



# SQL,PL/SQL(ORACLE)

## 3-2. 서브쿼리

## (2) 서브쿼리(Sub Query)

### 1) 설명

하나의 SQL문장절에 부속된(포함된) 또 다른 SELECT문장 따라서, 두 번 질의를 해야 얻을 수 있는 결과를 한번의 질의로 해결이 가능케하는 쿼리

### 2) 용어

<1> MAIN-QUERY or OUTER-QUERY

<2> SUB-QUERY or INNER-QUERY

### 예)

사원테이블에서 전체사원들 중 평균급여(반올림할것) // SUB-QUERY  
~보다 낮은 급여를 받는 사람들의 정보를 출력하시오.// MAIN-QUERY

```
SELECT * FROM employees WHERE salary<( ); //MAIN-QUERY
SELECT ROUND(AVG(salary),0) from employees; // SUB-QUERY
```

=> 위의 문장을 아래처럼 하나로 만들수 있다.

```
SELECT * FROM employees
WHERE salary<( SELECT ROUND(AVG(salary),0) from employees );
```



### 3) 특징

- <1> 괄호를 묶어야 한다.
- <2> 실행순서는 대부분 SUB-QUERY가 먼저 수행되고,  
MAIN-QUERY가 실행된다.
- <3> SUB-QUERY는 MAIN-QUERY의 다음 부분에 위치된다.
  - SELECT/DELETE/UPDATE 문장의 FROM절/WHERE절
  - INSERT 문장의 INTO절
  - UPDATE 문장의 SET절
- <4> SUB-QUERY는 ORDER BY절을 사용할 수 없다.  
( 예외: SELECT/DELETE/UPDATE 문장의 FROM절 )



#### 4) 서브쿼리 종류1 : 단일행, 단일컬럼을 반환하는 서브쿼리

SUB-QUERY의 실행결과가 하나의 컬럼과 하나의 행만을 리턴해 주는 경우의 쿼리 즉, 하나의 데이터를 리턴해 주는 쿼리 (WHERE 조건절에 비교값 형태로 오는 경우가 많으며 사용되는 연산자도 대부분 비교 연산자 등이 온다.)

- 사원테이블에서 급여가 제일 많은 사원의 정보를 출력하시오.  
즉 급여가 제일 많은 사람과 급여가 같은 사원을 의미.

```
SELECT * FROM employees WHERE salary=
```

```
( SELECT MAX(salary) FROM employees );
```

문제> 사원테이블에서 평균급여보다 급여가 많은 직원들을 출력하시오.

```
SELECT * FROM employees WHERE salary >  
( SELECT ROUND(AVG(salary)) FROM employees );
```



#### 4) 서브쿼리 종류2 : 복수행, 단일컬럼을 반환하는 서브쿼리

1) SUB-QUERY의 실행결과가 여러개의 행과 단일 컬럼 리턴하는 경우의 쿼리

2) 사용가능한 연산자

**IN** : 여러 개의 **SUB-QUERY** 결과값 중에서 일치되는 값을 찾을때

**>ANY, <ANY** : 비교연산자를 사용할때(서브쿼리의 반환값들 중에서 어느하나만 만족해도 됨)

**>ALL, <ALL** : 비교연산자를 사용할때(서브쿼리의 반환되는 로우들 전체에 대해 조건을 만족해야 함)

**EXISTS** : 서브쿼리 결과내용과는 상관없이 결과값이 있을 때만 결과를 보여줌.



예1) IN : 여러 개의 결과값 중에서 비교값과 같은 값을 찾을때  
 사원테이블에서 업무별로 최소 급여를 받는 사원의 정보를 사원번호,  
 이름,업무,입사일자,급여,부서번호를 출력하라.

