[정 의] 다른 SQL(SELECT,INSERT,DELETE,UPDATE,CREATE..) 문 안에 포함되는 SELECT문
MAIN QUERY, SUB QUERY

SQL(SELECT/INSERT/DELETE/UPDATE/CREATE)
SELECT ~

[용 도]

[종 류] Scalar Subquery → 1행,1컬럼의 값만을 리턴한다, Select List에서 사용되며 컬럼을 사용할수 있는곳에 사용.
Inline View → From 절에 테이블 대신 사용.
Nested Subquery → Where 조건절에 사용

* Query라는 단어의 2가지 의미 1.SQL 2.SELECT

[실행순서] SUBQUERY 실행 → MAIN QUERY실행
예외) CORRELATED SUBQUERY(상관서브쿼리)

[분 류]
1) RETURN STYLE에 의한
SINGLE ROW SUBQUERY : 리턴되는 데이터가 1건 이하, 단일행 비교 연산자와 함께 사용
Ex) = , > ,< , <=..
MULTIPLE ROWS SUBQUERY : 리턴되는 데이터가 1건 이상 , 리스트 연산자와 함께 사용
Ex) IN, ANY, ALL, SOME, EXISTS
SINGLE COLUMN SUBQUERY
MULTIPLE COLUMNS SUBQUERY
2) 동작하는 방식에 따른 분류
Un-Correlated Subquery ?? 이런 용어가 있는지 점검할 것...
서브쿼리가 메인쿼리의 컬럼을 참조하지 않는다.
메인쿼리에 값(서브쿼리의 실행결과)을 제공하는 목적으로 사용
Correlated Subquery
서브쿼리가 메인쿼리의 컬럼을 참조 한다.
메인 쿼리가 먼저 실행되고 서브쿼리에서 필터링 하는 목적으로 사용

SINGLE COLUMN,SINGLE ROW

① SELECT ENAME,JOB
FROM EMP
WHERE DEPTNO = (SELECT DEPTNO FROM EMP WHERE ENAME = 'SMITH');

② SELECT ENAME,SAL FROM EMP WHERE SAL < (SELECT AVG(SAL) FROM EMP);

SINGLE COLUMN, MULTIPLE ROW RETURN SUBQUERY

① SELECT ENAME,JOB FROM EMP WHERE DEPTNO = 10,30; // ??

② SELECT ENAME,JOB FROM EMP WHERE DEPTNO IN (10,30); // Multiple Rows

// 3명 이상 근무 하는 부서의 정보

③ SELECT DNAME,LOC FROM DEPT
WHERE DEPTNO = (SELECT DEPTNO FROM EMP GROUP BY DEPTNO HAVING COUNT(*) > 3);

MULTIPLE COLUMN, MULTIPLE ROW RETURN

④ SELECT DEPTNO,JOB,ENAME,SAL FROM EMP
WHERE (DEPTNO,JOB) IN (SELECT DEPTNO,JOB FROM EMP
GROUP BY DEPTNO,JOB HAVING AVG(SAL) > 2000);

Scalar Subquery

[장점] 편리성

[질문] 반복되는 실행을 하는가? 실행횟수 // 입/출력값 , Query Execution Cache , hashing

⑤ SELECT DEPTNO,ENAME,JOB,SAL,
(SELECT ROUND(AVG(SAL),0) FROM EMP S WHERE S.JOB=M.JOB) AS JOB_AVG_SAL
FROM EMP M
ORDER BY JOB;

CORRELATED SUBQUERY(상관서브쿼리)

[주의] Subquery는 Mainquery의 컬럼을 참조할수 있지만 Mainquery는 Subquery의 컬럼을 참조할수 없다

[질문] Mainquery에서 Subquery의 컬럼을 참조 하려면 → ① Join 으로 변환 ② Scalar Subquery

⑥ SELECT DEPTNO,ENAME,JOB,SAL FROM EMP M
WHERE SAL > (SELECT AVG(SAL) AS AVG_SAL FROM EMP WHERE JOB = M.JOB);

In-Line View (FROM 절에 사용된 SUBQUERY)

[설명] SQL이 실행되는 시점에 동적으로 생성되는 View의 역할을 한다고 해서 Dynamic View 라고도 한다. 일반으로 Subquery의 컬럼을 Mainquery에서 사용할수 없지만 Inline View에서 Subquery의 컬럼을 Mainquery 에서 사용이 가능하다.

① SELECT DEPTNO, ENAME,EMP.JOB,SAL,IV.AVG_SAL
FROM EMP, (SELECT JOB,ROUND(AVG(SAL)) AS AVG_SAL FROM EMP GROUP BY JOB) IV
WHERE EMP.JOB = IV.JOB AND SAL > IV.AVG_SAL
ORDER BY DEPTNO ,SAL DESC;

1:M vs M:1

// 결과 예측 해보면

② SELECT * FROM SCOTT.EMP WHERE DEPTNO IN (SELECT DEPTNO FROM SCOTT.DEPT);

③ SELECT * FROM SCOTT.DEPT WHERE DEPTNO IN (SELECT DEPTNO FROM SCOTT.EMP);

TOP-N,BOTTOM-M

④ SELECT * FROM (SELECT EMPNO,ENAME,SAL FROM EMP ORDER BY ORDER BY SAL ASC) BM
WHERE ROWNUM <= 5; // IN-LINE VIEW 에서 ORDER BY

⑤ SELECT TN.EMPNO,TN.ENAME,TN.SAL
FROM (SELECT EMPNO,ENAME,SAL FROM EMP ORDER BY SAL DESC) TN
WHERE ROWNUM < 5;

****** DML 연산과 Subquery // ???

지금까지는 SELECT 시에 사용되는 Subquery에 대한 실습을 했는데
DML 연산에도 Subquery를 사용할수 있으며 해당 실습은 DML 과정에서 다루며
여기에서는 DML 연산에 사용되는 Subquery의 예를 보기만 하겠다.

```
INSERT INTO BONUS(ENAME,JOB,SAL,COMM)
        SELECT ENAME,JOB,SAL,COMM FROM EMP;

INSERT INTO BONUS(ENAME,JOB,SAL,COMM)
        SELECT ENAME,JOB,SAL,DECODE(DEPTNO,10,SAL*0.3,20,SAL*0.2)+NVL(COMM,0)
```

```
        FROM   EMP
        WHERE DEPTNO IN (10,20);
```

-- 평상시에 COMM을 받지 못하는 사원들에게 평균 COMM 금액의 50%를 보너스로 지급

```
UPDATE EMP SET COMM = (SELECT AVG(COMM)/2 FROM EMP)
WHERE COMM IS NULL OR COMM = 0;
```

-- 평균 이상의 급여를 받는 사원들은 보너스 지급 대상자에서 제외

```
DELETE FROM   BONUS WHERE SAL > (SELECT AVG(SAL) FROM EMP ;
```