



SQL

✓ 원리를 알면 IT가 맛있다

SQL for Beginners

chapter 07.

서브쿼리

- 서브쿼리 개요
- 서브쿼리 종류
 - 단일행
 - 복수행
 - 다중 컬럼
 - 인라인 뷰 (inline view)
 - 상관 서브쿼리
- WITH 문

■ 서브쿼리 정의

- SELECT 문 (Mainquery) 에 포함되어 있는 또 하나의 별도 SELECT 문 (Subquery)이다.
- 여러 번의 SELECT문을 수행해야 얻을 수 있는 결과를 하나의 중첩된 SELECT 문으로 쉽게 얻을 수 있도록 해준다.

■ 서브쿼리 예

: 사원테이블에서 SCOTT 사원보다 많은 급여를 받는 사원의 이름을 검색하시오.

```
SQL> SELECT SAL
  2  FROM EMP
  3  WHERE ENAME = 'SCOTT';

      SAL
-----
      3000

SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL > 3000;

ENAME
-----
KING
```



```
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL > ( SELECT SAL
  4                  FROM EMP
  5                  WHERE ENAME = 'SCOTT' );

ENAME
-----
KING
```

■ 서브쿼리 사용방법

```
SELECT select_list
FROM table
WHERE expr operator
      (SELECT select_list
       FROM table);
```

- 바깥 쪽 쿼리를 Main query , 안쪽 쿼리를 Subquery 라고 한다.
- 서브쿼리가 먼저 실행되고, 그 결과가 메인 쿼리에 전달되어 실행된다.
따라서 메인 쿼리에서 서브 쿼리 컬럼을 사용할 수 없다.
- 서브쿼리는 SELECT, FROM, WHERE, HAVING, ORDER BY, UPDATE, INSERT INTO 절에도 사용될 수 있다.
- operator 는 단일 행 연산자 (> , = , >= , < , <= , !=)와 복수 행 연산자 (IN , ANY , ALL, EXISTS)를 사용 할 수 있다.
- SQL 문장의 서브쿼리가 단일 행 서브쿼리인 경우는 단일 행 연산자를 사용하고 복수 행 서브쿼리인 경우에는 복수 행 연산자를 사용해야 한다.
- 서브쿼리에는 반드시 괄호를 사용한다.
- 서브쿼리에는 ORDER BY 절을 사용불가

- 메인 쿼리와 연관성에 따라서
 - 연관성 없는 서브쿼리 (Non correlated subquery)
: 서브쿼리가 메인 쿼리 컬럼을 가지고 있지 않는 형태. 메인 쿼리에 값을 제공할 목적으로 주로 사용됨.
 - 연관성 있는 서브쿼리 (Correlated subquery) : 상관 서브쿼리
: 서브쿼리가 메인 쿼리 컬럼을 가지고 있는 형태. 메인 쿼리가 먼저 수행되어 읽혀진 데이터를 서브쿼리에서 조건이 맞는지 확인할 때 주로 사용됨.
- 서브쿼리 실행 결과 개수에 따라서
 - 단일 행 서브쿼리
 - 복수 행 서브쿼리
 - 다중 컬럼 서브쿼리
- 서브쿼리 위치에 따라서
 - 일반 서브쿼리
 - 인라인 뷰 (inline view) : from절에 위치.
 - 상관 서브쿼리 : 일반적으로 select절에 위치

■ 단일행 서브쿼리

- 서브쿼리가 한 개의 행을 리턴.
- 반드시 단일행 연산자를 사용해야 한다. (= , > , < , >= , <= , !=)

문제) EMP 테이블에서 SCOTT의 급여보다 많은 사원의 사원번호, 이름, 담당 업무, 급여를 출력 ?

```
SQL> SELECT empno,ename,job,sal
2 FROM emp
3 WHERE sal > (SELECT sal
4              FROM emp
5              WHERE ename = 'SCOTT');
```

EMPNO	ENAME	JOB	SAL
7839	KING	PRESIDENT	5000

문제) EMP 테이블에서 사원번호가 7521의 업무와 같고 급여가 7934보다 많은 사원의 사원번호, 이름, 담당업무,입사일자, 급여를 출력 ?

```
SQL> SELECT empno, ename, job, hiredate, sal
2 FROM emp
3 WHERE job = (SELECT job
4              FROM emp
5              WHERE empno = 7521)
6 AND
7      sal > (SELECT sal
8            FROM emp
9            WHERE empno = 7934);
```

EMPNO	ENAME	JOB	HIREDATE	SAL
7499	ALLEN	SALESMAN	20-FEB-81	1600
7844	TURNER	SALESMAN	08-SEP-81	1500

- 서브쿼리에서 그룹함수 사용

- 단일 행을 RETURN하는 SUBQUERY에 그룹 함수를 사용하여 MAIN QUERY로부터 데이터를 출력할 수 있다. 그룹함수도 단일행 리턴.

