

2. 조회를 위한 고급 기법들

1. 집합 연산자
2. 단일행 함수
3. 그룹 함수



1. 집합 연산자

- 여러 건의 결과들을 모아서 연산
- 집합 연산자 종류

연산자 종류	내 용
UNION	두 집합을 더해서 결과를 출력. 중복 값 제거하고 정렬함
UNION ALL	두 집합을 더해서 결과를 출력. 중복 값 제거 안하고 정렬 안 함
INTERSECT	두 집합의 교집합 결과를 출력함. 정렬함
MINUS	두 집합의 차집합 결과를 출력함. 정렬함. 쿼리의 순서 중요함

1. 집합 연산자

□ UNION / UNION ALL (두 집합을 더함)

- Student 테이블과 professor 테이블을 참조하여 101 번 학과에 소속되어 있는 학생과 교수들의 학번(교수님은 교수번호) , 이름, 학과번호를 출력하세요.

```
select studno, name, deptno1
from student
WHERE deptno1=101
UNION
select profno, name, deptno
from professor
WHERE deptno=101;
```

```
select studno, name, deptno1
from student
WHERE deptno1=101
UNION ALL
select profno, name, deptno
from professor
WHERE deptno=101;
```

- Student 테이블에서 101번 학과와 201번 학과를 전공하는 학생들의 이름을 출력하세요.

1. 집합연산자

□ INTERSECT 연산자 사용하기 - 교집합 찾기

- stduent 테이블을 사용하여 101 번 학과와 201 번 학과를 복수전공하는 사람의 이름을 출력하세요.

□ MINUS 연산자 사용하기 - 큰집합에서 작은 집합 빼기

- professor 테이블에서 전체 직원의 급여를 20 % 인상하기 위한 직원 명단을 출력하려 합니다. 단 직급이 전임강사인 사람들은 명단에서 제외하세요.

1. 집합 연산자

- **** 집합 연산자 사용시 주의사항**
 - ▣ 칼럼의 개수가 다를 경우 에러 발생함
 - ▣ 비교되는 칼럼끼리의 데이터 타입이 다를 경우

```
select studno, name  
from student  
UNION  
select profno  
from professor;
```

```
select studno, name  
from student  
UNION  
select name, profno  
from professor;
```

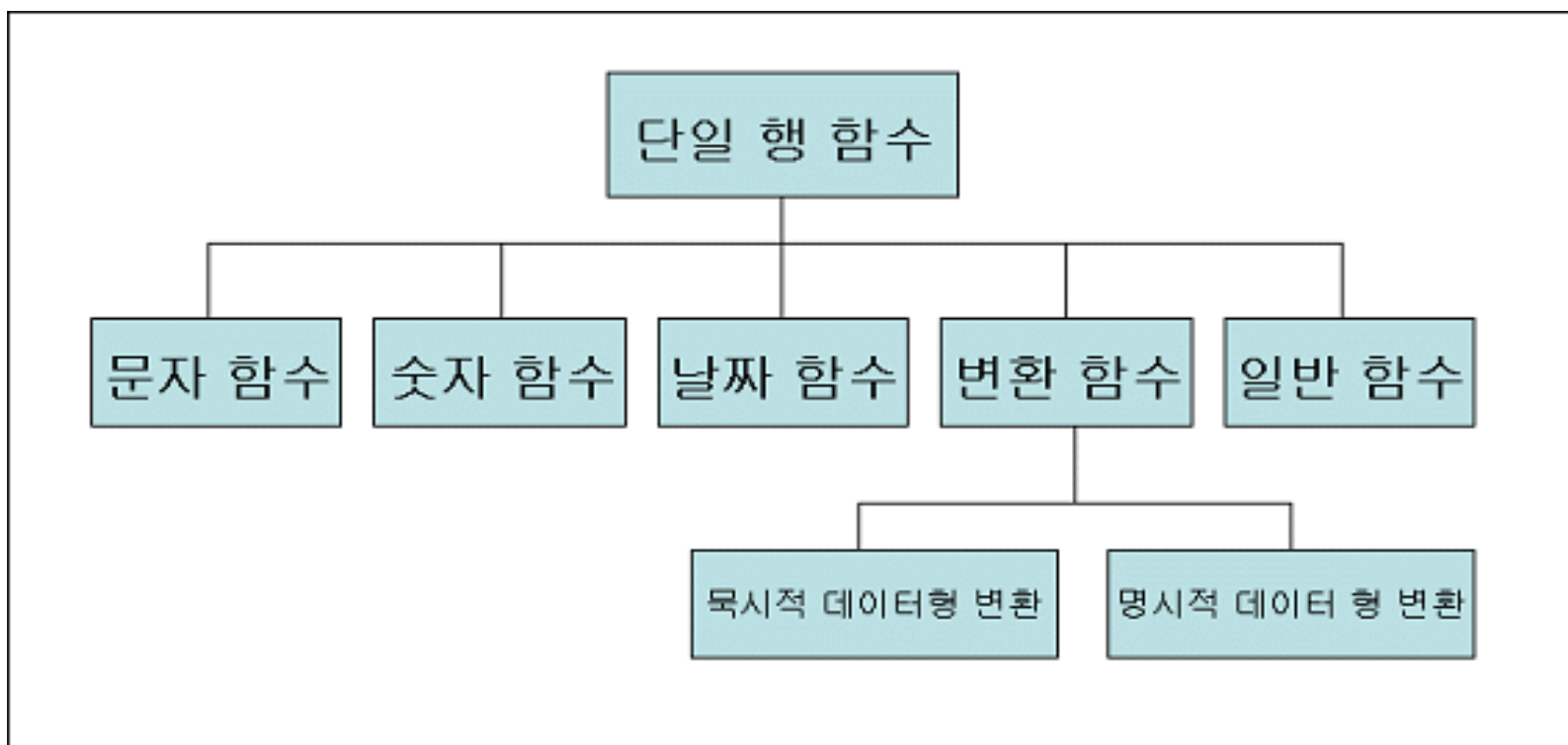
2. 함수

□ 함수의 종류

- ▣ 크게 단일 행 함수와 복수 행 함수로 구분.
 - 단일 행 함수 - 한번에 한 건씩 처리하는 함수
 - 복수 행 함수 - 여러 건의 데이터를 동시에 입력을 받아서 한꺼번에 처리한 후 결과값 1건을 만들어 주는 함수
- ▣ 복수 행 함수를 다른 말로 그룹함수나 집계함수 라고도 합니다.

