



Oracle 2강 – SQL 단일 행 함수

양 명 속

[now4ever7@gmail.com]



목차

- 문자 함수
- 숫자 함수
- 날짜 함수
- 형 변환 함수
- 일반 함수



SQL 단일행 함수

■ 함수

- 어떤 입력 값을 받아서 정해진 루틴에 의해 작업을 하고 결과값을 만들어서 출력해주는 것

■ SQL에서 사용되는 함수

- 단일 행 함수 - 데이터가 여러 건 존재하지만 단일 행 함수에 들어가는 데이터는 한번에 한 개
 - 여러 건의 데이터를 한꺼번에 처리하는 것이 아니라 한번에 하나씩 처리하는 함수
- 복수 행 함수 - 여러 건의 데이터를 동시에 입력 받아서 결과값 1건을 만들어 주는 함수
 - 그룹 함수라고도 함



SQL 단일행 함수

- 단일 행 함수 - 입력되는 데이터의 종류에 따라
 - 문자함수 - 함수에 입력되는 값이 문자이면 문자함수
 - 숫자함수
 - 날짜함수
 - 변환함수
 - 묵시적 데이터형 변환
 - 명시적 데이터형 변환
 - 일반함수

문자함수

함수명	의미	사용 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 특정 문자의 위치 추출	instr('A*B#','#') =>4
INSTRB	주어진 문자에서 특정 문자의 위치 바이트 값 추출	instrb('한글로','로')=>5

문자함수

함수명	의미	사용 예
LPAD	주어진 문자열에서 왼쪽으로 특정 문자를 채움	<code>lpad('love',6,'*')=>**love</code>
RPAD	주어진 문자열에서 오른쪽으로 특정 문자를 채움	<code>rpad('love',6,'*')=>love**</code>
LTRIM	주어진 문자열에서 왼쪽의 특정 문자를 삭제함	<code>ltrim('*love','*')=>love</code>
RTRIM	주어진 문자열에서 오른쪽의 특정 문자를 삭제함	<code>Rtrim('love*','*')=>love</code>
REPLACE	주어진 문자열에서 A를 B로 치환함	<code>replace('AB','A','E')=>EB</code>

INITCAP 함수

- 첫 글자만 대문자로 출력하고 나머지는 전부 소문자로 출력하는 함수

INITCAP(문자열 또는 칼럼명)

- student 테이블을 사용하여 1전공이 201번인 학생들의 ID를 첫 글자만 대문자로 출력

```
SQL> select initcap(id) from student  
2  where deptno1=201;
```

```
INITCAP(ID)  
-----  
Gunmandu  
Shyun1  
Nonnon1  
Munho  
Silverwt  
Youngmin
```

6 개의 행이 선택되었습니다.

```
SQL> select initcap('Pretty girl') from dual;
```

```
INITCAP('PR  
-----  
Pretty Girl
```

문자열 중간에 공백이 있는 경우에는 단어의 시작 부분이 대문자로 바뀜



LOWER 함수/ UPPER 함수

- LOWER 함수

- 입력되는 값을 전부 소문자로 변경하여 출력함

LOWER(문자열 또는 칼럼명)

- UPPER 함수

- 입력되는 값을 전부 대문자로 변경하여 출력

UPPER(문자열 또는 칼럼명)

LOWER 함수/ UPPER 함수

- STUDENT 테이블에서 1전공이 201번인 학생들의 ID를 이름과 함께 소문자, 대문자로 출력

```
SQL> select name 이름, id, lower(id) 소문자, upper(id) 대문자 from student  
2 where deptno1=201;
```

이름	ID	소문자	대문자
김재수	gunmandu	gunmandu	GUNMANDU
임세현	shyun1	shyun1	SHYUN1
아관훈	nonnon1	nonnon1	NONNON1
김민호	munho	munho	MUNHO
아이스	silverwt	silverwt	SILVERWT
이영민	youngmin	youngmin	YOUNGMIN

6 개의 행이 선택되었습니다.

- DB에 데이터를 저장하거나 불러올 때 대문자나 소문자로 통일해야 할 경우에 많이 사용됨
- 예) 회원 가입 받을 때 ID를 소문자로 받고 싶은 경우 사용자가 어떤 문자를 입력해도 저장할 때 lower 함수를 이용하여 소문자로 바꾼 후 저장하는 경우 유용

LENGTH/LENGTHB 함수

- 입력된 문자열의 길이(바이트 수)를 계산해주는 함수

LENGTH(칼럼 또는 문자열) / LENGTHB(칼럼 또는 문자열)

- STUDENT 테이블에서 ID가 9글자 이상인 학생들의 이름과 ID와 글자 수를 출력
- STUDENT 테이블에서 1전공이 201번인 학생들의 이름과 이름의 글자 수, 이름의 바이트 수를 출력

```
SQL> select name 이름, id, length(id) 글자수 from student  
2 where length(id) > 9;
```

이름	ID	글자수
이윤나	prettygirl	10

```
SQL> select name 이름, length(name), lengthb(name) from student  
2 where deptno=201;
```

이름	LENGTH(NAME)	LENGTHB(NAME)
김재수	3	6
김세현	3	6
김광호	3	6
김민수	3	6
김영민	3	6

6 개의 행이 선택되었습니다.