문제) EMP 테이블에서 급여의 평균보다 적은 사원의 사원번호, 이름, 담당업무, 급여, 부서번호를 출력 ?

```
SQL> SELECT empno,ename,job,sal,deptno
2 FROM emp
3 WHERE sal < (SELECT AVG(sal)
4             FROM emp);
```

EMPNO	ENAME	JOB	SAL	DEPTNO
7654	MARTIN	SALESMAN	1250	30
7499	ALLEN	SALESMAN	1600	30
7844	TURNER	SALESMAN	1500	30
7900	JAMES	CLERK	950	30
7521	WARD	SALESMAN	1250	30
7369	SMITH	CLERK	800	20
7876	ADAMS	CLERK	1100	20

.

8 rows selected.

■ HAVING 절에서 서브쿼리 사용

– SUBQUERY를 WHERE 절 뿐만 아니라 HAVING절에서도 사용 가능하다.
오라클 서버는 SUBQUERY를 실행하고 MAIN QUERY의 HAVING절에 RETURN 한다

문제) EMP 테이블에서 20번 부서의 최소 급여보다 많은 모든 부서를 출력 ?

```
SQL> SELECT deptno,MIN(sal)
```

```
2 FROM emp
```

```
3 GROUP BY deptno
```

```
4 HAVING MIN(sal) > (SELECT MIN(sal)
```

```
5 FROM emp
```

```
6 WHERE deptno = 20);
```

DEPTNO	MIN(SAL)
--------	----------

10	1300
----	------

30	950
----	-----

문제) EMP 테이블에서 업무별로 가장 적은 급여를 출력?

```
SQL> SELECT job,avg(sal)
2 FROM emp
3 GROUP BY job
4 HAVING AVG(sal) = (SELECT MIN(AVG(sal))
5 FROM emp
6 GROUP BY job);
```

JOB	AVG(SAL)
CLERK	1037.5

■ 복수행 서브쿼리

- 서브쿼리가 여러 개의 행을 리턴.
- 반드시 복수행 연산자를 사용해야 한다. (IN , ANY ,ALL ,EXISTS)

■ IN 연산자

- : WHERE 절에서 사용하는 일반 비교연산자와 동일하다.
- : 메인쿼리의 비교 조건이 서브쿼리의 결과 중에서 하나라도 일치하면 검색가능.

문제) EMP 테이블에서 업무별로 최소 급여를 받는 사원의 사원번호, 이름, 업무, 입사일자, 급여, 부서번호를 출력 ?

```
SQL> SELECT empno,ename,job,hiredate,sal,deptno
2 FROM emp
3 WHERE sal IN (SELECT MIN(sal)
4                FROM emp
5                GROUP BY job);
```

■ ALL 연산자

- : 복수행 서브쿼리 결과가 메인 쿼리의 WHERE 절에서 부등호 조건으로 비교될 때 사용된다. (원래 부등호 조건은 단일행에서 사용됨)
- : 서브쿼리에서 반환되는 행들 전체에 대해 조건을 만족해야 된다.

사용 예

: 사원 테이블에서 업무가 MANAGER인 사원의 최소급여보다 적은 급여를 받는 사원들의 이름 검색?

```
SQL> SELECT SAL
2 FROM EMP
3 WHERE JOB = 'MANAGER';
```

SAL
2975
2850
2450

```
SQL> SELECT EMPNO, ENAME, SAL
2 FROM EMP
3 WHERE SAL < 2450;
```

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7654	MARTIN	1250
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7934	MILLER	1300

최소값보다 작은

8 개의 행이 선택되었습니다.

```
SQL> SELECT EMPNO, ENAME, SAL
2 FROM EMP
3 WHERE SAL < ALL (SELECT SAL
4 FROM EMP
5 WHERE JOB = 'MANAGER' );
```

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7654	MARTIN	1250
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7934	MILLER	1300

8 개의 행이 선택되었습니다.

