예제는 text 중에서 '?' 의 위치를 찾아내는 화면입니다.



```
oracle@localhost:~  
SCOTT>SELECT text , REGEXP_INSTR(text,'#?') "? 위치"  
2 FROM reg_test ;  
  
TEXT          ? 위치  
-----  
ABC123        0  
ABC 123       0  
abc 123       0  
a1b2c3       0  
aabbcc123    0  
?/!@#*$&    1  
#~*().,      0  
  
7 rows selected.  
  
SCOTT>
```

2. SQL 단일행 함수

'aa bb cc dd ee' 에서 첫 시작이 공백이 아닌 ('[^]') 부분 중에서 1번 글자 ('aa') 부터 검사해서 3번째 위치가 출현되는 자리를 찾아내라



```
oracle@localhost:~  
SCOTT>SELECT REGEXP_INSTR('aa bb cc dd ee','[^ ]+',1,3)  
2 FROM dual;  
  
REGEXP_INSTR('AABBCCDDEE','[^ ]+',1,3)  
-----  
7  
  
SCOTT>
```

2. SQL 단일행 함수

- 사용 예제 2 : 여러 가지 옵션으로 검색하기

```

oracle@localhost:~
SCOTT>SELECT text,
2          REGEXP_INSTR(text , '[A-Z]{3}', --표현식
3                        1 , -- 검색시작위치
4                        1 , -- 찾는패턴 순번
5                        0 -- offset
6                        ) "offset 0"
7      ,
8          REGEXP_INSTR(text , '[A-Z]{3}', -- 표현식
9                        1 , -- 검색시작위치
10                       1 , -- 찾는패턴 순번
11                       1 -- offset
12                       ) "offset 1"
13      ,
14          REGEXP_INSTR(text , '[A-Z]{3}', -- 표현식
15                        1 , --검색시작위치
16                        1 , --찾는패턴 순번
17                        0 , -- offset
18                        'i' -- 대소문자구분안함
19                        ) "대소문자"
20 FROM reg_test ;

TEXT          offset 0  offset 1  대 소문자
-----
ABC123         1         4         1
ABC 123        1         4         1
abc 123        0         0         1
a1b2c3        0         0         0
aabbcc123      0         0         1
?/?!@#*$*&   0         0         0
W~*( ).,      0         0         0

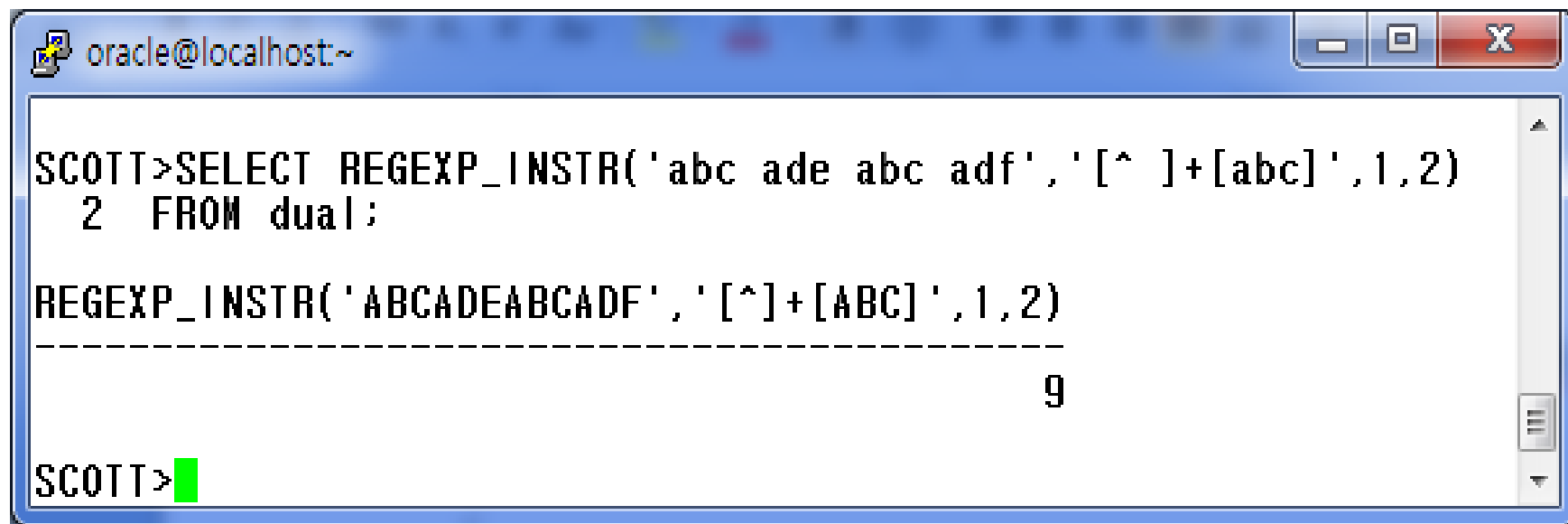
7 rows selected.

SCOTT>

```

2. SQL 단일행 함수

- 'abc ade abc def' 의 문자열에서 첫 글자가 공백이 아니고('[^]') abc 로 시작되는 두 번째 글자의 첫 위치를 표시하라

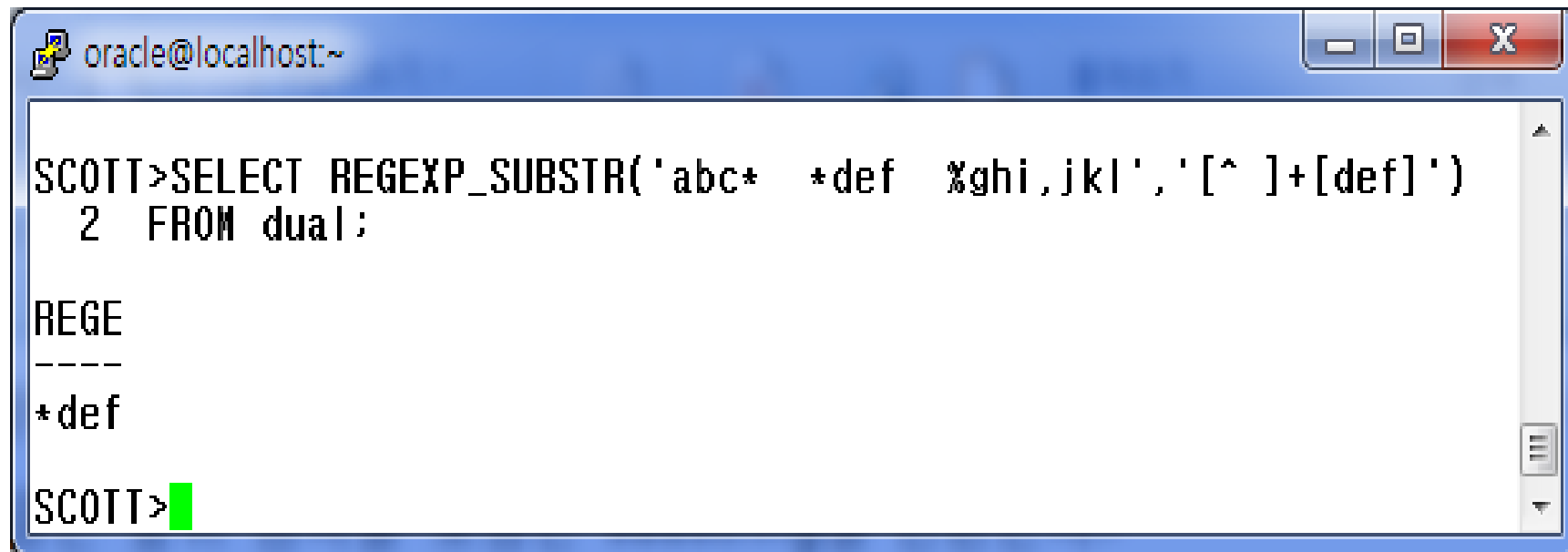


```
oracle@localhost:~  
SCOTT>SELECT REGEXP_INSTR('abc ade abc adf','[^ ]+[abc]',1,2)  
2 FROM dual;  
  
REGEXP_INSTR('ABCADEABCA DF','[^ ]+[ABC]',1,2)  
-----  
9  
  
SCOTT>
```

2. SQL 단일행 함수

17) REGEXP_SUBSTR 함수

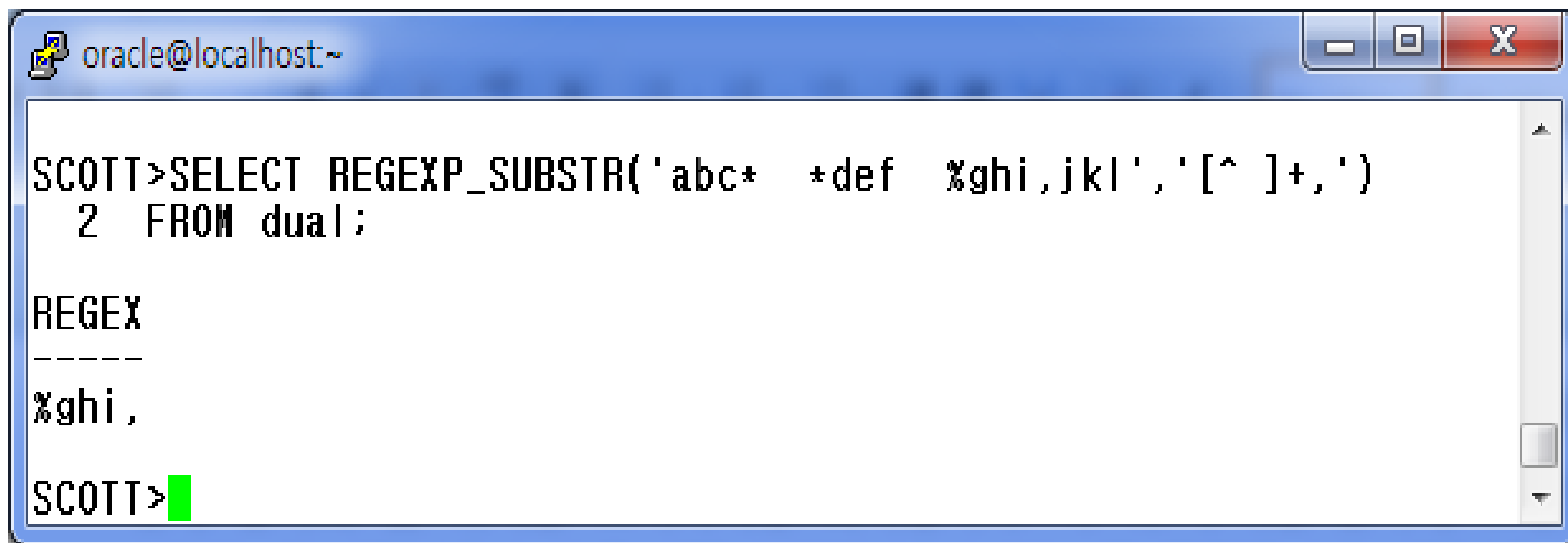
'abc* *def %ghijkl' 이란 문자열에서 첫 글자가 공백이 아니고('[^]') 그 후에 'def' 가 나오는 부분을 추출하라고 해서 '*def' 부분이 출력이 되는 예제



```
oracle@localhost:~  
SCOTT>SELECT REGEXP_SUBSTR('abc* *def %ghi,jkl','[^ ]+[def]')  
2 FROM dual;  
  
REGE  
----  
*def  
  
SCOTT>
```

2. SQL 단일행 함수

'abc* *def %ghi,jkl' 이란 문자열에서 첫 글자가 공백이 아니고('[^]')
그 후에 콤마(,) 로 구분되는 문자열을 출력하라고 해서 '%ghi,' 까지가 출
력된 예제



```
oracle@localhost:~  
SCOTT>SELECT REGEXP_SUBSTR('abc* *def %ghi,jkl','[^ ]+,')  
2 FROM dual;  
  
REGEX  
-----  
%ghi,  
  
SCOTT>
```

2. SQL 단일행 함수

- 교수테이블(professor)테이블에서 홈페이지(hpage) 주소가 있는 교수들만 조사해서 아래의 화면처럼 나오게 출력하세요.

```
oracle@localhost:~
SCOTT>col name for a10
SCOTT>col hpage for a20
SCOTT>
SCOTT>SELECT name,
2          LTRIM(REGEXP_SUBSTR(hpage,'/([[:alnum:]]+\\?.?){3,4}?'), '/') "hpage"
3 FROM professor
4 WHERE hpage IS NOT NULL ;
```

NAME	hpage
조인형	www.abc.net
박승곤	www.abc.net
송도권	www.power.com
김영조	num1.naver.com

```
SCOTT>
```


2. SQL 단일행 함수

```

oracle@localhost:~
SCOTT>COL name for a10
SCOTT>COL email for a20
SCOTT>SET pagesize 50
SCOTT>SELECT name,
       2      LTRIM(REGEXP_SUBSTR(email, '@([[:alnum:]]+#{3,4}?)', '@') "email"
       3 FROM professor ;

```

NAME	email
조인형	abc.net
박승권	abc.net
송도원	power.com
양선희	hamail.net
김영조	naver.com
주승재	naver.com
김도형	hanmir.com
나한열	empal.com
김현정	abc.com
심스기	daum.net
최슬기	def.com
박원철	hanmail.net
차범철	naver.com
바비	naver.com
전민호	jass.com
허은	daum.net

16 rows selected.

