

chapter02

기본 select문 과 단일행 함수

1. **SELECT / FROM 절**
2. WHERE 절
3. ORDER BY 절
4. 단일행 함수

01 SELECT / FROM 절

■ 모든 컬럼들 조회하기

```
select * from employees;
```

```
select * from departments;
```

- 마지막은 ; 로 표시
- 대소문자를 구분하지 않음
- select 문
 - select 절
 - from 절

01 SELECT / FROM 절

■ 원하는 컬럼만 조회하기(원하는 컬럼 이름을 지정)

```
select employee_id, first_name, last_name  
from employees;
```

[예제]

- 사원의 이름(first_name)과 전화번호, 입사일, 연봉을 출력하세요
- 사원의 이름(first_name)과 성(last_name), 급여, 전화번호, 이메일, 입사일을 출력하세요
→ 컬럼명이 많을때 표기팁

■ 출력할 때 컬럼에 별명 사용하기

```
select employee_id as empNO, first_name "E-name", salary "연 봉"  
from employees;
```

- 컬럼 이름을 변경하여 출력(원 컬럼이름은 변경되지 않으며 임시로 보여주는 형태)
- 컬럼명 뒤에 공백을 주고 별명을 쓰는 방식 → 대문자로 표기됨
- 컬럼명 뒤에 as 키워드 사용 후 별명을 쓰는 방식 → 대문자로 표기됨
- 별명에 **대소문자** 구분, **공백**, **특수문자** 등이 필요한 경우 " "(쌍 따옴표)로 감싸줌

[예제]

- 사원의 **이름**(first_name)과 **전화번호** **입사일** **급여** 로 표시되도록 출력하세요
- 사원의 **사원번호** **이름**(first_name) **성**(last_name) **급여** **전화번호** **이메일** **입사일**로 표시되도록 출력하세요

■ 연결 연산자(Concatenation)로 컬럼들 붙이기

```
select first_name, last_name  
from employees;
```

```
select first_name || last_name  
from employees;
```

```
select first_name || ' ' || last_name  
from employees;
```

```
select first_name || ' hire date is ' || hire_date  
from employees;
```

- ' '(따옴표)를 사용하여 문자를 추가하여 출력할 수 있다.

01 SELECT / FROM 절

■ 산술 연산자 사용하기

```
select first_name, salary  
from employees;
```

```
select first_name, salary, salary*12  
from employees;
```

```
select first_name, salary, salary*12, (salary+300)*12  
from employees;
```

• +, -, *, / 을 사용할 수 있음

[예제]

▪ 다음을 실행해 보고
오류를 분석해 보세요

```
select job_id*12  
from employees;
```

[예제]

- 전체직원의 정보를 다음과 같이 출력하세요

성명(first_name last_name)

