

[하위질의 (SubQuery)]

: 하나의 쿼리에 다른 쿼리가 포함되는 구조, ()로처리

1) 단일 행 서브쿼리(단일 행 반환) : > , < , >= , <= , <>(!=)

Main Query

Sub Query -----> 1개결과

2) 다중 행 서브쿼리(여러 행 반환) : in, any, all

Main Query

Sub Query -----> 여러 개의 결과

< any : 비교 대상 중 최대값보다 작음

> any : 비교 대상 중 최소값보다 큼

(ex. 과장직급의 급여를 받는 사원조회)

= any : in연산자와 동일

< all : 비교 대상 중 최솟값보다 작음

> all : 비교대상중 최대값보다 큼

(ex. 모든과장들의 직급보다 급여가 많은 사원조회)

3) 상관쿼리(correlated subquery)

: EXIST 연산자는 하위 쿼리에 레코드가 있는지 테스트하는 사용 된다

: EXIST 연산자는 하위 쿼리가 하나 이상의 레코드를 반환하면 true를 반환

: EXIST 연산자는 일반적으로 상관 관계가 있는 하위 쿼리와 함께 사용

: EXIST 연산자는 거의 * 로 구성된다

하위쿼리에 지정된 조건을 충족시키는 행이 있는지 없는지를 테스트하기 때문에 열 이름을 나열 할 의미가 없다

ex1) Neena 사원의 부서명을 알아내시오

```
select department_id from employees where first_name='Neena';  
select department_name from departments where department_id=90;
```

```
select department_name  
from departments  
where department_id = (select department_id from employees where first_name='Neena');
```

ex2) Neena 사원의 부서에서 Neena 직원보다 급여를 많이 받는 직원들의 last_name, department_id, salary 구하시오 (90, 17000)

```
select last_name, department_id, salary  
from employees  
where department_id = (select department_id from employees where first_name='Neena')  
and salary > (select salary from employees where first_name='Neena');
```

[문제1] 최저급여를 받는 직원들의 이름과 급여를 구하시오

[문제2] 부서별 급여 합계 중 최대급여를 받는 부서의 부서명과 급여합계를 구하시오

DEPARTMENT_NAME	SUM(SALARY)
-----	-----
Sales	304500

ex3) Austin과 같은 부서이면서 같은 급여를 받는 직원들의 이름, 부서명, 급여를 구하시오 (60부서, 4800달러)

```
select last_name, department_name, salary  
from employees  
left join departments using(department_id)  
where department_id = (select department_id from employees where last_name='Austin')  
and  
salary = (select salary from employees where last_name='Austin');
```

ex4) 'ST_MAN' 직급보다 급여가 많은 'IT_PROG' 직급의 last_name, job_id, salary 직원들을 조회하시오

```
select last_name, job_id, salary  
from employees  
where job_id = 'IT_PROG'  
and  
salary > any (select salary from employees where job_id='ST_MAN');
```

[문제3] 'IT_PROG' 직급 중 가장 많이 받는 사원의 급여보다 더 많은 급여를 받는 'FI_ACCOUNT' 또는 'SA_REP' 직급 직원들을 조회하시오

조건1) 급여 순으로 내림차순 정렬하시오

조건2) 급여는 세 자리마다 콤마(,) 찍고 화폐단위 '원'을 붙이시오

조건3) 타이틀은 직원명, 업무ID, 급여로 표시하시오

ex5) 'IT_PROG'와 같은 급여를 받는 직원들의 이름, 업무ID, 급여를 전부 구하시오

```
select last_name, job_id, salary
from employees
where salary in (select salary from employees where job_id='IT_PROG');
```

ex6) 전체직원에 대한 관리자와 직원을 구분하는 표시를 하시오

사원번호	이름	구분
100	King	관리자

★ 방법1 (in 연산자)

```
select employee_id as 사원번호,
       last_name as 이름,
       case
         when employee_id in (select manager_id from employees) then '관리자'
         else '직원'
       end as 구분
from employees
order by 3,1;
```

★ 방법2 (union, in, not in 연산자)

```
select employee_id as 사원번호, last_name as 이름, '관리자' as 구분
from employees
where employee_id in (select manager_id from employees)
union
select employee_id as 사원번호, last_name as 이름, '직원' as 구분
from employees
where employee_id not in (select manager_id from employees where manager_id is not null)
order by 3,1;
```

★ 방법3 (상관 쿼리 이용)

- 메인쿼리 한 행을 읽고 해당 값을 서브쿼리에서 참조하여 서브쿼리 결과에 존재하면 true를 반환

```
select employee_id as 사원번호, last_name as 이름, '관리자' as 구분
from employees e
where exists (select null from employees where e.employee_id=manager_id)
union
select employee_id as 사원번호, last_name as 이름, '직원' as 구분
from employees e
where not exists (select null from employees where e.employee_id=manager_id)
order by 3, 1;
```

[문제4] 자기 업무id(job_id)의 평균급여를 받는 직원들을 조회하시오

조건1) 평균급여는 100단위 이하 절삭하고 급여는 세자리마다 콤마, \$표시

조건2) 사원이름(last_name), 업무id(job_id), 직무(job_title), 급여(salary) 로 표시하시오

조건3) 급여순으로 오름차순 정렬하시오

ex7) group by rollup : a, b별 집계(Subtotal 구하기)

부서별, 직무ID별 급여평균구하기(동일부서에 대한 직무별 평균급여)

조건1) 반올림해서 소수 2째자리까지 구하시오

조건2) 제목은 Job_title, Department_name, Avg_sal로 표시하시오

```
select department_name, job_title, round(avg(salary), 2) as "Avg_sal"
from employees
join departments using(department_id)
join jobs using(job_id)
group by rollup(department_name, job_title);
```

ex8) group by cube : a별 집계 또는 b별 집계

부서별, 직무ID별 급여평균구하기(부서를 기준으로 나타내는 평균급여)

```
select department_name, job_title, round(avg(salary), 2) as "Avg_sal"
from employees
join departments using(department_id)
join jobs using(job_id)
group by cube(department_name, job_title);
```

ex9) group by grouping sets

직무별 평균급여와 전체사원의 평균급여를 함께 구하시오

```
select job_title, round(avg(salary), 2) as "Avg_sal"
from employees
join departments using(department_id)
join jobs using(job_id)
group by grouping sets((job_title), ( )); ← () All Rows의 역할
```