

7. 서브쿼리(Subquery)(1) [서브쿼리란]

7.1 서브쿼리(Subquery)란?

- SUBQUERY는 다른 하나의 SQL 문장의 절에 NESTED된 SELECT 문장 입니다.
- SELECT, UPDATE, DELETE, INSERT와 같은 DML문과 CREATE TABLE또는 VIEW에서 이용 될 수 있습니다.
- 알려지지 않은 조건에 근거한 값들을 검색하는 SELECT 문장을 작성하는데 유용 합니다.
- SUBQUERY는 MAIN QUERY가 실행되기 이전에 한번 실행 됩니다.

- SUBQUERY는 괄호로 묶어야 합니다.
- 두 종류의 비교 연산자들이 SUBQUERY에 사용 됩니다.
- 단일 행 연산자(=, >, >=, <, <=, <>, !=)
- 복수 행 연산자(IN, NOT IN, ANY, ALL, EXISTS)
- SUBQUERY는 연산자의 오른쪽에 나타나야 합니다.

** SUBQUERY의 유형

- 단일 행(Sing-Row) 서브쿼리 : SELECT문장으로부터 오직 하나의 행만을 검색하는 질의입니다.
- 다중 행(Multiple-Row) 서브쿼리 : SELECT문장으로부터 하나 이상의 행을 검색하는 질의입니다.
- 다중 열(Multiple-Column) 서브쿼리 : SELECT문장으로부터 하나 이상의 컬럼을 검색하는 질의입니다.
- FROM절상의 서브쿼리(INLINE VIEW) : FROM절상에 오는 서브쿼리로 VIEW처럼 작용 합니다.
- 상관관계 서브 쿼리 : 바깥쪽 쿼리의 컬럼 중의 하나가 안쪽 서브쿼리의 조건에 이용되는 처리 방식 입니다.

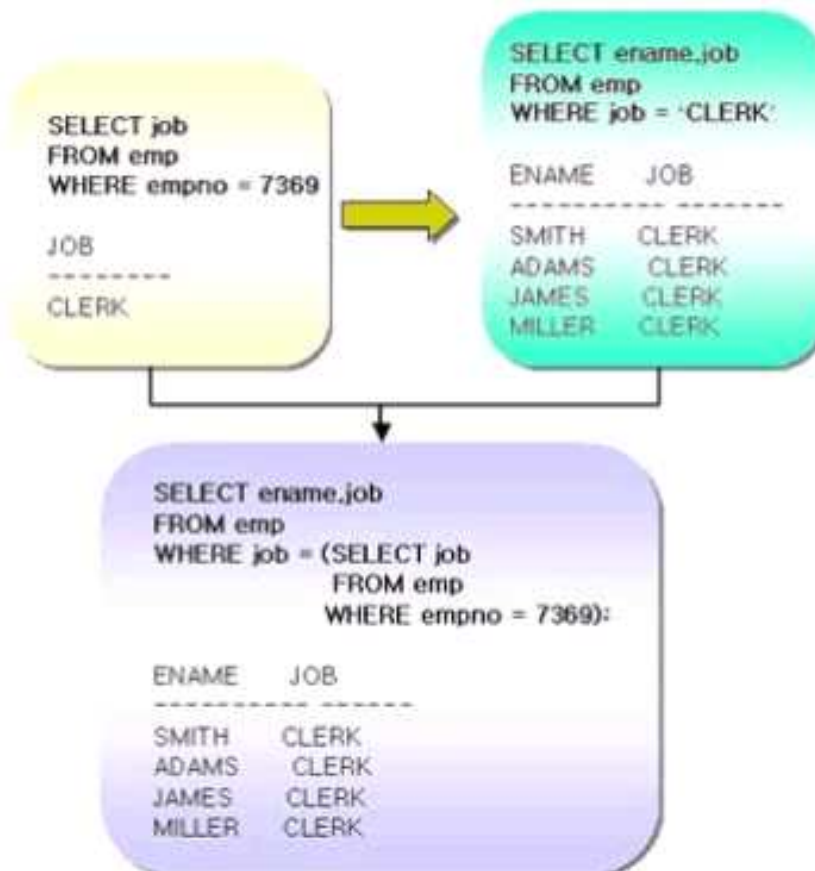
7.서브쿼리(2) [단일 행 서브쿼리]