1. SELECT MIN(salary) FROM employees GROUP BY job\_id ;

2. SELECT first\_name||' '||last\_name,job\_id,hire\_date,salary,department\_id  
 FROM employees  
 WHERE salary IN (4200,12008,8300,5800,110000,4400,17000,2500,.....);  
 //보유한 모든 급여중에서 하나하나 비교해서 같으면 출력

SELECT first\_name||' '||last\_name,job\_id,hire\_date,salary,department\_id  
 FROM employees  
 WHERE salary IN  
 ( SELECT MIN(salary) FROM employees GROUP BY job\_id );

	MIN(SALARY)
1	4200
2	12008
3	8300
4	5800
5	11000
6	4400
7	17000
8	2500
9	6900
0	12008
1	2500
2	10500
3	13000
4	10000
5	24000
6	6100
7	6000
8	2100
9	6500

EMPLOYEE_ID	LAST_NAME	FIRST_NAME	JOB_ID	HIRE_DATE	SALARY	DEPARTMENT_ID
1	Tyrene King	AD PRES	13006	2010		
2	Shanta Kochhar	AD VP	05/09/21	17000	90	
3	Luis De Raas	AD VP	01/01/19	17000	90	
4	Sharon Ebert	IT PROG	07/05/21	6000	40	
5	Olana Lorenz	IT PROG	07/02/07	4200	40	
6	Hansy Greenberg	IT MGR	02/08/19	37000	100	
7	Luis Popp	FI ACCOUNT	07/12/07	6900	100	
8	Don Nagasaki	FI MGR	02/12/07	6900	100	
9	Waren Coleman	FI CLERK	07/09/10	2500	30	
10	Shanta Sullivan	ST MAN	05/19/10	4500	50	
11	Kevin Hourigan	ST MAN	07/11/16	5800	50	
12	James Harlow	ST CLERK	01/02/16	2500	50	
13	OTO Olson	ST CLERK	07/04/10	2100	50	
14	Joshua Patel	ST CLERK	06/04/06	2500	50	
15	Patric Vempas	ST CLERK	06/07/09	2500	50	
16	Gerald Cambrault	SA MAN	07/18/15	11000	80	
17	Patric Tuckey	SA REP	05/01/20	10500	80	
18	Shanta King	SA REP	04/01/30	10000	80	
19	Shanta Vambay	SA REP	05/12/11	10500	80	
20	Harriane Bloom	SA REP	06/03/23	10000	80	
21	Shanta Roman	SA REP	09/04/21	6100	80	
22	Ellen Abel	SA REP	04/01/11	11000	80	
23	Shanta Sullivan	ST CLERK	07/06/10	2500	50	
24	Shanta Sarchand	ST CLERK	04/01/27	4200	50	
25	Shanta Perkins	ST CLERK	07/12/19	2500	50	
26	Shanta Whalen	AD ASST	13/09/17	4400	90	
27	Michael Hartstein	HR MGR	04/02/17	13000	20	
28	Pat Fay	HR REP	05/09/17	6200	20	
29	Shanta Harvie	HR REP	02/06/07	6500	20	
30	Shanta Day	HR REP	02/06/07	10000	20	
31	Shanta Higgins	AC MGR	02/06/07	12000	110	
32	Shanta Diaz	AC ACCOUNT	13/04/07	8100	110	

예2-1) >ANY : 비교연산자를 사용할 때(서브쿼리의 반환 값들 중에서  
어느 하나만 만족해도 됨)

사원 테이블에서 업무별로 평균 급여가 14000이상인 사원의 최소급여보다 많은 급여를 받는 사원의 정보를 사원번호,이름,업무,입사일자,급여,부서번호를 출력하라.

1. SELECT department\_id,first\_name||' '||last\_name,salary FROM employees  
WHERE salary >=14000;

DEPARTMENT_ID	FIRST_NAME	LAST_NAME	SALARY
1	90	Steven King	24000
2	90	Neena Kochhar	17000
3	90	Lex De Haan	17000
4	80	John Russell	14000

2. SELECT avg(salary) FROM employees  
WHERE salary >=14000  
GROUP BY job\_id;

	AVG(SALARY)
1	17000
2	14000
3	24000

3. SELECT first\_name||' '||last\_name,job\_id,hire\_date,salary,department\_id  
FROM employees  
WHERE salary > ANY (17000,14000,24000); //최소값인 14000보다 많은 급여를 의미.

