

※ select (해석 순서 1 → 2 → 3 → 4 → 5)

```
select [distinct] [컬럼1, 컬럼2,...][as 별명][ || 연산자][*] --- 6
from 테이블명 --- 1
[where 조건절] --- 2
[group by 컬럼명] --- 3
[having 조건절] --- 4
[order by 컬럼명 asc|desc ] --- 5
```

group by : 그룹함수(max, min, sum, avg, count..)와 같이 사용

having : 묶어놓은 그룹의 조건절

ex1) 사원테이블에서 급여의 평균을 구하시오

조건) 소수 이하는 절삭, 세자리 마다 콤마(,) 표시

사원급여평균
6,461

```
select to_char(trunc(avg(salary),0), '99,999') as 사원급여평균 from employees;
```

ex2) 부서별 급여평균을 구하시오

조건1) 소수 이하는 반올림

조건2) 세자리 마다 콤마, 화폐 단위(₩)로 표시

조건3) 부서별로 오름차순 정렬하시오

조건4) 평균급여가 5000이상인 부서만 표시하시오

부서코드	평균급여
10	₩8,600

```
select department_id as 부서코드,
       to_char(round(avg(salary), 0), 'L99,999,999') as 평균급여
from employees
group by department_id
having avg(salary) >= 5000
order by department_id asc;
```

ex3) 부서별 급여평균을 구해서 사원명(last\_name), 부서별 급여평균을 출력하시오 - X

```
select last_name, avg(salary)
from employees
group by department_id;
```

★ last\_name 때문에 error - group by절에 없는 것을 select에 조회하면 error

[문제1] job\_id별 급여의 합계를 구해서 job\_id, 급여합계를 출력하시오

ex4) 비효율적인 having절

10과 20부서에서 최대급여를 받는 사람의 최대급여를 구하여 정렬하시오

department_id	max_salary
10	4400
20	13000

```
select department_id, max(salary) as max_salary
from employees
group by department_id
having department_id in(10,20)
order by department_id;
```

전체 부서에 대해 그룹을 잡아서 최대급여를 구한 후에 부서가 10과 20인 것을 추려내기 때문인 것

```
select department_id, max(salary) as max_salary
from employees
where department_id in(10, 20)
group by department_id
order by department_id;
```

부서번호가 10과 20인 것만 골라내서 그룹잡기 때문에 속도가 좀 빠르다

## 조인(join)

EMPLOYEES			DEPARTMENTS			JOIN		
사 원 이 름	부 서 번 호	+	부 서 번 호	부 서 이 름	=	사 원 이 름	부 서 번 호	부 서 이 름

### ※ 종류

1. Inner join : 같은 것끼리만 연결
2. Outer join : 한쪽을 기준(모두포함)해서 연결
  - left join : 왼쪽 컬럼 모두포함
  - right join : 오른쪽 컬럼 모두포함
3. full join : 왼쪽, 오른쪽 모두 포함
4. self join : 자기 자신 테이블과 연결
5. cross join : 모든 경우의 수로 연결
6. non equijoin : 범위에 속하는지 여부를 확인
7. n개 테이블 조인 : 여러 개의 테이블 조인

### ※ 방법

1. 오라클 구문 전용
2. Ansi 표준 구문

ex5) inner join : 같은 것끼리만 조인

사원테이블과 부서테이블에서 부서가 같을 경우 사원번호, 부서번호, 부서이름을 출력하시오

방법1(오라클 전용 구문)

```
select employee_id,  
       employees.department_id,  
       department_name  
from employees, departments  
where employees.department_id = departments.department_id;
```

방법2(오라클 전용 구문)

```
select employee_id,  
       e.department_id,  
       department_name  
from employees e, departments d  
where e.department_id = d.department_id;
```

방법3(Ansi 표준)

```
select employee_id, department_id, department_name  
from employees  
join departments using(department_id);
```

[문제2] 부서테이블(DEPARTMENTS d)과 위치테이블(LOCATIONS l)을 연결하여  
부서가 위치한 도시를 알아내시오

department_id	city
10	Seattle

ex6) outer join(left) : 왼쪽 테이블은 모두 포함하여 조인

사원테이블과 부서테이블에서 부서번호가 같은 사원을 조인하시오 → 107레코드

방법1(오라클 전용 구문)

```
select e.last_name, d.department_id, d.department_name  
from employees e, departments d  
where e.department_id = d.department_id(+);
```

방법2(Ansi 표준)

```
select last_name, department_id, department_name  
from employees  
left join departments using(department_id);
```

ex7) outer join(right) : 오른쪽 테이블은 모두 포함하여 조인  
사원테이블과 부서테이블에서 부서번호가 같은 사원을 조인하시오 → 122 레코드

방법1(오라클 전용 구문)

```
select e.last_name, d.department_id, d.department_name
from employees e, departments d
where e.department_id(+) = d.department_id;
```

