

Chapter 2 함수 : 데이터의 가공

2장 . 데이터 가공을 위한 함수의 사용

1. 함수의 기능

- 1) QUERY문 더욱 강력하게 해줌
- 2) 데이터의 계산을 수행
- 3) 개별 데이터의 항목을 수정
- 4) 표시할 날짜 및 숫자 형식을 지정
- 5) 열 데이터의 유형을 변환

2. 함수의 사용

◆ 단일행 함수

- for 치환
- 검색 / 출력에서 쓰이는 것에 따라 select / where / order by 절에 다르게 사용

| 문자함수 | 문자를 입력하면 문자 또는 숫자를 변환한다.
| 숫자함수 | 숫자를 입력하면 숫자 값을 변환한다.
| 날짜함수 | 날짜 유형을 변환한다.
| 변환함수(데이터타입) | 데이터 유형을 변환한다.
| 일반함수(null치환용) | 기타 데이터에 필요한 함수(null 변환 함수 등)

◆ 복수행 함수 (그룹함수, 집계함수)

for 집합처리

1) 문자함수

(1) INITCAP () 함수

INITCAP ('문자열' 또는 컬럼명)

- 첫 문자만 대문자로 출력하는 함수
- 카멜 표기법 - 띄어쓰기로 되어있는 문자들의 첫 글자만 대문자로 쓸 때

(2) LOWER () / UPPER () 함수

Lower, Upper ('문자열' 또는 컬럼명)

- where절에서 많이 쓰임

(3) LENGTH () / LENGTHB () 함수

Length / Lengthb ('문자열' 또는 컬럼명)

- 글자 수 / 바이트수

★ (4) SUBSTR () 함수 - 위치기반 문자 추출 함수

SUBSTR('문자열' 또는 컬럼명, m [, n])

- 문자열 중 지정한 위치(m) 에서 지정한 길이(n) 만큼 문자추출

- n은 생략 가능하며, 생략할 시 맨 끝까지 추출

-m이 음수인 경우 시작하는 방향만 오른쪽에서 m만큼, n은 원래 방향으로 추출 cf) instr()

```
-----  
select jumin, substr(jumin,7,1) as 성별, substr(jumin,1,6) as 생년월일  
from STUDENT;  
-----
```

substr() 통해서 날짜 관련 (년,월,일) 뺄 수도 있지만, 완벽한 함수라고 보기는 어려움.
원데이터 기준으로 하는 함수이기 때문! (= like)

◆ SUBSTR() 함수 ~ 날짜 관련 쓰려면?

· alter session set nls_date_format='yyyy/mm/dd';

날짜의 포맷 형식을 일시적으로 변경시킴 (원본의 데이터는 그대로.) / 세션을 껐다 켜면 끝

* alter ; 변경하는 명령어

* session : 접속한 현재 창

* nls_data_format : 날짜의 파라미터 포맷

· select sysdate from dual;

* **sysdate** : 현재 날짜. 시간 출력하는 함수 (+시,분,초)

* **dual** ; 저장되어있지 않은 데이터를 출력할 때 from에 형식상 가정하여 씬 - **더미 테이블**
(select는 from이 필요함) / **함수 테스트용**

(# select * from dual ; => DUMMY column에 X만 나타남 / 있는 그대로 단 하나의 행에 표현)

(**cf.** select sysdate from emp; / select 'sysdate' from emp;

sysdate 함수를 **표현식**으로 인지 - 모든 행에 현재시간 나타남

/ sysdate라는 문자만 나타남)

-----**substr ~ 년/월/일 표시 틀린 예시**-----

```
select substr(sysdate, 1, 4) as year  
from dual
```

2018을 하고 싶었는데 18/0이 나옴 => 원래 데이터가 18/05/01로 저장되어있기 때문에!
따라서 **substr도 원데이터 기준으로** 해야 한다 (**cf) like**)

----- substr ~ 년/월/일 표시 맞는 예시 -----

```
select substr (sysdate, 1, 2) as 년, substr (sysdate, 4, 2) as 월, substr (sysdate, -2, 2) as 일
from dual;
```

일 : -2로 시작. / 마지막이 시분초 ~ 초인 것 같지만, 일 형태로 잘 나옴.
왜냐면 **yyymmdd**만 원데이터. 시분초는 오렌지의 형식이기 때문에!

따라서 원래의 데이터가 어떤지 알고 있을 때만 형식 변형 없이 사용!

원데이터를 모를 때에는!

```
alter session set nls_date_format = 'yyyy/mm/dd';
select sysdate, substr(sysdate,1,4)
from dual;
```

alter session으로 세션에서의 원본 형식을 잠시 바꾸고! 바뀐 형태에 따른 substr() 사용!

- 월은 substr(sysdate,5,2) / substr(sysdate,-2,2) 혹은 susbstr(sysdate,-2)만
- desc 테이블명; 에서 확인했을 때, **date type**으로 지정된 column만 가능.
- 문자포맷으로 저장되어있는 형태는 이렇게 불가능!

★ 원본이 어떤 형태로 있는지 자꾸 신경써야 함,

날짜 형식이 각 데이터마다 다르므로 like와 substr 그냥 쓰면 오류가 생길 가능성이 크다 ★

이것보다 더 강력한 날짜함수는 뒤에!

(5) SUBSTRB() 함수

SUBSTRB('문자열' 또는 칼럼명, m [, n]) 함수

- 문자열에서 원하는 바이트 수만큼 추출
- m바이트의 글자부터 n바이트만큼 추출
- n이 없을 시 끝까지 추출

★ (6)INSTR() : 특정글자의 위치를 찾기 / return : 숫자

INSTR('문자열' 또는 컬럼명, 찾는 글자 [, m] [, n])

- 문자열 중 지정한 위치 (m) 에서 특정 문자 (char2)가 특정 순서(n)로 나타나는 문자열의 위치를 반환
- m, n 생략시 둘 다 1 로 두고 계산됨
- m이 음수인 경우 위치를 지정하는 법 뿐만 아니라 스캔 하는 방향도 바뀜! cf substr
- 찾고자 하는 문자가 없으면 0을 return
- instr은 substr와의 결합으로 자주 쓰임
- 숫자는 앞에서부터의 순서 위치!