2. 함수

□ 단일행 함수



2. 함수- 문자함수

□ 문자함수

▣ 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

```
SELECT INITCAP(ID)
FROM STUDENT
WHERE DEPTNO1=201;
```

```
SELECT INITCAP('Pertty girl')
FROM DUAL;
```


2. 함수-문자 함수

□ LOWER 함수

- 입력되는 값을 전부 소문자로 변경하여 출력

문 법: LOWER(문자열 또는 컬럼명)

□ UPPER 함수

- 입력되는 값을 전부 대문자로 변경하여 출력합니다

문 법: UPPER(문자열 또는 컬럼명)

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

```
SELECT name, id, lower(id) 소문자, upper(id) 대문자  
FROM STUDENT  
WHERE DEPTNO1=201;
```

2. 함수-문자함수

□ LENGTH / LENGTHB 함수:

- ▣ 입력된 문자열 길이(바이트 수)를 계산해주는 함수

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

```
SELECT name, id  
FROM student  
WHERE LENGTH(ID) >= 9;
```

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

2. 함수-문자 함수

□ CONCAT 함수 (|| 연산자와 동일한 기능)

문 법: CONCAT('문자열1' , '문자열2')

- Professor 테이블에서 교수들의 이름과 직급을 출력하세요.

```
SELECT CONCAT(name, position)
FROM professor
WHERE deptno=101;
```

2. 함수

□ SUBSTR/SUBSTRB 함수

- 문자열에서 특정길이의 문자를 추출할 때 사용하는 함수입니다

문 법: SUBSTR('문자열' 또는 컬럼명 , 1 , 4)

```
SELECT SUBSTR('ABCDE', 2,3)  
FROM dual;
```

```
SELECT SUBSTR('ABCDE', -2,3)  
FROM dual;
```

- ① Student 테이블에서 jumin 컬럼을 사용해서 1 전공이 101번인 학생들의 이름과 생년월일을 출력하세요.
- ② Student 테이블에서 jumin 컬럼을 사용해서 태어난 달이 8월인 사람의 이름과 생년월일을 출력하세요.

2. 함수- 문자 함수

□ INSTR 함수

- 주어진 문자열이나 컬럼에서 특정 글자의 위치를 찾아주는 함수

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

```
SELECT INSTR('A*B*C*', '*', 1,2) FROM dual;
```

시작위치를 - (마이너스) 로 줄 경우

```
SELECT INSTR('A*B*C*', '*', -4,1) FROM dual;  
SELECT INSTR('A*B*C*', '*', -4,2) FROM dual;  
SELECT INSTR('A*B*C*', '*', -2,2) FROM dual;
```

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

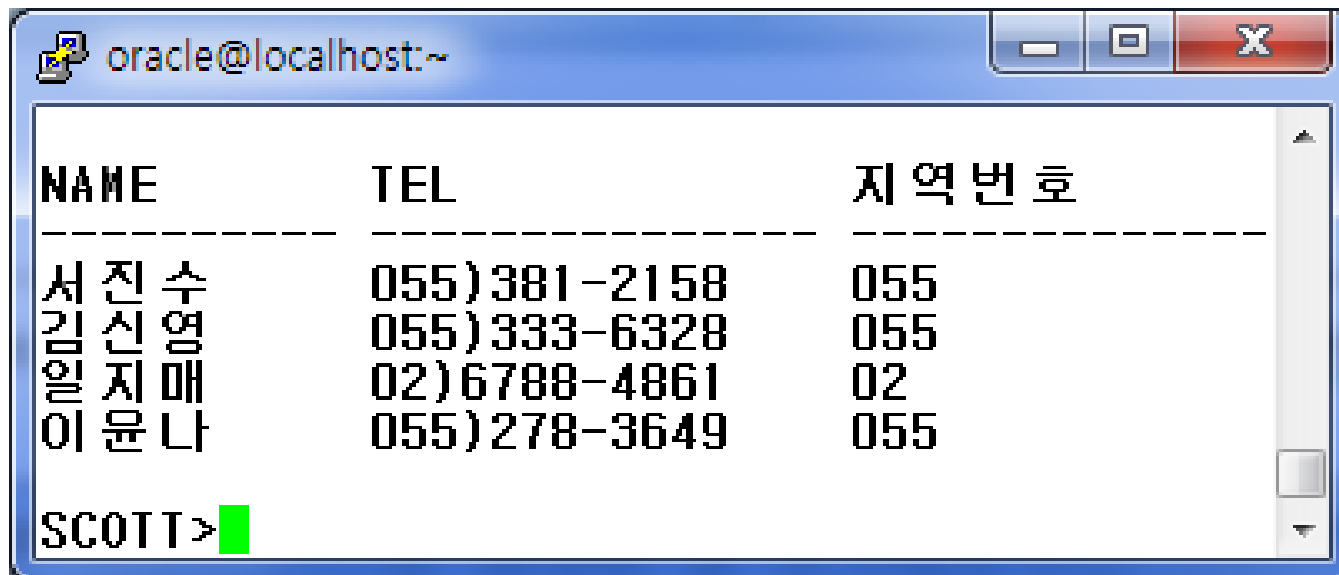
2. 함수- 문자함수

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

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)



