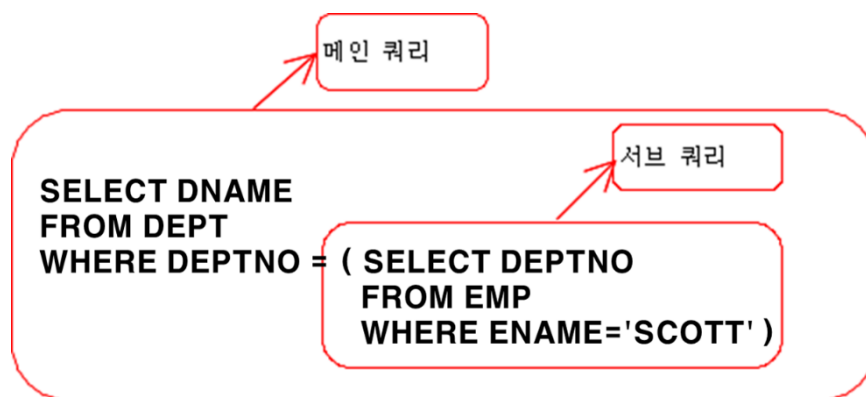


8장 서브 쿼리

8.1 서브 쿼리의 기본 개념

- 서브 쿼리는 하나의 SELECT 문장의 절 안에 포함된 또 하나의 SELECT 문장이다.
- 서브 쿼리를 포함하고 있는 쿼리문을 메인 쿼리, 포함된 또 하나의 쿼리를 서브 쿼리라 한다.
- 서브 쿼리는 비교 연산자의 오른쪽에 기술해야 하고 반드시 괄호로 둘러싸아야 한다
- 서브 쿼리는 메인 쿼리가 실행되기 이전에 한번만 실행이 된다.



8.2 단일행 서브 쿼리

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

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ( SELECT AVG(SAL)
               FROM EMP);
```

8.3 다중행 서브 쿼리

- 다중 행 서브 쿼리는 서브 쿼리에서 반환되는 결과가 하나 이상의 행일 때 사용하는 서브 쿼리이다. 다중 행 서브 쿼리는 반드시 다중 행 연산자(Multiple Row Operator)와 함께 사용해야 한다.

종류	의미
IN	메인 쿼리의 비교 조건('=' 연산자로 비교할 경우)이 서브 쿼리의 결과 중에서 하나라도 일치하면 참이다.

ANY, SOME	메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 하나 이상이 일치하면 참이다.
ALL	메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 모든 값이 일치하면 참이다.
EXIST	메인 쿼리의 비교 조건이 서브 쿼리의 결과 중에서 만족하는 값이 하나라도 존재하면 참이다.

8.3.1 IN 연산자

- IN 연산자는 메인 쿼리의 비교 조건에서 서브 쿼리의 출력 결과와 하나라도 일치하면 메인 쿼리의 WHERE절이 참이 되는 연산자이다.

```
SELECT ENAME, SAL, DEPTNO
FROM EMP
WHERE DEPTNO IN ( SELECT DISTINCT DEPTNO
FROM EMP
WHERE SAL >= 3000 );
```

8.3.2 ALL 연산자

- ALL 조건은 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 모든 값이 일치하면 참이다.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ALL (SELECT SAL
FROM EMP
WHERE DEPTNO = 30);
```

8.3.3 ANY 연산자

- ANY 조건은 메인 쿼리의 비교 조건이 서브 쿼리의 검색 결과와 하나 이상만 일치하면 참이다.

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL > ANY ( SELECT SAL
FROM EMP
WHERE DEPTNO = 30 );
```

8.3.4 EXISTS 연산자

- 서브쿼리에 결과 값이 하나 이상 존재하면 조건식이 모두 true, 존재하지 않으면 모두 false가 되는 연산자이다.

```
SELECT *
FROM EMP
WHERE EXISTS (SELECT DNAME FROM DEPT WHERE DEPTNO = 10);

SELECT *
```

```
FROM EMP
WHERE EXISTS (SELECT DNAME FROM DEPT WHERE DEPTNO = 50);
```

```
CONN MADANG/MADANG
```

```
SELECT NAME, ADDRESS
FROM CUSTOMER CS
WHERE EXISTS (SELECT * FROM ORDERS OD WHERE CS.CUSTID=OD.CUSTID);
```

```
-- EXISTS 연산자를 사용하여 대한민국에 거주하는 고객에게 판매한 도서의 총 판매액을 구하기
SELECT SUM(SALEPRICE) "TOTAL"
FROM ORDERS OD
WHERE EXISTS (
    SELECT *
    FROM CUSTOMER CS
    WHERE ADDRESS LIKE '%대한민국%' AND CS.CUSTID=OD.CUSTID
);
```

[과제] 과제-08-01.TXT

SQL> CONN ORA_USER/HONG 로 접속하여 SQL문을 작성하세요.

1. 101번 사원에 대해 아래의 결과를 산출하는 쿼리를 작성해 보자.

사번	사원명	job명칭	job시작일자	job종료일자	job수행부서명
----	-----	-------	---------	---------	----------

<정답>

2. 다음의 쿼리를 ANSI 문법으로 변경해 보자.

```
SELECT A.DEPARTMENT_ID, A.DEPARTMENT_NAME
FROM DEPARTMENTS A, EMPLOYEES B
WHERE A.DEPARTMENT_ID = B.DEPARTMENT_ID
AND B.SALARY > 3000
ORDER BY A.DEPARTMENT_NAME;
```

<정답>

[과제] 과제-08-02.TXT

1. 다음의 두 쿼리를 ANY, ALL을 사용해서 동일한 결과를 추출하도록 변경해보자.
(단, SQL> CONN ORA_USER/HONG 로 접속하여 SQL문을 작성하세요.)

```
SELECT EMPLOYEE_ID, SALARY
FROM EMPLOYEES
WHERE SALARY IN (2000, 3000, 4000)
ORDER BY EMPLOYEE_ID;
```

```
SELECT EMPLOYEE_ID, SALARY
FROM EMPLOYEES
WHERE SALARY NOT IN (2000, 3000, 4000)
```

ORDER BY EMPLOYEE_ID;

<정답>

2. IN연산자를 이용하여 부서별로 가장 많은 급여를 받는 사원의 정보(사원번호,사원명,급여,부서번호)를 출력하는 SQL문을 작성 하세요?
(단, SQL> CONN SCOTT/TIGER 로 접속하여 SQL문을 작성하세요.)

<정답>