CONCAT 함수

- || 연산자와 동일한 기능

```
CONCAT('문자열1', '문자열2')
```

- PROFESSOR 테이블에서 교수들의 이름과 직급 출력

```
SQL> select concat(name, position) from professor  
2  where deptno=101;
```

```
CONCAT(NAME, POSITION  
-----
```

```
조인형정교수  
박승곤조교수  
송도권전임강사
```

SUBSTR 함수

- 문자열에서 특정 길이의 문자를 추출할 때 사용하는 함수

SUBSTR('문자열' 또는 칼럼명, 1, 4)

- 앞의 숫자 : 시작 위치
- 뒤의 숫자 : 추출할 글자수
- 시작위치의 숫자를 - (마이너스)로 하게 되면 뒤에서부터 자릿수를 계산해서 문자를 추출

```
SQL> select substr('ABCDE', 2, 3) from dual;
```

```
SUB  
---  
BCD
```

```
SQL> select substr('ABCDE', -2, 3) from dual;
```

```
SU  
--  
DE
```

SUBSTR 함수

- student 테이블에서 JUMIN 칼럼을 사용하여 1전공이 101번인 학생들의 이름과 생년월일을 출력

```
SQL> select name, substr(jumin, 1, 6) "생년월일" from student  
2 where deptno1=101;
```

NAME	생년월일
-----	-----
서진수	751023
김신영	760123
일지매	771129
이윤나	780819

- student 테이블에서 JUMIN 칼럼을 사용하여 태어난 달이 8월인 사람의 이름과 생년월일을 출력

```
SQL> select name, substr(jumin, 1, 6) "생년월일" from student  
2 where substr(jumin, 3, 2)='08';
```

NAME	생년월일
-----	-----
이윤나	780819
인영민	780809



SUBSTRB 함수

- SUBSTR 함수와 문법은 동일
- 차이점은 추출할 자릿수가 아니라 추출할 바이트 수를 지정하는 것

```
SQL> select substr(name, 1, 2), substrb(name, 1, 2) from student  
2  where deptno1=101;
```

SUBS	SU
------	----

----	---
------	-----

서진	서
김신	김
일지	일
이운	이

- substr 사용 => 2글자가 출력
- substrb 사용 => 2바이트(1글자)만 출력



INSTR 함수

- 주어진 문자열이나 칼럼에서 특정 글자의 위치를 찾아주는 함수

INSTR('문자열' 또는 칼럼, 찾는 글자, 시작 위치, 몇 번째인지(기본값은 1))

```
SQL> select instr('A×B×C×','×',1,2) from dual;
```

```
INSTR('A×B×C×','×',1,2)
```

```
-----  
4
```

- 두 번째 '*'가 나오는 위치의 값을 출력
- 시작 위치를 변경한 후 테스트

```
SQL> select instr('A×B×C×','×',3,2) from dual;
```

```
INSTR('A×B×C×','×',3,2)
```

```
-----  
6
```

INSTR 함수

■ 시작 위치를 마이너스(-)로 변경

```
SQL> select instr('A*B*C*','*",-4,1) from dual;
```

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

```
-----  
2
```

```
SQL> select instr('A*B*C*','*",-4,2) from dual;
```

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

```
-----  
0
```

```
SQL> select instr('A*B*C*','*",-2,2) from dual;
```

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

```
-----  
2
```

- 시작위치를 마이너스로 주면 오른쪽에서 왼쪽으로 결과값을 검색
- 해당 문자열을 찾을 수 없으면 0 반환

자리번호	1	2	3	4	5	6
문자열	A	*	B	*	C	*
1,2번 예제			시작위치	-3	-2	-1
3번 예제		2번째 *표		1번째 *표	시작위치	-16

INSTR 함수

- student 테이블의 TEL 칼럼을 사용하여 학생의 이름과 전화번호, ')'가 나오는 위치를 출력

```
SQL> select name, tel, instr(tel, ')', 1,1) from student  
2 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

- 실습) student 테이블을 참조해서 1전공이 101번인 학생의 이름과 전화번호와 지역번호를 출력. 단, 지역번호는 숫자만 나와야 함

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

LPAD 함수

LPAD('문자열' 또는 칼럼명, 자릿수, '채울문자')

- student 테이블에서 1전공이 101번인 학과 학생들의 ID를 총 10자리로 출력하되 왼쪽 빈 자리는 '\$'기호로 채우세요

```
SQL> select id, lpad(id, 10, '$') "LPAD" from student  
2 where deptno1=101;
```

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

LPAD 연습

1234사장실
경영지원부
재무관리팀
1234총무팀
1234기술부
123H/W지원팀
123S/W지원팀
1234영업부
영업기획
123영업1
123영업2
123영업3
123영업4

13 개의 행이 선택되었습니다.

- 실습) DEPT2 테이블을 사용하여 DNAME을 다음 결과가 나오도록 쿼리 작성하기
 - dname을 총 10바이트로 출력하되 원래 dname이 나오고 나머지 빈 자리는 해당 자리의 숫자가 나오면 됨. 즉, 사장실은 이름이 총 6바이트이므로 숫자가 1234까지 나옴

```
select lpad(dname, 10, '1234567890') as "LPAD 연습"  
from dept2;
```

RPAD 함수

RPAD('문자열' 또는 칼럼명, 자릿수, '채울문자')

- student 테이블에서 ID를 12자리로 출력하되 오른쪽 빈 자리에는 '*'기호로 채우세요

```
SQL> select id, rpad(id, 12, '*') "LPAD" from student  
2  where deptno1=101;
```

ID	LPAD
75true	75true*****
bingo	bingo*****
onejima	onejima*****
prettygirl	prettygirl**

LTRIM 함수

```
select 'java' || 'oracle 교육'
      , 'java' || ltrim('oracle 교육')
      , rtrim('java o ') || 'racle 교육'
from dual;
```

LTRIM('문자열' 또는 칼럼명, '제거할 문자')

- LPAD, RPAD 함수가 채우는 함수였다면, LTRIM, RTRIM 함수는 제거하는 함수
- DEPT2 테이블에서 DNAME을 출력하되 왼쪽에 '영'이란 글자를 모두 제거하고 출력