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

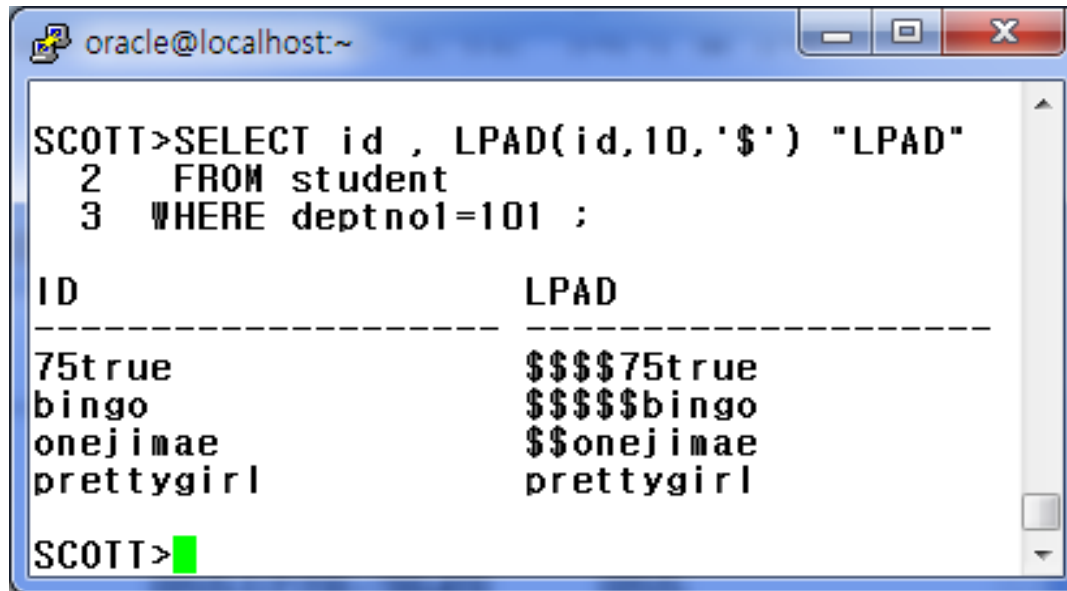
SCOTT>

2. 함수-문자 함수

□ 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            $$$$75true  
bingo             $$$$bingo  
onejima           $$onejima  
prettygirl        prettygirl  
  
SCOTT>
```

2. 함수-문자함수

□ LPAD

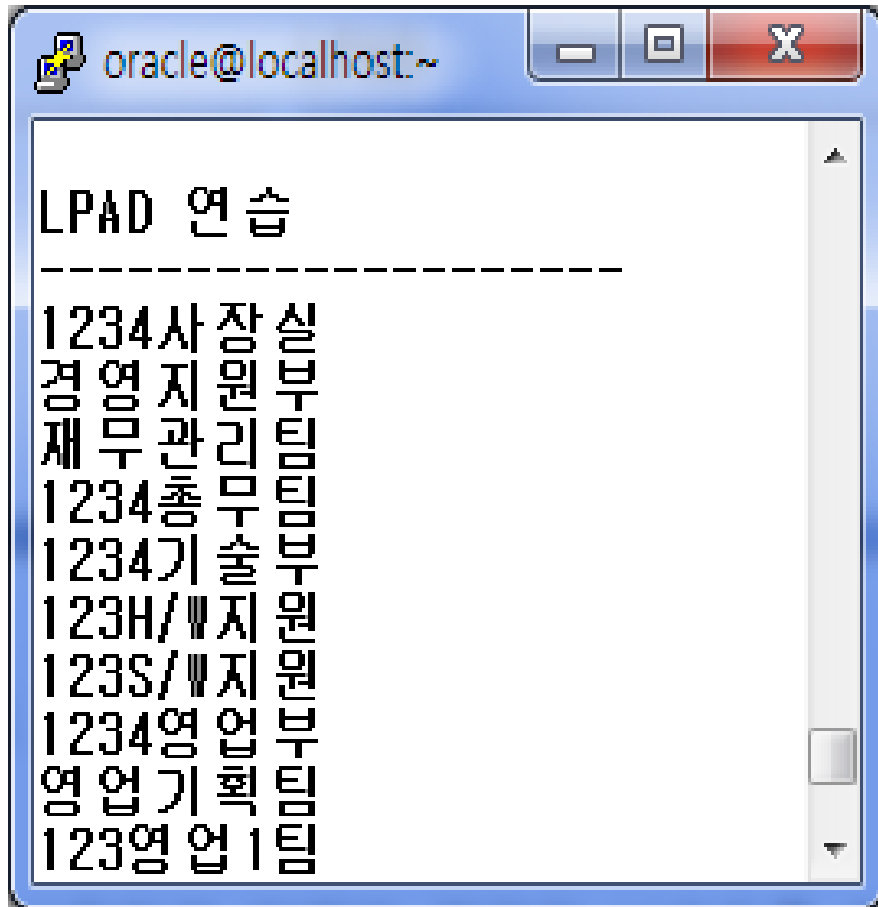
문자함수 퀴즈 2)

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)



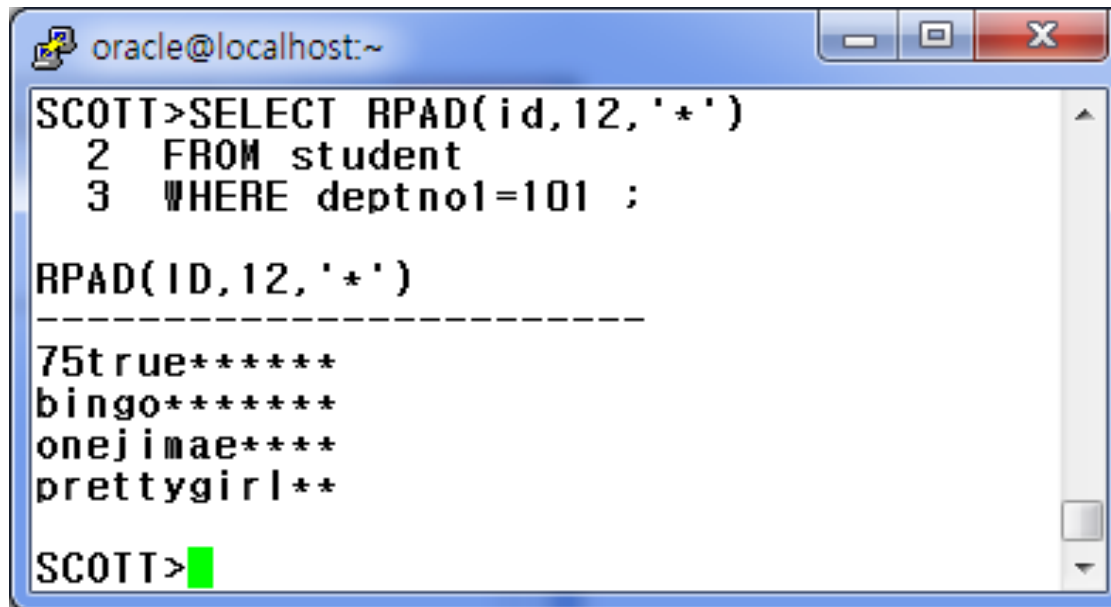
```
oracle@localhost:~  
LPAD 연습  
-----  
1234사장실  
1234경영지원부  
1234재무관리팀  
1234총무팀  
1234기술부  
123H/지원  
123S/지원  
1234영업부  
1234영업기획팀  
1234영업1팀
```


2. 함수- 문자함수

□ 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*****  
onejmae*****  
prettygirl**  
  
SCOTT>
```