성과 이름사이에 - 로 구분 ex) William-Gietz

급여

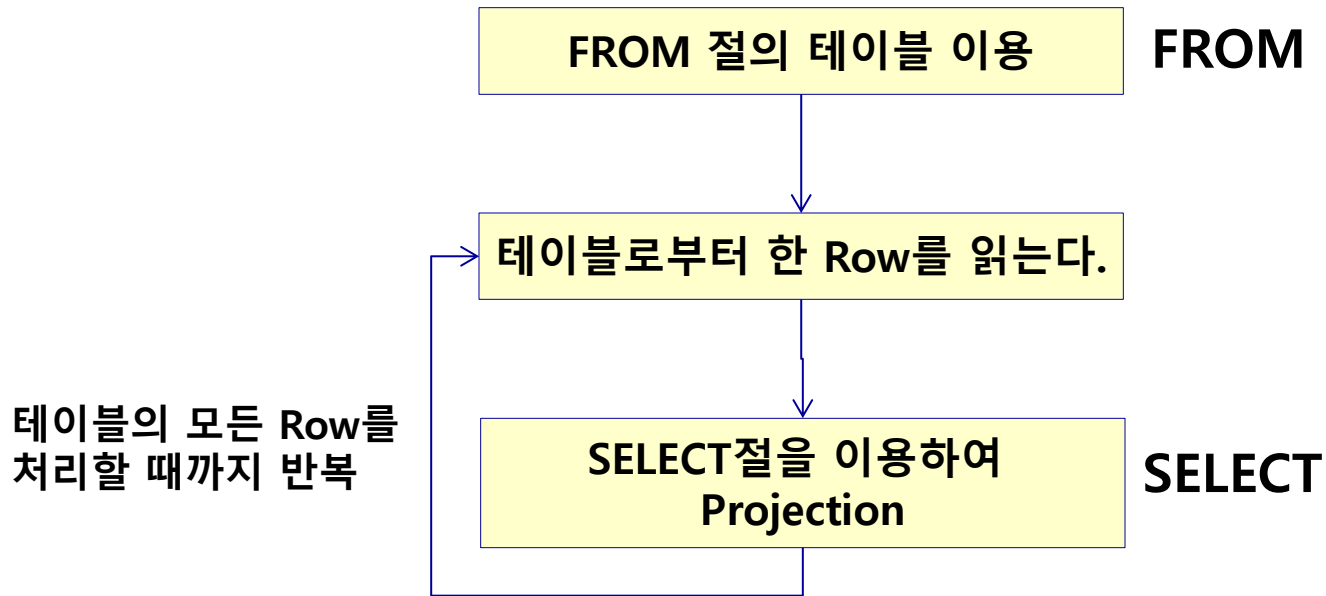
연봉(급여*12)

연봉2(급여*12+5000)

전화번호

01 SELECT / FROM 절

■ select from 절 처리방법



chapter02

기본 select문 과 단일행 함수

1. SELECT / FROM 절
2. **WHERE** 절
3. ORDER BY 절
4. 단일행 함수

■ 비교연산자

```
select first_name  
from employees  
where department_id = 10;
```

• 부서번호가 10인 사원의 이름을 구하시오

• =, !=, >, <, >=, <= 연산자 사용

[예제]

- 연봉이 15000 이상인 직원들의 이름과 연봉을 출력하세요
- 07/01/01 일 이후에 입사한 직원들의 이름과 입사일을 출력하세요
- 이름이 Lex인 직원의 연봉을 출력하세요

→ 문자, 날짜는 ' ' 로 감싸줌 ' ' 는 대소문자를 구분한다

■ 조건이 2개이상 일때 한꺼번에 조회하기

```
select first_name, salary
from employees
where salary >= 14000
and salary <= 17000;
```

- 연봉이 14000 이상 17000이하인 사원의 이름과 연봉을 구하시오

[예제]

- 연봉이 14000 이하이거나 17000 이상인 사원의 이름과 연봉을 출력하세요
- 입사일이 04/01/01 에서 05/12/31 사이의 사원의 이름과 입사일을 출력하세요

■ BETWEEN 연산자로 특정구간 값 출력하기

```
select first_name, salary
from employees
where salary between 14000 and 17000;
```

• 연봉이 14000 이상 17000이하인 사원의
이름과 연봉을 구하시오

```
select first_name, salary
from employees
where salary >= 14000
and salary <= 17000;
```

• 연봉이 14000 이상 17000이하인 사원의
이름과 연봉을 구하시오

- 작은 값을 앞에, 큰 값을 뒤에
- 두 값을 모두 포함하는 결과를 출력(경계 값을 포함하지 않는 경우 사용하면 안됨)
- 느린 연산자에 속함

■ IN 연산자로 여러 조건을 검사하기

```
select first_name, last_name, salary
from employees
where first_name in ('Neena', 'Lex', 'John');
```

[예제]

