Sub Query 는 단일행 서브쿼리 ; 서브쿼리의 결과가 단일 행

단일 행 연산자 : =, >, >=, <, <=, <>

다중행 서브쿼리 ; 서브쿼리의 결과가 2 행 이상의 행

복수 행 연산자 : IN, NOT IN, ANY, SOME, ALL, EXISTS

. 서브 쿼리의 개념

I SCOTT 의 부서명을 알아내기 위한 서브 쿼리문부터 살펴보자

SELECT dname FROM dept WHERE deptno=(SELECT deptno

FROM emp

WHERE ename='SCOTT'); -- ( )부분 : 서브쿼리

(1) 서브 쿼리는 하나의 SQL 문장의 절 안에 포함된 또 하나의 SELECT 문장입니다.

(2) 그렇기에 서브 쿼리를 포함하고 있는 쿼리문을 메인 쿼리, 포함된 또 하나의 쿼리를 서브 쿼리라 합니다.

(3) 서브 쿼리는 비교 연산자의 오른쪽에 기술하고 반드시 괄호로 둘러 쌓아야 합니다.

(4) 서브 쿼리는 메인 쿼리가 실행되기 이전에 한번만 실행이 됩니다

2. 단일행 서브쿼리

(1) 단일 행(Single Row) 서브 쿼리는 수행 결과가 오직 하나의 로우(행, row)만을 반환하는 서브 쿼리를 갖는 것을 말합니다.

(2) 단일 행 서브 쿼리문에서는 이렇게 오직 하나의 로우(행, row)로 반환되는 서브 쿼리의 결과는 메인 쿼리에 보내게 되는데 메인 쿼리의 WHERE 절에서는 단일 행 비교 연산자인 =, >, >=, <, <=, <>를 사용해야 합니다

. 다중행 서브쿼리

(1) 다중 행 서브 쿼리는 서브쿼리에서 반환되는 결과가 하나 이상의 행일 때 사용하는 서브 쿼리입니다. 다중 행 서브 쿼리는 반드시 다중 행 연산자(Multiple Row Operator)와 함께 사용해야 합니다.

(2) 다중행 연산자의 종류

I IN : 메인 쿼리의 비교 조건('=' 연산자로 비교할 경우)이 서브 쿼리의 결과 중에서 하나라도 일치하면 참

I ANY, SOME : 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 하나 이상이 일치하면 참

I ALL : 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 모든 값이 일치하면 참

I EXISTS : 메인 쿼리의 비교 조건이 서브 쿼리의 결과 중에서 만족하는 값이 하나라도 존재하면 참

```sql
--1. 사원테이블에서 가장 먼저 입사한 사람의 이름, 급여, 입사일
SELECT MIN(HIREDATE) FROM EMP;
SELECT ENAME, SAL, HIREDATE
FROM EMP
WHERE HIREDATE = (SELECT MIN(HIREDATE) FROM EMP);

-- 2. 회사에서 가장 급여가 적은 사람의 이름, 급여
SELECT MIN(SAL) FROM EMP;
SELECT ENAME, SAL
FROM EMP
WHERE SAL = (SELECT MIN(SAL) FROM EMP);
-- 3. 회사 평균보다 급여를 많이 받는 사람의 이름, 급여, 부서코드
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE SAL > (SELECT AVG(SAL) FROM EMP);
--4. 회사 평균 이하의 급여를 받는 사람의 이름, 급여, 부서명
SELECT ENAME, SAL, DNAME
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO AND SAL <= (SELECT AVG(SAL) FROM EMP);
--5. SCOTT보다 먼저 입사한 사람의 이름, 급여, 입사일, 급여 등급
SELECT ENAME, SAL, HIREDATE, GRADE
FROM EMP, SALGRADE
WHERE SAL BETWEEN LOSAL AND HISAL
AND HIREDATE < (SELECT HIREDATE FROM EMP WHERE ENAME = 'SCOTT');
SELECT * FROM EMP;
```

```sql
--6. 5번(SCOTT보다 먼저 입사한 사람의 이름, 급여, 입사일, 급여 등급)에 부서명 추가하고 급여가 큰 순 정렬
SELECT ENAME, SAL, HIREDATE, GRADE, DNAME
FROM EMP E, DEPT D, SALGRADE
WHERE SAL BETWEEN LOSAL AND HISAL
AND E.DEPTNO = D.DEPTNO
AND HIREDATE < (SELECT HIREDATE FROM EMP WHERE ENAME = 'SCOTT')
ORDER BY SAL DESC;
SELECT * FROM EMP;
--7. BLAKE 보다 급여가 많은 사원들의 사번, 이름, 급여
SELECT DEPTNO, ENAME, SAL
FROM EMP
WHERE SAL > (SELECT SAL FROM EMP WHERE ENAME = 'BLAKE');
--8. MILLER보다 늦게 입사한 사원의 사번, 이름, 입사일
SELECT DEPTNO, ENAME, HIREDATE
FROM EMP
WHERE HIREDATE > (SELECT HIREDATE FROM EMP WHERE ENAME = 'MILLER');
--9. 사원전체 평균 급여보다 급여가 많은 사원들의 사번, 이름, 급여
SELECT EMPNO, ENAME, SAL
FROM EMP
WHERE SAL > (SELECT AVG(SAL) FROM EMP);
```

```sql
--10. CLARK와 같은 부서번호이며, 사번이 7698인 직원의 급여보다 많은 급여를 받는 사원의 사번, 이름, 급여
SELECT EMPNO, ENAME, SAL
FROM EMP
WHERE SAL > (SELECT SAL FROM EMP WHERE EMPNO = 7698)
AND DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME = 'CLARK');
--11.  응용심화. CLARK와 같은 부서명이며, 사번이 7698인 직원의 급여보다 많은 급여를 받는 사원의 사번, 이름, 급여
SELECT EMPNO, ENAME, SAL
FROM EMP E, DEPT D
WHERE E.DEPTNO = D.DEPTNO
AND DNAME = (SELECT DNAME FROM EMP E2, DEPT D2 WHERE E2.DEPTNO=D2.DEPTNO AND ENAME = 'CLARK')
AND SAL > (SELECT SAL FROM EMP WHERE EMPNO = '7698');
--12. BLAKE와 같은 부서에 있는 모든 사원의 이름과 입사일자
SELECT ENAME, HIREDATE
FROM EMP
WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME = 'BLAKE') AND ENAME != 'BLAKE';
--13. 평균 급여 이상을 받는 모든 종업원에 대해서 사원번호와 이름 단 급여가 많은 순으로 출력)
SELECT EMPNO, ENAME
FROM EMP
WHERE SAL > (SELECT AVG(SAL) FROM EMP)
ORDER BY SAL DESC;
```