SCOTT>

- Professor 테이블에서 교수의 이름과
메일 주소를 출력하세요.
단 메일 주소는 @뒤에 있는 주소만 출력
하세요.

2. SQL 단일행 함수

3. 11g 에서 추가된 정규식 함수

2. SQL 단일행 함수

1) REGEXP_COUNT 함수 : 특정 문자의 개수를 세는 함수

```
oracle@localhost:~
SCOTT>SELECT text ,
2          REGEXP_COUNT(text, 'a')
3  FROM reg_test ;

TEXT          REGEXP_COUNT(TEXT, 'A')
-----
ABC123                0
ABC 123               0
abc 123               1
a1b2c3               1
aabbcc123            2
?/!@#$*&            0
₩~*()..             0
123123               0
123abc               1

9 rows selected.

SCOTT>
```

소문자 'a' 의 개수를
세는 경우

2. SQL 단일행 함수

```
oracle@localhost:~
SCOTT>SELECT text ,
2          REGEXP_COUNT(text,'a',3)
3  FROM reg_test ;

TEXT          REGEXP_COUNT(TEXT,'A',3)
-----
ABC123                0
ABC 123               0
abc 123               0
a1b2c3               0
aabbcc123            0
?/!@#*$*&           0
₩~*()..              0
123123               0
123abc               1

9 rows selected.

SCOTT>
```

검색 위치를 3으로 지정해서 3번째 문자 이후부터 해당 소문자 'a'가 나오는 개수를 세는 예제입니다

2. SQL 단일행 함수

```
oracle@localhost:~
SCOTT>SELECT text ,
2          REGEXP_COUNT(text, 'A') "R1" ,
3          REGEXP_COUNT(text, 'A', 1, 'i') "R2"
4 FROM reg_test ;
```

TEXT	R1	R2
ABC123	1	1
ABC 123	1	1
abc 123	0	1
a1b2c3	0	1
aabbcc123	0	2
?/!@#\$*&	0	0
#~*() ,	0	0
123123	0	0
123abc	0	1

```
9 rows selected.

SCOTT>
```

대소문자 구분 여부를 테스트 한 화면입니다.

R2 결과 컬럼을 보면 'i' 옵션으로 대소문자를 무시하고 검색하는 것을 확인할 수 있습니다

2. SQL 단일행 함수

```
oracle@localhost:~
SCOTT>SELECT text ,
2          REGEXP_COUNT(text, '.') "R1",
3          REGEXP_COUNT(text, '₩.') "R2"
4  FROM reg_test ;
```

TEXT	R1	R2
ABC123	6	0
ABC 123	7	0
abc 123	7	0
a1b2c3	6	0
aabbcc123	9	0
?/!@#\$*&	8	0
₩~*().,	7	1
123123	6	0
123abc	6	0

```
9 rows selected.

SCOTT>
```

탈출문자(Escape)를 사용하는
예를 보여줍니다.
R1 결과에서는 . (점) 이 모든
것이란 뜻으로 작동했지만 R2
결과는 ₩(탈출문자) 를 썼기
때문에 .(점)으로 인식해서 결
과를 출력 한 것입니다.

2. SQL 단일행 함수

```
oracle@localhost:~
SCOTT>SELECT text,
2          REGEXP_COUNT(text, 'aa') "R1",
3          REGEXP_COUNT(text, 'a{2}') "R2",
4          REGEXP_COUNT(text, '(a)(a)') "R3",
5  FROM reg_test ;
```

TEXT	R1	R2	R3
ABC123	0	0	0
ABC 123	0	0	0
abc 123	0	0	0
a1b2c3	0	0	0
aabbcc123	1	1	1
?/!@#\$*&	0	0	0
₩~*().,	0	0	0
123123	0	0	0
123abc	0	0	0

```
9 rows selected.
SCOTT>
```

어떤 문자를 검색할 때 사용하는 방법을 3가지로 살펴 보았습니다.

어떤 방법으로 검색하든 동일한 결과가 나오음을 알 수 있습니다.

2. SQL 단일행 함수

2) 11g REGEXP_SUBSTR 추가 기능(Sub Expression 사용하기)

```

oracle@localhost:~
SCOTT>
SCOTT>SELECT hpage ,
2          REGEXP_SUBSTR(hpage, '(\\.)([a-z]+)(\\.)' , -- 찾고싶은 패턴
3          1 , -- 검색시작위치
4          1 , -- 검색 occurrence
5          'i' , -- 대소문자 구분 없음
6          1 -- 11g 부터 추가된 기능
7          ) "REGEXP"
8 FROM professor
9 WHERE hpage is not null ;

HPAGE                                REGEXP
-----                                -
http://www.abc.net                   .
http://www.abc.net                   .
http://www.power.com                 .
http://num1.naver.com                .

SCOTT>

```


2. SQL 단일행 함수

```

oracle@localhost:~
SCOTT>SELECT hpage ,
2          REGEXP_SUBSTR(hpage, '(\\.)([a-z]+)(\\.)' , -- 찾고싶은 패턴
3          1 , -- 검색시작위치
4          1 , -- 검색 occurrence
5          'i' , -- 대소문자 구분 없음
6          2 -- 11g 부터 추가된 기능
7          ) "REGEXP"
8 FROM professor
9 WHERE hpage is not null ;

HPAGE                                REGEXP
-----                                -
http://www.abc.net                  abc
http://www.abc.net                  abc
http://www.power.com                power
http://num1.naver.com                naver

SCOTT>

```

2. SQL 단일행 함수

4. 숫자 함수

2. SQL 단일행 함수

이 름	의 미	사 용 예
ROUND	주어진 숫자를 반올림 한 후 출력함	ROUND(12.345,2) -> 12.35
TRUNC	주어진 숫자를 버림 한 후 출력함	TRUNC(12.345,2) -> 12.34
MOD	주어진 숫자를 나누기 한 후 나머지 값 출력함	MOD(12,10) -> 2
CEIL	주어진 숫자와 가장 근접한 큰 정수 출력함	CEIL(12.345) -> 13
FLOOR	주어진 숫자와 가장 근접한 작은 정수 출력함	FLOOR(12.345) -> 12
POWER	주어진 숫자1의 숫자2 승을 출력함	POWER(3,2) -> 9

2. SQL 단일행 함수

1) ROUND 함수

- 문 법: ROUND(숫자 , 원하는 자리수)

```
oracle@localhost:~
SCOTT>SELECT ROUND(12.345) ROUND1,
2          ROUND(12.345,2) ROUND2,
3          ROUND(12.345,-1) ROUND3
4  FROM dual;
```

ROUND1	ROUND2	ROUND3
12	12.35	10

```
SCOTT>
```

자리 수에 주의하세요!

2. SQL 단일행 함수

2) TRUNC 함수

- 문 법: TRUNC(숫자 , 원하는 자리수)

```
oracle@localhost:~
SCOTT>SELECT TRUNC(12.345) TRUNC1,
2            TRUNC(12.345,2) TRUNC2,
3            TRUNC(12.345,-1) TRUNC3
4 FROM dual ;

      TRUNC1      TRUNC2      TRUNC3
-----
          12      12.34         10

SCOTT>
```

2. SQL 단일행 함수

3) MOD , CEIL , FLOOR 함수

```
oracle@localhost:~
SCOTT>SELECT MOD(12,10) "Mod" ,
2          CEIL(12.345) "Ceil",
3          FLOOR(12.345) "Floor"
4  FROM dual;

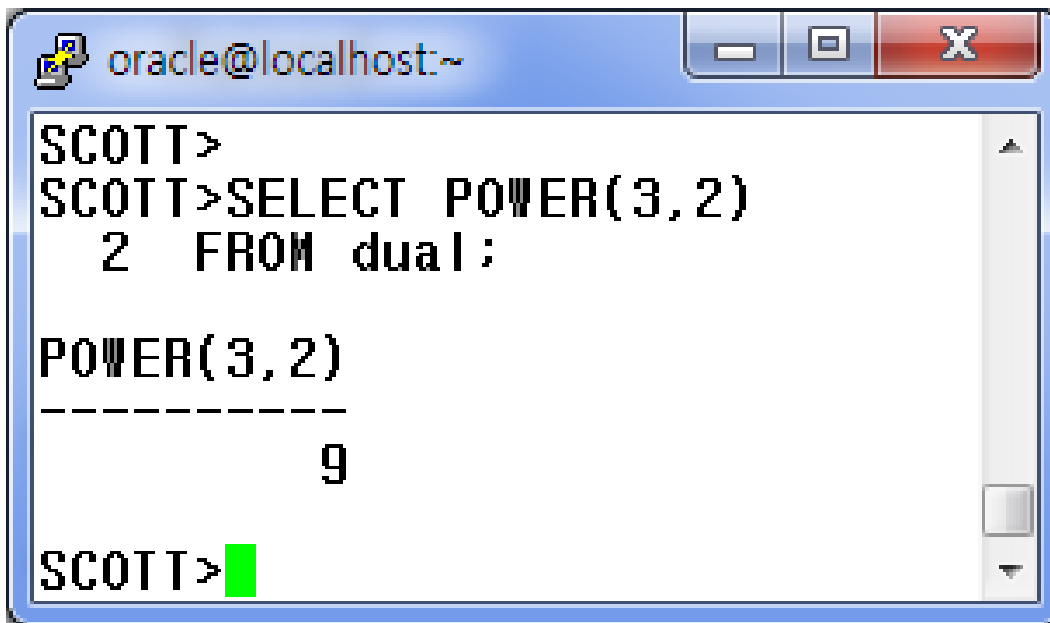
      Mod      Ceil      Floor
-----
         2         13         12

SCOTT>
```

2. SQL 단일행 함수

4) POWER 함수

- 문 법: POWER(숫자1 , 숫자2)



```
oracle@localhost:~  
SCOTT>  
SCOTT>SELECT POWER(3,2)  
2 FROM dual;  
  
POWER(3,2)  
-----  
9  
  
SCOTT>
```

2. SQL 단일행 함수

5. 날짜 함수

2. SQL 단일행 함수

- 10월 23일 + 3 은 10월 26일로 생각합니다.
- 10월 23일 - 3 은 10월 20일로 생각합니다.
- 10월 23일 - 10월 20일 은 3일 차이 난다 라고 생각합니다.