- 연봉이 2100, 3100, 4100, 5100 인 사원의 이름과 연봉을 구하시오

■ Like 연산자로 비슷한것들 모두 찾기

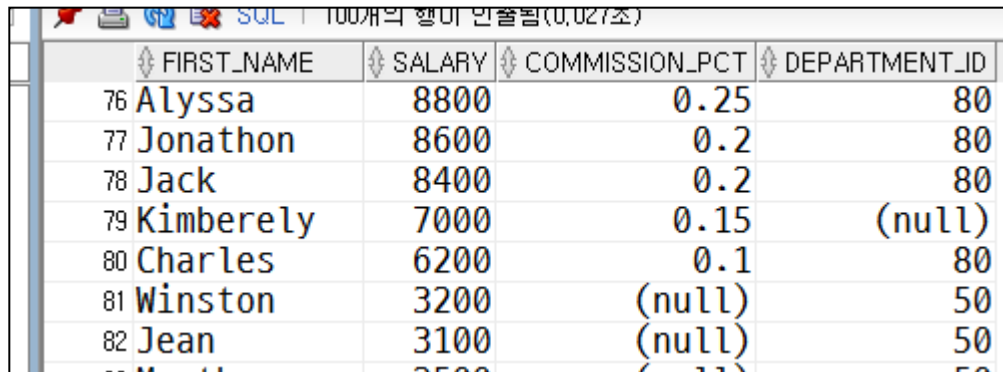
```
select first_name, last_name, salary
from employees
where first_name like 'L%';
```

- % : 임의의 길이의 문자열(공백 문자 가능)
- _ : 한글자 길이

[예제]

- 이름에 am 을 포함한 사원의 이름과 연봉을 출력하세요
- 이름의 두번째 글자가 a 인 사원의 이름과 연봉을 출력하세요
- 이름의 네번째 글자가 a 인 사원의 이름을 출력하세요
- 이름이 4글자인 사원중 끝에서 두번째 글자가 a인 사원의 이름을 출력하세요

■ NULL



100개의 행이 반환됨(0.027초)

	FIRST_NAME	SALARY	COMMISSION_PCT	DEPARTMENT_ID
76	Alyssa	8800	0.25	80
77	Jonathon	8600	0.2	80
78	Jack	8400	0.2	80
79	Kimberely	7000	0.15	(null)
80	Charles	6200	0.1	80
81	Winston	3200	(null)	50
82	Jean	3100	(null)	50



- 아무런 값도 정해지지 않았음을 의미 (0이 아님)
- 어떠한 데이터타입에도 사용가능
- not null 이나 primary key 속성에는 사용할 수 없음
- null을 포함한 산술식은 null

```
select first_name, salary, commission_pct, salary*commission_pct
from employees
where salary between 13000 and 15000
```