2. 함수- 문자함수

□ LTRIM/ RTRIM 함수

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

문 법: LTRIM('문자열' 또는 컬럼명 , '제거할 문자')

문 법: RTRIM('문자열' 또는 컬럼명, '제거할 문자')

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

2. 함수-문자 함수

□ REPLACE 함수

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

- Student 테이블에서 deptno1이 101인 학생들의 이름을 출력하되 성 부분은 '#' 으로 표시되게 출력하세요.

```
SELECT REPLACE(name, substr(name,1,1),'#')  
FROM student  
WHERE deptno1=101;
```

2. 함수-문자 함수

□ 실습 예제

1. Student 테이블에서 아래와 같이 1 전공(deptno1)이 101 번인 학생들의 이름과 주민등록번호를 출력하되 주민등록번호의 뒤 7자리는 '*' 로 표시되게 출력하세요
2. Student 테이블에서 아래 그림과 같이 1 전공이 102 번인 학생들의 이름과 전화번호, 전화번호에서 국번 부분만 '#' 처리하여 출력하세요. 단 모든 국번은 3자리로 간주합니다.

2. 함수- 숫자 함수

□ 숫자함수의 종류

이름	의미	사용예
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. 함수-숫자 함수

□ ROUND

- 문 법: ROUND(숫자 , 원하는 자리수)

```
SELECT ROUND(12.345) ROUND1,  
       ROUND(12.345,2) ROUND2,  
       ROUND(12.345, -1) ROUND3  
FROM dual;
```

□ TRUNC

- 문 법: TRUNC(숫자 , 원하는 자리수)

```
SELECT TRUNC(12.345) TRUNC1,  
       TRUNC(12.345,2) TRUNC2,  
       TRUNC(12.345, -1) TRUNC3  
FROM dual;
```

1. 함수- 숫자 함수

□ MOD, CEIL, FLOOR 함수

```
SELECT MOD(12,10)"MOD",  
       CEIL(12.345) "CEIL",  
       FLOOR(12.345) "FLOOR"  
FROM dual;
```

□ POWER 함수

- 문 법: POWER(숫자1 , 숫자2)

```
SELECT POWER(3,2)  
FROM dual;
```

2. 함수- 날짜 함수

□ 날짜의 연산

- 10월 23일 + 3 은 10월 26일
- 10월 23일 - 3 은 10월 20일.
- 10월 23일 - 10월 20일 은 3일 차이 .

□ 날짜 함수의 종류

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. 함수- 날짜 함수

□ SYSDATE 함수 (현재 날짜/시간 출력)

```
SELECT SYSDATE FROM dual;
```

날짜 및 시간 형식 변경

```
ALTER SESSION SET NLS_DATE_FORMAT='yyyy-mm-dd hh24:mi:ss';
```

```
SELECT SYSDATE FROM dual;
```

2. 함수-날짜 함수

□ MONTHS_BETWEEN 함수

- ▣ 두 날짜 사이의 개월 수 계산

```
SELECT MONTHS_BETWEEN('2012-03-01', '2012-01-01')  
FROM dual;
```

- ▣ MONTHS_BETWEEN 함수 주요 특징

- 큰 날짜를 먼저 써야 합니다

```
SELECT MONTHS_BETWEEN('2014-01-01', '2014-03-01')  
FROM dual;
```