함 수 명	의 미	결 과
SYSDATE	시스템의 현재 날짜와 시간	날짜
MONTHS_BETWEEN	두 날짜 사이의 개월 수	숫자
ADD_MONTHS	주어진 날짜에 개월을 더함	날짜
NEXT_DAY	주어진 날짜를 기준으로 돌아오는 날짜 출력	날짜
LAST_DAY	주어진 날짜가 속한 달의 마지막 날짜 출력	날짜
ROUND	주어진 날짜를 반올림	날짜
TRUNC	주어진 날짜를 버림	날짜

2. SQL 단일행 함수

1) SYSDATE 함수 (현재 날짜/시간 출력)

```
oracle@localhost:~
SCOTT>SELECT SYSDATE FROM dual;

SYSDATE
-----
20-MAR-12

SCOTT>alter session set NLS_DATE_FORMAT='YYYY-MM-DD:HH24:MI:SS';
Session altered.

SCOTT>SELECT SYSDATE FROM DUAL ;

SYSDATE
-----
2012-03-20:15:06:36

SCOTT>
```

2. SQL 단일행 함수

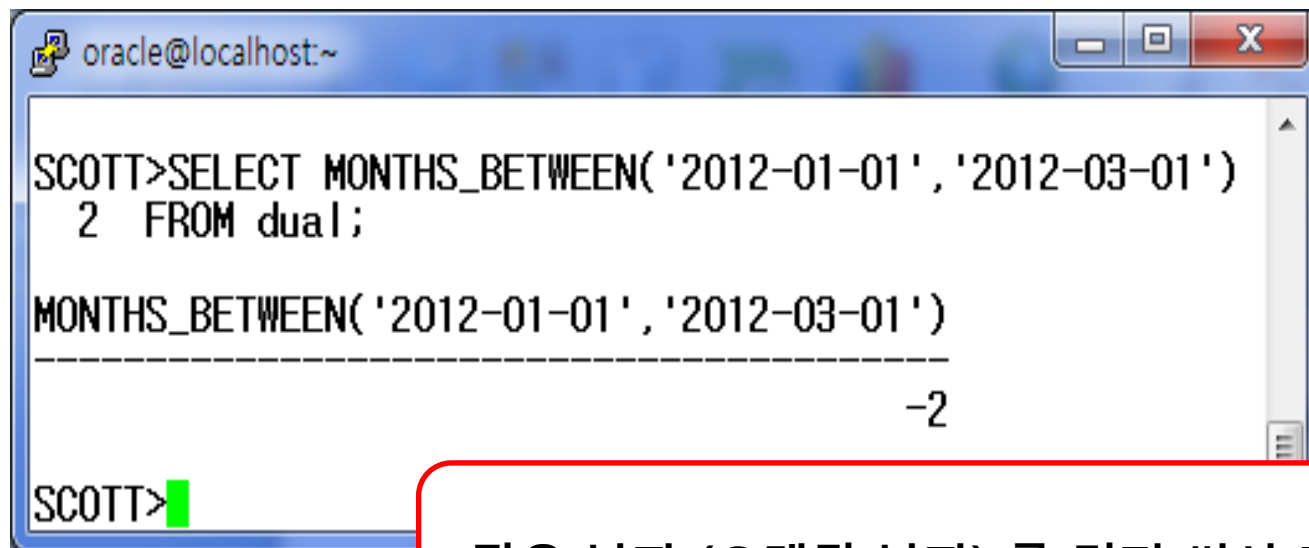
2) MONTHS_BETWEEN 함수 (두 날짜 사이의 개월 수 계산)

```
oracle@localhost:~  
SCOTT>SELECT MONTHS_BETWEEN( '2012-03-01' , '2012-01-01' )  
2 FROM DUAL ;  
  
MONTHS_BETWEEN( '2012-03-01' , '2012-01-01' )  
-----  
2  
  
SCOTT>
```

2. SQL 단일행 함수

- MONTHS_BETWEEN 함수 주요 특징

1. 큰 날짜를 먼저 써야 합니다.



```

oracle@localhost:~
SCOTT>SELECT MONTHS_BETWEEN('2012-01-01','2012-03-01')
        2  FROM dual;

MONTHS_BETWEEN('2012-01-01','2012-03-01')
-----
-2

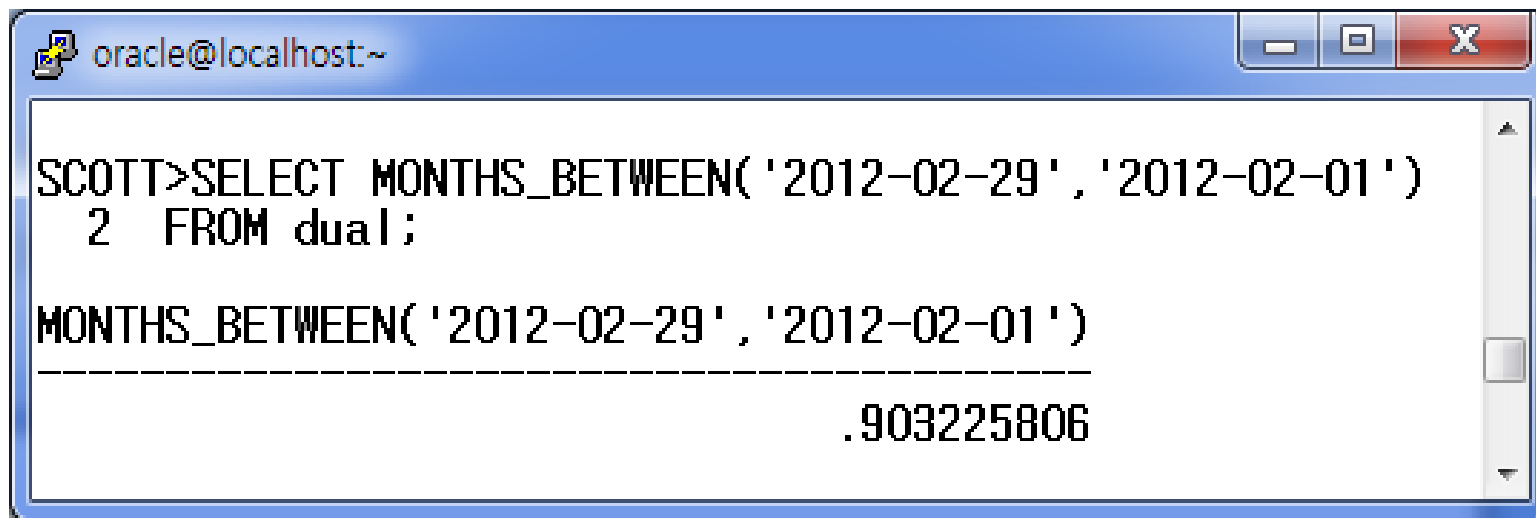
SCOTT>
    
```

작은 날짜 (오래된 날짜) 를 먼저 써서 마이너스 값 나옴

2. SQL 단일행 함수

2. 두 날짜가 같은 달에 속해 있으면 특정 규칙으로 계산된 값이 나옵니다

* 1개월이 29일 인 2012년 2월을 조회한 화면



```

oracle@localhost:~
SCOTT>SELECT MONTHS_BETWEEN('2012-02-29','2012-02-01')
2 FROM dual;

MONTHS_BETWEEN('2012-02-29','2012-02-01')
-----
.903225806
    
```

2. SQL 단일행 함수

* 1개월이 30일 인 2012년 4월을 조회한 화면

```
oracle@localhost:~
SCOTT>SELECT MONTHS_BETWEEN('2012-04-30','2012-04-01')
         2 FROM dual;

MONTHS_BETWEEN('2012-04-30','2012-04-01')
-----
.935483871
```

* 1 개월이 31일 인 2012년 1월을 조회한 화면

```
oracle@localhost:~
SCOTT>SELECT MONTHS_BETWEEN('2012-01-31','2012-01-01')
         2 FROM dual;

MONTHS_BETWEEN('2012-01-31','2012-01-01')
-----
.967741935
```

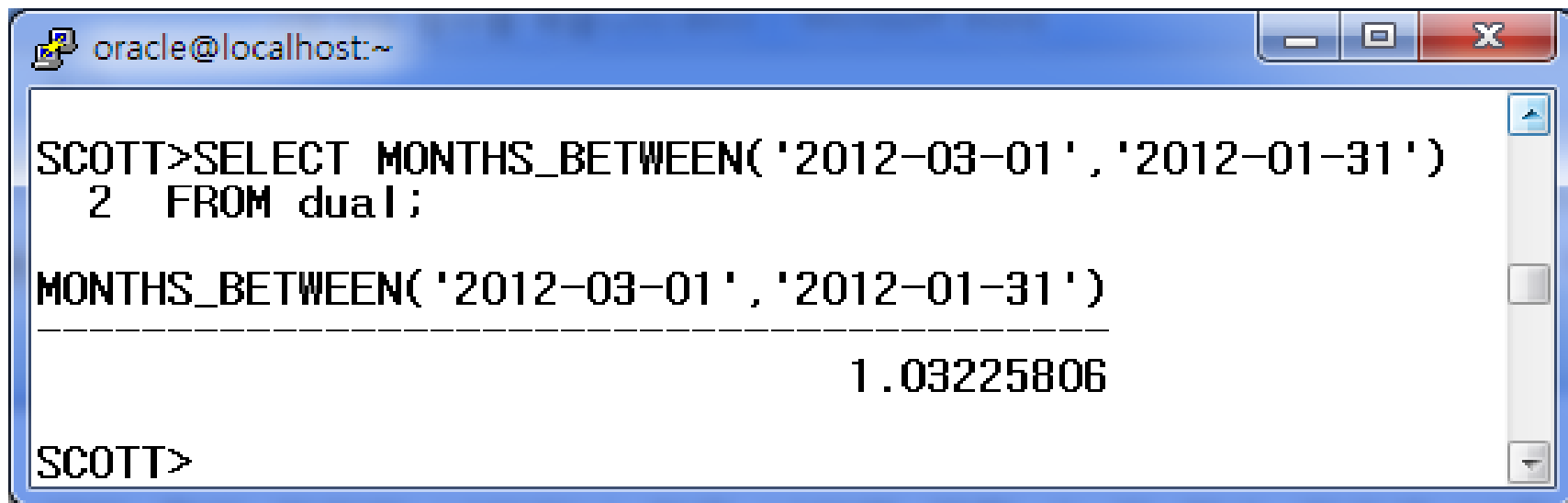
2. SQL 단일행 함수

3. 두 날짜가 모두 해당 월의 마지막 날이거나 처음 날이면 개월 수 가 정수 값으로 나옵니다.

```
oracle@localhost:~  
SCOTT>SELECT MONTHS_BETWEEN( '2012-07-01' , '2012-01-01' )  
2 FROM dual ;  
  
MONTHS_BETWEEN( '2012-07-01' , '2012-01-01' )  
-----  
6  
  
SCOTT>
```

2. SQL 단일행 함수

4. 두 날짜 중 하나는 시작일이고 하나는 마지막 일이면 특정 규칙으로 계산된 값이 나옵니다.



```
oracle@localhost:~  
SCOTT>SELECT MONTHS_BETWEEN( '2012-03-01' , '2012-01-31' )  
2 FROM dual ;  
  
MONTHS_BETWEEN( '2012-03-01' , '2012-01-31' )  
-----  
1.03225806  
  
SCOTT>
```