■ is null / is not null

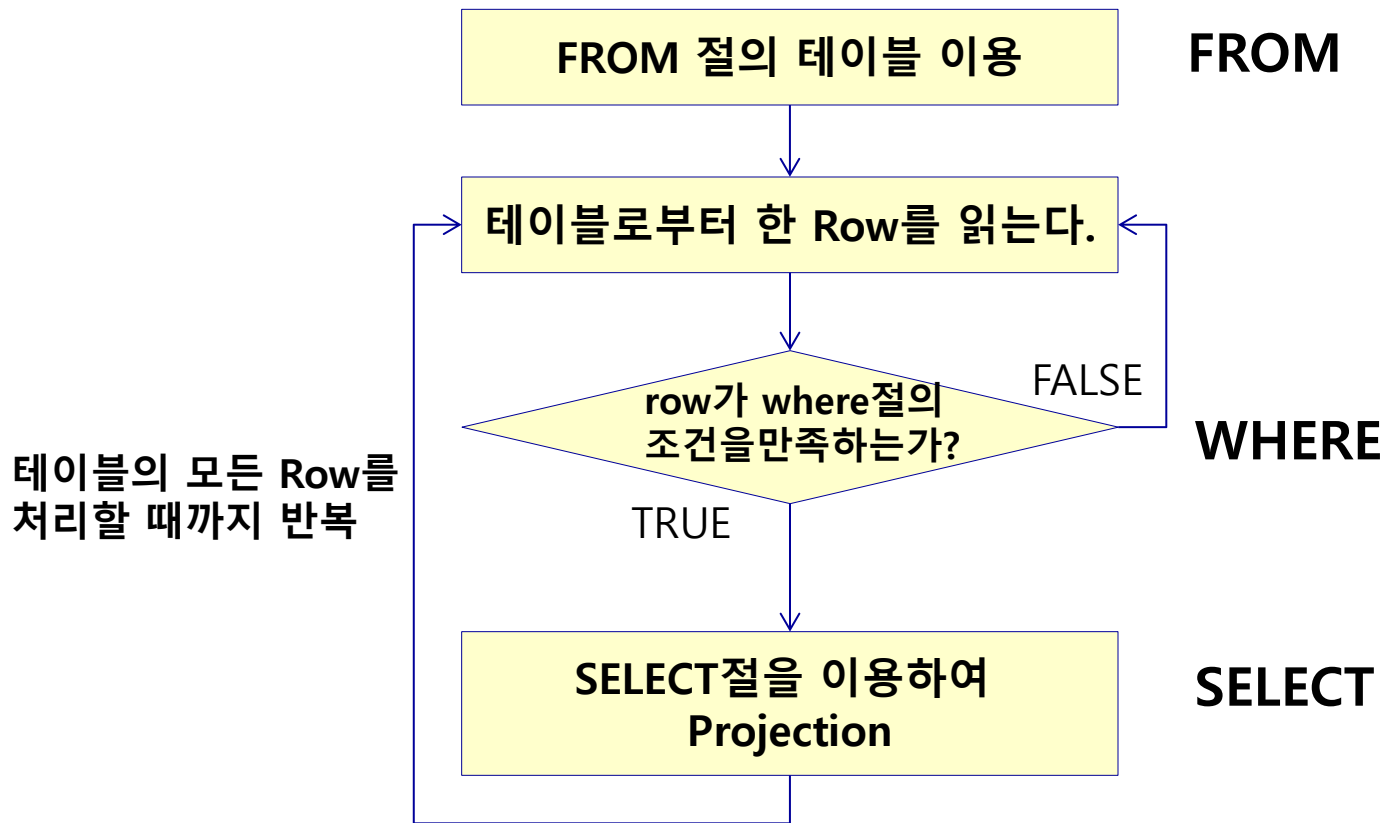
```
select first_name, salary, commission_pct  
from employees  
where commission_pct is null;
```

where commision_pct = null 이 아님

[예제]

- 커미션비율이 있는 사원의 이름과 연봉 커미션비율을 출력하세요
- 담당매니저가 없고 커미션비율이 없는 직원의 이름을 출력하세요

■ select from where 절 처리방법



chapter02

기본 select문 과 단일행 함수

1. SELECT / FROM 절
2. WHERE 절
3. **ORDER BY** 절
4. 단일행 함수

■ order by 절을 사용해 보기 좋게 정렬하기

```
select first_name, salary
from employees
order by salary desc;
```

```
select first_name, salary
from employees
where salary >= 9000
order by salary desc;
```

- 기본값 오름차순(작은거→큰거)

한글: 가, 나, 다, 라 ...

영어: A, B, C, D ...

숫자: 1, 2, 3, 4 ...