ex. 이메일에서 id만 빼내기 => substr + instr 결합

◆ instring - stress 예시

```
select instr('stress','s',1,2) from dual
select instr('stress','s',-1,2) from dual
select instr('stress','s',3) from dual
select instr('stress','s',6) from dual
select instr('stress','s',-3) from dual
```

◆ 실습문제 - emp테이블

1. scott의 직원번호, 성명, 담당업무(소문자로), 담당업무(대문자로), 담당업무(첫글자만 대문자로) 변환하여 출력하여라.

sol)

```
select empno, ename, lower(job), upper(job), initcap(job)
  from emp
 where upper(ename) = upper('scott');
```

2. 이름의 첫글자가 'K'보다 큰 사원의 직원번호, 이름, 업무를 출력하여라.

sol)

```
select empno, ename, job
  from EMP
 where substr(ename,1,1) > 'K';
```

3. 이름이 6자리 이상인 사원의 이름과 업무를 출력하여라.

sol)

```
select ename, job
  from EMP
 where length(ename) >=6;
```

4. 이름 중 'L'자의 첫 위치를 출력하여라.

```
select instr(ename,'L',1,1)
  from EMP;
```

(7) LPAD() 함수 - 왼쪽 공백에 특별한 문자로 채우기

(8) RPAD() 함수 - 오른쪽 공백에 특별한 문자로 채우기

LPAD('문자열' 또는 컬럼명, 자리수 [, 채울 문자])

RPAD('문자열' 또는 컬럼명, 자리수 [, 채울 문자])

not PAD 함수

-자리수 : 채웠을 때 총 자리수 (not 위치에 따른 순서)

-문자열이 자리 수보다 클 경우에는 자리수만큼 줄여서 반환

-채울 문자 생략시 공백을 문자 대신 끼워서 반환함

PAD ; 삽입 목적 / LPAD-왼쪽, RPAD-오른쪽

동시 삽입하려면 두 함수를 써야함 , 한 번에 둘 다 삽입하는 함수는 X

채울문자를 사용해서 총 바이트수를 만들어라

원본의 문자열보다 넉넉한 자리수를 만들어야 함

/한글로 했을 때 조금 이상함.. 아마 바이트수로 자리 수를 계산하는 것 같다.

cf.

--문제8) EMP 테이블을 이용하여 사원이름을 표시하고 급여 총액을 별표로 나타내는 질의를 작성한다.

--각 별표는 100단위를 나타낸다. (예, sal+comm이 100이면 '*'가 된다)

```
select ename, lpad('*', (sal+nvl(comm,0))/100, '*')
from emp;
```

(9) LTRIM() 함수 - 왼쪽에 있는 대상 제거하기

(10) RTRIM() 함수 - 오른쪽에 있는 대상 제거하기

LTRIM('문자열' 또는 컬럼명, '제거할 문자')

RTRIM('문자열' 또는 컬럼명, '제거할 문자')

- 제한적임. 중간의 문자를 삭제하지 않음. 끝에 존재하는 문자만 제거하는 함수!

- 제거하고 또 그 제거할 문자가 있으면 계속 제거, 없으면 stop.

(11) TRIM() 함수 - 양쪽의 공백 제거하기

TRIM('문자열' 또는 컬럼명)

- 앞뒤 공백만 제거함.

★ 치환함수 - replace, translate

★ (12) REPLACE() 함수 - 문자열에서 특정 단어를 다른 단어로 변환

REPLACE('문자열' 또는 컬럼명, '문자1', '문자2')

-정확하게 일치하는 문자열만 통째로 치환함

단어 대 단어로 치환

★ (13) TRANSLATE() 함수 - 문자열에서 특정 문자를 다른 문자로 변환

TRANSLATE('문자열' 또는 컬럼명, '문자1', '문자2')

- '12' 'ab'이면 1->a, 2->b로 번역하는 기능을 가짐

-문자1의 개수 < 문자2의 개수이면 문자2에 없는 패턴은 치환되지 않는다.

-문자1의 개수 > 문자2의 개수이면 문자2에 없는 패턴은 삭제된다. (for 응용)

글자 대 글자로 치환하는 번역하는 함수

비교

```
-----
select replace ('abccba', 'ab', 'AB'),
       translate ('abccba', 'ab', 'AB')
from dual;
```

```
-----
replace  translate
ABccba  ABccBA
```

```
-----
select replace ('abccba', 'ab', 'ABC'),
       translate ('abccba', 'ab', 'ABC')
from dual;
```

```
-----
replace  translate
ABCccba  ABccBA
```

----- ★ 문제발생 ★

```
select replace ('abccba', 'abc', 'AB'),
       translate ('abccba', 'abc', 'AB')
from dual;
```

```
-----
replace translate
ABcba  ABBA
```

- 매칭되지 않는 c를 삭제하도록 하고 있다. => 중간문자 삭제하는 방법 활용!

--문제 단 하나의 선택문으로 ID에 있는 특수문자 '-' 삭제하기 in professor 테이블

sol1) replace() 사용

```
select replace(id, '-', '')
from professor;
```

sol2) translate() 사용

select translate(id, '-', '') (X) 뒤가 공백이면 모두가 삭제됨

select translate(id, '1-', '1') (O)

만일 - 말고도 삭제하고 싶으면

select translate(id, '1-!@#%^&*()_+|W', '1')

매칭되지 않는 정보는 삭제됨. 따라서 삭제할 것 앞에만 아무 글자 사용해서 매칭시켜 써주기

◆실습 문제 - 문자 관련 함수, emp테이블

1 이름에 *를 왼쪽에 채워 모두 동일한 15개의 이름으로 변환하고, 업무와 급여를 출력한다.

sol)

```
select LPAD(ename, 15, '*'), job, sal
from emp;
```

2. 급여를 모두 동일하게 만단위로 보여지게 변환하고, 이름과 부서번호를 함께 출력한다.

sol)

```
select RPAD(sal, 5, 0), ename, deptno
from emp;
```

3. 담당 업무 중 좌측의 'A'를 삭제하고, 급여 중 좌측의 1(숫자)를 삭제하여 출력하여라

sol)

```
select LTRIM(job, 'A'), LTRIM(sal, 1)
from emp;
```

4. 담당 업무 중 우측의 'T'를 삭제하고, 급여 중 우측의 0을 삭제하여 출력하여라.