2. SQL 단일행 함수

* 2 월은 일수가 달라도 반환 값은 같습니다.

```
oracle@localhost:~
SCOTT>SELECT MONTHS_BETWEEN('2011-03-01','2011-02-28')
2 FROM dual
3 ;

MONTHS_BETWEEN('2011-03-01','2011-02-28')
-----
.129032258

SCOTT>
```

2월이 28일까지 있는 **2011 년 2월**
을 조회한 화면 입니다

```
oracle@localhost:~
SCOTT>SELECT MONTHS_BETWEEN('2012-03-01','2012-02-28')
2 FROM dual;

MONTHS_BETWEEN('2012-03-01','2012-02-28')
-----
.129032258
```

2월이 29일까지 있는 **2012 년 2월**
을 조회한 화면 입니다

2. SQL 단일행 함수

- MONTHS_BETWEEN 과 전통적인 방법의 차이

oracle@localhost:~

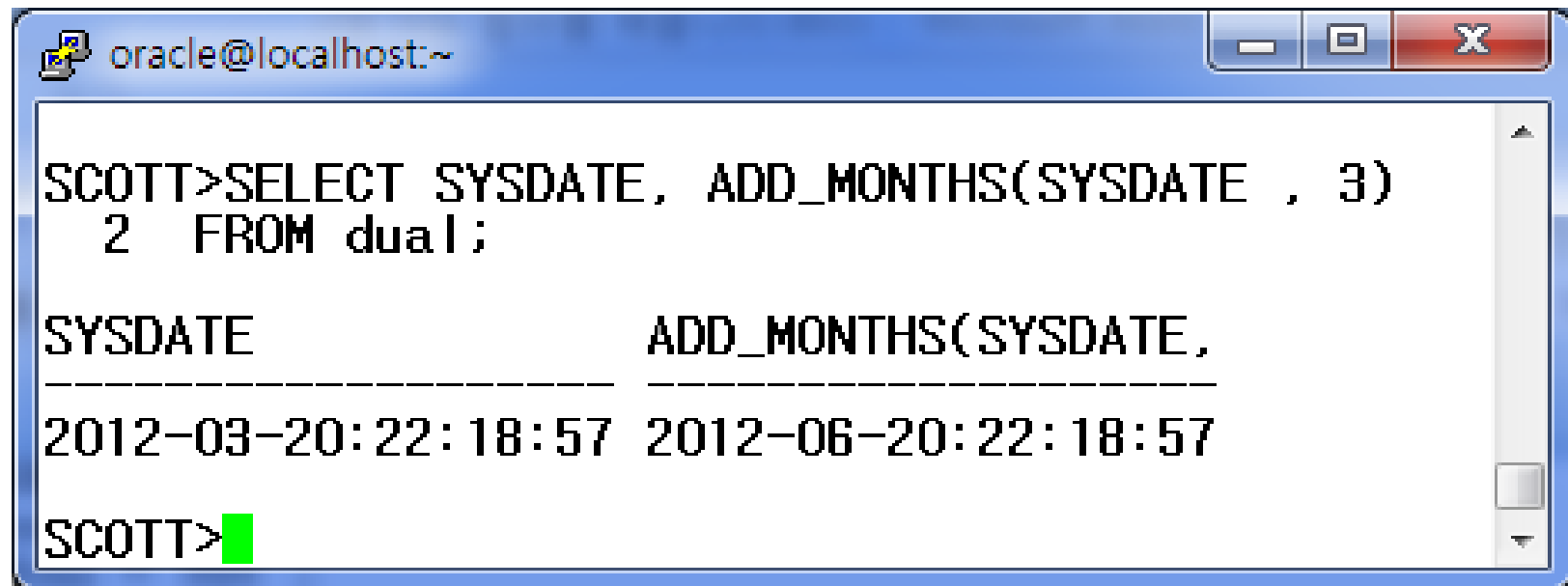
```
SCOTT>SELECT name,SYSDATE,hiredate ,
2          ROUND(MONTHS_BETWEEN(SYSDATE,hiredate),2) Date_1 ,
3          ROUND(((SYSDATE - hiredate) / 31),2) Date_2
4 FROM professor
5 WHERE deptno = 101 ;
```

NAME	SYSDATE	HIREDATE	DATE_1	DATE_2
조인형	2012-03-20:20:32:47	1980-06-23:00:00:00	380.93	374
박승곤	2012-03-20:20:32:47	1987-01-30:00:00:00	301.71	296.19
송도권	2012-03-20:20:32:47	1998-03-22:00:00:00	167.96	164.93

SCOTT>

2. SQL 단일행 함수

3) ADD_MONTHS 함수



The screenshot shows a terminal window titled 'oracle@localhost:~'. The user 'SCOTT' has entered the following SQL query:

```
SCOTT>SELECT SYSDATE, ADD_MONTHS(SYSDATE , 3)
2 FROM dual;
```

The output of the query is displayed in two columns:

SYSDATE	ADD_MONTHS(SYSDATE,
2012-03-20:22:18:57	2012-06-20:22:18:57

The prompt 'SCOTT>' is followed by a green cursor.

2. SQL 단일행 함수

4) NEXT_DAY 함수

```
oracle@localhost:~
SCOTT>SELECT SYSDATE , NEXT_DAY(SYSDATE, 'MON')
2 FROM dual;

SYSDATE                NEXT_DAY(SYSDATE, 'M
-----
2012-03-20:22:20:20  2012-03-26:22:20:20

SCOTT>
```

2. SQL 단일행 함수

5) LAST_DAY 함수

```
oracle@localhost:~  
SCOTT>SELECT SYSDATE , LAST_DAY(SYSDATE)  
2 FROM dual;  
  
SYSDATE                LAST_DAY(SYSDATE)  
-----  
2012-03-20:22:22:10  2012-03-31:22:22:10  
  
SCOTT>
```

2. SQL 단일행 함수

6) 날짜의 ROUND , TRUNC 함수

```
oracle@localhost:~
SCOTT>SELECT SYSDATE,ROUND(SYSDATE),TRUNC(SYSDATE)
2 FROM dual;

SYSDATE                ROUND(SYSDATE)        TRUNC(SYSDATE)
-----
2012-03-20:22:28:52    2012-03-21:00:00:00    2012-03-20:00:00:00
SCOTT>
```

2. SQL 단일행 함수

6. 형 변환 함수

2. SQL 단일행 함수

- 오라클의 주요 데이터 형

데이터 타입	설 명
CHAR(n)	고정길이의 문자를 저장합니다. 최대값은 2000 bytes 입니다
VARCHAR2(n)	변하는 길이의 문자를 저장합니다. 최대값은 4000 bytes 입니다
NUMBER(p,s)	숫자 값을 저장합니다.p 는 전체 자리수로 1-38 자리까지 가능하고 s는 소수점 이하 자리수로 -84 ~ 127 자리까지 가능합니다.
DATE	총 7Byte로 BC 4712년 1월 1일부터 AD 9999년 12월 31일까지의 날짜를 저장할 수 있습니다.
LONG	가변 길이의 문자를 저장하며 최대 2GB 까지 저장 할 수 있습니다
CLOB	가변 길이의 문자를 저장하며 최대 4GB 까지 저장 할 수 있습니다
BLOB	가변 길이의 바이너리 데이터를 최대 4GB 까지 저장 할 수 있습니다
RAW(n)	원시 이진 데이터로 최대 2000 bytes 까지 저장할 수 있습니다
LONG RAW(n)	원시 이진 데이터로 최대 2GB까지 저장할 수 있습니다
BFILE	외부 파일에 저장된 데이터로 최대 4GB 까지 저장 할 수 있습니다

2. SQL 단일행 함수

- 묵시적 (자동) 형 변환

```
oracle@localhost:~
SCOTT>SELECT 1 + '1' FROM dual;

      1+'1'
-----
         2

SCOTT>
```

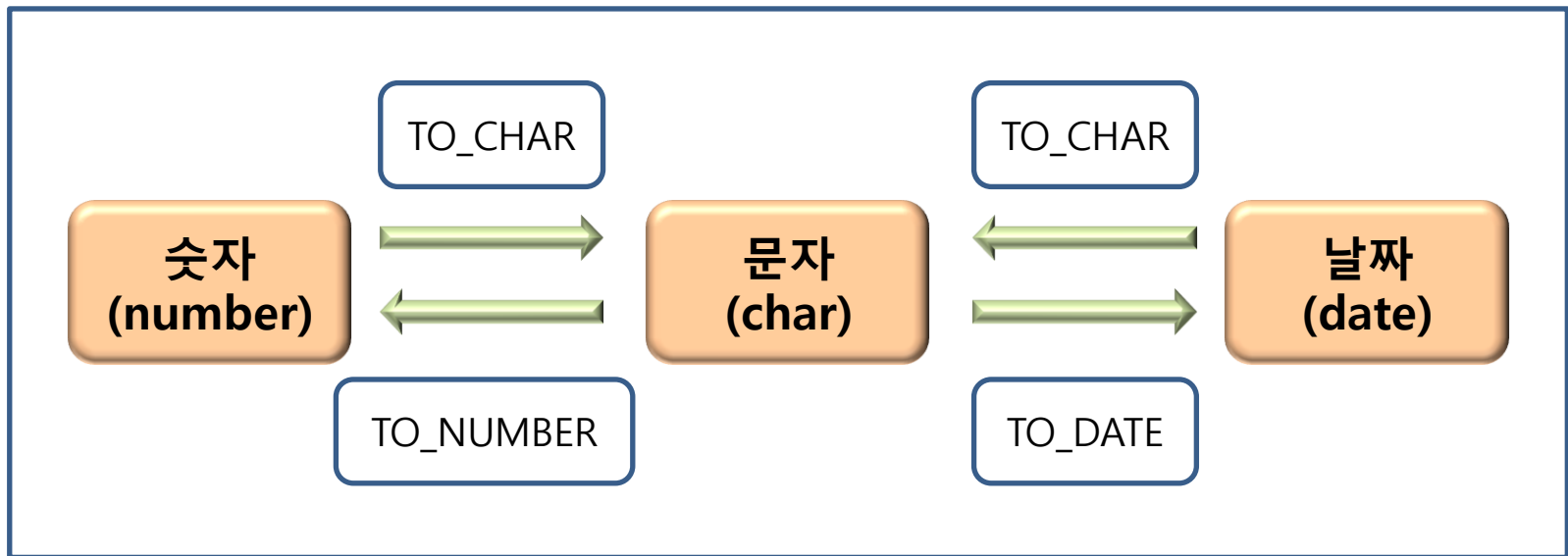
SQL> SELECT 1 + to_number('1')
2 FROM dual ;

```
oracle@localhost:~
SCOTT>SELECT 1 + 'A' FROM dual;
SELECT 1 + 'A' FROM dual
          *
ERROR at line 1:
ORA-01722: invalid number

SCOTT>
```

2. SQL 단일행 함수

- 명시적 (수동) 형 변환



2. SQL 단일행 함수

1) TO_CHAR 함수 (날짜를 문자로 형 변환하기)

문 법: TO_CHAR(원래 날짜 , '원하는 모양')

- * **년도** : 'YYYY' – 연도를 4자리로 표현합니다. 예: 1999
- 'RRRR' – 2000 년 이후에 등장한 새로운 날짜 표기법입니다.
- 'YY' – 연도를 끝의 2 자리만 표시합니다. 예 : 99
- 'YEAR' – 연도의 영문 이름 전체를 표시합니다.

```

oracle@localhost:~
SCOTT>SELECT TO_CHAR(SYSDATE, 'YYYY') "YYYY" ,
2            TO_CHAR(SYSDATE, 'RRRR') "RRRR" ,
3            TO_CHAR(SYSDATE, 'YY') "YY" ,
4            TO_CHAR(SYSDATE, 'YEAR') "YEAR"
5 FROM dual ;

YYYY RRRR YY YEAR
-----
2012 2012 12 TWENTY TWELVE

SCOTT>
    
```