날짜: 예전날짜에서 최근날짜로 정렬

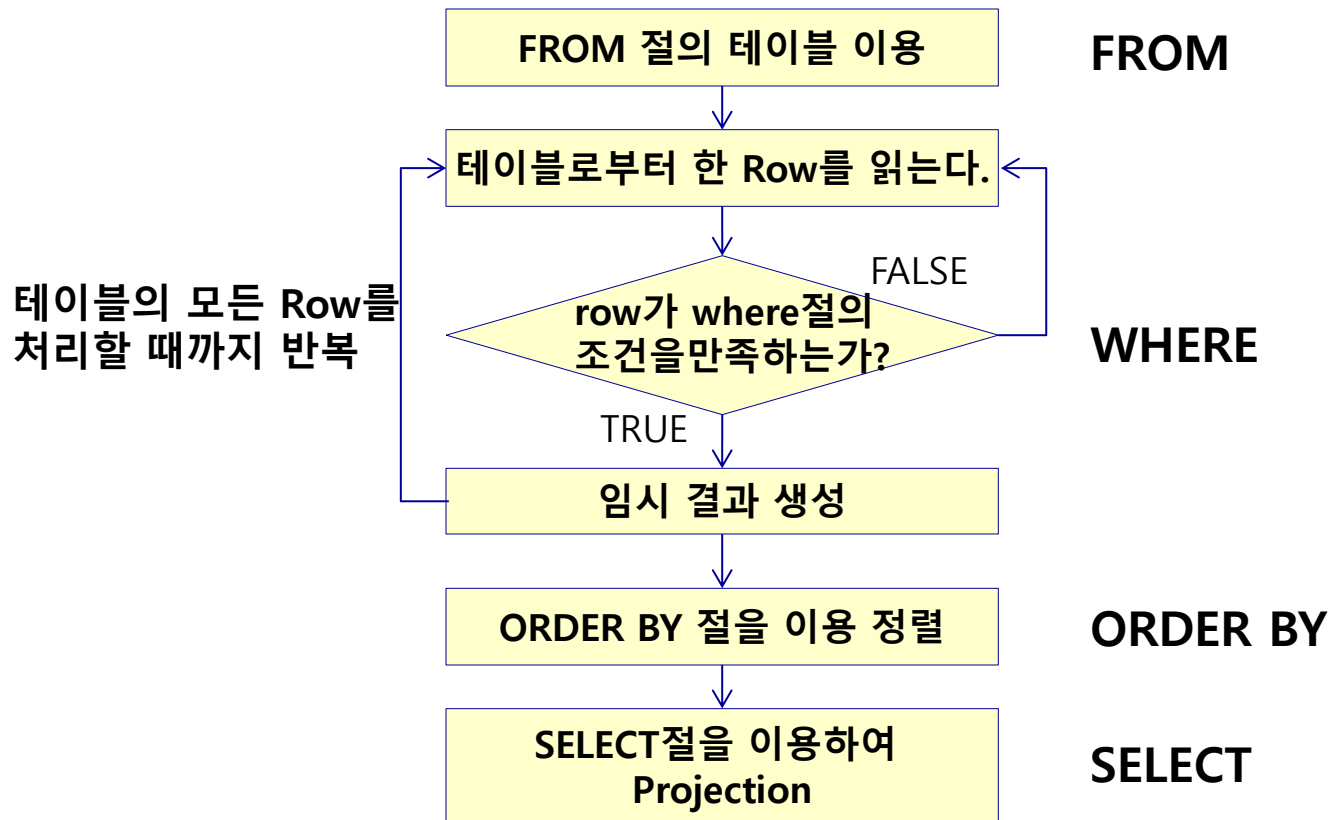
↔ 내림차순은 반대

- 정렬조건이 복수일때는 , 로 구분하여 나열한다.

[예제]

- 부서번호를 오름차순으로 정렬하고 부서번호, 급여, 이름을 출력하세요
- 급여가 1000 이상인 직원의 이름 급여를 급여가 큰직원부터 출력하세요
- 부서번호를 오름차순으로 정렬하고 부서번호가 같으면 급여가 높은 사람부터 부서번호 급여 이름을 출력하세요