```
SQL> select dname, ltrim(dname,'영') from dept2;
```

DNAME	LTRIM(DNAME,'영')
사장실	사장실
경영지원부	경영지원부
재무관리팀	재무관리팀
총무팀	총무팀
기술부	기술부
H/W지원	H/W지원
S/W지원	S/W지원
영업부	영업부
영업기획팀	영업기획팀
영업1팀	영업1팀
영업2팀	영업2팀
영업3팀	영업3팀
영업4팀	영업4팀

13 개의 행이 선택되었습니다.

RTRIM 함수

RTRIM('문자열' 또는 칼럼명, '제거할 문자')

- DEPT2 테이블에서 DNAME을 출력하되 오른쪽 끝에 '부'라는 글자는 제거하고 출력

```
SQL> select dname, rtrim(dname,'부') from dept2;
```

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

13 개의 행이 선택되었습니다.



reverse()

- reverse ()

- 어떤 문자열을 거꾸로 보여주는 것

```
select 'ORACLE', reverse('ORACLE'),  
       '대한민국', reverse('대한민국'), --한글은 깨짐  
       reverse( reverse('대한민국') ) --한글도 정상으로  
from dual;
```

☰ 'ORACLE'	REVERSE('ORACLE')	'대한민국'	REVERSE('대한민국')	REVERSE(REVERSE('대한민국'))
▶ ORACLE	ELCARO	대한민국	ㄹ꺤꺤꺤꺤	대한민국

REPLACE 함수

REPLACE('문자열' 또는 칼럼명, '문자1', '문자2')

- 첫번째 문자열에서 문자1이 있으면 문자2로 바꾸어 출력하는 함수
- student 테이블에서 학생들의 이름을 출력하되 성 부분은 '#'으로 표시되게 출력

```
SQL> select name, replace(name, substr(name,1,1), '#') from student  
2 where deptno1=101;
```

NAME	REPLACE(NA
-----	-----
서진수	#진수
김신영	#신영
일지매	#지매
이윤나	#윤나

REPLACE 함수

- 실습) student 테이블에서 1전공이 101번인 학생들의 이름을 출력하되 가운데 글자만 '#'으로 표시되게 출력

replace	
서	#수
김	#영
일	#매
이	#나

- 실습) student 테이블에서 1전공이 101번인 학생들의 이름과 주민번호를 출력하되 주민번호의 뒤 7자리는 '*'로 표시되게 출력

NAME	replace2
서재수	750224*****
신은경	760412*****
김진욱	770402*****
김주현	780324*****

- 실습) student 테이블에서 1전공이 102번인 학생들의 이름과 전화번호, 전화번호에서 국번 부분만 '#' 처리하여 출력. 단, 모든 국번은 3자리로 간주함

NAME	TEL	replace3
서재수	051)426-1700	051)###-1700
신은경	051)418-9627	051)###-9627
김진욱	055)488-2998	055)###-2998
김주현	055)423-9870	055)###-9870



숫자 함수

- 숫자 함수 - 함수에 입력되는 값이 숫자인 함수

함수명	의미	사용 예
ROUND	주어진 숫자를 반올림한 후 출력	ROUND(12.345, 2)=>12.35
TRUNC	주어진 숫자를 버림한 후 출력	TRUNC(12.345, 2)=>12.34
MOD	주어진 숫자를 나누기한 후 나머지 값 출력	MOD(12, 10)=>2
CEIL	주어진 숫자와 가장 근접한 큰 정수 출력	CEIL(12.345)=>13
FLOOR	주어진 숫자와 가장 근접한 작은 정수 출력	FLOOR(12.345)=>12
POWER	주어진 숫자 1의 숫자 2승을 출력	POWER(3,2)=>9

ROUND / TRUNC 함수

ROUND(숫자, 원하는 자릿수)

- 자릿수를 안 준 경우,
자릿수를 2자리로 준 경우,
자릿수를 마이너스(-)로 준 경우

```
SQL> 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

TRUNC(숫자, 원하는 자릿수)

- 자릿수를 안 준 경우,
자릿수를 2자리로 준 경우,
자릿수를 마이너스(-)로 준 경우

```
SQL> 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

MOD, CEIL, FLOOR 함수

```
SQL> select MOD(12, 10) "Mod", CEIL(12.345) "Ceil", FLOOR(12.345) "Floor"  
2 from dual;
```

Mod	Ceil	Floor
2	13	12

■ POWER 함수

POWER(숫자1, 숫자2)

```
SQL> select power(3, 2) from dual;
```

```
POWER(3,2)
```

```
-----  
9
```

날짜 함수

- 날짜를 사용하는 경우
 - 2월 5일 +3 은 2월 8일
 - 2월 5일 -3 은 2월 2일
 - 2월 5일 - 2월 2일은 3일 차이 난다
 - => 날짜 + 숫자 = 날짜로 인식 (며칠 후)
 - 날짜 - 숫자 = 날짜로 인식 (며칠 전)
 - 날짜 - 날짜 = 숫자로 인식 (두 날짜 사이의 간격)

함수명	의미	결과
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



SYSDATE 함수

- 현재 시스템의 시간을 출력해 주는 함수
 - 오라클에서는 보통 날짜 체계가 미국을 기준으로 되어 있어서 우리나라와 다른 형식으로 출력
 - 한국에서 익숙한 형태로 표기하기 위해서는 다음 작업 후 조회

```
SQL> select sysdate from dual;
```

```
SYSDATE
```

```
-----  
13/01/28
```

```
SQL> alter session set NLS_DATE_FORMAT='YYYY-MM-DD HH24:MI:SS';
```

세션이 변경되었습니다.

