

6장 서브쿼리 향상

목적

- 다중 열 서브쿼리를 작성한다.
- 널 값이 검색되었을 때 서브쿼리의 수행을 기술하고 설명한다.
- **FROM**절에서 서브쿼리를 작성한다.

다중 열 서브쿼리

Main query
MANAGER 10

Subquery
SALESMAN 30
MANAGER 10
CLERK 20

```
SELECT column, column, .....  
FROM table  
WHERE (column, column, ....) IN  
      (SELECT column, column, .....  
       FROM table  
       WHERE condition);
```

다중 열 서브쿼리 사용

- 급여와 보너스가 부서 **30**에 있는 어떤 종업원의 보너스와 급여와 같은 종업원의 이름, 부서 번호, 급여 그리고 보너스를 디스플레이 한다.

```
SQL> SELECT  ename, deptno, sal, comm
2    FROM    emp
3    WHERE    (sal, NVL(comm,-1)) IN
4                SELECT sal, NVL(comm-1)
5                FROM    emp
6                WHERE deptno = 30 ;
```

열 비교

Pairwise	
SAL	COMM
1600	300
1250	500
1250	1400
2850	
1500	0
950	

Nonpairwise	
SAL	COMM
1600	300
1250	500
1250	1400
2850	
1500	0
950	

다중 열 서브쿼리에서의 열 비교는 pairwise비교 또는 nonpairwise비교이다. 앞장의 예는 pairwise이고 WHERE절에서 실행된다. SELECT문자의 각 후보행은 부서 30에있는 종업원의 급여와 보너스에 일치한다.

Nonpairwise 비교 서브쿼리

- 급여와 보너스가 부서 **30**에 있는 어떤 종업원의 보너스와 급여에 일치하는 종업원의 이름, 부서 번호, 급여 그리고 보너스를 디스플레이 한다.

```
SQL> SELECT      ename, deptno, sal, comm
2      FROM        emp
3      WHERE      sal IN      ( SELECT sal
4                                FROM emp
5                                WHERE deptno = 30 ) ;
6      AND
7      NVL(comm, -1) IN ( SELECT NVL(comm, -1)
8                                FROM emp
9                                WHERE deptno = 30 ) ;
```

ENAME	DEPTNO	SAL	COMM
JAMES	30	950	
BLAKE	30	2850	
TURNER	30	1500	0
ALLEN	30	1600	300
WARD	30	1250	500
MARTIN	30	1250	1400

EMP 테이블 변경

- 종업원 **CLERK**의 급여와 보너스를 부서 **30**에 있는 어떤 종업원의 급여와 같고, 부서 **30**에 있는 다른 종업원의 보너스와 같게 수정한다고 가정
- **CLERK**의 급여는 **Turner(\$1500)**의 급여와 같고, **CLERK**의 보너스는 **Allen(\$300)**의 보너스와 같다.

ENAME	SAL	COMM
....		
CLERK	1500	300
....		
ALLEN	1600	300
TURNER	1500	0
....		

Pairwise 서브쿼리

```
SQL> SELECT  ename, deptno, sal, comm
2    FROM    emp
3    WHERE    (sal, NVL(comm,-1)) IN
4
5              SELECT sal, NVL(comm-1)
6              FROM    emp
              WHERE deptno = 30 ;
```

ENAME	DEPTNO	SAL	COMM
JAMES	30	950	
WARD	30	1250	500
MARTIN	30	1250	1400
TURNER	30	1500	0
ALLEN	30	1600	300
BLAKE	30	2850	

Nonpairwise 서브쿼리

```
SQL> SELECT      ename, deptno, sal, comm
2      FROM      emp
3      WHERE      sal  IN      ( SELECT sal
4                                FROM emp
5                                WHERE deptno = 30 ) ;
6      AND
7      NVL(comm, -1) IN ( SELECT NVL(comm, -1)
8                          FROM emp
9                          WHERE deptno = 30 ) ;
```