sol)

```
select RTRIM(job, 'T'), RTRIM(sal, 0)
from emp;
```

5. job의 'A'를 '\$'로 바꾸어 출력하여라.

sol)

```
select replace(job, 'A', '$')
from emp;
```


2) 숫자 관련 함수들

(1) ROUND() / TRUNC()

ROUND(숫자 [, m]) / TRUNC(숫자 [, m])

- $m > 0$ 일 경우 소수점 **m+1** 번째 자리에서 반올림 / 버림
- $m < 0$ 일 경우 정수의 **-m** 번째 자리에서 반올림 / 버림
- $m = 0$, m 생략일 경우 소수점 첫 자리 반올림 / 버림

(2) MOD, FLOOR 함수

MOD(m,n) : m을 n으로 나눈 후 나머지 값

FLOOR(n) : 주어진 값보다 작거나 같은 최대 정수

CEIL(n) : 주어진 값보다 크지만 가장 근접하는 최소 정수

(3) ABS, SIGN 함수

ABS(n) : n의 절대값

SIGN(n) : 숫자가 양수면 1, 음수면 -1, 0이면 0을 반환하는 함수 **for 응용 ★ decode**

- 조건을 결합하여 쓸 때 ! 나중에 decode 함수에서 응용하는 방법

3) 날짜 관련 함수들

(1) SYSDATE 함수 - 현재 날짜와 시간, dual(더미테이블)과 함께 쓰임

```
-----  
select sysdate  
  from dual;  
-----
```

(2) MONTHS_BETWEEN - 개월 수

MONTHS_BETWEEN(날짜1, 날짜2)

- 두 날짜 간의 개월 수 차이 값
- 두 날짜 중 큰 날짜를 먼저 써야 양수가 나옴

```
-----  
select month_between(sysdate, hiredate)  
  from emp;  
-----
```

- **개월**은 나머지 소수점 **버림!** (왜냐면 sysdate에 시.분.초 존재해서!) => **trunc**사용

날짜 default 계산 : 일단위 -> 개월 수 함수 따로 존재!

(3) ADD_MONTHS n개월 이후

ADD_MONTHS(날짜1, n)

- 날짜1에서 n개월 이후 값을 반환
- n이 음수인 경우 n개월 이전 값을 반환함

(4) NEXT_DAY - 지정 날짜 이후 돌아오는 지정 요일

NEXT_DAY(날짜, 지정요일)

- 일요일 : 1, 월요일 : 2, 화요일 : 3, 수요일 : 4, 목요일 : 5, 금요일 : 6, 토요일 : 7
- db마다 날짜에 대한 표현식을 영어/한국어 등으로 함
 - > 즉, nls_date_language [파라미터]= Korean인 경우 지정요일을 '월', '화' 등으로도 이용 가능
 - > =American 인 경우 'MON', 'TUE' , ...

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

```
-----  
select sysdate, next_day(sysdate, 2)  
from dual;
```

cf. 언어를 바꾸는 방법

alter session set nls_date_language = american;

=> next_day(sysdate,'mon') 가능

(5) LAST_DAY 함수

LAST_DAY(날짜)

- 해당 월의 마지막 날짜를 반환

(6) ROUND, TRUNC 함수 in 날짜

ROUND(날짜, 단위)

- 날짜를 년, 월 단위로 반올림, 버림 (단위 : 'year', 'month') - default : 일 단위

```
-----  
select sysdate, round(sysdate),  
        round(sysdate,'month'),  
        round(sysdate, 'year')  
from dual;
```

round(sysdate) - 시간단위에서 반올림해서 일 단위를 맞춰줌

round(sysdate, 'month') - 일 단위에서 반올림해서 월 단위를 맞춰줌

round(sysdate, 'year') - 월 단위에서 반올림해서 년 단위를 맞춰줌

◆실습 문제 - 날짜 관련 문제, emp 테이블

1. emp 테이블에서 현재까지 근무일수가 몇 주, 몇 일인가를 출력하여라. 단, 근무일수가 많은 사람 순으로 출력하여라.

sol)

```
select trunc((sysdate - hiredate)/7) as 근무주 , trunc(mod (sysdate - hiredate, 7)) as 근무일
from emp
order by 근무주 desc, 근무일 desc;
```

2. 10번 부서원의 현재까지의 근무 월수를 계산하여 출력하여라.

sol)

```
select trunc(months_between(sysdate,hiredate))
from emp
where deptno=10;
```

3. 입사한 달의 남은 근무 일수를 계산하여 출력하여라.

단, 토요일과 일요일도 근무 일수에 포함한다.

sol)

```
select last_day(hiredate)-hiredate
from emp;
```

4. 10번 부서원의 입사 일자로부터 돌아오는 금요일을 계산하여 출력하여라.

sol)

```
select next_day(hiredate,6)
from emp
where deptno=10;
```

◆총괄 - 심화 문제

1. emp 테이블에서 급여(sal)와 커미션(comm) 합계가 10% 인상된 급여보다 많은 사원의 이름, 급여 및 커미션을 표시하도록 한다.

단, comm이 null이 아닌 경우만 나타내도록 한다

--1번

```
select ename, sal, comm
from EMP
where sal < (sal+comm)/1.1
and comm is not null;
```

2. emp 테이블에서 커미션을 받지 않는 사원의 급여를 10% 인상하여 표시하되, 다음의 형식으로 출력하여라.

The salary of SMITH after a 10% raise is 880

-- 2번

```
select 'The Salary of ' || ename || ' after a 10% raise is ' || sal*1.1
from EMP
where comm = 0;
```

3. EMP 테이블에서 이름의 첫 글자가 'K'보다 큰 사원의 사원번호, 이름, 업무를 출력하여라.

-- 3번

```
select empno, ename, job
from EMP
where substr(ename,1,1) > 'K';
```

4. PROFESSOR 테이블에서 email컬럼을 이용하여 email id를 각 교수의 이름과 함께 출력하여라.

-- 4번

```
select name, substr(email,1,instr(email,'@',1,1)-1)
from professor;
```

5. student 테이블에서 성이 '사'인 학생의 학번, 이름, 학년을 출력하여라.