```
SQL> select sysdate from dual;
```

```
SYSDATE
```

```
-----  
2013-01-28 22:39:40
```

MONTH_BETWEEN 함수

- 두 날짜를 입력 받아서 두 날짜 사이의 개월 수를 출력하는 함수

```
SQL> select months_between('2013-03-01', '2013-01-01')
       2 from dual;

MONTHS_BETWEEN('2013-03-01', '2013-01-01')
-----
2
```

- 두 날짜를 입력해서 두 날짜 사이의 개월 차이를 알려주는데 MONTH_BETWEEN 함수와 관련된 기본적 규칙
 - 1) 두 날짜 중 큰 날짜를 먼저 써야 양수가 나옴

```
SQL> select months_between('2013-01-01', '2013-03-01')
       2 from dual;

MONTHS_BETWEEN('2013-01-01', '2013-03-01')
-----
-2
```

- 2) 두 날짜가 같은 달에 속해 있으면 특정 규칙으로 계산된 값이 나옴
 - 단, 1개월이 28일, 30일, 31일일 경우 모두 값이 다르게 계산됨

```
SQL> select months_between('2013-02-28', '2013-02-01')
       2 from dual;

MONTHS_BETWEEN('2013-02-28', '2013-02-01')
-----
.870967742
```

```
SQL> select months_between('2013-04-30', '2013-04-01')
       2 from dual;

MONTHS_BETWEEN('2013-04-30', '2013-04-01')
-----
.935483871
```

```
SQL> select months_between('2013-01-31', '2013-01-01')
       2 from dual;

MONTHS_BETWEEN('2013-01-31', '2013-01-01')
-----
.967741935
```

MONTH_BETWEEN 함수

- 3) 두 날짜가 모두 해당 월의 마지막 날이거나 처음 날이면 개월 수가 정수 값으로 나옴

```
SQL> select months_between('2013-07-01', '2013-01-01')
       2 from dual;

MONTHS_BETWEEN('2013-07-01', '2013-01-01')
-----
6
```

- 4) 두 날짜 중 하나는 시작일이고 하나는 마지막 일이면 특정 규칙으로 계산된 값이 나옴

```
SQL> select months_between('2013-03-01', '2013-01-31')
       2 from dual;

MONTHS_BETWEEN('2013-03-01', '2013-01-31')
-----
1.03225806
```

- 2월이 28일까지 있는 달과 29일까지 있는 달을 구분하지 못함

```
SQL> select months_between('2013-03-01', '2013-02-28')
       2 from dual;

MONTHS_BETWEEN('2013-03-01', '2013-02-28')
-----
.129032258
```

2월이 28일까지 있는 2013년 2월 조회

```
SQL> select months_between('2012-03-01', '2012-02-28')
       2 from dual;

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

2월이 29일까지 있는 2012년 2월 조회

ADD_MONTHS 함수

■ ADD_MONTHS 함수

- 주어진 날짜에 숫자 만큼의 달을 추가하는 함수

```
SQL> select sysdate, add_months(sysdate, 3)  
2 from dual;
```

SYSDATE	ADD_MONT
13/01/30	13/04/30

■ NEXT_DAY 함수

- 주어진 날짜를 기준으로 돌아오는 가장 최근 요일의 날짜를 반환해 주는 함수
- 요일명 대신 숫자를 입력할 수 도 있다
 - 1:일, 2:월 ..7:토

```
SQL> select sysdate,next_day(sysdate, '월') from dual;
```

SYSDATE	NEXT_DAY(SYSDATE, '월')
2013-02-20 23:25:15	2013-02-25 23:25:15

```
SQL>
```

```
SQL> select sysdate,next_day(sysdate, 2) from dual
```

SYSDATE	NEXT_DAY(SYSDATE,2)
2013-02-20 23:29:58	2013-02-25 23:29:58

```
SQL> select sysdate,next_day(sysdate, '월요일') from dual;
```

SYSDATE	NEXT_DAY(SYSDATE, '월요일')
2013-02-20 23:25:18	2013-02-25 23:25:18

LAST_DAY 함수

```
select sysdate+to_yinterval('02-04') as "2년4개월 후",  
sysdate+to_dsinterval('1 03:10:20') as"1일 3시간 10분  
20초 후",  
sysdate+to_yinterval('02-04')+to_dsinterval('1  
03:10:20')  
as "2-4-1 3:10:20후", sysdate from dual;
```

■ LAST_DAY 함수

- 주어진 날짜가 속한 달의 가장 마지막 날을 출력해주는 함수

```
SQL> select sysdate, last_day(sysdate)  
2 from dual;
```

SYSDATE	LAST_DAY
13/01/30	13/01/31

■ to_yinterval(), to_dsinterval()

to_yinterval

- 문자 형태를 년, 월의 기간 차이로 변환
- 예) 입사일 이후 3년 4개월 지난 직원 조회

to_dsinterval

- 문자 형태를 일, 시간의 기간 차이로 변환
- 예) 입사일 이후 15일 10시간 지난 직원 조회

LAST_DAY 함수

ROUND/ TRUNC 함수
⇒ 시간은 제외됨
⇒ 00:00:00 으로 셋팅

■ ROUND/ TRUNC 함수

- 날짜의 ROUND 는 하루의 반에 해당되는 시간은 낮(정오) 12:00:00 인데 주어진 날짜가 이 시간을 넘어서는 경우에 다음 날짜로 출력하고 이 시간이 안될 경우에는 당일로 출력
- 날짜의 TRUNC 함수는 무조건 당일로 출력함
 - 주로 원서 접수나 상품 주문 등에서 오전까지 접수된 건은 당일 접수 처리하고 오후에 접수된 건은 익일로 처리할 경우에는 접수 날짜를 ROUND 함수로 처리
 - 접수 시간에 상관없이 당일로 처리하는 경우는 TRUNC 함수로 처리하는 등의 경우에 사용될 수 있음

```
SQL> alter session set nls_date_format='YYYY-MM-DD HH24:Mi:SS';
```

세션이 변경되었습니다.

```
SQL> select sysdate, round(sysdate), trunc(sysdate)
2 from dual;
```