■ ANY 연산자

: 복수행 서브쿼리 결과가 메인 쿼리의 WHERE 절에서 부등호 조건으로 비교될 때 사용된다.

: 서브쿼리에서 반환되는 행들 중에서 어느 하나만 만족해도 된다.

사용 예

: 사원 테이블에서 업무가 MANAGER인 사원의 최소급여보다 많은 급여를 받는 사원들의 이름 검색?

```
SQL> SELECT SAL  
2 FROM EMP  
3 WHERE JOB = 'MANAGER';
```

SAL
2975
2850
2450

```
SQL> SELECT EMPNO, ENAME, SAL  
2 FROM EMP  
3 WHERE SAL > 2450;
```

최소값보다 큰

EMPNO	ENAME	SAL
7566	JONES	2975
7698	BLAKE	2850
7788	SCOTT	3000
7839	KING	5000
7902	FORD	3000

```
SQL> SELECT EMPNO, ENAME, SAL  
2 FROM EMP  
3 WHERE SAL > ANY ( SELECT SAL  
4 FROM EMP  
5 WHERE JOB = 'MANAGER');
```

EMPNO	ENAME	SAL
7839	KING	5000
7902	FORD	3000
7788	SCOTT	3000
7566	JONES	2975
7698	BLAKE	2850

■ EXISTS 연산자

: 서브쿼리에서 검색된 결과가 하나라도 존재하는지 여부를 확인하는 조건을 의미한다.

만일 서브쿼리에서 검색된 결과가 하나도 없으면 메인 쿼리의 조건절이 거짓이 되어 '선택된 레코드가 없습니다' 라는 결과를 출력한다.

문제) 만일 사원중에서 comm을 받는 사원이 한 명이라도 있으면 모든 사원출력?

```
SQL> SELECT *  
2  FROM emp  
3  WHERE EXISTS ( SELECT empno  
4                  FROM emp  
5                  WHERE comm IS NOT NULL );
```

■ 다중 컬럼 서브쿼리

: 서브쿼리에서 여러개의 컬럼값을 검색하여 메인쿼리의 조건절과 비교하는 서브쿼리이다.

메인쿼리의 조건절에서도 서브쿼리의 컬럼수만큼 지정해야 된다.

■ 다중 컬럼 서브쿼리 비교 방법 2가지

1) pairwise

: 컬럼을 쌍으로 묶어서 동시에 비교하는 방식

1	2
1	3
4	2
3	2

1	2
4	3

2) unpairwise

: 컬럼별로 나누어 비교하고 나중에 AND 연산 처리하는 방식

1	2
1	3
4	2
3	2

1	2
4	3

문제) 부서별로 가장 많은 sal을 받는 사원 정보 출력 ? – pairwise 이용한 경우

```
SQL> SELECT deptno, empno, ename, sal
2  FROM emp
3  WHERE (deptno, sal ) IN ( SELECT deptno, MAX(sal)
4                             FROM emp
5                             GROUP BY deptno );
```

DEPTNO	EMPNO	ENAME	SAL
30	7698	BLAKE	2850
20	7902	FORD	3000
20	7788	SCOTT	3000
10	7839	KING	5000

문제) 부서별로 가장 많은 sal을 받는 사원 정보 출력? – unpairwise 이용한 경우

```
SQL> SELECT deptno, empno, ename, sal
2  FROM emp
3  WHERE deptno IN ( SELECT deptno
4                     FROM emp
5                     GROUP BY deptno )
6  AND sal IN ( SELECT MAX(sal)
7                FROM emp
8                GROUP BY deptno );
```

DEPTNO	EMPNO	ENAME	SAL
30	7698	BLAKE	2850
20	7902	FORD	3000
20	7788	SCOTT	3000
10	7839	KING	5000

* from절 뒤에 테이블 명이 나와야 되지만, 서브 쿼리가 하나의 가상 테이블을 반환하는 형태로 사용되는 경우를 의미한다.

```
SELECT column_list
FROM (subquery) alias
WHERE condition;
```

예제> emp 와 dept 테이블에서 부서별 sal 총합과 평균을 출력?

```
select e.deptno , total_sum , total_avg , cnt
from ( select deptno , sum(sal) total_sum, avg(sal) total_avg , count(*) cnt
      from emp
      group by deptno ) e , dept d
where e.deptno = d.deptno;
```