--5번

```
select studno, name, grade
from student
where substr(name,1,1) >= '사'
and substr(name,1,1) < '아' ;
```

--한글의 초성검색은 like로 불가!

4) 데이터 형 변환 함수들

->TO_CHAR	<-TO_CHAR
숫자	문자 날짜
<-TO_NUMBER	->TO_DATE
+ 숫자->날짜 : TO_DATE	

숫자 앞에 0000, 공백, -등의 특수문자가 포함된 숫자 : 문자

◆ TO_CHAR(숫자 / 날짜, '문자열') / 두 번째 인수 홀따옴표 사용!!

(1) TO_CHAR 함수 (숫자형-> 문자형으로 변환하기)

종류	의미	사용 예	결과
9	9의 개수만큼 자리 수	TO_CHAR(9876, '99999')	9876 (부족한 자리수 공백 생김)
0	빈자리를 0으로 채우기	TO_CHAR(9876, '0999999')	009876 (0은 9 앞에만 쓸 수 있다)
\$	\$표시를 붙여서 표시	TO_CHAR(9876, '\$99999')	\$9876 (앞에 한 칸 공백 띄움)
.	소수점 이하를 표시	TO_CHAR(9876, '9999.99')	9876.00
,	천 단위 구분기호 표시	TO_CHAR(9876, '99,999')	9,876 (앞에 한 칸 공백 띄움)

(2) TO_CHAR (날짜->문자형) - **format 변경**

TO_CHAR(원래 날짜, '원하는 모양')

◆ in 원하는 모양

- 'Q' : 분기

[년도]

- 'YYYY' 현재 기준 년도 표현 - 18 : 2018년

- 'RRRR' 더 이전 년도 표현 - 18 : 1918

- 'YY'

- 'RR'

- 'YEAR' 연도의 영문 이름 전체를 표시 / 날짜가 영어로 되어있을 때만 가능

[월]

- 'MM' 월을 숫자 두 자리

- 'MON' 월의 축약형 이름 ex. JAN

- 'MONTH' 월의 이름 전체를 표시 ex. JANUARY

[일]

- 'DD' 일을 숫자 두 자리로 표현

- 'DAY' 요일로 표현

- 'DDTH' 몇 번째 날인지 서수로 표현

- 'DDSPTH' 몇 번째 날인지 서수 영어로 표현

[시간]

- 'HH24'

- 'HH'

- 'MI'

- 'SS'

- 날짜 관련 의미 문자에 담긴 대/소문자대로 표현됨

ex)

alter session set nls_date_language=american; --영어로 표현 방식 변경

TO_CHAR(sysdate, 'Day') -> Monday

TO_CHAR(sysdate, 'DAY') -> MONDAY

TO_CHAR(sysdate, 'day') -> monday

- 한글일 때랑 영문일 때 표현법이 다르다

ex. TO_CHAR(sysdate, 'day') -> 월요일

날짜를 다룰 땐 TO_CHAR를 사용하자.

```
-----
select hiredate,
       substr(hiredate,1,4),
       substr(to_char(hiredate,'yyyy-mm-dd'),1,4)
from emp;
-----
```

- TO_CHAR라는 날짜 포맷을 바꿔주는 강력한 함수 사용하여 날짜 handling
(substr을 통해 날짜를 다르면 불완전함. 데이터의 원래 형식을 다루는 함수여서!)

- 결과는 문자!!

영어로 변경했을 때에 표시 예시

----->영어로 변경-----

```
alter session set nls_date_language=american;
```

```
select sysdate,
       to_char(sysdate, 'year-Mon'),
       to_char(sysdate, 'DAY'),
       to_char(sysdate, 'Q'), --분기
       to_char(sysdate, 'ddth'),
       to_char(sysdate, 'ddspth') -- 일자를 all 영문 서수로 표현
from dual;
```

```
-----
2018/05/02 11:08:30      twenty eighteen-May    WEDNESDAY      2      02nd      second
```

-->TO_CHAR를 통해 바꾸고 싶은 날짜 포맷 : 05월 02일, 2018

```
-----
select to_char(sysdate, 'mm"월" dd"일", yyyy')
from dual;
```

-format과 무관한 텍스트를 같이 쓰고 싶을 때 " "를 통해 텍스트 작성

cf. " " 사용 예시- alias : " ", to_char : " "

- , 는 기호여서 밖에 써도 괜찮음.

Q 2018년 8월 10일의 요일을 구하기??

```
-----  
select to_char('2018/08/10', 'day')    --(X)  
    from dual;  
-----
```

'2018/08/10'을 날짜가 아닌 문자열로 인식.

오라클이 앞에 있는 인수를 날짜로 인식하게 하는 과제 필요!

-> **TO_DATE ()** 사용해야 날짜로 인식하게끔!

(3) TO_NUMBER

TO_NUMBER('문자,')

(4) TO_DATE()

TO_DATE('날짜처럼 생긴 문자/숫자' [, 원본의 날짜 타입])

-두 번째 인수를 쓰지 않으면 대개 **rrrr-mm-dd** 순서 (DB의 원래 포맷)

-앞에 생긴 문법이 어떻게 해석해야 하는지 두 번째 인수에 파싱

-날짜와 관련된 문자가 아니면 두 번째 인수에 "" 형태로 써야 함.

```
-----  
select to_date('08/10/2018', 'dd/mm/yyyy')  
-----
```

- **날짜로 인식하게** 하는 함수(변환함수)

--> 따라서, **날짜 포맷 변경 목적으로 TO_DATE를 쓰는 게 아님!** 포맷 변경 안 됨

TO_CHAR에서 포맷을 변경!

날짜로 인식시키는 함수는 TO_DATE

날짜 포맷의 변경은 TO_CHAR

```
-----  
select to_char(to_date('08/10/2018', 'dd/mm/yyyy'),mm/dd/yy')  
    from dual;  
-----
```

표시 형식은 **to_char** 함수에!