7.2 단일 행(Sing-Row) 서브쿼리

- 오직 한개의 행(값)을 반환 합니다.(Return only one row)
- 단일 행 연산자(=,>,>=,<,<=,<>,<>!=)만 사용할 수 있습니다.

```
SQL> SELECT ename, job FROM emp
      WHERE job = (SELECT job
                   FROM emp
                   WHERE empno = 7369);
```

위의 서브쿼리를 실행 시키면, 아래의 그림처럼, empno가 7369인 job을 먼저 검색하고 job이 'CLERK'인 사원의 이름과 직업을 반환 합니다.



[실습]

사원테이블에서 scott의 급여보다 많은 사원의 사원번호, 이름, 업무, 급여를 출력하세요.

먼저

```
SELECT sal FROM emp WHERE ename='SCOTT';
```

 결과 ==>3000

와

```
SELECT empno,ename,job,sal FROM emp WHERE sal > 3000
```

을 합쳐서 아래와 같이

```
SELECT empno,ename,job,SAL FROM EMP WHERE SAL >
(SELECT sal FROM EMP WHERE ename='SCOTT');
```

[문제1] 사원테이블에서 사원명, 입사일, 담당업무, 부서코드를 선택하되 'ADAMS'와 같은 업무를 하는 사원이거나 'SCOTT'과 같은 부서인 사원을 선택하라.

```
SELECT ENAME, HIREDATE, JOB, DEPTNO FROM EMP
WHERE JOB=(SELECT JOB FROM EMP WHERE ENAME='ADAMS' OR
DEPTNO=(SELECT DEPTNO FROM EMP WHERE ENAME='SCOTT');
```

[문제2] 사원테이블에서 사원번호가 7521인 사원과 업무가 같고 급여가 7934인 사원보다 많은 사원의 사번,이름,업무,입사일자,급여를 출력하세요.

```
SELECT empno,ename,job,hiredate, sal FROM EMP
WHERE job =(SELECT job FROM EMP WHERE empno=7521) AND
sal > (SELECT sal FROM EMP WHERE empno=7934);
```

* SUBQUERY에서 그룹함수 사용

4_1) 단일행을 반환하는 subquery

단일행을 반환하는 subquery에 그룹함수를 사용하여 메인쿼리에서 데이터를 출력할 수 있다.

[실습] 사원테이블에서 급여의 평균보다 적은 사원의 사번, 이름, 업무, 급여, 부서번호를 선택하라.

```
SELECT empno,ename,job,sal,deptno FROM EMP  
WHERE sal < (SELECT AVG(sal) FROM EMP);
```

4_2) subquery를 갖는 having절

[실습] 사원테이블에서 사원의 급여가 20번부서의 최소급여보다 많이 받는 부서를 선택하라.

- > 1. 먼저 부서별 최소급여를 구한다.
2. 최소급여가 20번부서의 최소급여 보다 큰 조건식을 세운다.

```
SELECT deptno, MIN(sal) FROM EMP  
GROUP BY deptno  
HAVING MIN(sal) >  
(SELECT MIN(sal) FROM EMP WHERE deptno=20);
```

[문제1] 사원테이블의 사원의 급여가 10번부서의 평균급여보다 많이 받는 업무를 하는 업무별 평균급여를 선택하라.

7. 서브쿼리(3) [다중 행 서브쿼리]

7.3 다중 행(Multiple-Row) 서브쿼리

- 하나 이상의 행을 Return하는 Subquery를 다중 행 Subquery라고 합니다.
- 복수 행 연산자(**IN, NOT IN, ANY, ALL, EXISTS**)를 사용 할 수 있습니다.