3. 함수-날짜 함수

□ MONTHS_BETWEEN 함수

▣ MONTHS_BETWEEN 함수 주요 특징

- 두 날짜가 같은 달에 속해 있으면 특정 규칙으로 계산된 값이 나옵니다
 - 1개월이 29일 인 2012년 2월을 조회
 - 1개월이 30일 인 2012년 4월을 조회
 - 1개월이 31일 인 2012년 1월을 조회

```
SELECT MONTHS_BETWEEN('2012-02-29', '2012-02-01')  
FROM dual;
```

```
SELECT MONTHS_BETWEEN('2012-04-30', '2012-04-01')  
FROM dual;
```

```
SELECT MONTHS_BETWEEN('2012-01-31', '2012-01-01')  
FROM dual;
```

3. 함수- 날짜 함수

□ MONTHS_BETWEEN 함수

- ▣ 두 날짜가 모두 해당 월의 마지막 날이거나 처음 날이면 개월 수가 정수 값으로 나옵니다.

```
SELECT MONTHS_BETWEEN('2012-03-01', '2012-02-01')  
FROM dual;
```

```
SELECT MONTHS_BETWEEN('2012-05-31', '2012-04-30')  
FROM dual;
```

- ▣ 실습 예 : 교수테이블에서 deptno=101인 경우 hiredate와 SYSDATE를 이용하여 근무 개월 수를 구하라. (소수점 2자리에서 반올림)

2. 함수-날짜 함수

□ ADD_MONTHS 함수

```
SELECT SYSDATE, ADD_MONTHS(SYSDATE, 3)  
FROM dual;
```

□ NEXT_DAY 함수

```
SELECT SYSDATE, NEXT_DAY(SYSDATE, '월')  
FROM dual;
```

□ LAST_DAY 함수

```
SELECT SYSDATE, LAST_DAY(SYSDATE)  
FROM dual;
```

2. 함수-날짜 함수

□ 날짜의 ROUND , TRUNC 함수

```
SELECT SYSDATE, ROUND(SYSDATE),  
TRUNC(SYSDATE)  
FROM dual;
```

2. 함수- 형변환 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

□ 묵시적 (자동) 형 변환

```
SELECT 1 + '1' FROM dual;
```

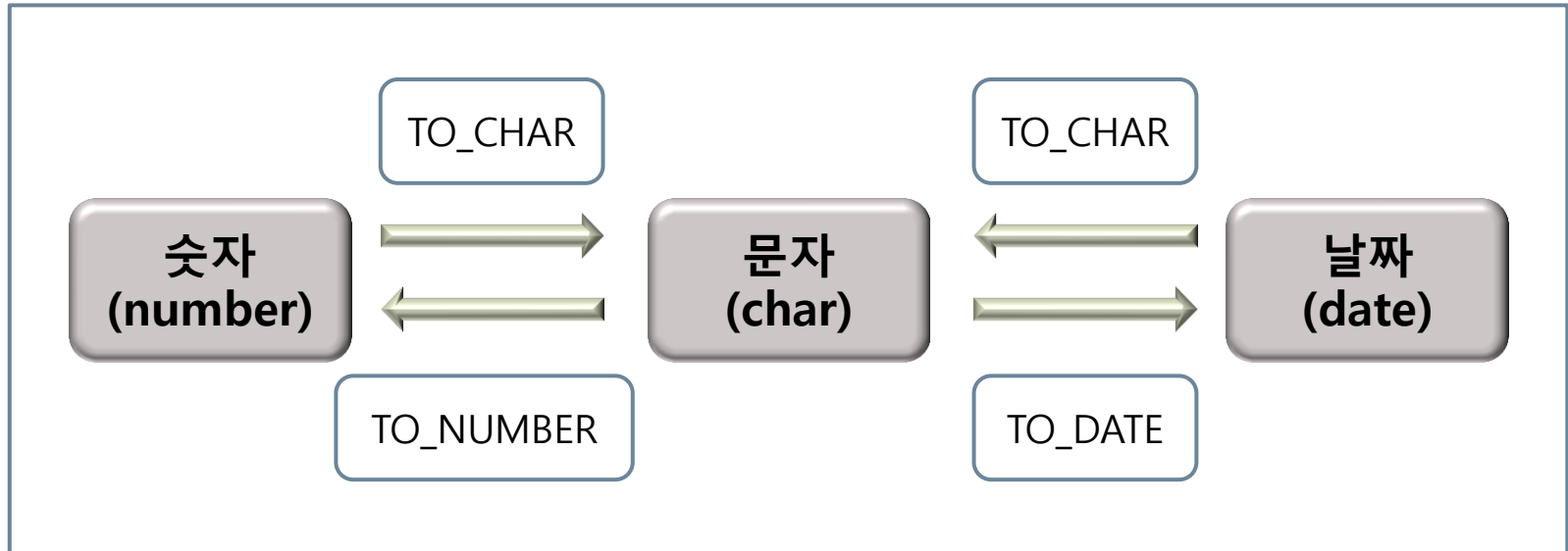
```
SELECT 1 + to_number('1')  
FROM dual ;
```

```
SELECT 1 + 'A' FROM dual;
```

오류 발생

2. 함수 -형 변환 함수

□ 명시적(수동) 형 변환



2. 함수 - 형변환 함수

□ 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.

```
SELECT TO_CHAR(SYSDATE, 'YYYY') "YYYY",  
       TO_CHAR(SYSDATE, 'RRRR') "RRRR",  
       TO_CHAR(SYSDATE, 'YY') "YY",  
       TO_CHAR(SYSDATE, 'YEAR') "YEAR"  
FROM dual;
```

2. 함수 - 형변환 함수

□ TO_CHAR 함수 (날짜를 문자로 형 변환하기)-2

월 : MM – 월을 숫자 2자리로 표현합니다. 예: 10

MON – 월을 뜻하는 영어 3글자로 표시합니다. 예: OCT

MONTH – 월을 뜻하는 영어 이름 전체를 표시합니다

```
SELECT TO_CHAR(SYSDATE, 'MM') "MM",  
       TO_CHAR(SYSDATE, 'MON') "MON",  
       TO_CHAR(SYSDATE, 'MONTH') "MONTH"  
FROM dual;
```

일 : DD – 일을 숫자 2자리로 표시합니다. 예: 12

DAY – 요일에 해당하는 영어 명칭을 표시합니다.