Q 2018년 8월 10일의 요일을 구하기

```
-----  
select to_char(to_date('2018/08/10'), 'day')  
    from dual;  
-----
```


◆ 추가문제

student 테이블에서 **jumin**에서 생년월일 구하고, 태어난 요일 구하기

```
-----  
select to_char(to_date(substr(jumin,1,6), 'rrmmdd'), 'day') as "태어난 요일"  
from student;  
-----
```

- to_date에서 두 번째 인수 생략했을 때 **rrmmdd**로 선택됨
- 그런데 인수를 작성하면서 **yymmdd**로 포맷을 쓰면, 앞을 2000년대로 표현하게 됨
=> 평소 **rr**을 인수로 쓰는 것을 권장! for 오류 발생 피하기, 파이썬 항상 사용

(5) 묵시적(자동)형 변환

데이터형을 자동으로 변환해주는 기능

```
SELECT 2 + '2'
```

```
FROM DUAL;
```

-> 저절로 `SELECT 2 + TO_NUMBER('2') FROM DUAL;`로 자동 변경됨

◆ 실습문제 - EMP 테이블

1. 10번 부서원의 입사 일자를 '1 MAY 1981'와 '1998년 01월 01일'의 형태로 각각 출력하여라.
SOL)

```
select to_char(hiredate, 'dd MON yyyy'),  
       to_char(hiredate, 'yyyy"년" mm"월" dd"일"')  
from emp  
where deptno = 10;
```

2. 20번 부서원의 급여 앞에 \$를 삽입하고 3자리마다 ,를 출력하여라.
sol)

```
select to_char(sal, '$99,999')  
from emp  
where deptno = 20;
```

3. february 22, 1981에 입사한 사원의 이름, 업무, 입사일자를 출력하여라.

* nls_date_language=american 인 경우

* 'february 22, 1981'의 형태는 바꿀 수 없다

SOL

1. 날짜-> 문자로 변경해서 hiredate를 문자로 = 'february 22, 1981' 과 비교

2. 문자-> 날짜로 변경해서 hiredate = 날짜와 비교

1) hiredate를 문자로 변경(to_char)

```
alter session set nls_date_language=american;
```

```
select ename, job, hiredate
```

```
from EMP
```

```
where to_char(hiredate,'month dd, yyyy') = 'february 22, 1981';
```

-> 불필요한 공백이 중간에 삽입이 됨. 매칭이 되지 않아 결과가 나오지 않음

가장 큰 글자 크기에 맞춰서 나머지 부족한 데이터들이 공백이 삽입되는 현상.

/ 만일 mon형태였으면 가능했을 것..!

- 대소문자 구별 구분 주의

(이 방법으로 하려면=> 중간 공백을 삭제하는 추가 조건 필요

month trim...!)

2) 문자를 날짜로 변경(to_date)

```
alter session set nls_date_language=american;
```

```
select ename, job, hiredate
```

```
from EMP
```

```
where hiredate = to_date('february 22, 1981','month dd, yyyy');
```

- 2번이 더 강력한 형식! 왜냐하면 컬럼을 가공하지 않는 방법이 더 성능이 좋은 방법.

- 데이터를 빠르게 찾기 위해서, **where절**에서 column 부분에 가공하지 않은 상태로 하는 편이 좋음 (인덱스 때문)

5) 다양한 일반 함수들

null 치환 함수 (NULL있으면 NULL이 포함되었을 때 사칙연산 결과도 NULL, 데이터 소멸)

->두 데이터 타입이 일치되도록 변경 !!중요!!

(1) NVL() 함수

NVL(조사할 컬럼, null일 경우 치환할 값)

- 모든 데이터 타입에 적용 가능

- 조사할 컬럼과 치환할 값의 데이터 타입이 같아야 함

```
select sal, comm, nvl(to_char(comm),'null')
from emp;
```

(comm은 숫자column, null값에 'null'이라는 문자열로 치환하기 위해, TO_CHAR(comm)로 변경!)

ex. 날짜 치환

NVL(hiredate, '2014/05/01')

(2) NVL2() 함수

NVL2(조사 대상 컬럼, null 아닐 때 치환할 값, null일 경우 치환할 값)

- 조사 대상 컬럼과 null 치환 값의 데이터 타입은 일치하지 않아도 된다.

- null일 경우 치환 값과, null 아닐 때 치환 값의 데이터 타입이 서로 일치해야 한다.

- 반환되는 데이터 타입은, 두 번째 인자인, null 아닐 때 치환할 값의 데이터 타입이 된다.

-----1-----

```
select comm, nvl2(comm, 'a','b')
from emp;
```

-----2-----

```
select comm, nvl2(comm, 0,'b') -- XXXX불가능
from emp;
```

-----3-----

```
select comm, nvl2(comm, '0','b')
from emp;
```

in 2,

nvl2 함수에 따라 숫자타입의 데이터로 comm이 새로 정의됨,

그런데 숫자 데이터 타입은 문자 데이터를 포함할 수 없음. 그래서 **error 발생**

에러발생 이유 -> 어떻게 변경해야 하는지

-----4-----

```
select comm, nvl2(comm, 'b',0)
from emp;
```

in 4, 'b'와 0의 데이터 형식이 다른데 가능함!

왜냐하면, 문자 데이터 타입에 숫자 데이터 표시 가능 (문자 데이터: 숫자, 날짜 데이터 상위개념)

=> NULL 아닐 때 치환할 값의 데이터 형태 우선적으로 신경쓰기!

(3) DECODE 함수

- only in 오라클
- 조건문(if)이 축약된 형식의 함수 표현식.
- 각 조건이 맞는 경우 각각 문자 치환 가능

ex.

