

chapter04

SubQuery rownum

1. **SubQuery**
2. rownum

■ SubQuery : 하나의 SQL 질의문 속에 다른 SQL 질의문이 포함되어 있는 형태

[예제]

- 'Den' 보다 급여를 많은 사람의 이름과 급여는?
 - 'Den' 의 급여?

```
select salary 11000
from employees
where first_name='Den'
```

급여를 많이 받는 사람의 이름과 급여?
→ 급여가 11000보다 많이받는...

```
select first_name, salary
from employees
where salary > ?? 11000
```

```
select employee_id, first_name, salary
from employees
where salary > (select salary
                 from employees
                 where first_name='Den');
```

■ 서브쿼리 작성시 주의사항

```
select employee_id, first_name, salary
from employees
where salary > (select salary
                 from employees
                 where first_name='Den');
```

11000

- SubQuery 부분은 where 절의 연산자 오른쪽에 위치해야 하며 괄호로 묶어야한다.
 - 가급적 order by 를 하지 않는다.
 - 단일행 SubQuery 와 다중행 SubQuery 에 따라 연산자를 잘 선택해야함
- 현재 예제는 단일행**

■ 단일행 SubQuery

- subQuery의 결과가 한 Row인 경우
- 연산자 : = , > , >= , < , <= , <> (같지않다)

```
select first_name, salary
from employees
where salary < (select avg(salary)
                from employees);
```

[예제]

- 급여를 가장 적게 받는 사람의 이름, 급여, 사원번호는?

```
select first_name, salary, employee_id
from employees
where salary = (select min(salary)
                from employees);
```

- 평균 급여보다 적게 받는 사람의 이름, 급여를 출력하세요?

■ 다중행 SubQuery

- subQuery의 결과가 여러 Row인 경우
- 연산자 : **ANY, ALL, IN ...**

```
select first_name, salary
from employees
where salary IN (select salary
                  from employees
                  where department_id = 110);
```

SALARY
12008
8300



```
select first_name, salary
from employees
where salary > (select salary
                 from employees
                 where department_id = 110);
```

SALARY
12008
8300



■ 다중행 SubQuery - IN

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

```
where salary IN (select salary  
                  from employees  
                  where department_id = 110);
```

SALARY
12008
8300

```
where salary = 12008  
or salary = 8300
```

	FIRST_NAME	SALARY
1	Shelley	12008
2	Nancy	12008
3	William	8300

01 SubQuery

■ 다중행 SubQuery - ANY (or)

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

```
where salary >ANY (select salary  
                    from employees  
                    where department_id = 110);
```

SALARY
12008
8300

```
where salary > 12008  
or salary > 8300
```

	FIRST_NAME	SALARY
1	Steven	24000
2	Neena	17000
3	Lex	17000
4	John	14000
5	Karen	13500
6	Michael	13000
7	Nancy	12008
8	Shelley	12008
9	Alberto	12000
10	Lisa	11500
11	Den	11000
12	Gerald	11000

21	Danielle	9500
22	David	9500
23	Patrick	9500
24	Peter	9000
25	Alexander	9000
26	Allan	9000
27	Daniel	9000
28	Alyssa	8800
29	Jonathon	8600
30	Jack	8400

■ 다중행 SubQuery - **ALL (and)**

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

```
where salary >ALL (select salary  
                   from employees  
                   where department_id = 110);
```

⚡ SALARY
12008
8300

```
where salary > 12008  
and salary > 8300
```

	⚡ FIRST_NAME	⚡ SALARY
1	Michael	13000
2	Karen	13500
3	John	14000
4	Lex	17000
5	Neena	17000
6	Steven	24000

[예제]

- 각 부서별로 최고급여를 받는 사원을 출력하세요

[예제]

- 각 부서별로 최고급여를 받는 사원을 출력하세요 -(조건절에서 비교)

```
select department_id, employee_id, first_name, salary
from employees
where (department_id, salary) in (select department_id, max(salary)
                                from employees
                                group by department_id)
```

employees 테이블

	DEPARTME...	EMPLOYEE_ID	FIRST_NAME	SALARY
1	10	200	Jennifer	4400
2	20	201	Michael	13000
3	20	202	Pat	6000
4	30	114	Den	11000
5	30	115	Alexander	3100
9	30	119	Karen	2500
10	40	203	Susan	6500
11	50	120	Matthew	8000
12	50	121	Adam	8200
13	50	122	David	7000