2. SQL 단일행 함수

월 : MM – 월을 숫자 2자리로 표현합니다. 예: 10

MON – 월을 뜻하는 영어 3글자로 표시합니다. 예: OCT

MONTH – 월을 뜻하는 영어 이름 전체를 표시합니다

```
oracle@localhost:~
SCOTT>SELECT TO_CHAR(SYSDATE, 'MM') "MM" ,
2          TO_CHAR(SYSDATE, 'MON') "MON" ,
3          TO_CHAR(SYSDATE, 'MONTH') "MONTH"
4  FROM dual;

MM MON    MONTH
-- --
03 MAR    MARCH

SCOTT>
```

2. SQL 단일행 함수

일 : DD – 일을 숫자 2자리로 표시합니다. 예: 12
 DAY – 요일에 해당하는 영어 명칭을 표시합니다.
 DDTH – 몇 번째 날인지를 표시합니다.

```
oracle@localhost:~
SCOTT>SELECT TO_CHAR(SYSDATE, 'DD') "DD",
2          TO_CHAR(SYSDATE, 'DAY') "DAY",
3          TO_CHAR(SYSDATE, 'DDTH') "DDTH"
4  FROM dual;

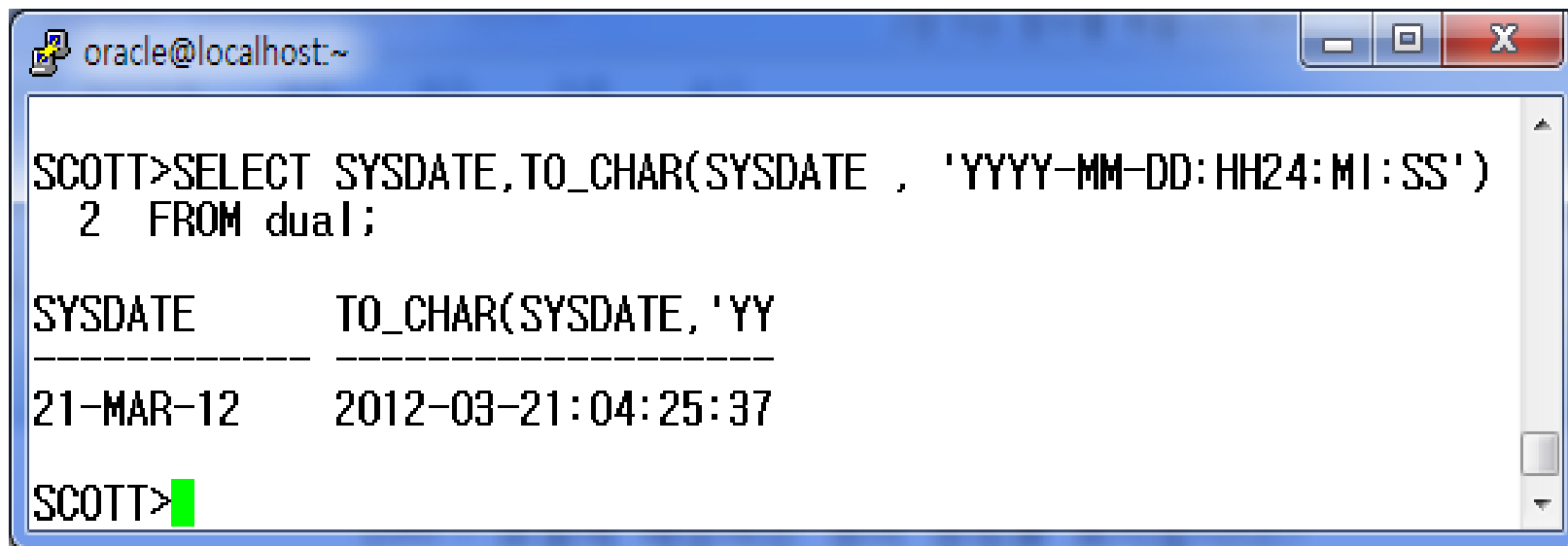
DD DAY          DDTH
-- --
21 WEDNESDAY    21ST

SCOTT>
```

2. SQL 단일행 함수

시간 : HH24 – 하루를 24시간으로 표시합니다
HH – 하루를 12 시간으로 표시합니다.

분 : MI 로 표시합니다.
초 : SS 로 표시합니다.



```

oracle@localhost:~
SCOTT>SELECT SYSDATE,TO_CHAR(SYSDATE , 'YYYY-MM-DD:HH24:MI:SS')
2  FROM dual;

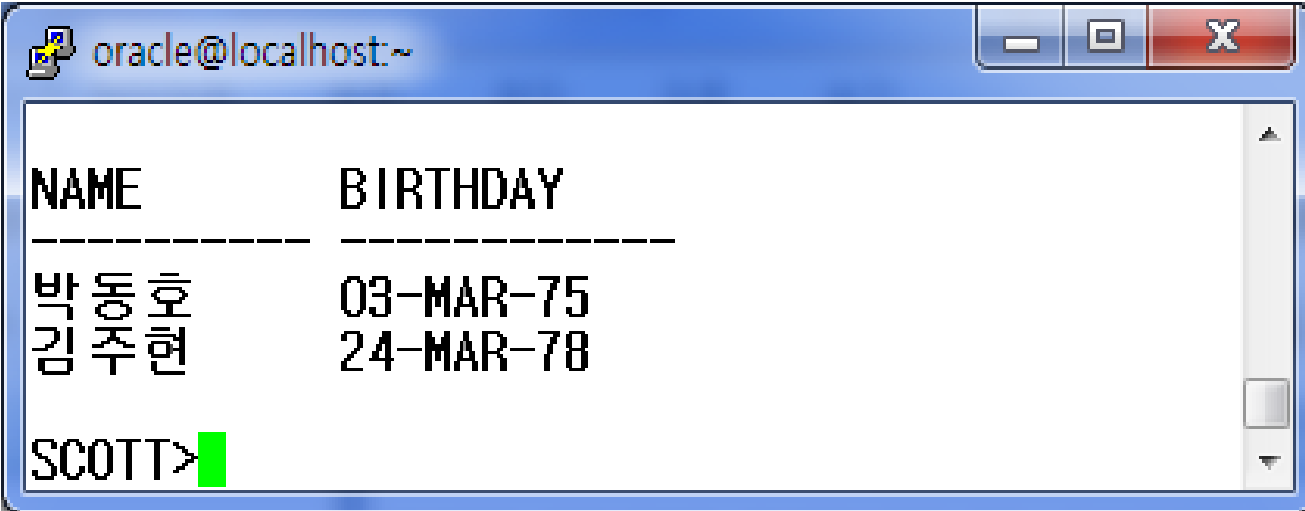
SYSDATE          TO_CHAR(SYSDATE,'YY
-----
21-MAR-12        2012-03-21:04:25:37

SCOTT>
  
```

2. SQL 단일행 함수

** 형 변환 함수 퀴즈 1

-Student 테이블의 birthday 칼럼을 참조하여 생일이 3월인 학생의 이름과 birthday 를 출력하세요.



```

oracle@localhost:~
NAME          BIRTHDAY
-----
박동호        03-MAR-75
김주현        24-MAR-78

SCOTT>
  
```

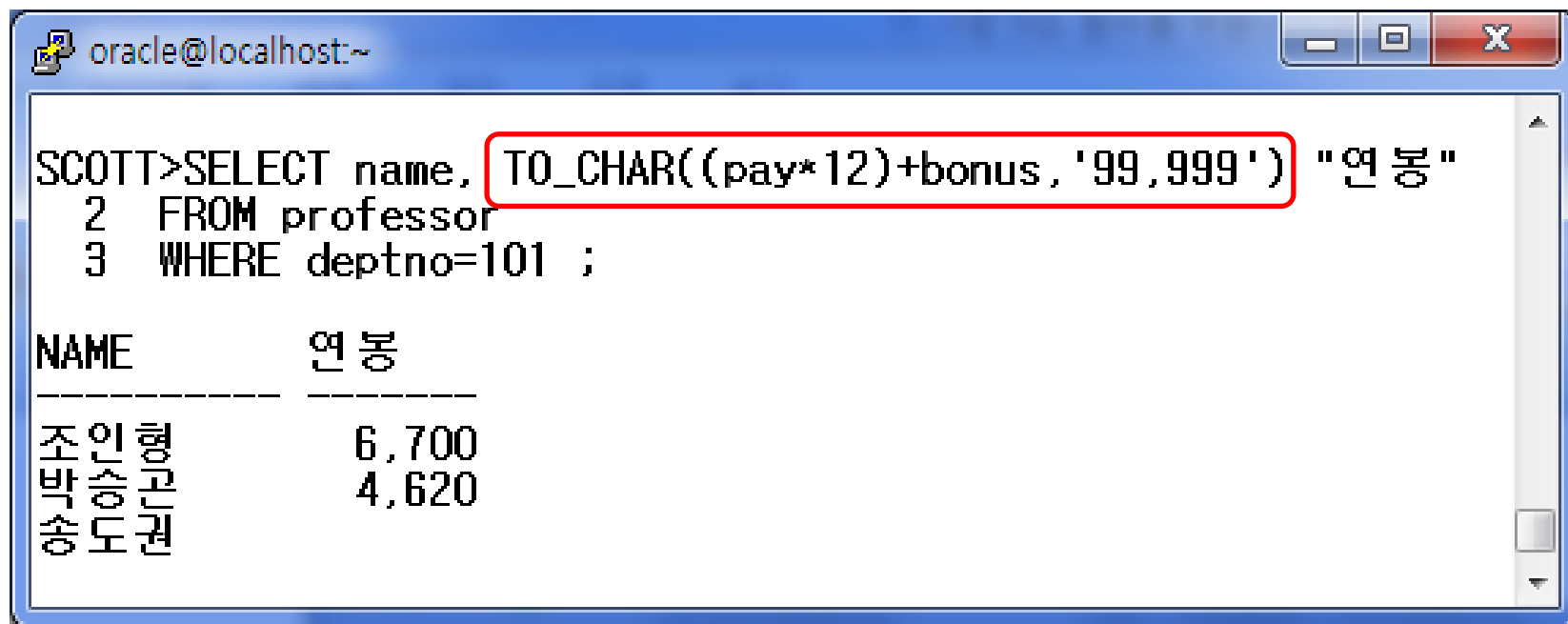
2. SQL 단일행 함수

2) TO_CHAR 함수 (숫자를 문자로 형 변환하기)

종류	의 미	사 용 예	결 과
9	9 하나당 1 자리를 의미	TO_CHAR(1234,'99999')	1234
0	빈자리를 0으로 표시	TO_CHAR(1234,'099999')	001234
\$	\$ 표시를 붙여서 표시	TO_CHAR(1234,'\$9999')	\$1234
.	소수점 이하를 표시	TO_CHAR(1234,'9999.99')	1234.00
,	천 단위 구분기호를 표시	TO_CHAR(12345,'99,999')	12,345

2. SQL 단일행 함수

- professor 테이블을 참조하여 101번 학과 교수들의 이름과 연봉을 출력하세요.
단 연봉은 $(pay * 12) + bonus$ 로 계산하시고 천 단위 구분기호로 표시하세요.