SYSDATE	ROUND(SYSDATE)	TRUNC(SYSDATE)
2013-01-30 20:37:54	2013-01-31 00:00:00	2013-01-30 00:00:00



예

```
select sysdate, round(sysdate), trunc(sysdate) from dual;
```

-- 오늘부터 2년 3개월 1일 2시간 10분 5초 후 출력

```
select to_char(sysdate, 'yyyy-mm-dd hh24:mi:ss') ,  
       to_char(add_months(sysdate, 2*12 + 3) + 1 + 2/24 + 10/(24*60) + 5/(24*60*60)  
              , 'yyyy-mm-dd hh24:mi:ss') ,  
       to_char(sysdate + to_ymininterval('02-03') + to_dsinterval('001 02:10:5')  
              , 'yyyy-mm-dd hh24:mi:ss')  
from dual;
```



예제

- emp테이블에서 사원의 입사일 90일 후의 날짜?
- emp테이블에서 사원의 입사후 1년이 되는 날짜?
- 오늘부터 크리스마스까지 남은 일수는?
- 오늘부터 크리스마스까지 남은 달수는?
(months_between)



예제

- emp테이블에서 입사한지 오늘까지 몇일 되었나?
- emp테이블에서 입사한지 오늘까지 몇달 되었나?
- emp테이블에서 입사한지 오늘까지 몇 년이 되었나?



예제

- 현재 날짜에 해당하는 달의 마지막 날짜는?
(last_day)
- 2016-02-13에 해당하는 달의 마지막 날짜는?

형 변환 함수

문자 - char, varchar2
숫자 - number
날짜 - date

- 오라클에서 사용하는 데이터 타입

데이터 타입	설명
CHAR(n)	고정길이의 문자를 저장함. 최대값은 2000 bytes
VARCHAR2(n)	변하는 길이의 문자를 저장함. 최대값은 4000 bytes
NUMBER(p, s)	숫자값을 저장함. p는 전체 자릿수로 1~38자리까지 가능, s는 소수점 이하 자릿수로 -84~127자리가 가능
DATE	총 7byte로 BC 4712년 1월1일 부터 AD 9999년 12월 31일까지의 날짜를 저장할 수 있음
LONG	가변 길이의 문자를 저장, 최대 2GB까지 저장 가능
CLOB	가변 길이의 문자를 저장, 최대 4GB까지 저장 가능
BLOB	가변 길이의 바이너리 데이터를 최대 4GB까지 저장 가능
RAW(n)	원시 이진 데이터로 최대 2000 bytes 까지 저장 가능
LONG_ROW(n)	원시 이진 데이터로 최대 2 GB까지 저장 가능
BFILE	외부 파일에 저장된 데이터로 최대 4GB까지 저장 가능

형 변환 함수

```
SQL> select 1+ 'A' from dual;  
select 1+ 'A' from dual  
*  
1행에 오류:  
ORA-01722: 수치가 부적합합니다
```

- 각종 데이터들을 저장하기 위해서 오라클에서 다양한 형태의 데이터 타입을 제공함
- 오라클에서의 형변환
 - 묵시적(자동)형 변환 - 오라클이 자동으로 형변환을 시킨다는 뜻
 - 명시적(수동)형 변환 - 수동으로 지정해 주어야 한다는 의미
- 자동 형변환이 이루어지는 예
 - 숫자 + '문자' => 에러 발생
 - 예외) 숫자처럼 생긴 문자는 자동으로 문자를 숫자로 변환 후 연산

```
select * from emp2  
where birthday >= '1982-04-01'
```

```
SQL> select 1+ '1' from dual;
```

1+'1'
2

- 내부적으로 묵시적 형변환
Select 1+ to_number('1')
from dual;

- 묵시적 형변환
 - 튜닝에서 뜻하지 않게 속도가 느려지는 주범이 될 수 있으므로 조심해야 함

- 형변환 함수 - 명시적으로 데이터 형을 변환할 때 사용하는 것
 - TO_CHAR() - 숫자와 날짜를 **문자로 변환**
 - TO_NUMBER() - 문자를 **숫자로**
 - TO_DATE() - 문자를 **날짜로**

TO_CHAR 함수

■ 날짜를 문자로 형 변환하기

TO_CHAR(원래 날짜, '원하는 모양')

■ 날짜를 표현하는 방법

- 연도 : 'YYYY' - 연도를 4자리로 표현 예) 1999

'RRRR' - 2000년 이후에 Y2K 버그로 인해 등장한 새로운 날짜 표기법

'YY' - 연도를 끝의 2자리만 표시함 예) 99

'YEAR' - 연도의 영문 이름 전체를 표시

```
SQL> 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
```

```
-----  
2013 2013 13 TWENTY THIRTEEN
```

TO_CHAR 함수

- 월 : MM - 월을 숫자 2자리로 표현 예) 10
MON - 월을 뜻하는 영어 3글자로 표시 예) oct
MONTH - 월을 뜻하는 영어 이름을 전체를 표시

PARAMETER	VALUE
NLS_LANGUAGE	KOREAN
NLS_TERRITORY	KOREA
NLS_CURRENCY	₩
NLS_ISO_CURRENCY	KOREA
NLS_NUMERIC_CHARACTERS	.,
NLS_CALENDAR	GREGORIAN
NLS_DATE_FORMAT	RR/MM/DD
NLS_DATE_LANGUAGE	KOREAN
NLS_SORT	BINARY
NLS_TIME_FORMAT	HH24:MI:SSXFF

```
SQL> select to_char(sysdate, 'MM') 'MM',
2          to_char(sysdate, 'MON') 'MON',
3          to_char(sysdate, 'MONTH') 'MONTH'
4  from dual;
```