DDTH – 몇 번째 날인지를 표시합니다.

```
SELECT TO_CHAR(SYSDATE, 'DD') "DD",  
       TO_CHAR(SYSDATE, 'DAY') "DAY",  
       TO_CHAR(SYSDATE, 'DDTH') "DDTH"  
FROM dual;
```

2. 함수 - 형 변환 함수

□ TO_CHAR 함수 (날짜를 문자로 형 변환하기)-3

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

분 : MI 로 표시합니다.
초 : SS 로 표시합니다.

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

2. 함수-형 변환 함수

□ 실습문제

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

NAME	BIRTHDAY
박동호	1975-03-03 00:00:00
김주현	1978-03-24 00:00:00

2. 함수-형 변환 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. 함수-형 변환 함수

□ 실습

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

2. 함수-형 변환 함수

□ TO_NUMBER 함수

문 법: TO_NUMBER('1')

```
select to_number('1')  
from dual;
```

□ TO_DATE 함수

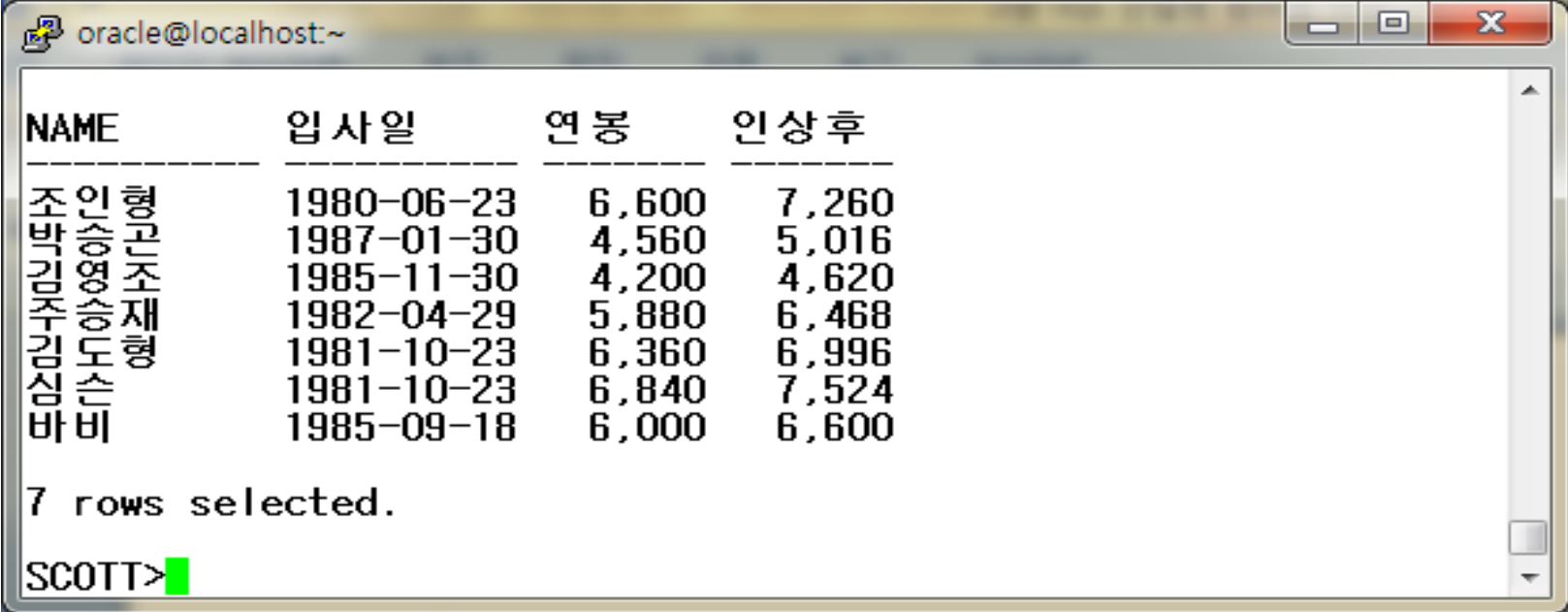
문 법: TO_DATE('문자','날짜 포맷')

```
select to_date('2014-01-01', 'yyyy-mm-dd')  
from dual
```


2. 함수-형 변환 함수

□ 실습

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



The screenshot shows a terminal window titled 'oracle@localhost:~'. It displays the output of an SQL query. The output is a table with four columns: NAME, 입사일, 연봉, and 인상후. There are 7 rows of data. Below the table, it says '7 rows selected.' and the prompt 'SCOTT>' is visible.

