

<예제-1>

“갑을병의 직책과 같은 직책을 갖는 직원들의 이름과 직책을 모두 검색하라”

```
SELECT emp_name, title
FROM employee
WHERE title = (SELECT title
               FROM employee
               WHERE emp_name = '갑을병')
```

Advice

nested(or embeded) query는 넓게는 위 그림의 내부질의와 외부질을 합친 뜻으로 쓰이기도 하고 좁게는 ‘내포된 질의’ 즉, 위 그림의 ‘내부질의’의 뜻으로 쓰이기도 한다. 여기서는 nested query를 ‘중첩질의’라고 부를 것이며, 특별한 언급이 없는 한, ‘외부질의 + 내부질의’의 뜻으로 사용하고, subquery(or nested subquery)를 ‘내부질의’의 뜻으로 사용한다. nested query는 포대기형 질의, 함유된 질의 등으로 불려지기도 한다. 명칭이야 어찌됐든, ‘query 속의 query’에는 어떤 형태가 있으며 어떻게 쓰이고 있는지가 이 단원의 핵심이다.

<예제-1>

“갑을병과 같은 직책을 갖는 직원들의 이름과 직책을 모두 검색하라”

<employee>테이블에서 갑을병의 직책이 과장이므로 <예제-1>의 query는

“직책이 과장인 직원들의 이름과 직책을 모두 검색하라”

라고 하면 될 것 같은데, 왜 이리 복잡하게 중첩질의를 만들어서 표현하는 것일까. 이것은 우리가 query를 작성할 때에 ‘갑을병’의 직책을 아직 모를 수도 있고, 또는 ‘갑을병’의 직책이 지금 과장인 것은 알고 있지만, 직책의 변화가 예상되고 직책이 변화되었을 때에도 역시 이 query가 유효하려면 <예제-1>과 같이 작성하는 것이 보다 적절하기 때문이다. 보다 근본적으로 말하자면 우리가 query를 작성할 때, 다른 query의 결과로 나온 값이나 애트리뷰트, 테이블 등을 참조해야 하는 경우가 있다. 이러한 경우 query 속에 query가 있는 중첩질의가 나타나게 되는 것이다. 외부질의에 포함되는 내부질의는 그 자체로 다시 내부질의를 가질 수도 있다.. 중첩질의는 <예제-1>의 형태뿐만 아니라, 다른 형태도 추가적으로 가지고 있으며, 매우 다양한 query에 쓰여진다.

서브쿼리 종류

네스티드 서브쿼리(Nested Subquery)

서브쿼리가 where 절에서 사용된 경우 Nested 서브쿼리라고 한다.

Nested 서브쿼리는 독립적으로 실행된 후, 메인쿼리는 그 결과를 이용하여 실행되는 형태를 취한다.

(서브쿼리 1번만 실행)

예를들어 영업부에서 일하는 사원을 출력하고 싶은 경우 아래와 같은 질의문을 작성할 수 있다.

```
select empno, ename
```

```
from emp
```

```
where deptno( select deptno from dept
```

```
where dname = 'sales');
```

(deptno에 인덱스가 있어야지만 서브쿼리가 먼저 돈다)