■ select from where order by 절 처리방법



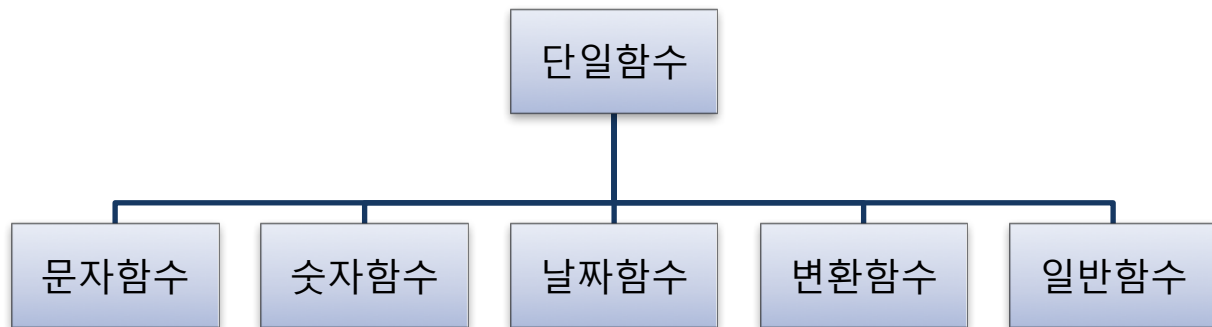
chapter02

기본 select문 과 단일행 함수

1. SELECT / FROM 절
2. WHERE 절
3. ORDER BY 절
4. **단일행 함수**

■ 단일행 함수의 특징

- 단일행 함수 : 각각의 데이터를 한건씩 처리
- 복수행 함수 : 여러건의 데이터를 한꺼번에 처리 후 1개의 결과로 처리
그룹함수, 집계함수 라고도 불림



■ 문자함수의 종류

Function	설명
CONCAT(s1,s2)	문자열 결합
INITCAP(s)	첫글자만 대문자로 변경
LOWER(s)	소문자로 변경
UPPER(s)	대문자로 변경
LPAD(s1,n,s2)	문자열의 왼쪽 채움 (길이:n, 채움문자 s2)
RPAD(s1,n,s2)	문자열 오른쪽 채움 (길이:n, 채움문자 s2)
LTRIM(s,c)	문자열 왼쪽 c문자열 제거
RTRIM(s,c)	문자열 오른쪽 c문자열 제거
CHR(n)	ASCII값이 n인 문자 반환
REPLACE(s,p,r)	문자열 치환, S속의 p문자열을 r로 치환
SUBSTR(s,m,n)	부분 문자열, m번째부터 길이 n인 문자열 반환
TRANSLATE(s,from,to)	s에서 from 문자열의 각 문자를 to문자열의 각 문자로 변환
ASCII(s)	ASCII값 반환
INSTR(s1,s2,m,n)	문자열 검색, s1의 m번째부터 s2 문자열이 나타나는 n번째 위치 반환
LENGTH(s)	문자열 길이 반환

■ 문자함수 – INITCAP(컬럼명)

- 영어의 첫 글자만 대문자로 출력하고 나머지는 전부 소문자로 출력하는 함수

```
select email, initcap(email), department_id
from employees
where department_id = 100;
```

■ 문자함수 – LOWER(컬럼명) / UPPER(컬럼명)

- 입력되는 값을 전부 소문자/대문자로 변경하는 함수

```
select first_name, lower(first_name), upper(first_name)
from employees
where department_id = 100;
```

■ 문자함수 – SUBSTR(컬럼명, 시작위치, 글자수)

- 주어진 문자열에서 특정길이의 문자열을 구하는 함수

```
select first_name, substr(first_name,1,3), substr(first_name,-3,2)
from employees
where department_id = 100;
```

- 양수인 경우 왼쪽→오른쪽으로 검색해서 글자수 만큼 추출
- 음수인 경우 오른쪽→왼쪽 검색을 한 후 왼쪽→오른쪽으로 글자수 만큼 추출

■ 문자함수 – LPAD(컬럼명, 자리수, '채울문자') / RPAD(컬럼명, 자리수, '채울문자')

- LPAD() :왼쪽 공백에 특별한 문자로 채우기
- RPAD() :오른쪽 공백에 특별한 문자로 채우기

```
select first_name,  
       lpad(first_name,10,'*'),  
       rpad(first_name,10,'*')  
from employees;
```

■ 문자함수 – REPLACE (컬럼명, 문자1, 문자2)

- 컬럼명에서 문자1을 문자2로 바꾸는 함수

```
select first_name,  
       replace(first_name, 'a', '*')  
from employees  
where department_id =100;
```

```
select first_name,  
       replace(first_name, 'a', '*'),  
       replace(first_name, substr(first_name, 2, 3), '***')  
from employees  
where department_id =100;
```