MM	MON	MONTH
01	1월	1월

```
SQL> SELECT *
2   FROM NLS_SESSION_PARAMETERS
3   WHERE PARAMETER = 'NLS_DATE_LANGUAGE';
```

PARAMETER

VALUE

NLS_DATE_LANGUAGE
KOREAN

```
SQL> select to_char(sysdate, 'MM') 'MM',
2          to_char(sysdate, 'MON', 'NLS_DATE_LANGUAGE=AMERICAN') 'MON',
3          to_char(sysdate, 'MONTH', 'NLS_DATE_LANGUAGE=AMERICAN') 'MONTH'
4  from dual;
```

MM	MON	MONTH
01	JAN	JANUARY

TO_CHAR 함수

- 일 : DD - 일을 숫자 2자리로 표시 예)12
DAY - 요일에 해당하는 영어 명칭을 표시
DDTH - 몇 번째 날인지를 표시

```
SQL> select to_char(sysdate, 'DD') "DD",  
2          to_char(sysdate, 'DAY') "DAY",  
3          to_char(sysdate, 'DDTH') "DDTH"  
4  from dual;
```

DD	DAY	DDTH
30	수요일	30TH

```
SQL> select to_char(sysdate, 'DD') "DD",  
2          to_char(sysdate, 'DAY', 'NLS_DATE_LANGUAGE=AMERICAN') "DAY",  
3          to_char(sysdate, 'DDTH') "DDTH"  
4  from dual;
```

DD	DAY	DDTH
30	WEDNESDAY	30TH

TO_CHAR 함수

- 시간 : **HH24** - 하루를 24시간으로 표시
HH - 하루를 12시간으로 표시
- 분 : **MI** 로 표시
- 초 : **SS** 로 표시

```
SQL> select sysdate, to_char(sysdate, 'YYYY-MM-DD HH24:MI:SS')  
2 from dual;
```

SYSDATE	TO_CHAR(SYSDATE, 'YY
2013-01-30 21:28:19	2013-01-30 21:28:19

- 실습) STUDENT 테이블의 birthday 칼럼을 참조하여 생일이 3월인 학생의 이름과 birthday를 출력

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



TO_CHAR

■ 숫자를 문자로 형 변환하기

종류	의미	사용 예	결과
9	9 하나당 1자리를 의미	TO_CHAR(1234, '99999')	1234
0	빈 자리를 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 구분 기호를 표시	TO_CHAR(12345, '99,999')	12,345



TO_CHAR

- Professor 테이블을 참조하여 101번 학과 교수들의 이름과 연봉을 출력하시오. 단 연봉은 $(pay*12)+bonus$ 로 계산하고 천 단위 구분기호로 표시하시오.

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

NAME	연봉
조인성	6,700
박승민	4,620
송도원	

$$\text{pay} * 12 + \text{pay} * 12 * 0.1 \Rightarrow \text{pay} * 12 * (1 + 0.1) \Rightarrow \text{pay} * 12 * 1.1$$

TO_NUMBER, TO_DATE 함수

TO_NUMBER('1')

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

TO_NUMBER('1')

1

TO_DATE ('문자', '날짜 포맷')

```
SQL> select to_date('2013-01-01', 'YYYY-MM-DD')
2 from dual;
```

TO_DATE('2013-01-01

2013-01-01 00:00:00

- [실습] Professor 테이블을 사용하여 1990년 이전에 입사한 교수명과 입사일, 현재 연봉과 10% 인상 후 연봉을 출력하시오.

연봉은 상여금(bonus)를 제외한 (pay*12)로 계산하고 연봉과 인상 후 연봉은 천 단위 구분 기호를 추가하여 출력하시오.

NAME	입사일	연봉	10% 인상 후 연봉
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

62 ▶ select to_date('2013-05-17') from dual;

Data Grid

Messages | Data Grid | Trace | DBMS Output (disabled)

Row# TO_DATE('2013-05-17')

1 2013-05-17

select to_date('2015-08-07')+2 from dual;



TO_DATE 함수

- pd 테이블에서 regdate(등록일) 가 '2015-07-17' ~ '2015-08-06' 일 사이의 데이터만 조회하기

```
select * from pd
where regdate >='2015-07-17' and regdate< to_date('2015-08-06')+1;

select * from pd
where regdate >='2015-07-17'
and regdate<= to_date('2015-08-06 23:59:59', 'yyyy-mm-dd hh24:mi:ss');
```

- 24시간 이내에 등록한 상품을 구하기 위해서 현재날짜와 등록일 사이의 경과된 일수 구하기

```
select pdname, regdate, sysdate-regdate from pd
where regdate>='2015-01-01';
```




일반함수

- 일반함수 - 함수의 입력되는 값이 숫자, 문자, 날짜 구분 없이 모두 사용할 수 있는 함수

- (1) NVL 함수

- NULL 값을 만나면 다른 값으로 치환해서 출력하는 함수

NVL(칼럼, 치환할 값)

- NULL 값을 0을 포함한 다른 값으로 바꾸어 주는 함수
- 치환값이 숫자일 경우
 - NVL(sal, 100) => sal 칼럼의 값이 null 일 경우 null 대신 100으로 치환
- 치환값이 문자일 경우
 - NVL(job, '기타') => job 값이 null 일 경우 '기타'로 치환
- 치환값이 날짜일 경우
 - NVL(hiredate, sysdate) => hiredate 값이 null 일 경우 sysdate 날짜로 치환

일반함수

- Professor 테이블에서 101번 학과 교수들의 이름과 급여, bonus, 연봉을 출력하시오. 단, 연봉은 (pay*12+bonus)로 계산하고 bonus가 없는 교수는 0으로 계산하시오.

```
SQL> select name, pay, bonus, (pay*12+NVL(bonus,0)) "연봉"  
2  from professor  
3  where deptno=101;
```