4. SELECT first\_name||' '||last\_name,job\_id,hire\_date,salary,department\_id  
FROM employees  
WHERE salary > ANY( SELECT avg(salary) FROM employees  
WHERE salary >=14000 GROUP BY job\_id );

	FIRST_NAME	LAST_NAME	JOB_ID	HIRE_DATE	SALARY	DEPARTMENT_ID
1	Steven	King	AD PRES	03/06/17	24000	90
2	Neena	Kochhar	AD VP	05/09/21	17000	90
3	Lex	De Haan	AD VP	01/01/13	17000	90



예2-2) <ANY : 비교연산자를 사용할 때(서브쿼리의 반환 값들 중에서  
어느 하나만 만족해도 됨)

사원 테이블에서 업무별로 평균 급여가 14000이상인 사원 최대급여보다 적은 급여를  
받는 사원의 정보를 사원번호,이름,업무,입사일자,급여,부서번호를 출력하라.

1. SELECT department\_id,first\_name||' '||last\_name,salary FROM employees  
WHERE salary >= 14000;

DEPARTMENT_ID	FIRST_NAME	LAST_NAME	SALARY
1	90	Steven King	24000
2	90	Neena Kochhar	17000
3	90	Lex De Haan	17000
4	80	John Russell	14000

2. SELECT avg(salary) FROM employees  
WHERE salary >= 14000  
GROUP BY job\_id;

	AVG(SALARY)
1	17000
2	14000
3	24000

3. SELECT first\_name||' '||last\_name,job\_id,hire\_date,salary,department\_id  
FROM employees  
WHERE salary < ANY (17000,14000,24000); // 최대값인 24000보다 적은 급여들만~

4. SELECT first\_name||' '||last\_name,job\_id,hire\_date,salary,department\_id  
FROM employees  
WHERE salary < ANY (SELECT avg(salary) FROM employees  
WHERE salary >= 14000 GROUP BY job\_id);





예3) ALL : 비교연산자를 사용할때(서브쿼리의 반환되는 모든행을 비교)

즉, > ALL(모든행보다 큰값 : 최대값보다 큰값),  
< ALL(모든행보다 작값 : 최소값보다 작은값)

\*\*\*\* 비교값 > ALL(모든행보다 큰값 : 최대값보다 큰값)

80번 부서 직원들의 급여보다 많은 급여를 받는 직원을 출력하라.

1. SELECT salary FROM employees WHERE department\_id=80;  
결과 >>> 오른쪽 그림참고.

2. SELECT employee\_id,first\_name||' '||last\_name, job\_id,  
hire\_date, salary, department\_id  
FROM employees  
WHERE salary > ALL

( SELECT salary FROM employees WHERE department\_id=80 ) ;

EMPLOYEE_ID	FIRST_NAME  ' '  LAST_NAME	JOB_ID	HIRE_DATE	SALARY	DEPARTMENT_ID
1	102Lex De Haan	AD VP	01/01/13	17000	90
2	101Neena Kochhar	AD VP	05/09/21	17000	90
3	100Steven King	AD PRES	03/06/17	24000	90

\*\*\*\* 비교값 < ALL(모든행보다 작값 : 최소값보다 작은값)

80번 부서 직원들의 급여보다 적은 급여를 받는 직원을 출력하라.

1. 오른쪽 그림 참고

2. SELECT employee\_id,first\_name||' '||last\_name, job\_id,  
hire\_date, salary, department\_id  
FROM employees  
WHERE salary < ALL

( SELECT salary FROM employees WHERE department\_id=80 ) ;

SALARY
14000
13500
12000
11000
10500
10000
9500
9000
8000
7500
7000
10000
9500
9000
8000
7500
7000
10500
9500
7200
6800
6400
6200
11500
10000
9600
7400
7300
6100
11000
8800
8600
8400
6200

예3) EXISTS :서브쿼리 결과내용과는 상관없이  
결과값이 있을 때만 결과를 보여줌.