IN연산자의 사용 예제

- 부서별로 급여를 가장 많이 받는 사원과 같은 급여를 받는 사원의 사원번호, 사원명, 급여, 부서코드를 출력하라.

```
SQL> SELECT empno, ename, job, sal, deptno
      FROM emp
      WHERE sal IN (SELECT MAX(sal)
                    FROM emp
                    GROUP BY deptno);
```

| | empno | ename | job | sal | |
|---|-------|-------|-----------|---------|----|
| ▶ | 7698 | BLAKE | MANAGER | 2850.00 | 30 |
| | 7788 | SCOTT | ANALYST | 3000.00 | 20 |
| | 7839 | KING | PRESIDENT | 5000.00 | 10 |
| | 7902 | FORD | ANALYST | 3000.00 | 20 |

[문제] 업무별로 최대 급여를 받는 사원과 같은 급여를 받는 사원의 사원번호와 이름, 업무, 급여를 출력하세요.

**** ANY 연산자의 사용 예제 ****

ANY 연산자는 서브쿼리의 결과값 중 어느 하나의 값이라도 만족이 되면 결과값을 반환 합니다.

업무가 'SALESMAN'인 사원의 최소급여보다

많으면서 부서번호가 20번이 아닌 사원의 이름과 급여, 부서코드를 출력하라.

```
SQL> SELECT ename, sal, deptno FROM emp
      WHERE deptno !=20
      AND sal >ANY(SELECT sal FROM emp WHERE job='SALESMAN');
```

| | ename | sal | deptno |
|---|--------|---------|--------|
| ▶ | CLARK | 2450.00 | 10 |
| | KING | 5000.00 | 10 |
| | MILLER | 1300.00 | 10 |
| | ALLEN | 1600.00 | 30 |
| | BLAKE | 2850.00 | 30 |
| | TURNER | 1500.00 | 30 |

[문제] 사원테이블의 사원중 KING속한 부서의 사원보다 늦게 입사한 사원의 사원명, 업무, 급여, 입사일을 선택하라.

**** ALL 연산자의 사용 예제 ****

- ALL연산자는 서브쿼리의 **결과 값 중 모든 결과 값이 만족** 되어야만 결과값을 반환한다.

업무가 'SALESMAN'인 사원의 최대급여보다

많으면서 부서번호가 20번이 아닌 사원의 이름과 급여를 출력하라.

```
SQL> SELECT ename, sal FROM emp
      WHERE deptno != 20
      AND sal > ALL(SELECT sal FROM emp WHERE job='SALESMAN');
```

| | ename | sal |
|---|-------|---------|
| ▶ | CLARK | 2450.00 |
| | KING | 5000.00 |
| | BLAKE | 2850.00 |

**** EXISTS 연산자의 사용 예제 ****

- EXISTS 연산자를 사용하면 서브쿼리의 데이터가 존재하는가의 여부를 먼저 따져 존재하는 값들만을 결과로 반환해 줍니다.

- SUBQUERY에서 적어도 1개의 행을 RETURN하면 논리식은 참이고 그렇지 않으면 거짓입니다.

예제) 사원을 관리할 수 있는 사원의 정보를 선택한다.

```
SQL> SELECT empno, ename, sal
      FROM emp e
      WHERE EXISTS (SELECT empno FROM emp WHERE e.empno = mgr);
```

| | empno | ename | sal |
|---|-------|-------|---------|
| ▶ | 7566 | JONES | 2975.00 |
| | 7698 | BLAKE | 2850.00 |
| | 7782 | CLARK | 2450.00 |
| | 7788 | SCOTT | 3000.00 |
| | 7839 | KING | 5000.00 |
| | 7902 | FORD | 3000.00 |

7. 서브쿼리(4) [다중 열 서브쿼리]

7.4 다중 열(Multiple-Column) 서브쿼리

- 다중 열 서브쿼리란 서브쿼리의 결과값이 두개 이상의 컬럼을 반환하는 서브쿼리 입니다.