The screenshot shows a terminal window titled 'oracle@localhost:~'. The user 'SCOTT' has entered the following SQL query:

```
SCOTT>SELECT name, TO_CHAR((pay*12)+bonus, '99,999') "연 봉"
2 FROM professor
3 WHERE deptno=101 ;
```

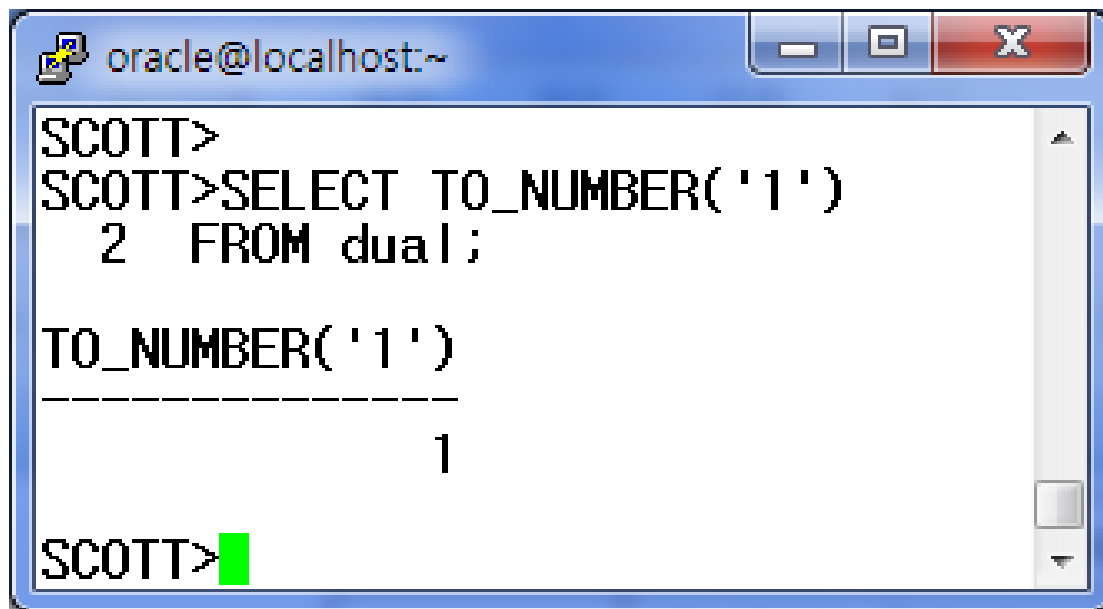
The query results are displayed in a table format:

NAME	연 봉
조인형	6,700
박승권	4,620
송노권	

2. SQL 단일행 함수

3) TO_NUMBER 함수

문 법: TO_NUMBER('1')



```

oracle@localhost:~
SCOTT>
SCOTT>SELECT TO_NUMBER('1')
           2  FROM dual;

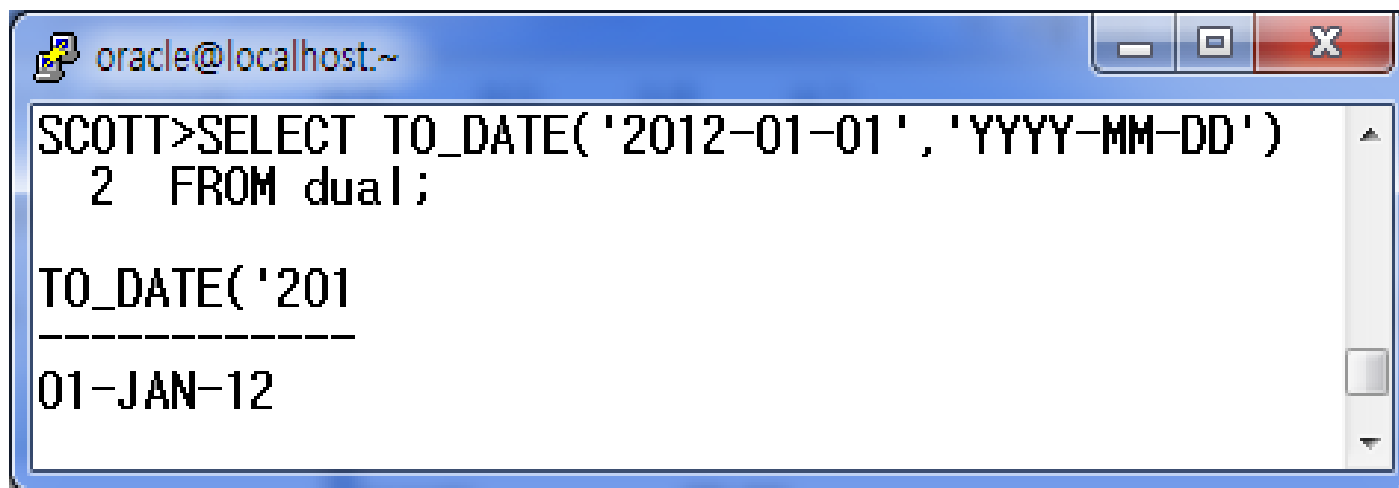
TO_NUMBER('1')
-----
                1

SCOTT>
  
```

2. SQL 단일행 함수

4) TO_DATE 함수

문 법: TO_DATE('문자','날짜 포맷')



```

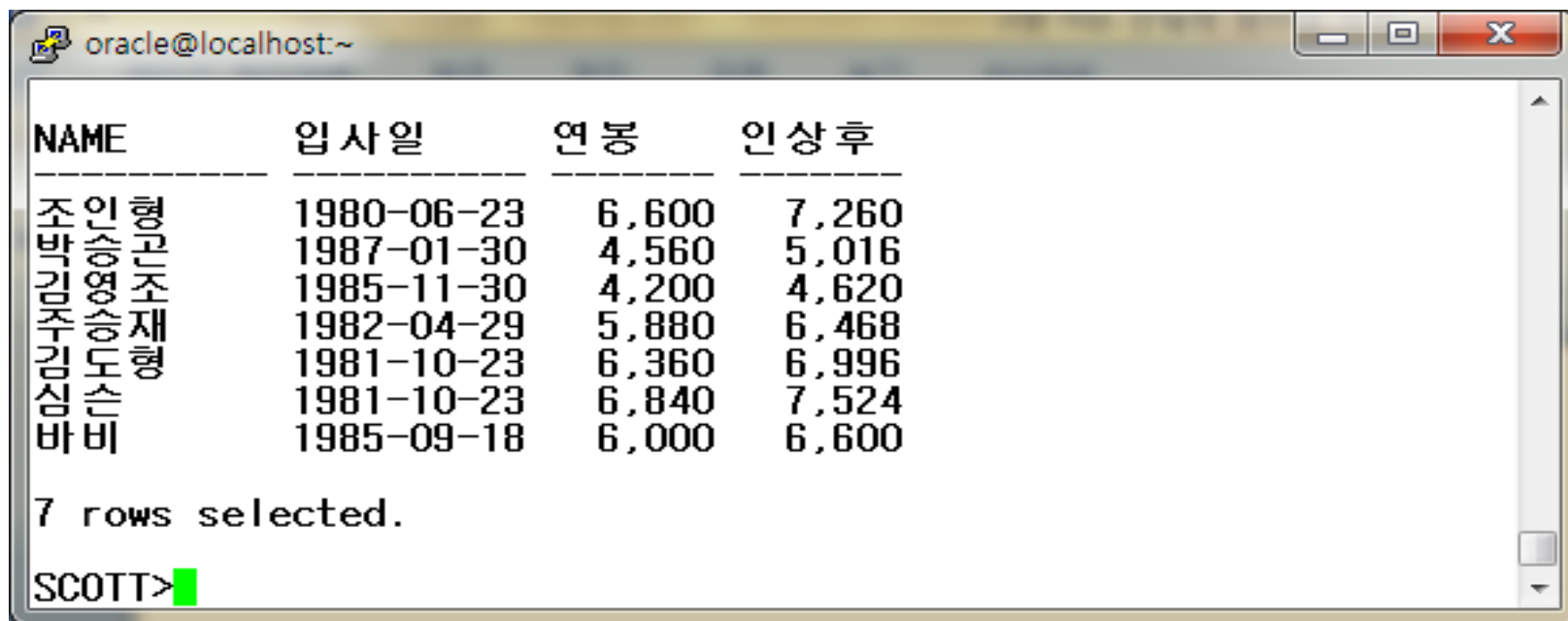
oracle@localhost:~
SCOTT>SELECT TO_DATE('2012-01-01','YYYY-MM-DD')
2 FROM dual;

TO_DATE('201
-----
01-JAN-12
    
```

2. SQL 단일행 함수

** 형 변환 함수 퀴즈 2

-Professor 테이블을 사용하여 1990년 이전에 입사한 교수명 과 입사일, 현재 연봉과 10% 인상 후 연봉을 출력하세요. 연봉은 상여금(bonus)를 제외한 (pay*12) 로 계산하고 연봉과 인상 후 연봉은 천 단위 구분 기호를 추가하여 출력하세요.



NAME	입사일	연봉	인상후
조인형	1980-06-23	6,600	7,260
박승곤	1987-01-30	4,560	5,016
김영조	1985-11-30	4,200	4,620
주승재	1982-04-29	5,880	6,468
김도형	1981-10-23	6,360	6,996
심슨	1981-10-23	6,840	7,524
바비	1985-09-18	6,000	6,600

7 rows selected.

SCOTT>

2. SQL 단일행 함수

7. 일반 함수

2. SQL 단일행 함수

1) NVL 함수 : NULL 값을 만나면 다른 값으로 치환해서 출력하는 함수

문 법: NVL(컬럼 , 치환할 값)

* 치환값이 숫자일 경우

NVL(sal , 0) -> sal 컬럼의 값이 null 일 경우 null 대신 0 으로 치환하라

NVL(sal , 100) -> sal 컬럼의 값이 null 일 경우 null 대신 100 으로 치환하라

* 치환값이 문자일 경우

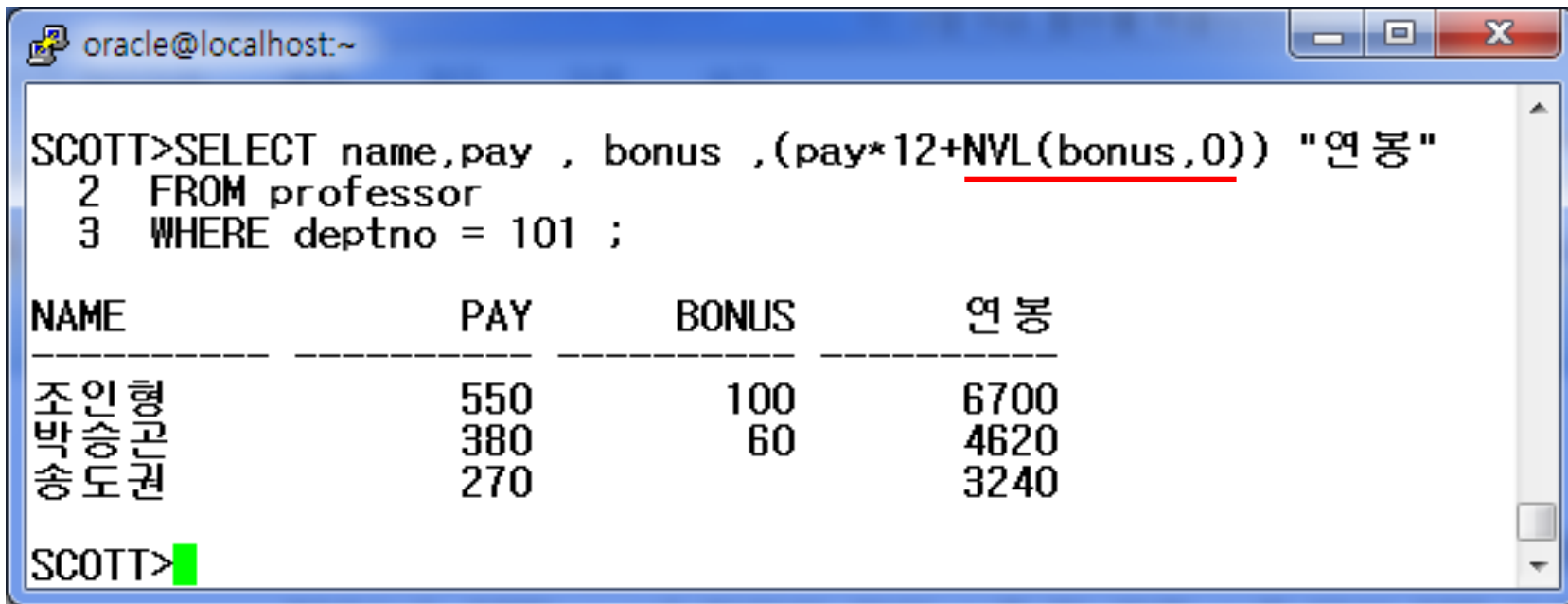
NVL(job , '무직') -> job 값이 null 일 경우 '무직' 으로 치환하라