--sql에서 if문 안됨. 해석을 위한 if들

if when deptno = 10 then '인사부'

when deptno = 20 then '총무부'

when deptno = 30 then '분석부'

=> decode(deptno, **10, '인사부'**, 20, '총무부', **30, '분석부'**)

두 개씩 세트로 보고 처리하기

```
select deptno,
       decode (deptno, 10, '인사부')
from emp;
```

짜이 맞지 않은 경우 인수가 생략되어있으면, null로 리턴됨.

```

-----
select deptno,
       decode (deptno, 10, '인사부', '기타')
from emp;
-----

```

짜이 맞지 않은 경우 모두 '기타'로 리턴됨.

```

-----
select deptno,
       decode (deptno, 10, '인사부', 20, '총무부', 30, '분석부', '기타')
from emp;
-----

```

else값이 필요하다면 '기타'로 리턴됨

◆ 중첩 decode

해석을 위한 중첩 if문

```

if when deptno = 20 then if when job = 'ANALYST' then '분석부'
                        else '기타부'
                        else null

```

=> decode(deptno, 20, decode(job, 'ANALYST', '분석부', '기타부'), null)

```

-----
select deptno, job, decode(deptno, 20, decode(job, 'ANALYST', '분석부', '기타부'), null)
from emp;
-----

```

decode의 단점 : 1. 등식(=)만 성립함, 대소비교가 불가능함. => sign()함수 이용 / case문 사용
 2. 비교대상이 하나로 통일되어야 함. => case문 사용

◆ 실습문제 - decode, emp테이블

1 부서번호가 10이면 'A'로, 20이면 'B'로, 30이면 'C'로 바꾼 뒤 이름, 업무, 부서번호와 함께 출력하여라.

sol)

```
select ename, job, decode(deptno, 10, 'A', 20, 'B', 30, 'C') as 부서번호
      from emp;
```

2. JOB = ANALYST -> 급여증가 10%, CLERCK -> 15%, MANAGER -> 20%, 나머지는 급여증가X. 이름, 업무, 급여, 증가된 급여를 출력하여라.

sol)

```
select empno,
       ename,
       job,
       sal,
       decode(job, 'ANALYST', sal*1.1,
               'CLERK', sal*1.15,
               'MANAGER', sal*1.2,
               sal) as "증가된 급여"
      from emp;
```

대/소문자 잘 표현!

3. 사원이름, 매니저번호(MGR)을 출력하고, 매니저번호가 null이면 '상위관리자', 있으면 ex. '7869담당'으로 표시

sol)

1. nvl2()로 풀기

```
select ename, mgr, nvl2(mgr, MGR||'담당', '상위관리자')
      from emp;
```

nvl2 -> #숫자 + 텍스트 => 문자형식

2. decode () - 나는 이렇게!

```
select ename, MGR, decode(MGR, null, '상위관리자', MGR||'담당')
      from emp;
```

※ **DECODE()** 에서 대소비교 하기 위해서 ~ **SIGN()** 결합

2000 초과 : A등급, 나머지 : B등급

-----1-----

```
select sal, sign(sal - 2000)
  from emp;
```

결과: -1, 0, 1 나옴.

-----2-----

```
select sal, decode(sign(sal - 2000), 1, 'A', 'B') as 등급
  from emp;
```

대소비교 가능!

(4) case() 함수

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

- case -> end로 끝내기

*조건이 안으로 들어가는 경우 : 대소비교 등

decode-> case로 바꿔보기 (실습 2번문제)

JOB = ANALYST -> 급여증가 10%, CLERCK -> 15%, MANAGER -> 20%, 나머지는 급여증가X.
이름, 업무, 급여, 증가된 급여를 출력하여라.

```
select case job when 'ANALYST' then sal * 1.1
              when 'CLERCK'   then sal * 1.15
              when 'MANAGER'  then sal * 1.2
              else SAL
            end as "증가된 급여"
  from emp;
```

- =(equal) 일 때만 case ____ when 에 넣기
eqaul 조건 생략, 영문자 대/소문자 잘 쓰기

비교식 등 연산자를 쓰려면 when 안으로 들여보내기

```
-----  
select case when job = 'ANALYST'    then sal * 1.1  
           when job = 'CLERCK'     then sal * 1.15  
           when job = 'MANAGER'    then sal * 1.2  
           else SAL  
        end as "증가된 급여"  
from emp;  
-----
```

sal>200 A, 나머지 B등급 case로 바꿔보기

```
-----  
select  
    case  when sal > 2000 then 'A'  
          else 'B'  
        end as 등급  
from emp;  
-----
```


3. 복수행 함수 - 여러 건의 데이터를 한꺼번에 처리하는 그룹함수

1) 기본적인 그룹 함수들

(1) COUNT() 함수 (+ group by -> 그룹별 건수)

COUNT(컬럼명)

- 숫자, 문자, 날짜 모두 셀 수 있음. NULL 여부만 확인.
 - COUNT() : **NULL을 포함하지 않은** 모든 건수
 - COUNT(*) : 전체 건수 모든 column이 null인 데이터는 없음.
- cf. count(not null인 column명) : 훨씬 빠르고, 정확함.**

=>

**count(*)보다 desc 테이블명; 확인 후,
not null인 column 이름을 통해 count () 쓸 것!**

```
-----
select count(*), count(empno), count(comm)
from emp;
-----
14      14      4
```

(2) SUM() 함수

SUM(컬럼명)

(3) AVG() 함수

AVG(컬럼명)

- NULL을 count하지 않는다.
- > 평균값이 원하는 값과 다른 경우도 있음. null 처리를 잘 해야 함

※ **avg(nvl(컬럼명,0))** : null값을 0으로 해서 전체에 대한 평균 만들기 cf) **avg(컬럼명)**

※ **null여부 먼저 체크 필요!** 필요에 따라 avg 바로 사용 or nvl로 치환 후 avg 사용

-----예시-----

```
select sum(comm) / 14 as 평균1,
       avg(comm) as 평균2,
       avg(nvl(comm,0)) as 평균1
from emp;
```

```
157.142857142857142857142857142857142857      550
```

```
157.142857142857142857142857142857142857
```

null = 0으로 간주했을 때의 평균, / null이 아닌 사람들만의 평균

(4) MAX(), MIN()

MAX(컬럼명), MIN(컬럼명)

※ 정리 !

COUNT (), AVG () : NULL여부에 따라 달라지는 함수!

◆배운 것 심화문제

-- 1) student 테이블에서 1전공(deptno1)이 101번인 학생들의 이름을 출력하되,
-- 두번째 글자만 '*'로 표시되게 출력하세요.

sol)

```
select replace(name, substr(name,2,1), '*')
  from STUDENT
where deptno01 = 101;
```

Q!!???!!만일 두 번째 글자가 뒤에도 나타날 수 있음!