**** Pairwise(쌍비교) Subquery ****

- 서브쿼리가 한번 실행되면서 모든 조건을 검색해서 주 쿼리로 넘겨 줍니다.

사원테이블에서 급여와 보너스가 부서 30에 있는 사원의 급여, 보너스와 일치하는 사원의 이름, 부서번호, 급여, 보너스를 출력하세요.

```
SQL> SELECT empno, sal, deptno
      FROM emp
      WHERE (sal, deptno) IN (SELECT sal, deptno
                             FROM emp
                             WHERE deptno = 30
                             AND comm is NOT NULL);
```

| | empno | sal | deptno |
|---|-------|---------|--------|
| ▶ | 7499 | 1600.00 | 30 |
| | 7521 | 1250.00 | 30 |
| | 7654 | 1250.00 | 30 |
| | 7844 | 1500.00 | 30 |

[문제]업무별로 최소 급여를 받는 사원의 사번, 이름, 업무, 부서번호를 출력하세요.
단, 업무별로 정렬하세요.

**** Nonpairwise(비쌍비교) Subquery ****

- 서브쿼리가 여러 조건별로 사용 되어서 결과값을 주 쿼리로 넘겨 줍니다.

```
SQL> SELECT empno, sal, deptno
      FROM emp
      WHERE sal IN (SELECT sal
                    FROM emp
                    WHERE deptno = 30
                    AND comm is NOT NULL)
      AND deptno IN (SELECT deptno
                    FROM emp
                    WHERE deptno= 30
                    AND comm is NOT NULL);
```

| EMPNO | SAL | DEPTNO |
|-------|------|--------|
| 7521 | 1250 | 30 |
| 7654 | 1250 | 30 |
| 7844 | 1500 | 30 |
| 7499 | 1600 | 30 |

7. 서브쿼리(5) [FROM절상의 서브쿼리]

7.5 FROM절상의 서브쿼리(INLINE VIEW)

SUBQUERY는 FROM절에서도 사용 가능하다.

하나의 테이블에서 자료의 양이 많을 경우 FROM절에 테이블 전체를 기술하여 사용하면 효율이 떨어질 수 있다.

이런 경우 필요한 행과 열만 선택하여 FROM절에 기술하면 오라클 서버가 최적화 단계에서 효율적인 검색을 할 수 있다.

이처럼 FROM절에 기술한 서브쿼리는 마치 VIEW와 같은 역할을 한다. 이런 것을 VIEW라고 한다.

예제] 급여가 20부서의 평균 급여보다 크고 사원을 관리하는 사원으로서 20부서에 속하지 않은 사원의 정보를 보여주는 SQL문 입니다.

```
SQL> SELECT b.empno, b.ename, b.job, b.sal, b.deptno, b.mgr
      FROM (SELECT empno
            FROM emp
            WHERE sal > (SELECT AVG(sal) FROM emp WHERE deptno = 20)) a, emp b
      WHERE a.empno = b.empno
      AND b.mgr is NOT NULL
      AND b.deptno != 20;
```

| | empno | ename | job | sal | mgr | deptno |
|---|-------|-------|---------|---------|------|--------|
| ▶ | 7698 | BLAKE | MANAGER | 2850.00 | 7839 | 30 |
| | 7782 | CLARK | MANAGER | 2450.00 | 7839 | 10 |

[실습]

EMP와 DEPT 테이블에서 업무가 MANAGER인 사원의 이름, 업무, 부서명, 근무지를 출력하세요.

- 아래는 inline view를 이용하여

```
SELECT ENAME, JOB, A.DEPTNO, B.LOC FROM (
SELECT ENAME, JOB, DEPTNO FROM EMP WHERE JOB='MANAGER') A, DEPT B
WHERE A.DEPTNO=B.DEPTNO;
```

- 또는 join절을 이용해서 구할 수도 있다.

```
SELECT  ENAME,JOB,A.DEPTNO,LOC FROM EMP A
JOIN DEPT B ON A.DEPTNO= B.DEPTNO
AND A.JOB='MANAGER';
```