*치환값이 날짜일 경우

NVL(hiredate,'sysdate') -> hiredate 값이 없을 경우 sysdate 날짜로 치환하라

2. SQL 단일행 함수

- Professor 테이블에서 101번 학과 교수들의 이름과 급여, bonus , 연봉을 출력하세요. 단 연봉은 $(pay*12+bonus)$ 로 계산하고 bonus 가 없는 교수는 0으로 계산하세요.



```
oracle@localhost:~  
SCOTT>SELECT name,pay , bonus ,(pay*12+NVL(bonus,0)) "연 봉"  
2 FROM professor  
3 WHERE deptno = 101 ;  
  
NAME                PAY      BONUS      연 봉  
-----  
조인형              550        100      6700  
박승곤              380         60      4620  
송도권              270         00      3240  
  
SCOTT>
```

NAME	PAY	BONUS	연 봉
조인형	550	100	6700
박승곤	380	60	4620
송도권	270		3240

2. SQL 단일행 함수

2) NVL2 함수

문 법: NVL2(COL1 , COL2 , COL3)

- professor 테이블에서 교수의 이름과 pay , bonus , 연봉을 출력하세요
단 연봉은 (pay*12+bonus) 로 계산하고 만약 bonus 가 없으면 급여를 0
으로 처리하세요.

```
oracle@localhost:~
SCOTT>SELECT name,pay , bonus ,NVL2(bonus,pay*12+bonus,pay*0) "연 봉"
2   FROM professor
3  WHERE deptno = 101 ;
```

NAME	PAY	BONUS	연 봉
조인형	550	100	6700
박승곤	380	60	4620
송도권	270		0

```
SCOTT>
```


2. SQL 단일행 함수

3) DECODE 함수

유형 1.

A 가 B 일 경우 '1' 을 출력하는 경우

IF 문장

```
IF A = B THEN  
  RETURN '1';  
END IF;
```

DECODE
함수

```
DECODE (A, B, '1', null)  
(마지막 null은 생략 가능 합니다)
```

이 예는 A 가 B 라면 1
을 출력합니다. 가장 기
본적인 DECODE 함수
문형입니다. 그렇다면 A
가 B가 아니라면 무엇
을 출력할까요? 답은
NULL 을 출력합니다.

2. SQL 단일행 함수

유형 1 예제: professor 테이블에서 교수명과 학과번호, 학과명을 출력하되 deptno 가 101 번인 교수만 컴퓨터 공학과로 출력하고 101번이 아닌 교수들은 학과명에 아무것도 출력하지 마세요

```
oracle@localhost:~
SCOTT>SELECT name, deptno , DECODE(deptno,101,'컴퓨터공학과') "학과명"
2 FROM professor ;
```

NAME	DEPTNO	학과명
조인형	101	컴퓨터공학과
박승권	101	컴퓨터공학과
송도원	101	컴퓨터공학과
양선희	102	
김영조	102	
주승재	102	
김도형	103	
나한열	103	
김현성	103	
심승기	201	
최원철	201	
박범철	202	
차범비	202	
바민호	203	
전민호	301	
허민호	301	

16 rows selected.

SCOTT>

deptno 가 101 번 이면 주어진 출력
내용인 '컴퓨터공학과' 를 출력하고
그 외에는 전부 NULL 값으로 출력이
되었습니다.

2. SQL 단일행 함수

유형 2.

A 가 B 일 경우 '1' 을 출력하고 아닐 경우 '2' 를 출력하는 경우

IF 문장

```
IF A = B THEN  
  RETURN '1' ;  
ELSE  
  RETURN '2';  
END IF ;
```

DECODE
함수

DECODE (A, B, '1' , '2')

2. SQL 단일행 함수

유형 2 예제: professor 테이블에서 교수명과 학과번호, 학과명을 출력하되 deptno 가 101 번인 교수만 컴퓨터 공학과로 출력하고 101번이 아닌 교수들은 학과명에 "기타학과" 로 출력하세요

```
oracle@localhost:~
SCOTT>SELECT name, deptno , DECODE(deptno,101,'컴퓨터공학과','기타학과') "학과"
2 FROM professor ;
```

NAME	DEPTNO	학과
조인형	101	컴퓨터공학과
박승권	101	컴퓨터공학과
송도권	101	컴퓨터공학과
양선희	102	기타학과
김영조	102	기타학과
주승재	102	기타학과
김도형	103	기타학과
나한열	103	기타학과
김현성	103	기타학과
심슬기	201	기타학과
최원범	201	기타학과
박범철	202	기타학과
차범철	202	기타학과
바비	203	기타학과
전민호	301	기타학과
허은	301	기타학과

```
16 rows selected.
SCOTT>
```

2. SQL 단일행 함수

유형 3.

A 가 B 일 경우 '1' 을 출력하고 A 가 C 일 경우 '2' 를 출력하고
둘 다 아닐 경우 '3' 을 출력하는 경우

IF 문장

```
IF A = B THEN
  RETURN '1' ;
ELSIF A = C THEN
  RETURN '2' ;
ELSE
  RETURN '3' ;
END IF ;
```

DECODE
함수

```
DECODE ( A , B , '1' , C , '2' , '3' )
```

2. SQL 단일행 함수

유형 3 예제: Professor 테이블에서 교수의 이름과 학과명을 출력하되 학과 번호가 101 번 이면 '컴퓨터 공학과', 102 번이면 '멀티미디어 공학과', 103 번이면 '소프트웨어공학과' 나머지는 '기타학과' 로 출력하세요.

```

oracle@localhost:~
SCOTT>SELECT name, deptno , DECODE(deptno,101,'컴퓨터 공학과',
2                                     102,'멀티미디어 공학과',
3                                     103,'소프트웨어 공학과',
4                                     '기타학과') "학과명"
5 FROM professor ;

NAME                DEPTNO 학과명
-----
조인형              101 컴퓨터 공학과
박승도              101 컴퓨터 공학과
송도권              101 컴퓨터 공학과
양희조              102 멀티미디어 공학과
김영선              102 멀티미디어 공학과
김주승              102 멀티미디어 공학과
김도형              103 소프트웨어 공학과
나한영              103 소프트웨어 공학과
김현수              103 소프트웨어 공학과
심스기              201 기타학과
심원철              201 기타학과
박원철              202 기타학과
차범철              202 기타학과
바비민              203 기타학과
전미은              301 기타학과
허은                301 기타학과

16 rows selected.

SCOTT>

```

2. SQL 단일행 함수

유형 4.

A 가 B 일 경우 중에서 A 가 C 를 만족하면 '1'을 출력하고 A 가 C가 아닐 경우 '2' 를 출력 하는 경우 (DECODE 함수 안에 DECODE 함수가 중첩되는 경우)

IF 문장

```
IF A = B THEN
  IF A = C THEN
    RETURN '1' ;
  ELSE
    RETURN '2' ;
  END IF ;
ELSE
  RETURN '2' ;
END IF ;
RETURN '3' ;
END IF ;
```

DECODE
함수

```
DECODE ( A , B , DECODE( A , C , '1' , '2' ) '2' )
```

2. SQL 단일행 함수

유형 4 예제 : professor 테이블에서 교수의 이름과 부서번호를 출력하고 101 번 부서 중에서 이름이 조인형 교수에게 석좌교수 후보라고 출력하세요. 나머지는 NULL 값을 출력합니다.

```
oracle@localhost:~
SCOTT>SELECT name, deptno,
2          DECODE(deptno,101,DECODE(name,'조인형','석좌교수후보')) "비 고"
3 FROM professor ;
```

NAME	DEPTNO	비 고
조인형	101	석좌교수후보
송고	101	
박승도	101	
송도권	101	
양선희	102	
김영조	102	
주승재	102	
김도형	103	
나한열	103	
김현정	103	
심스	201	
최슬기	201	
박원범	202	
차범철	202	
바비	203	
전민호	301	
허은	301	

```
16 rows selected.
SCOTT>
```


2. SQL 단일행 함수

유형 5.

A 가 B 일 경우 중에서 A 가 C 를 만족하면 '1'을 출력하고 A 가 C가 아닐 경우 '2' 를 출력 하고 A 가 B가 아닐 경우 '3' 을 출력하는 경우
(DECODE 함수 안에 DECODE 함수가 중첩되는 경우)

DECODE
함수

DECODE (A , B , DECODE(A , C , '1' , '2') '3')

2. SQL 단일행 함수

유형 5 예제: professor 테이블에서 교수의 이름과 부서번호를 출력하고 101 번 부서 중에서 이름이 조인형 교수에게 비고란에 "석좌교수 후보"라고 출력하세요. 101 번 학과의 조인형 교수 외에는 비고란에 "후보아님"을 출력하고 101번 교수가 아닐 경우는 비고란이 공란이 되도록 출력하세요.

```

oracle@localhost:~
SCOTT>SELECT name, deptno,
2          DECODE(deptno,101,DECODE(name, '조인형', '석좌교수 후보', '후보아님')) "비  고"
3  FROM professor;

NAME          DEPTNO 비  고
-----
조인형        101  석좌교수 후보
조박승        101  후보아님
송도권        101  후보아님
양선희        102
김영조        102
주승재        102
김도형        103
나한열        103
김현정        103
심슨         201
최슬기        201
박원철        202
차범철        202
바비         203
전민호        301
허은         301

16 rows selected.

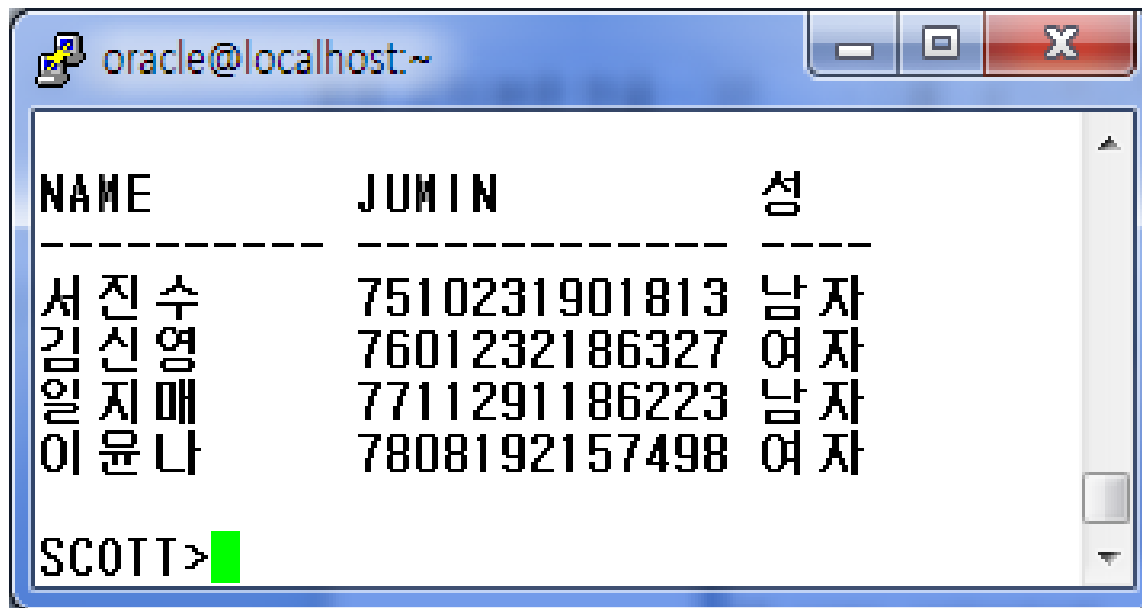
SCOTT>

```

2. SQL 단일행 함수

** DECODE 연습 문제 1

-Student 테이블을 사용하여 제 1 전공 (deptno1) 이 101 번인 학과 학생들의 이름과 주민번호, 성별을 출력하되 성별은 주민번호(jumin) 칼럼을 이용하여 7번째 숫자가 1일 경우 "남자" , 2일 경우 "여자" 로 출력하세요.