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. 함수-일반 함수

□) 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. 함수-일반 함수

□ 실습

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

```
select name, pay, bonus,  
to_char((pay*12)+nvl(bonus,0), '99,999') "연봉"  
from professor  
where deptno=101;
```

2. 함수-일반 함수

□ 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.

```
select name, pay, bonus,  
nvl2(bonus,pay*12+bonus, pay*0) "연봉"  
from professor  
where deptno=101;
```

2. 함수-일반 함수

□ 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. 함수-일반 함수

□ 유형1 예제

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

```
select name, deptno,  
       decode(deptno, 101, '컴퓨터공학과') "학과명"  
from professor;
```

2. 함수-일반 함수

□ DECODE 함수-2

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. 함수-일반 함수

□ 유형2 실습 예제

professor 테이블에서 교수명과 학과번호 , 학과명을 출력하되 deptno 가 101 번인 교수만 컴퓨터 공학과로 출력하고 101번이 아닌 교수들은 학과명에 “기타학과” 로 출력하세요

```
select name, deptno,  
       decode(deptno, 101, '컴퓨터공학과','기타학과') "학과명"  
from professor;
```


2. 함수-일반 함수

□ DECODE 함수-3

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. 함수-일반 함수

□ 유형3 예제

Professor 테이블에서 교수의 이름과 학과명을 출력하되 학과 번호가 101 번이면 '컴퓨터 공학과', 102 번이면 '멀티미디어 공학과', 103 번이면 '소프트웨어공학과' 나머지는 '기타학과' 로 출력하세요.

```
select name, deptno,  
       decode(deptno, 101, '컴퓨터공학과',  
              102, '멀티미디어 공학과',  
              103, '소프트웨어 공학과',  
              '기타학과') "학과명"  
from professor;
```

2. 함수-일반 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. 함수-일반 함수

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

```
select name, deptno,  
       decode(deptno, 101, decode(name, '조인형','석좌교수')) "비고"  
from professor;
```

2. 함수-일반 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')

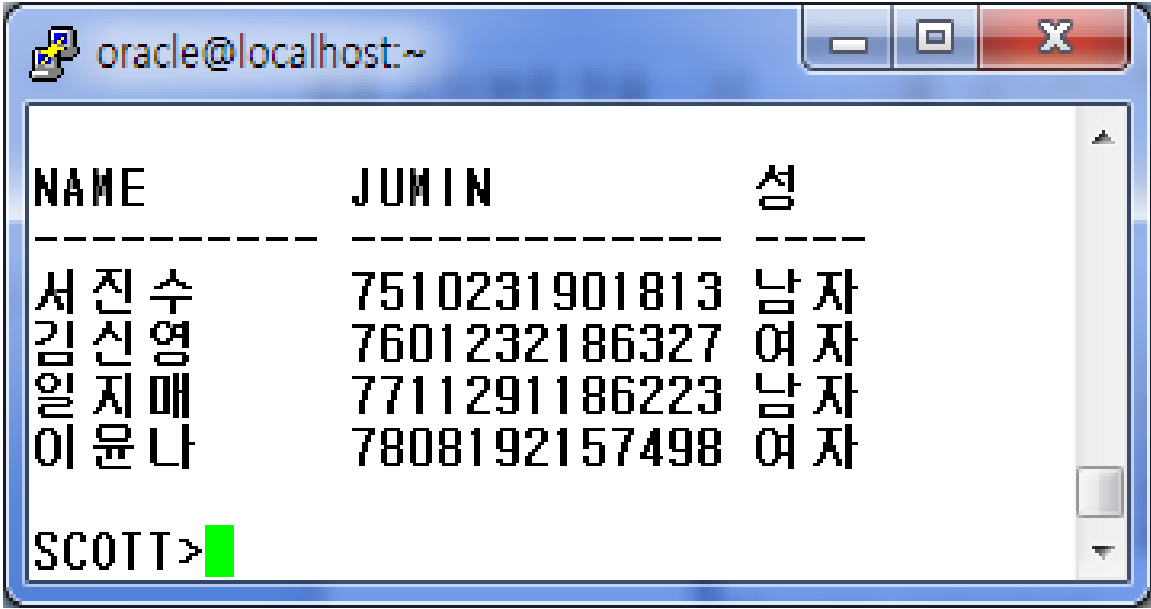
예제: professor 테이블에서 교수의 이름과 부서번호를 출력하고 101 번 부서 중에서 이름이 조인형 교수에게 비고란에 "석좌교수 후보"라고 출력하세요. 101번 학과의 조인형 교수 외에는 비고란에 "후보아님"을 출력하고 101 번 교수가 아닐 경우는 비고란이 공란이 되도록 출력하세요.

```
select name, deptno,  
       decode(deptno, 101, decode(name, '조인형','석좌교수', '후보아님')) "비고"  
from professor;
```

2. 함수-일반 함수

□ decode 연습문제

-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 image 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:

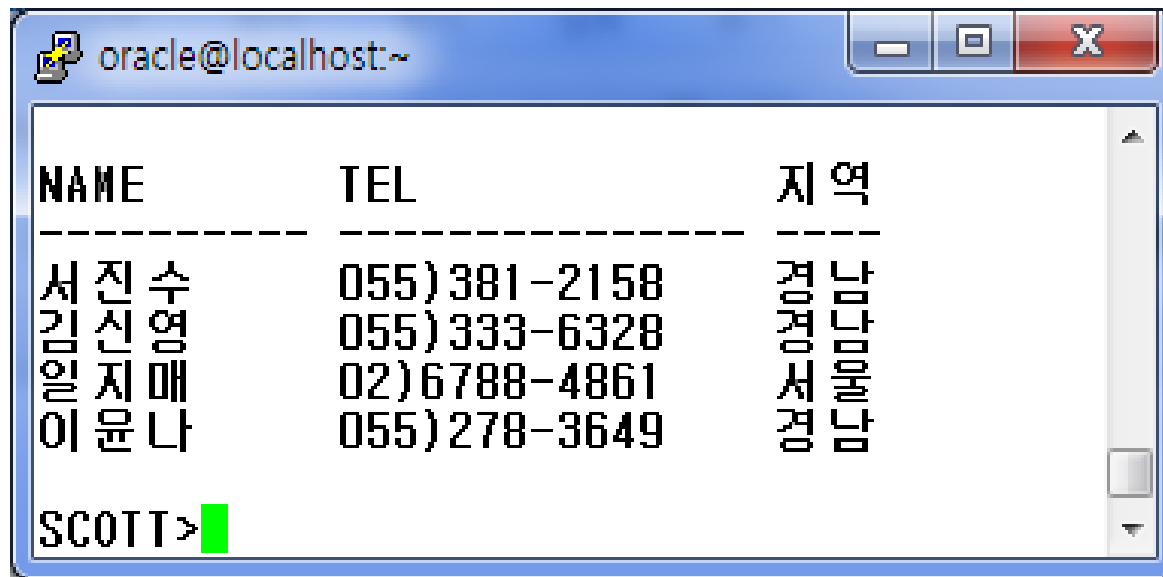
NAME	JUMIN	성
서진수	7510231901813	남자
김신영	7601232186327	여자
일지매	7711291186223	남자
이운나	7808192157498	여자

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

2. 함수-일반 함수

□ 연습문제-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. 함수-일반 함수

□ CASE 표현식

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


2. 함수-일반 함수

□ case 예제

예제 1: DECODE 와 동일하게 '=' 조건으로 사용되는 경우

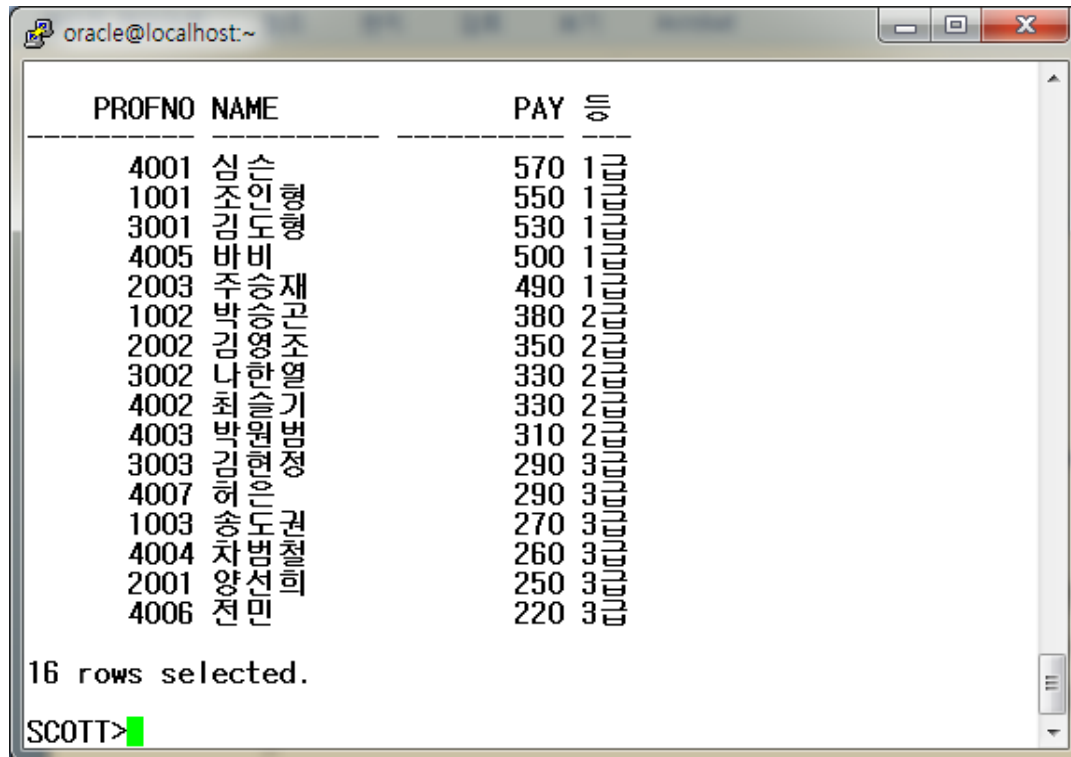
- Student 테이블을 참조하여 학생의 이름과 전화번호, 지역 명을 출력하세요. 단 지역번호가 02 면 서울, 031 이면 경기, 051 이면 부산, 052 이면 울산, 055 이면 경남, 나머지는 기타로 표시하세요.

```
select name, tel,  
       case(substr(tel,1, instr(tel, '-')-1))  
         when '02' then '서울'  
         when '031' then '경기'  
         when '051' then '부산'  
         when '052' then '울산'  
         when '055' then '경남'  
         else '기타'  
       end "지역명"  
from student;
```

2. 함수-일반 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	심수민	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	나열	330	2급
4002	최슬기	330	2급
4003	박원범	310	2급
3003	김현정	290	3급
4007	허영진	290	3급
1003	송도권	270	3급
4004	차범철	260	3급
2001	양선희	250	3급
4006	전민	220	3급

16 rows selected.

SCOTT>

2. 함수-일반 함수

- 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

```
select name, substr(jumin,3,2) "출생일",  
       case  
         when(substr(jumin,3,2)) between '01' and '03' then '1/4 분기'  
         when(substr(jumin,3,2)) between '04' and '06' then '2/4 분기'  
         when(substr(jumin,3,2)) between '07' and '09' then '3/4 분기'  
         when(substr(jumin,3,2)) between '10' and '12' then '4/4 분기'  
         else '안태어 남'  
       end "분기별"  
from student;
```