-> 해결

```
select substr(name,1,1)||'*'||substr(name,3),name
  from student
where deptno1=101;
```

-- 2) emp 테이블에서 '1981년 09월 28일'에 입사한 학생의 이름, 직업, 입사일자를 출력하여라.
-- 단, '1981년 09월 28일'을 그대로 사용하여야 한다.

sol)

```
select ename, job, hiredate
  from EMP
where hiredate = to_date('1981년 09월 28일', 'yyyy"년" mm"월" dd"일"');
```

-- 3) student 테이블에서 전화번호 형식을 다음과 같이 변경하세요.
-- 051)426-1700 => 051 426 1700

sol)

```
select translate(tel, '-')-, ' ')
  from student;
```

-- 4)emp 테이블에서 각 연봉을 기준으로 아래의 연봉 등급을 구하세요 (decode 사용)

--sal < 2000 : C

--2000 =< sal < 3500 : B

--sal >= 3500 : A

sol)

```
select sal, decode(sign(sal-2000), -1, 'C', decode(sign(sal-3500), -1, 'B', 'A'))
  from emp;
```

-- 초과나 미만 되는 걸로 계산하는게 편하군! 만약에 이상, 이하면 0,1 / 0,-1 둘 다 포함해야하니까!

+ 이중디코드를 안 해도

sign(sal - 2000) + sign(sal - 3500)

= > 2 : 3500 초과

= > 0 : 2000~3500

= > -2 : 2000 미만

/ 대개 **decode** 중첩 권하지 않음

-- 5) **student** 테이블에서 전화번호 국번만 '#' 처리하여 출력하세요.

-- ex) 051) ###-1700

sol)

처음 잘못된 것 같은 접근!

```
select case instr(tel,'-')-instr(tel,'')-1 when 3 then replace(tel, substr(tel,instr(tel,'')+1,3), '###')
                                             when 4 then replace(tel, substr(tel,instr(tel,'')+1,4), '####')
        end as "#표시"
from student;
```

힌트

instr(tel,'-')-instr(tel,'')-1 = 3, 4

replace(tel, substr(**tel,instr(tel,'')+1**,instr(tel,'-')-instr(tel,'')-1), '###')

replace(tel, substr(**tel,instr(tel,'')+1**,instr(tel,'-')-instr(tel,'')-1), '####')

Q!!!!만일 국번이 뒤에도 같은 형태로 나타날 수 있음!

해결

```
select case instr(tel,'-')-instr(tel,'')-1 when 3 then substr(tel, 1, instr(tel, '-') ) || '###' || substr(tel,
instr(tel, '-') )
                                             when 4 then substr(tel, 1, instr(tel, '-') ) || '####' || substr(tel,
instr(tel, '-') )
        end as "#표기"
from student;
```

+(L/R)PAD함수로 푸는 방법!!!

LPAD('#', instr(tel,'-')-instr(tel,'-')-1, '#') 사용

```
select replace(tel,substr(tel, instr(tel,'')+1, instr(tel,'-')-instr(tel,'-')-1),LPAD('#', instr(tel,'-')-instr(tel,'-')-1, '#'))  
from student;
```

Q. 국번이 같은 경우로 나타날 경우? + 엄청 다양한 경우???

* instr(tel,'-')-instr(tel,'-')-1만 order by desc나 asc로 해서 범위를 볼 수 있겠지만.. 엄청 많으면 어떡하지?

-- 6) EMP 테이블을 이용하여 사원이름, 입사일 및 급여검토일을 표시합니다.

-- 급여검토일은 여섯달 근무후 해당되는 첫번째 월요일입니다.

-- 날짜는 "Sunday the Seventh of September, 1981" 형식으로 표시하도록 한다.

-- 열 이름은 check로 한다.(alter session set nls_date_language=american)

sol)

```
alter session set nls_date_language = american;
```

```
select ename,  
       hiredate,  
       to_char(next_day(add_months(hiredate,6),2), 'Day" the"Ddspth "of" Month, yyyy') as  
"CHECK"  
from emp;
```

Q 띄어쓰기 마구잡이 형태??? (January 등 긴 단어 중심으로 공백이 맞춰짐)

```
trim(to_char(hiredate,'Day'))||' the '||
```

+정규식 표현으로 공백 삭제 할 수 있음

180503 목

2) GROUP BY 절 - 그룹별로 단 하나의 데이터 / 이미 GROUPING 된 데이터로, 개별 DATA 못 찾음

기술순서	실행순서
SELECT 그룹함수 ([DISTINCT] {COLUMN,,,}	⑤
FROM 테이블명	①
[WHERE 조건]	②
[GROUP BY COLUMN1, COLUMN2, ...]	③
[HAVING 그룹조건]	④
[ORDER BY COLUMN1, COLUMN2, ...]	⑥

- **GROUP BY** vs **ORDER BY**

뒤에 **COLUMN** 순서 변경 시

결과 변경 **X**

결과 변경 **O**

- **WHERE** vs **HAVING**

개별 데이터 선택/제거

그룹 선택/제거

- " " (Alias) 사용되는 곳 - only in **SELECT**절, **ORDER BY** 절

GROUP BY + HAVING 중복 제거 in 현업

ex. job 중복 확인

```
select job, count(job)
  from emp
 group by job
 having count(job) >=2;
```

cf. 정렬 : 컴퓨터 부하되기 쉬움. 따라서 불필요한 경우는 사용하지 않는 것이 좋음.

-----학년별 키,몸무게 평균1-----

```
select avg(HEIGHT), avg(WEIGHT)
  from student
 group by grade;
```

출력된 행 = 그룹 수 = 4개, 그런데 어떤 grade가 결과값인지?

-----2-----

```
select grade, avg(HEIGHT), avg(WEIGHT)
  from student
group by grade;
```

+grade 정렬하고 싶으면 order by 사용

```
select grade, max(HEIGHT), avg(WEIGHT)
  from student
group by grade;
```

- 그룹 내 max/min값은 나타나나, 그 값을 가진 **사람 이름**은 찾을 수 없다.
group by로 쓴 순간, 그룹으로 묶여서 개인의 특성을 확인할 수 없게 되었음.

=> **group**으로 데이터 출력 + **개인**으로 데이터 출력 : 동시에 불가

※ **group by** 절에 표현(기술)되지 않은 **column**은 **select**에서 단독으로 쓸 수 없다.
group 형식의 함수와 함께 써야 한다.