■ 숫자함수의 종류

Function	설명	Example	Result
ABS(n)	절대값	ABS(-5)	5
CEIL(n)	n 보다 크거나 같은 최소 정수	CEIL(-2.4)	-2
FLOOR(n)	n 보다 작거나 같은 최대 정수	FLOOR(-2.4)	-3
MOD(m,n)	나머지	MOD(13,2)	1
POWER(m,n)	m의 n승	POWER(2,3)	8
ROUND(m,n)	소수점아래 n자리까지 반올림	ROUND(4.567,2)	4.57
TRUNC(m,n)	소수점아래 n자리미만 버림	TRUNC(4.567,2)	4.56
SIGN(n)	부호 (1, 0, -1)	SIGN(-10)	-1

■ 숫자함수 – ROUND(숫자, 출력을 원하는 자리수)

- 주어진 숫자의 반올림을 하는 함수

```
select round(123.346, 2) "r2",  
       round(123.456, 0) "r0",  
       round(123.456, -1) "r-1"  
from dual;
```

■ 숫자함수 – TRUNC(숫자, 출력을 원하는 자리수)

- 주어진 숫자의 버림을 하는 함수

```
select trunc(123.346, 2) "r2",  
       trunc(123.456, 0) "r0",  
       trunc(123.456, -1) "r-1"  
from dual;
```

■ 날짜함수의 종류

Function	Purpose
ADD_MONTHS(d,n)	d날짜에 n달 더함
LAST_DAY(d)	d의 달의 마지막 날
MONTHS_BETWEEN(d1,d2)	d1, d2사이의 달 수
NEW_TIME(d,z1,z2)	z1타임존의 d에서 z2타임존의 날짜 생성
NEXT_DAY(d,day)	d날 후의 첫 day요일의 날짜
ROUND(d,fmt)	fmt에 따른 날짜반올림
TRUNC(d,fmt)	fmt에 따른 날짜내림
SYSDATE	현재 날짜 시간 반환

■ 날짜함수의 종류

FUNCTION	RESULT
MONTHS_BETWEEN ('01-SEP-95','11-JAN-94')	19.677419
ADD_MONTHS ('11-JAN-94',6)	'11-JUL-94'
NEXT_DAY ('01-SEP-95','FRIDAY')	'08-SEP-95'
LAST_DAY('01-FEB-95')	'28-FEB-95'

현재날짜를 '25-JUL-95'가정	
ROUND(SYSDATE,'MONTH')	01-AUG-95
ROUND(SYSDATE , 'YEAR')	01-Jan-96
TRUNC(SYSDATE , 'MONTH')	01-JUL-95
TRUNC(SYSDATE , 'YEAR')	01-JAN-95

■ 날짜함수 – SYSDATE()

- 현재 날짜와 시간을 출력해주는 함수

```
select sysdate  
from dual;
```

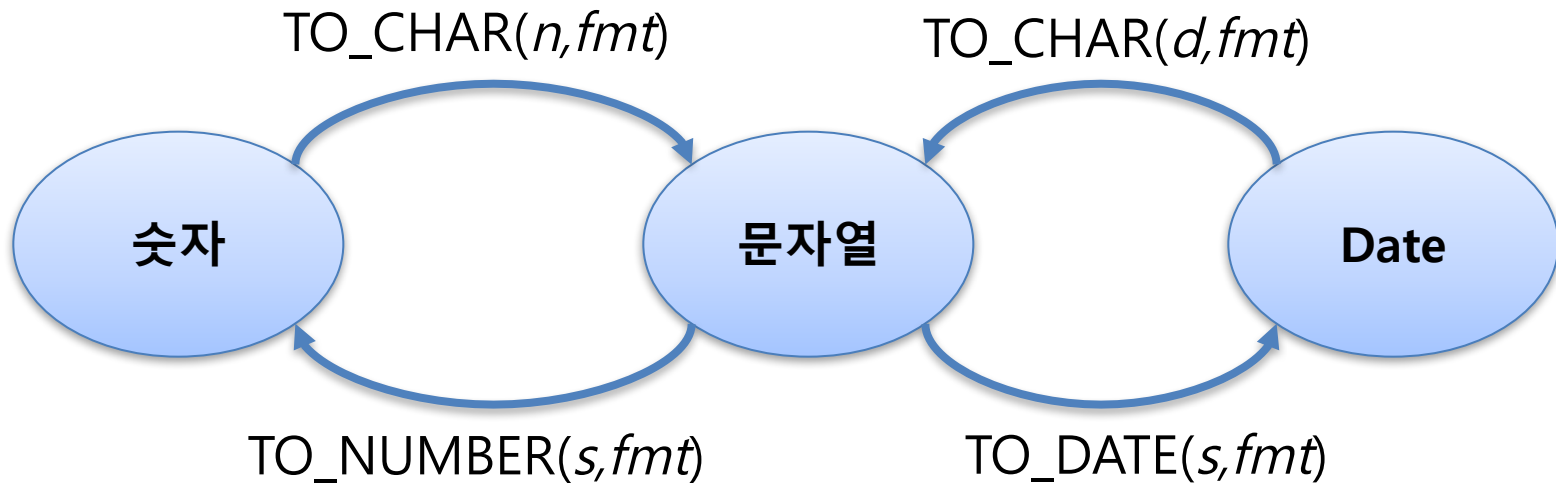
```
select sysdate  
from employees;
```