NAME	PAY	BONUS	연봉
조인형	550	100	6700
박승근	380	60	4620
송도권	270		3240

일반함수

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

■ (2) NVL2 함수

NVL2(col1, col2, col3)

- Col1의 값이 null이 아니면 col2를, null 이면 col3를 출력

- Professor 테이블에서 교수의 이름과 pay, bouns, 연봉을 출력 하시오. 단, 연봉은 (pay*12+bonus)로 계산하고 bonus가 없으면 급여를 0으로 처리하시오.

```
SQL> select name, pay, bonus, NVL2(bonus, pay*12+bonus, 0) "연봉"  
2  from professor  
3  where deptno=101;
```

NAME	PAY	BONUS	연봉
조인형	550	100	6700
박승근	380	60	4620
송도권	270		0

일반함수

■ (3) DECODE 함수

- IF 문을 오라클 SQL안으로 가져온 함수
- 오라클에서만 사용되는 함수로 IF 문을 사용해야 하는 조건문을 처리

■ IF 문과 DECODE문 비교

- A가 B일 경우 '1'을 출력하는 경우

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

```
DECODE(A, B, '1', NULL)
```

마지막 NULL은 생략가능

- A가 B일 경우 '1'을 출력하고 아닐 경우 '2'를 출력하는 경우

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

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

일반함수

- A가 B일 경우 '1'을 출력하고 A가 C일 경우 '2'를 출력하고 둘 다 아닐 경우 '3'을 출력하는 경우

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

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

- A가 B일 경우 중에서 A가 C를 만족하면 '1'을 출력하고 A가 C가 아닐 경우 '2'를 출력하는 경우 (DECODE함수 안에 DECODE 함수가 중첩)

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

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

일반함수

- A가 B일 경우 중에서 A가 C를 만족하면 '1'을 출력하고, A가 C가 아닐 경우 '2'를 출력하고 A가 B가 아닐 경우 '3'을 출력하는 경우

(DECODE 함수 안에 DECODE 함수가 중첩되는 경우)

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

- Professor 테이블에서 교수명, 학과번호, 학과명을 출력하되 deptno가 101번인 교수만 컴퓨터 공학과로 출력하고 101번이 아닌 교수들은 학과명에 아무것도 출력하지 마세요.

```
SQL> select name, deptno, DECODE(deptno, 101, '컴퓨터 공학과', "학과명"
2 from professor;
```

NAME	DEPTNO	학과명
조인형	101	컴퓨터 공학과
조박승근	101	컴퓨터 공학과
조승도권	101	컴퓨터 공학과
조양선희	102	
조영조	102	
조승재	102	
김도형	103	
김나하	103	
김현정	103	
심승	201	
최슬기	201	

NAME	DEPTNO	학과명
박원범	202	
차범철	202	
바비	203	
전민호	301	
허은	301	

일반함수

- Professor 테이블에서 교수명, 학과번호, 학과명을 출력하되 deptno가 101번인 교수만 컴퓨터 공학과로 출력하고 101번이 아닌 교수들은 학과명에 '기타학과'로 출력하세요.

- Professor 테이블에서 교수명, 학과명을 출력하되 deptno가 101번이면 '컴퓨터 공학과', 102번이면 '멀티미디어 공학과', 103번이면 '소프트웨어 공학과', 나머지는 '기타학과'로 출력하세요.

```
SQL> select name, deptno, DECODE(deptno, 101, '컴퓨터 공학과', '기타학과') "학과"
2 from professor;
```

NAME	DEPTNO	학과
조인성	101	컴퓨터 공학과
박승도	101	컴퓨터 공학과
송영선	101	컴퓨터 공학과
김주호	102	기타학과
양희재	102	기타학과
김영준	102	기타학과
김나현	103	기타학과
정승기	103	기타학과
김현성	201	기타학과
최슬기	201	기타학과

NAME	DEPTNO	학과
박원범	202	기타학과
차범철	202	기타학과
바비	203	기타학과
전민익	301	기타학과
허영	301	기타학과

```
SQL> 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
조인성	101	컴퓨터 공학과
박승도	101	컴퓨터 공학과
송영선	101	컴퓨터 공학과
김주호	102	멀티미디어 공학과
양희재	102	멀티미디어 공학과
김영준	102	멀티미디어 공학과
김나현	103	소프트웨어 공학과
정승기	103	소프트웨어 공학과
김현성	201	기타학과
최슬기	201	기타학과

NAME	DEPTNO	학과명
박원범	202	기타학과
차범철	202	기타학과
바비	203	기타학과
전민익	301	기타학과
허영	301	기타학과

16 개의 행이 선택되었습니다.

일반함수

- Professor 테이블에서 교수명, 부서번호를 출력하고, deptno가 101번인 부서 중에서 이름이 조인형인 교수에게 '석좌교수 후보'라고 출력하세요. 나머지는 null 값 출력.

- Professor 테이블에서 교수명, 부서번호를 출력하고, deptno가 101번인 부서 중에서 이름이 조인형인 교수에게 비교란에 '석좌교수 후보'라고 출력하세요. 101번 학과의 조인형 교수 외에는 비교란에 '후보아님'을 출력하고 101번 교수가 아닐 경우는 비교란이 공란이 되도록

```
SQL> select name, deptno,
2      DECODE(deptno, 101, DECODE(name, '조인형', '석좌교수후보')) "비교"
3 from professor;
```

NAME	DEPTNO	비교
조인형	101	석좌교수후보
박승도	101	
송도권	101	
양선희	102	
김영조	102	
주승재	102	
김도형	103	
나한영	103	
김현정	103	
김신기	201	
최슬기	201	

NAME	DEPTNO	비교
박원범	202	
차범철	202	
바비	203	
전민호	301	
허인	301	