-----1-----

```
select grade, name, max(HEIGHT), avg(WEIGHT)
  from student
group by grade;
```

-> 오류 발생

-----2-----

```
select grade, min(name), max(HEIGHT), avg(WEIGHT)
  from student
group by grade;
```

=> group by **한계** ; select 절에 표시 한계. -> **subquery**로 해결 가능

ex. **subquery** (추후에)

그룹에 대한 추가 정보를 쓰고 싶을 때.. ex. 제일 큰 사람 이름

=> SQL 구문 1개 만으로는 동시에 표현 불가능. => subquery 를 통해서!

----- 평균이 175 이상인 그룹만 출력하고 싶음-----

```
select grade, avg(HEIGHT)
  from student
 group by grade
having avg(height) >= 175;
```

-----..-----.

where절 사용시 오류 발생. where절에 group 관련 사용 불가능

----- 키가 180 이상인 학생들을 제외한 각 학년별 평균 키-----

```
select grade, avg(HEIGHT)
  from student
where height < 180
```

```
group by grade;
```

-----1,2학년의 키 평균-----

```
select grade, round(avg(height))
  from student
```

```
where grade in (1,2)
```

```
group by grade;
```

-- where 대신 having grade in (1,2) 써도 결과값 동일함,

※ **where, having** 둘 다 쓸 수 있는 경우

where 처리 후 group by 처리 / group by는 모든 데이터를 처리한 후 having에서 처리. (부하)

※ 둘 다 쓸 수 있을 때, **where절** 조건에서 거를 수 있으면 먼저 거르는 게 더 좋음.

cf. scott 계정에서 hr의 employees 테이블 가져오기

① 시작 - cmd

② sqlplus / as sysdba;

③ alter user scott account unlock;

alter user scott identified by oracle;

④ alter user hr account unlock;

alter user hr identified by oracle;

⑤ grant select on hr.employees to scott;

⑥ create public synonym employees for hr.employees;

◆ 실습문제 - emp테이블

1. 부서별로 직원수, 최대연봉, 최소연봉, 평균연봉을 구하고 급여평균으로 정렬하여라.

```
select deptno, count(empno), max(sal), min(sal), avg(sal)
  from EMP
 group by deptno
 order by 5;
```

2. 각 업무별로 직원수, 급여평균을 구하고 업무별로 정렬하라.

```
select job, count(empno), avg(sal)
  from EMP
 group by job
 order by job;
```

3. 업무별, 부서별로 그룹화하여 각각의 인원수와 급여평균을 구하고, 급여평균이 많은 순으로 정렬하여라.

```
select job, deptno, count(empno), avg(sal)
  from EMP
 group by job, deptno
 order by avg(sal) desc;
```

4. 부서 인원이 4명보다 많은 부서의 부서번호, 인원수, 급여의 합을 출력하여라

```
select deptno, count(empno), sum(sal)
  from EMP
 group by deptno
 having count(empno) >= 4;
```

5. 업무별 급여의 평균이 3000 이상인 업무에 대해 업무명, 평균급여, 급여의 합을 구하여라.

```
select job, avg(sal), sum(sal)
  from EMP
 group by job
 having avg(sal) >= 3000;
```

4. 집합연산자

- 두 개 이상의 집합에 대한 처리를 가능하게 하는 연산자
- 두 개 이상의 SQL 구문 사이에 쓸 수 있다
- 조건
 - . 두 개 사이에 자리해야 함
 - . 합치거나 비교하려는 집합 사이의 컬럼 개수가 일치해야 함 (양 집단의 크기 동일)
 - . **각 집단의 컬럼의 순서, 데이터 형식, 컬럼의 갯수가 똑같이 대응해야 함.**

/ 맞지 않으면 맞춰주기

위의 데이터 셋트에 아래 데이터 셋트가 들어감. (상하로 합쳐짐)

1) UNION, UNION ALL

- 합집합
 - UNION : **중복된 값 제거**된 단 하나의 value - distinct 처럼
 - UNION ALL : **중복된 값도 전체** 나타내줌
 - UNION : 중복된 값을 제거하기 위해 정렬을 수행
 - > **결과값이 같을 경우에는 UNION ALL 사용**
- / 전체 집단에 대해서 (교집합도, 각각의 중복 값 자체도)

2) INTERSECT

- 완전히 교집합. 두 집합에 공통으로 존재하는 데이터 출력

3) MINUS

- 위의 집합에서 아래 집합의 결과값을 제거한 나머지 출력.
- $A-B \neq B-A$, 따라서 **순서 중요!!**

-- emp랑 emp2를 합친다 생각 -> column 순서 뒤죽박죽, 다른 것들 많음 -> 못 합쳐짐

```
select EMPNO, ENAME, DEPTNO, SAL
  from emp
```

union all

```
select EMPNO, NAME, DEPTNO, PAY
  from emp2;
```

=> 에러발생

컬럼의 개수, 컬럼의 순서 맞추기

+ 컬럼의 데이터타입 같아야 하고, 컬럼의 의미도 같아야 함

조회가 안 될 경우

1. 컬럼의 사이즈가 다르거나

ex. empno가 같아도 서로 정의된 empno 개수가 다름

-> 큰 empno를 위로 올리거나, 위에 있는 empno를 확장시키기

확장 : `to_char(empno,'09999999')`

2. 데이터 타입이 다르거나

ex. deptno 숫자 / 문자

cf. 앞에 0이 있는 숫자는 '문자'임. 숫자에서 000043 경우는 존재 X

/ `to_char(empno,'09999999')` (문자형)에 대응하기 위해 emp2에 `to_char(empno)`

cf. `to_number(to_char(,))` 하면 다시 0 없어짐

```
-----
select to_char(EMPNO,'099999999'), ENAME, DEPTNO, SAL
  from emp
union all
select to_char(EMPNO), NAME, TO_NUMBER(DEPTNO), PAY
  from emp2;
```

+ job을 하고 싶다. 그런데 아래 emp2에는 job X, => 임의로 표현식을 통해 맞춰줌

```
-----
select to_char(EMPNO,'099999999'), ENAME, DEPTNO, SAL, job
  from emp
union all
select to_char(EMPNO), NAME, TO_NUMBER(DEPTNO), PAY, 'intern'
  from emp2;
```

+-----

서로 공유하지 않지만 꼭 넣고 싶은 것이 있을 때, 임의로 표현식을 통해 만들어주면 됨