방법2(Ansi 표준)

```
select last_name, department_id, department_name
from employees right join departments using(department_id);
```

ex8) full join : 왼쪽, 오른쪽 테이블을 모두 포함하여 조인  
사원테이블과 부서테이블에서 부서번호가 같은 사원을 조인하시오 → 123레코드

--방법1(오라클 전용 구문) : 없다

--방법2(Ansi표준)

```
select last_name, department_id, department_name
from employees
full join departments using(department_id);
```

ex9) inner join : 두 개의 컬럼이 일치 하는 경우  
부서ID와 매니저ID가 같은 사원을 연결 하시오 → 32 레코드  
(관련 테이블 : departments, employees)  
last\_name      department\_id      manager\_id

방법1(오라클 전용 구문)

```
select e.last_name, d.department_id, d.manager_id
from employees e, departments d
where e.department_id=d.department_id and e.manager_id=d.manager_id;
```

방법2(Ansi표준)

```
select last_name, department_id, manager_id
from employees
inner join departments using(department_id, manager_id);
```

ex10) 내용은 같은데 컬럼명이 다른 경우에 조인으로 연결하기  
departments(location\_id), locations2(loc\_id)

테이블 복사

```
create table locations2 as select * from locations;
select * from locations2;
alter table locations2 rename column location_id to loc_id;
```

방법1(오라클 전용 구문)

```
select d.department_id, l.city
from departments d, locations2 l
where d.location_id=l.loc_id;
```

방법2(Ansi 표준)

```
select department_id, city
from departments
join locations2 on(location_id=loc_id);
```

방법3(Ansi 표준)

```
select department_id, city
from departments d
join locations2 l on(d.location_id=l.loc_id);
```

ex11) self 조인 : 자기 자신의 테이블과 조인하는 경우 사원과 관리자를 연결하시오

사원번호	사원이름	관리자
101	Kochhar	King

EMPLOYEES	EMPLOYEES
employee_id, last_name(사원이름)	last_name(관리자)
조건 employee_id = manager_id	

```
select employee_id, manager_id, last_name from employees; → e
select employee_id, last_name from employees; → m
```

방법1

```
select e.employee_id as 사원번호,
       e.last_name as 사원이름,
       m.last_name as 관리자
from employees e, employees m
where m.employee_id=e.manager_id;
```

방법2

```
select  e.employee_id as 사원번호,
        e.last_name as 사원이름,
        m.last_name as 관리자
from employees e
join employees m on(m.employee_id=e.manager_id);
```

ex12) cross join : 모든 행에 대해 가능한 모든 조합을 생성하는 조인

```
select * from countries, locations; → 575레코드
```

```
select * from countries cross join locations;
```

ex13) Non Equijoin (년 이큐조인)

컬럼값이 같은 경우가 아닌 범위에 속하는지 여부를 확인 할 때  
on ( 컬럼명 between 컬럼명1 and 컬럼명2)

```
create table salgrade(
salvel varchar2(2),
lowst number,
highst number);
```

```
insert into salgrade values('A', 20000, 29999);
insert into salgrade values('B', 10000, 19999);
insert into salgrade values('C', 0, 9999);
commit;
```

```
select * from salgrade;
```

```
select last_name, salary, salvel
from employees
join salgrade on(salary between lowst and highst)
order by salary desc;
```

ex14) n(여러)개의 테이블은 조인

업무ID 같은 직원들의 사원이름, 업무내용, 부서이름을 출력하시오  
(EMPLOYEES, JOBS, DEPARTMENTS 테이블을 조인)

<분석>

EMPLOYEES	JOBS	DEPARTMENTS
department_id	job_id	department_id
job_id		

<출력>

last\_name          job\_title          department\_name

```
select last_name, job_title, department_name
from employees
join departments using(department_id)
join jobs using(job_id);
```

[문제3] 위치ID, 부서ID를 연결해서 사원이름, 도시, 부서이름을 출력하시오

(관련 테이블 : EMPLOYEES, LOCATIONS2, DEPARTMENTS)

조건1 : 사원이름, 도시, 부서이름으로 제목을 표시하시오

조건2 : Seattle 또는 Oxford 에서 근무하는 사원

조건3 : 도시 순으로 오름차순 정렬하시오

사원이름	도	시	부서이름
Hall	Oxford		Sales

[문제4] 부서ID, 위치ID, 국가ID를 연결해서 다음과 같이 완성하시오

(관련 테이블 : EMPLOYEES, LOCATIONS2, DEPARTMENTS, COUNTRIES)

조건1 : 사원번호, 사원이름, 부서이름, 도시, 도시주소, 나라이름으로 제목을 표시하시오

조건2 : 도시주소에 Ch 또는 Sh 또는 Rd가 포함되어 있는 데이터만 표시하시오

조건3 : 나라이름, 도시별로 오름차순 정렬하시오

조건4 : 모든 사원을 포함한다