■ 단일함수 > 날짜함수 – MONTH_BETWEEN(d1, d2)

- d1 날짜와 d2 날짜의 개월수를 출력하는 함수

```
select months_between(sysdate, hire_date)  
from employees  
where department_id = 110;
```

■ 변환함수의 종류



■ TO_CHAR(숫자, '출력모양') 숫자형→문자형으로 변환하기

종류	의미	사용예	결과
9	9의 개수 자리수 만큼표시 (5자리까지표시)	to_char(9876, '99999')	9876
0	빈자리를 0으로 채우기	to_char(9876, '099999')	009876
\$	\$ 표시를 붙여서 표시	to_char(9876, '\$9999')	\$9876
.	소수점 이하를 표시	to_char(9876, '9999.99')	9876.00
,	천 단위 구분기호를 표시	to_char(9876, '99,999')	9,876

```
select first_name, to_char(salary*12, '$999,999.99') "SAL"  
from employees  
where department_id =110;
```

■ 단일함수>변환함수>TO_CHAR(날짜, '출력모양') 날짜→문자형으로 변환하

출력모양	의미	결과
'YYYY'	연도를 4자리로 표현합니다.	2017
'YY'	연도를 2자리로 표현합니다.	17
'MM'	월을 숫자2자리로 표현합니다.	07
'MON'	유닉스에서는 월을 뜻하는 영어 3글자를 표현합니다. 윈도우는 'MONTH'와 동일 합니다.	JUL 7월
'MONTH'	월을 뜻하는 이름 전체를 표현합니다.	7월
'DD'	일을 숫자 2자리로 표현합니다.	15
'DAY'	유닉스에서는 요일을 영문으로 표현 합니다. 윈도우에서는 요일을 한글로 표현 합니다.	SUN 토요일
'DDTH'	몇번째 날인지 표현합니다.	15TH
'HH24'	하루를 24시간으로 표현합니다.	19
'HH'	하루를 12시간으로 표현합니다.	07
'MI'	분으로 표현합니다.	56
'SS'	초로 표현합니다.	41

■ 변환함수>TO_CHAR(날짜, '출력모양') 날짜→문자형으로 변환하기

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

원하는 문자는 "년" 형태로
쌍따옴표로 추가 가능

■ 일반함수 > NVL(컬럼명, null일때값)/NVL2(컬럼명, null아닐때값, null일때 값)

- NVL(조사할 컬럼, NULL 일 경우 치환할 값)
- NVL2(조사할 컬럼, NULL이 아닐때 치환할 값, NULL일때 치환할 값)

```
select commission_pct, nvl(commission_pct, 0)
from employees;
```

평균같이 계산에 참여해야할 경우
null을 0으로 변경해준다

```
select commission_pct, nvl2(commission_pct, 100, 0)
from employees;
```

null이 아닐때