```
SQL> select name, deptno,
2      DECODE(deptno, 101, DECODE(name, '조인형', '석좌교수후보', '후보아님')) "비교"
3 from professor;
```

NAME	DEPTNO	비교
조인형	101	석좌교수후보
박승도	101	후보아님
송도권	101	후보아님
양선희	102	
김영조	102	
주승재	102	
김도형	103	
나한영	103	
김현정	103	
김신기	201	
최슬기	201	

NAME	DEPTNO	비교
박원범	202	
차범철	202	
바비	203	
전민호	301	
허인	301	

일반함수 - 실습

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

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

decode(), substr() 이용

- Student 테이블에서 제 1전공(deptno1)이 101번인 학생들의 이름(name)과 전화번호(tel), 지역명을 출력하세요. 지역명은 지역번호가 02는 서울, 031은 경기, 051은 부산, 052는 울산, 055는 경남으로 출력하세요

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

decode(), substr(), instr() 이용

CASE 표현식

- case 함수 - 동일 값 뿐만 아니라 범위 값을 판단할 수 있는 함수
- 오라클 8i부터 지원됨

- DECODE 함수는 주로 = 인 값을 처리하는데 사용됨
 - 즉, 크거나 작은 조건을 처리할 경우 decode 함수는 불편한 부분이 많음
=> CASE 표현식에서는 쉽게 처리

```
CASE 조건 WHEN 결과1 THEN 출력1  
           [WHEN 결과2 THEN 출력2]  
           ELSE 출력3  
END "칼럼명"
```

- DECODE - 함수 내부에 콤마로 조건들이 구분,
- CASE 표현식 내부는 콤마가 사용되지 않음

- DECODE 와 동일하게 '=' 조건으로 사용되는 경우
 - Student 테이블에서 학생들의 이름과 전화번호, 지역명을 출력하세요. 단, 지역번호가 02는 서울, 031은 경기, 051은 부산, 052는 울산, 055는 경남, 나머지는 기타로 출력하세요

```
SQL> select name, tel, CASE<SUBSTR<tel, 1, INSTR<tel, '>'>-1>> WHEN '02' THEN '서울'
2      WHEN '031' THEN '경기' WHEN '051' THEN '부산'
3      WHEN '052' THEN '울산' WHEN '055' THEN '경남'
4      ELSE '기타'
5      END "지역명"
6 from student;
```

NAME	TEL	지역
서진수	055>381-2158	경남
서재수	051>426-1700	부산
이미경	053>266-8947	기타
이재수	02>6255-9875	서울
김박동호	031>740-6388	경기
김신영	055>333-6328	경남
김이영	051>418-9627	부산
오나라	051>724-9618	부산
구유미	055>296-3784	경남
임세현	02>312-9838	서울
이지매	02>6788-4861	서울

NAME	TEL	지역
김지욱	055>488-2998	경남
김관훈	053>736-4981	기타
김민정	02>6175-3945	서울
김노호	051>785-6984	부산
이정호	055>278-3649	경남
이유나	02>381-5440	서울
이영미	031>345-5677	경기
이주우	055>423-9870	경남
이재우	02>6122-2345	서울

CASE 표현식

- 비교 조건이 '=' 이 아닌 경우
 - Student 테이블의 JUMIN 칼럼을 참조하여 학생들의 이름과 태어난 달, 분기를 출력하세요. 태어난 달이 01~03월은 1/4분기, 04~06월은 2/4분기, 07~09월은 3/4분기, 10~12월은 4/4분기로 출력하세요

```
SQL> select name, SUBSTR(jumin, 3,2) "출생월",
2      CASE WHEN <SUBSTR(jumin, 3, 2)> BETWEEN '01' AND '03' THEN '1/4분기',
3            WHEN <SUBSTR(jumin, 3, 2)> BETWEEN '04' AND '06' THEN '2/4분기',
4            WHEN <SUBSTR(jumin, 3, 2)> BETWEEN '07' AND '09' THEN '3/4분기',
5            WHEN <SUBSTR(jumin, 3, 2)> BETWEEN '10' AND '12' THEN '4/4분기',
6            ELSE '안태어남'
7      END "분기별"
8  from student;
```

NAME	출생	분기별
서진수	10	4/4분기
서재수	02	1/4분기
이미경	06	2/4분기
김재수	12	4/4분기
박동호	03	1/4분기
김신영	01	1/4분기
신은경	04	2/4분기
오나라	09	3/4분기
구유미	01	1/4분기
임세현	10	4/4분기
일지매	11	4/4분기

NAME	출생	분기별
김지욱	04	2/4분기
안광훈	09	3/4분기
김민호	02	1/4분기
김민호	12	4/4분기
노정호	08	3/4분기
이안수	01	1/4분기
이영민	08	3/4분기
김현우	03	1/4분기
허우	02	1/4분기

CASE WHEN 조건1 THEN 출력1
 [WHEN 조건2 THEN 출력2]
 ELSE 출력3
 END "칼럼명"

CASE 표현식

- 실습 - Professor 테이블을 조회하여 교수의 급여액수(pay)를 기준으로 200 미만은 4급, 201~300은 3급, 301~400은 2급, 401 이상은 1급으로 표시하여 교수의 번호(profno), 교수이름(name), 급여(pay), 등급을 출력하세요. 단, pay 칼럼을 내림차순으로 정렬하세요.

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

CASE 표현식

- emp 테이블에서 sal 이 2000 보다 크면 보너스는 1000, 1000보다 크면 500, 나머지는 0 으로 표시하세요

ENAME	SAL	보너스
SMITH2	800	0
ALLEN	1600	500
WARD	1250	500
JONES	2975	1000
MARTIN	1250	500
BLAKE	2850	1000
CLARK	2450	1000
SCOTT	3000	1000
KING	5000	1000
TURNER	1500	500