DEPARTME...	MAX(SALARY)
10	4400
20	13000
30	11000
40	6500
50	8200
60	9000
70	10000
80	14000
90	24000

[예제]

- 각 부서별로 최고급여를 받는 사원을 출력하세요 -(테이블에서 조인)

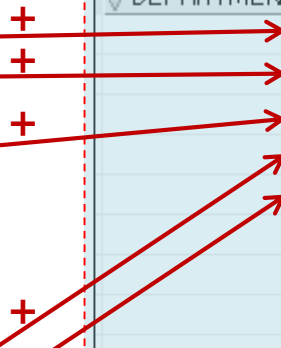
```
select e.department_id, e.employee_id, e.first_name, e.salary
from employees e, (select department_id, max(salary) salary
from employees
group by department_id) s
where e.department_id = s.department_id
and e.salary = s.salary;
```

employees e

	DEPARTMENT_ID	EMPLOYEE_ID	FIRST_NAME	SALARY
1	10	200	Jennifer	4400
2	20	201	Michael	13000
3	20	202	Pat	6000
4	30	114	Den	11000
5	30	115	Alexander	3100
6	30	116	Shelli	2900
7	30	117	Sigal	2800
8	30	118	Guy	2600

salary s

DEPARTMENT_ID	SALARY
10	4400
20	13000
30	11000
40	6500
50	8200
60	9000
70	10000
80	14000



employees e

	DEPARTME...	EMPLOYEE_ID	FIRST_NAME	SALARY
1	10	200	Jennifer	4400
2	20	201	Michael	13000
3	20	202	Pat	6000
4	30	114	Den	11000
5	30	115	Alexander	3100
6	30	116	Shelli	2900
7	30	117	Sigal	2800
8	30	118	Guy	2600
9	30	119	Karen	2500
10	40	203	Susan	6500
11	50	120	Matthew	8000
12				

salary s

DEPARTMENT_ID	SALARY
10	4400
20	13000
30	11000
40	6500
50	8200
60	9000
70	10000
80	14000
90	24000
100	12008
110	12008
	000

DEPARTMENT_ID	EMPLOYEE_ID	FIRST_NAME	SALARY	DEPARTMENT_ID_1	SALARY_1
10	200	Jennifer	4400	10	4400
20	201	Michael	13000	20	13000
30	114	Den	11000	30	11000
40	203	Susan	6500	40	6500
50	121	Adam	8200	50	8200
60	103	Alexander	9000	60	9000
70	204	Hermann	10000	70	10000
80	145	John	14000	80	14000
90	100	Steven	24000	90	24000
100	108	Nancy	12008	100	12008

chapter04

SubQuery rownum

1. SubQuery
2. **rownum**

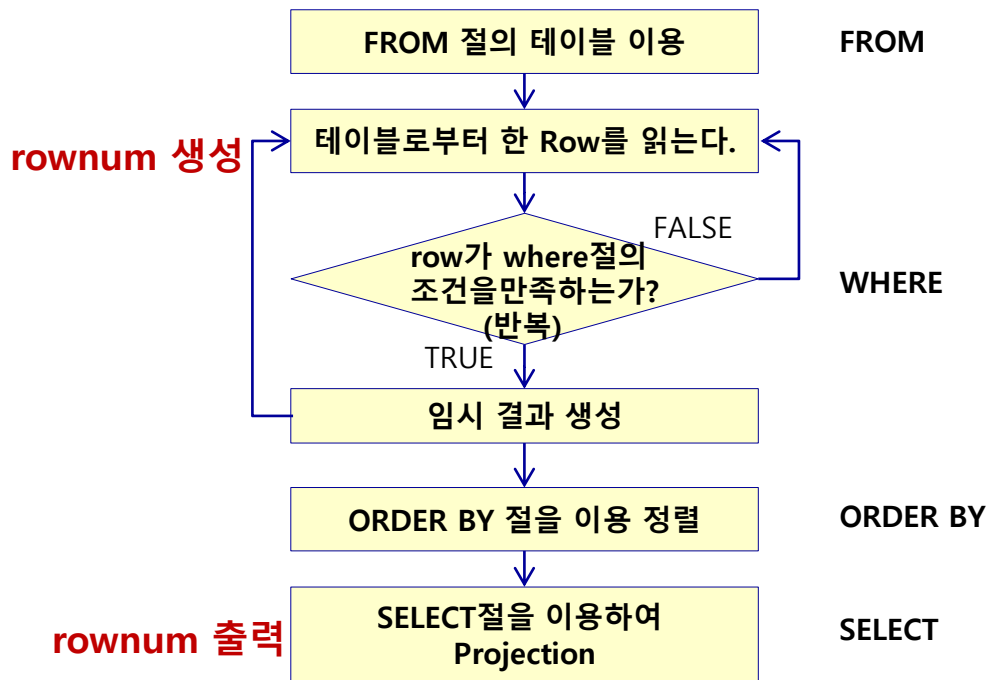
■ rownum : 질의의 결과에 가상으로 부여되는 Oracle의 가상(Pseudo)의 Column (일련번호)