The screenshot shows a terminal window titled 'oracle@localhost:~'. It displays the output of a SQL query. The output is a table with three columns: NAME, JUMIN, and 성. The data is as follows:

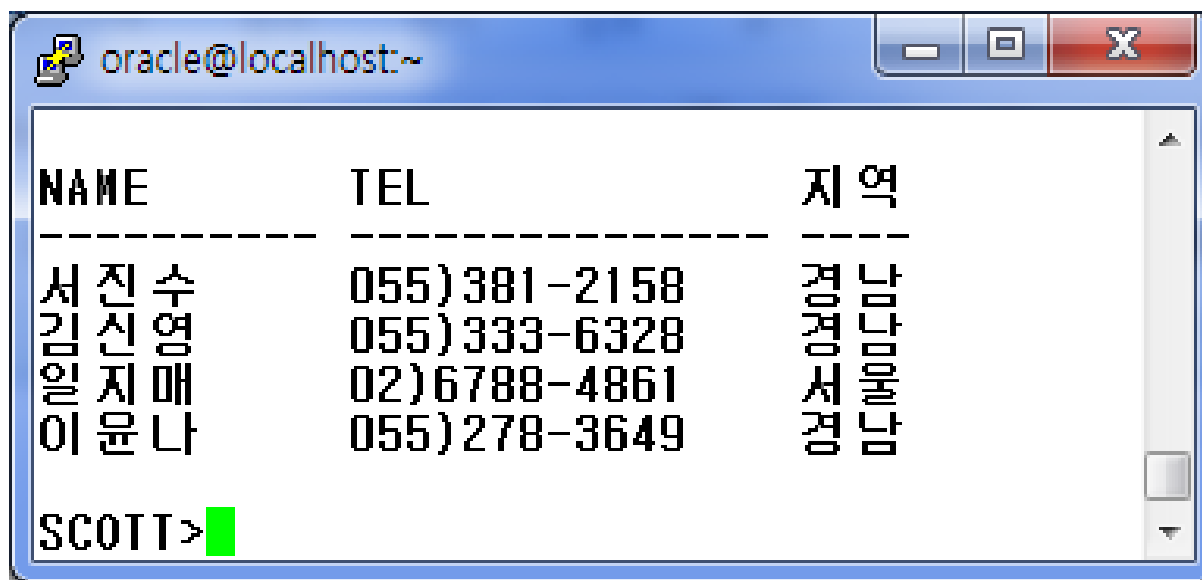
NAME	JUMIN	성
서진수	7510231901813	남자
김신영	7601232186327	여자
일지매	7711291186223	남자
이윤나	7808192157498	여자

Below the table, the prompt 'SCOTT>' is visible with a green cursor.

2. SQL 단일행 함수

** DECODE 연습 문제 2

Student 테이블에서 1 전공이 (deptno1) 101번인 학생의 이름과 전화번호와 지역명을 출력하세요. 지역번호가 02 는 서울 , 031 은 경기 , 051 은 부산 , 052 는 울산 , 055 는 경남으로 출력하세요.



The screenshot shows a terminal window titled 'oracle@localhost:~'. It displays the output of a SQL query. The output is a table with three columns: NAME, TEL, and 지역. The data is as follows:

NAME	TEL	지역
서진수	055)381-2158	경남
김신영	055)333-6328	경남
일지매	02)6788-4861	서울
이윤나	055)278-3649	경남

Below the table, the prompt 'SCOTT>' is visible with a green cursor.

2. SQL 단일행 함수

4) CASE 표현식

```
CASE 조건 WHEN 결과1 THEN 출력1  
           [WHEN 결과2 THEN 출력2]  
           ELSE 출력3  
END “컬럼명”
```

2. SQL 단일행 함수

```

oracle@localhost:~
SCOTT>SELECT name ,tel ,CASE (substr(tel,1,INSTR(tel,'') -1)) WHEN '02' THEN '서울'
2                                     WHEN '031' THEN '경기'
3                                     WHEN '051' THEN '부산'
4                                     WHEN '052' THEN '울산'
5                                     WHEN '055' THEN '경남'
6                                     ELSE '기타'
7                                     END "지역명"
8 FROM student ;

```

NAME	TEL	지역
서진수	055)381-2158	경남
서재수	051)426-1700	부산
이미경	053)266-8947	기타
김재수	02)6255-9875	서울
김동신	031)740-6388	경기
김신은	055)333-6328	경남
김은나	051)418-9627	부산
오유미	051)724-9618	부산
구세현	055)296-3784	경남
임지숙	02)312-9838	서울
임지숙	02)6788-4861	서울
김광훈	055)488-2998	경남
김광훈	053)736-4981	기타
김광훈	02)6175-3945	서울
김광훈	051)785-6984	부산
이영수	055)278-3649	경남
이영수	02)381-5440	서울
이영수	031)345-5677	경기
이영수	055)423-9870	경남
이영수	02)6122-2345	서울

20 rows selected.

SCOTT>

예제 1: DECODE 와 동일하게 '=' 조건으로 사용되는 경우

- Student 테이블을 참조하여 학생의 이름과 전화번호, 지역 명을 출력하세요. 단 지역번호가 02 면 서울, 031 이면 경기, 051 이면 부산, 052 이면 울산, 055 이면 경남, 나머지는 기타로 표시하세요.

2. SQL 단일행 함수

```

oracle@localhost:~
SCOTT>SELECT name, SUBSTR(jumin,3,2) "출생월",
2         CASE WHEN (SUBSTR(jumin,3,2)) BETWEEN '01' AND '03' THEN '1/4 분기'
3              WHEN (SUBSTR(jumin,3,2)) BETWEEN '04' AND '06' THEN '2/4 분기'
4              WHEN (SUBSTR(jumin,3,2)) BETWEEN '07' AND '09' THEN '3/4 분기'
5              WHEN (SUBSTR(jumin,3,2)) BETWEEN '10' AND '12' THEN '4/4 분기'
6              ELSE '안 태 어 남'
7              END "분기별"
8 FROM student ;

```

NAME	출생	분기별
서진수	10	4/4
서재수	02	1/4
이재수	06	2/4
김재수	12	4/4
김동호	03	1/4
김신영	01	1/4
김신라	04	2/4
오나미	09	3/4
유현미	01	1/4
유세현	10	4/4
김지매	11	4/4
김관호	04	2/4
김관호	09	3/4
김관호	02	1/4
김관호	12	4/4
김관호	08	3/4
김관호	01	1/4
김관호	08	3/4
김관호	03	1/4
김관호	02	1/4

20 rows selected.

SCOTT>

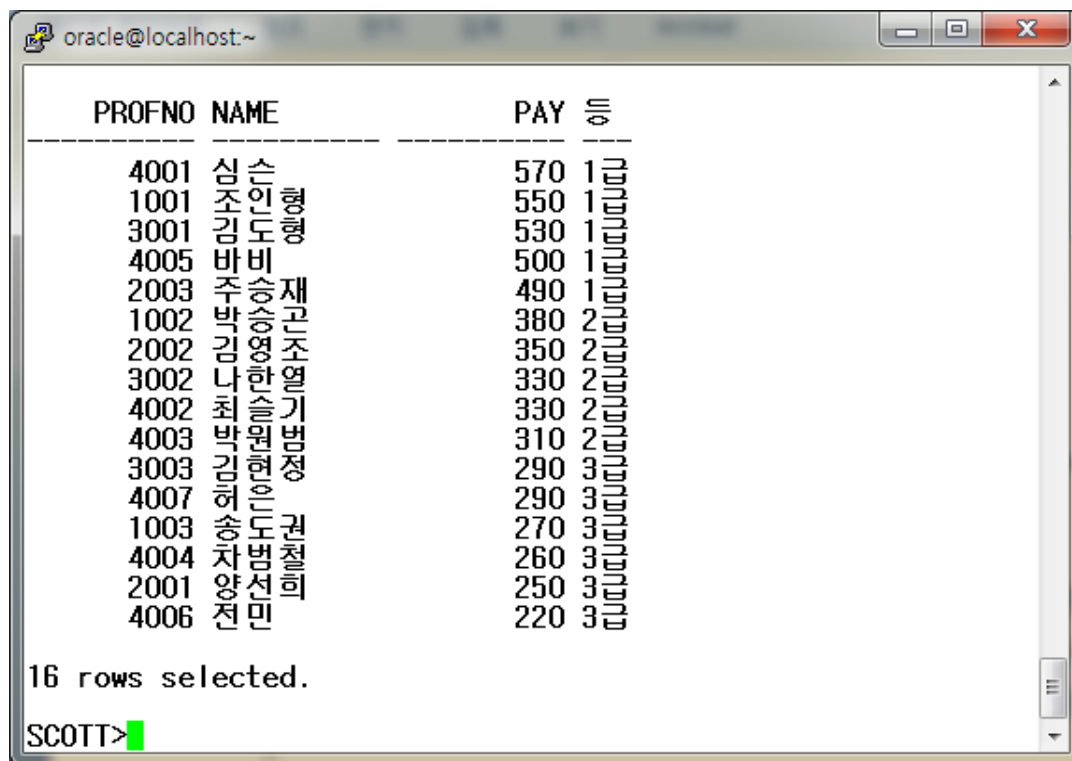
- 예제 2 : 비교 조건이 '=' 이 아닌 경우

Student 테이블의 jumin 컬럼을 참조하여 학생들의 이름과 태어난 달, 그리고 분기를 출력하세요. 태어난 달이 01-03월 은 1/4분기, 04 - 06월 은 2/4 분기, 07 - 09 월 은 3/4 분기, 10 - 12 월은 4/4 분기로 출력하세요

2. SQL 단일행 함수

** CASE 연습 문제

교수테이블 (professor) 를 조회하여 교수의 급여액수를 기준으로 200 미만은 4급, 201 – 300 까지는 3급, 301 – 400 까지는 2급, 401 이상은 1급으로 표시하여 교수의 번호, 교수이름, 급여, 등급을 출력하세요. (단 pay 컬럼을 내림차순으로 정렬하세요)



PROFNO	NAME	PAY	등급
4001	심승민	570	1급
1001	조인형	550	1급
3001	김도형	530	1급
4005	바비	500	1급
2003	주승재	490	1급
1002	박승곤	380	2급
2002	김영조	350	2급
3002	나한열	330	2급
4002	최슬기	330	2급
4003	박원범	310	2급
3003	김현정	290	3급
4007	허은정	290	3급
1003	송도권	270	3급
4004	차범철	260	3급
2001	양선희	250	3급
4006	전민	220	3급

16 rows selected.

SCOTT>