* 앞서 배운 인라인 뷰에 별칭(alias)을 부여하고, FROM 절에서 별칭이 부여된 인라인 뷰를 사용할 수 있다. 서브쿼리 재사용이 가능하다.

```
WITH 별칭1 AS ( subquery ),  
     별칭2 AS ( subquery )
```

```
SELECT 컬럼1, 컬럼2  
FROM 별칭1, 별칭2 ;
```

예제 1> 하나의 서브쿼리

```
WITH AA AS  
( SELECT ROWNUM, 'TEST1', SYSDATE FROM DUAL  
  
  UNION ALL  
  
  SELECT ROWNUM, 'TEST2', SYSDATE FROM DUAL  
  
  UNION ALL  
  
  SELECT ROWNUM, 'TEST3', SYSDATE FROM DUAL )  
  
SELECT * FROM AA;
```

	ROWNUM	'TEST1'	SYSDATE
1	1	TEST1	15/05/23
2	1	TEST2	15/05/23
3	1	TEST3	15/05/23

예제 2 > 2개의 서브쿼리

```
WITH AA AS
  (SELECT ROWNUM AS SEQ, 'TEST1' AS NAME, SYSDATE
    FROM DUAL
   UNION ALL
   SELECT ROWNUM AS SEQ, 'TEST2' AS NAME, SYSDATE
    FROM DUAL
   UNION ALL
   SELECT ROWNUM AS SEQ, 'TEST3' AS NAME, SYSDATE FROM DUAL),

BB AS
  (SELECT ROWNUM AS SEQ, 'TEST1' AS NAME, SYSDATE
    FROM DUAL
   UNION ALL
   SELECT ROWNUM AS SEQ, 'TEST2' AS NAME, SYSDATE
    FROM DUAL
   UNION ALL
   SELECT ROWNUM AS SEQ, 'TEST3' AS NAME, SYSDATE FROM DUAL)

SELECT * FROM AA, BB WHERE AA.NAME = BB.NAME;
```

	SEQ	NAME	SYSDATE	SEQ_1	NAME_1	SYSDATE_1
1	1	TEST1	15/05/23	1	TEST1	15/05/23
2	1	TEST2	15/05/23	1	TEST2	15/05/23
3	1	TEST3	15/05/23	1	TEST3	15/05/23

예제 3 > 특정 서브쿼리에서 다른 서브쿼리 접근 가능 (중첩 사용 가능)

```
create table emp1
as
select * from emp;

create table emp2
as
select * from emp
where job != 'CLERK';

WITH AA AS
( SELECT * FROM emp1 ),
BB AS
( SELECT * FROM emp2 ),
CC AS
( SELECT AA.empno, AA.ename, BB.deptno FROM AA, BB WHERE AA.empno = BB.empno )
SELECT * FROM CC;
```

□ 3) 서브쿼리 – 상관 서브쿼리 (스칼라 서브쿼리: 1 row 1 column 반환) SQL

- select절에 하나의 컬럼 처럼 동작되는 서브쿼리로서, 메인 쿼리와 상관(조인 등)이 있다고 해서 상관 서브쿼리라고 한다.
즉, 바깥쪽 쿼리의 컬럼 중의 하나가 안쪽 서브쿼리의 조건에 이용되는 처리 방식 이다.

```
SELECT column_list , (subquery) alias  
FROM table  
WHERE condition;
```

예제> 사원이름과 소속 부서 정보 ?

```
select e.empno, e.ename, e.sal, dname "부서명"  
from emp e, dept d  
where e.deptno = d.deptno  
order by 1;
```

```
select e.empno, e.ename, e.sal,  
      ( select dname from dept where dept.deptno = e.deptno ) "부서명"  
from emp e  
order by 1;
```


■ 실습 문제

1. 사원 테이블에서 BLAKE 보다 급여가 많은 사원들의 사번, 이름, 급여를 검색하시오.
2. 사원 테이블에서 MILLER 보다 늦게 입사한 사원의 사번, 이름, 입사일을 검색하시오.
3. 사원 테이블에서 사원 전체 평균 급여보다 급여가 많은 사원들의 사번, 이름, 급여를 검색하시오.
4. 사원 테이블에서 부서별 최대 급여를 받는 사원들의 사번, 이름, 부서코드, 급여를 검색하시오.
5. Salgrade가 2등급인 사원들의 평균 급여보다 적게 받는 사원 정보를 검색하시오.



Thank you