[예제]

- 급여를 가장 많이 받는 3명의 직원의 이름을 출력하시오.

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

ROWNUM	FIRST_NAME	SALARY
1	Steven	24000
2	Neena	17000
3	Lex	17000
4	Alexander	9000
5	Bruce	6000
6	David	4800
7	Valli	4800
8	Diana	4200
9	Nancy	12008
10	Daniel	9000
11	John	8200
12	Ismael	7700



[예제]

- 급여를 가장 많이 받는 3명의 직원의 이름을 출력하시오.

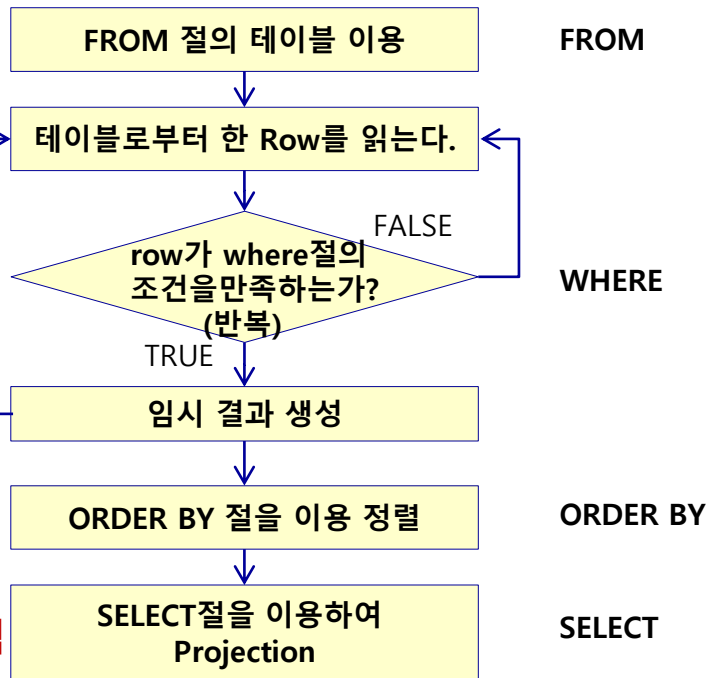
select **rownum**,
first_name,
salary
from employees
order by salary desc;

ROWNUM	FIRST_NAME	SALARY
1	Steven	24000
2	Neena	17000
3	Lex	17000
46	John	14000
47	Karen	13500
102	Michael	13000
9	Nancy	12008
106	Shelley	12008
48	Alberto	12000
69	Lisa	11500
15	Den	11000

rownum 생성

정렬시 rownum
이 섞인다.