예1)부서테이블의 부서코드,부서이름을 출력하라.

```
select department_id,department_name  
from departments;
```

예2)사원테이블의 부서코드를 출력하라.(중복제거)

```
select distinct(department_id)  
from employees;
```

DEPARTMENT_ID
100
30
(null)
90
20
70
110
50
80
40
60
10

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
30	Purchasing
40	Human Resources
50	Shipping
60	IT
70	Public Relations
80	Sales
90	Executive
100	Finance
110	Accounting
120	Treasury
130	Corporate Tax
140	Control And Credit
150	Shareholder Services
160	Benefits
170	Manufacturing
180	Construction
190	Contracting
200	Operations
210	IT Support
220	NOC
230	IT Helpdesk
240	Government Sales
250	Retail Sales
260	Recruiting
270	Payroll

예3)사원테이블에 존재하는 부서코드와 부서이름을 출력하라.

```
select department_id,department_name  
from departments  
where EXISTS  
(select distinct(department_id) from employees  
where departments.department_id = employees.department_id);
```

DEPARTMENT_ID	DEPARTMENT_NAME
10	Administration
20	Marketing
30	Purchasing
40	Human Resources
50	Shipping
60	IT
70	Public Relations
80	Sales
90	Executive
100	Finance
110	Accounting

문제)사원 테이블에서 부하직원이 없는 사원의 모든 정보를 exists를 이용하여 작성하라.

```
select employee_id, first_name
from employees e1
where not exists
(select employee_id
from employees e2
where e1.employee_id = e2.manager_id);
```

#### 4) 서브쿼리 종류3 : 복수행, 복수컬럼을 반환하는 서브쿼리

복수 컬럼 SUB-QUERY :

SUB-QUERY의 실행결과가 여러개의 컬럼 AND  
여러개의 행을 리턴해주는 쿼리

예1)

업무별로 최소급여를 받는 사원의 정보를 사원번호,이름,업무,급여,  
부서번호를 출력하여라.

```
SELECT employee_id,first_name||' '||last_name "Name",job_id,salary,department_id
FROM employees
WHERE (job_id,salary) IN(SELECT job_id,min(salary) FROM employees GROUP BY
job_id);
```

#### 4) 서브쿼리 종류4 : 상호 관련 SUB-QUERY

MAIN-QUERY절에 사용된 테이블이 SUB-QUERY절에 다시  
재사용되는 경우의 서브쿼리로 .(컴마)를 통해 연결되어 있는 쿼리

예1)

테이블에서 적어도 한명의 사원으로부터 보고를 받을수 있는 사원의정보를  
사원번호,이름,업무,입사일자,급여를 출력하라.

```
SELECT employee_id,first_name||' '||last_name "Name",job_id,hire_date,salary  
FROM employees e  
WHERE EXISTS(SELECT * FROM employees WHERE manager_id=e.employee_id);
```

강사 양성  
원

예2)

소속부서의 평균 급여보다 많은 급여를 받는 사원을 출력하라.

```
SELECT * FROM employees e  
WHERE salary > (select avg(salary)  
from employees  
where department_id = e.department_id);
```



## 5) UPDATE 에서 서브쿼리 사용해 보기

### 예1-Update, 변경, 수정)

'David Austin'의 직업을 'John Chen'인 사람의 직업과 같은 직업으로 변경하세요

#### 1. 변경전)

```
select job_id from employees  
where first_name = 'David'  
      and last_name = 'Austin'; //IT_PROG
```

```
select job_id from employees  
where first_name = 'John'  
      and last_name = 'Chen'; //FI_ACCOUNT
```

#### 2. 변경진행)

```
UPDATE employees  
SET job_id = ( SELECT job_id FROM employees  
                WHERE first_name = 'John' and last_name = 'Chen')  
WHERE first_name = 'David' and last_name = 'Austin';
```

#### 3. 변경후)

```
select job_id from employees where first_name = 'David'  
      and last_name = 'Austin'; //FI_ACCOUNT
```



## 5) DELETE 에서 서브쿼리 사용해 보기


예2-Delete, 삭제, 제거)

최소급여를 받는 **사원들을 삭제 하라.**

1. 삭제 전)

```
SELECT min(salary) FROM employees;
```

```
select employee_id,first_name||' '||last_name,salary FROM employees  
WHERE salary=(SELECT min(salary) FROM employees);
```



2. 삭제 진행)

```
DELETE FROM employees  
WHERE salary=( SELECT min(salary) FROM employees );
```

3. 삭제 후)

```
select employee_id,first_name||' '||last_name,salary FROM employees  
WHERE salary=(SELECT min(salary) FROM employees);
```

**참고>>> 테스트 완료 후 rollback;**