rownum 출력

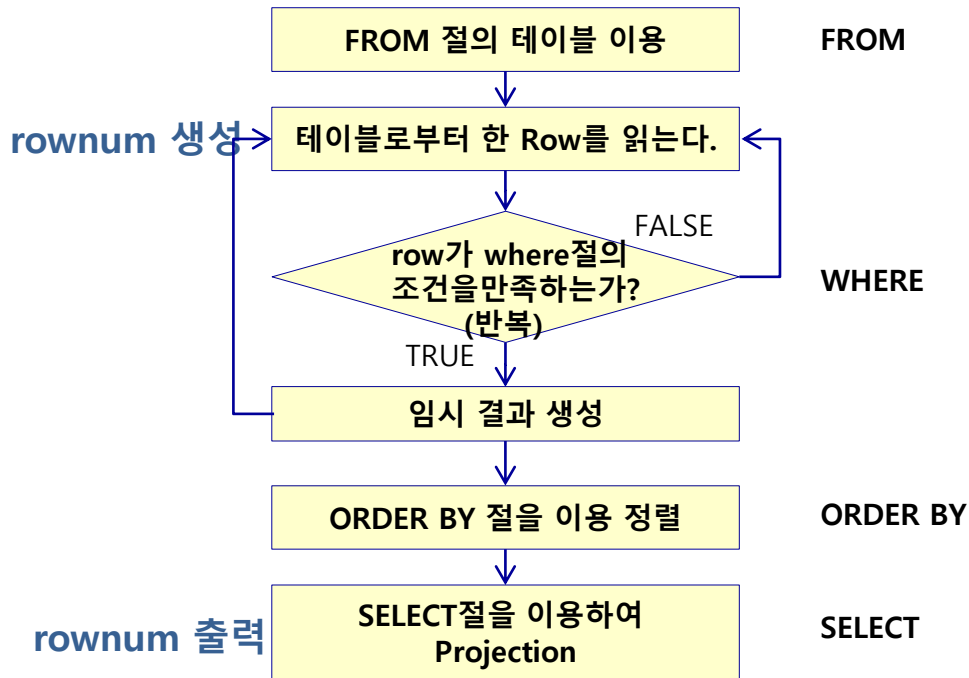


[예제]

- 급여를 가장 많이 받는 3명의 직원의 이름을 출력하시오.


```
select rownum,
       first_name,
       salary
from (
  select first_name,
         salary
  from employees
  order by salary desc
) ;
```

ROWNUM	FIRST_NAME	SALARY
1	Steven	24000
2	Neena	17000
3	Lex	17000
4	John	14000
5	Karen	13500
6	Michael	13000



[예제]

- 급여를 가장 많이 받는 3명의 직원의 이름을 출력하시오.



```
select rownum,
       first_name,
       salary
from (
  select first_name,
         salary
  from employees
  order by salary desc
)
where rownum >=3 ;
```

rownum = 1
조건을 만족하는
row가 없음

rownum 값을 생성후에
where절을 실행한다.

FROM 절의 테이블 이용

FROM

테이블로부터 한 Row를 읽는다.

row가 where절의
조건을 만족하는가?
(반복)

FALSE

WHERE

TRUE

임시 결과 생성

ORDER BY 절을 이용 정렬

ORDER BY

SELECT절을 이용하여
Projection

SELECT

ROWNUM	FIRST_N...	SALARY
--------	------------	--------

[예제]

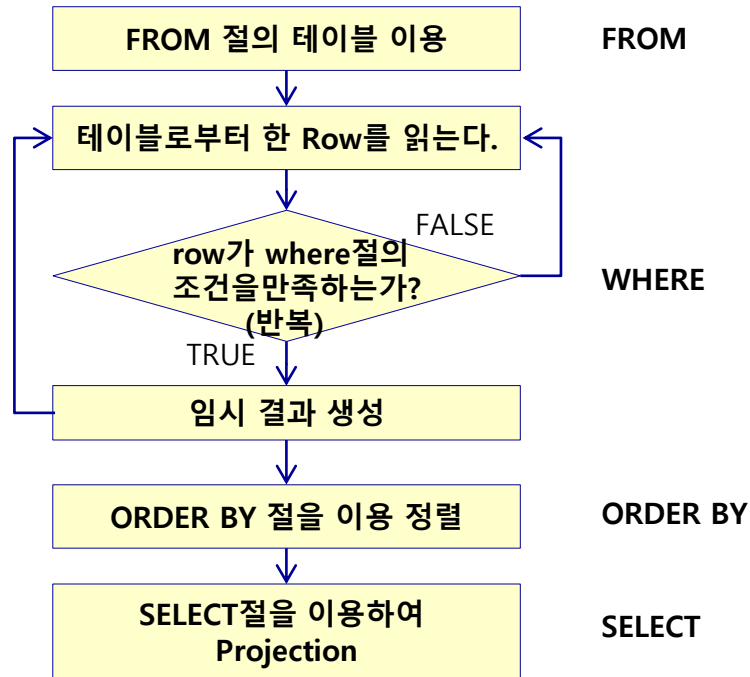
- 급여를 가장 많이 받는 3명의 직원의 이름을 출력하시오.

```

select rn,
       first_name,
       salary
from
  (select rownum rn,
          first_name,
          salary
   from
     (select first_name,
            salary
      from employees
      order by salary desc )
  )
where rn >= 11
and rn <= 20
  
```

rownum 값을 생성후에

where절을 실행한다.



[예제]

- 07년에 입사한 직원중 급여가 많은 직원중 3에서 7등의 이름 급여 입사일은?