ENAME	DEPTNO	SAL	COMM
JAMES	30	950	
BLAKE	30	2850	
TURNER	30	1500	0
CLERK	10	1500	300

서브쿼리에서의 NULL값

```
SQL> SELECT      employee.ename
2  FROM          emp employee
3  WHERE         employee.empno NOT IN
4                ( SELECT manager . mgr
5                  FROM    emp manager) ;
```

위의 SQL문장은 어떤 종속하는 직원을 가지지 않는 종업원을 디스플레이 하려고 한다. 논리적으로 이 SQL문장은 8개의 행을 리턴해야 한다. 그러나 SQL문장은 어떤 행도 리턴하지 않는다. 내부 질의에 의해 리턴된 값 중의 하나가 null값이므로 전체 질의는 행이 없다고 리턴한다. 이 유는 null값을 비교하는 모든 조건은 null이기 때문이다. 그래서 서브쿼리의 결과 집합의 일부가 null값일 때 NOT IN연산자를 사용하지 마십시오. NOT IN연산자는 !=ALL과 똑같다.

서브쿼리 결과 집합의 일부분으로서의 null값은 IN연산자를 사용할때는 문제가 되지 않는다. IN 연산자는 =ANY와 똑같다.

FROM 절에서 서브쿼리 사용

```
SQL> SELECT      a.ename, a.sal, a.deptno, b.salavg
2      FROM      emp a, ( SELECT deptno, avg(sal) salavg
3                      FROM      emp
4                      GROUP BY deptno) b
5      WHERE      a.deptno = b.deptno
6      AND        a.sal > b.salavg;
```

ENAME	SAL	DEPTNO	SALAVG
KING	5000	10	2916.6667
JONES	2975	20	2175
SCOTT	3000	20	2175
.....			

요 약

- 다중 열 서브쿼리는 하나 이상의 열을 리턴한다.
- 다중 열 비교에서의 열 비교는 **pairwise** 또는 **nonpairwise**일 수 있다.
- 또한 다중 열 서브쿼리는 **SELECT** 문장의 **FROM** 절에서 사용될 수 있다.

SQL: 1999의 스칼라 서브 쿼리

- 스칼라 서브 쿼리는 쿼리 수식으로부터 유도된 스칼라 값을 지정하기 위해 사용됩니다.
- **Oracle8i**에서는 다음과 같이 제한된 경우에만 지원이 되었습니다.
 - 삽입 문장의 **VALUES** 리스트
 - 데이터 형 생성자에 대한 인자
- **Oracle9i**에서 스칼라 서브 쿼리는 유효한 수식이 쓰일 수 있는 모든 곳에서 사용될 수 있습니다.
- 스칼라 서브 쿼리는 오직 하나의 값만 반환합니다.
- 반환되는 값의 데이터 형은 서브 쿼리에서 선택되는 데이터 형과 일치합니다.

스칼라 서브 쿼리의 사용

- GROUP BY를 제외한 모든 SELECT 절
- INSERT 문의 VALUES 절
- UPDATE 문의 SET 절 및 WHERE 절
- DECODE 및 CASE의 조건 내지 수식

예: SELECT 리스트에서의 서브 쿼리

Oracle SQL에서의 스칼라 서브 쿼리는 **SELECT** 리스트의 항목으로 사용될 수 있습니다.

```
SELECT employee_id, last_name, (  
    SELECT department_name  
    FROM departments d  
    WHERE e.department_id = d.department_id  
    ) department_name  
FROM employees e  
ORDER BY department;
```

예: WHERE 절에서의 스칼라 서브 쿼리

Oracle SQL: 1999에서는 스칼라 서브 쿼리를 **WHERE** 절의 조건에서 쓸 수 있습니다.

```
SELECT employee_id, last_name
FROM employees e
WHERE ((SELECT location_id
        FROM departments d
        WHERE e.department_id = d.department_id)
      =
      (SELECT location_id
        FROM locations l
        WHERE STATE_province = 'California'))
;
```


예: ORDER BY 절에서의 스칼라 서브 쿼리

스칼라 서브 쿼리를 ORDER BY 절에서도 사용할 수 있습니다.

```
SELECT employee_id, last_name
  FROM employees e
ORDER BY (
  SELECT department_name
    FROM departments d
   WHERE e.department_id = d.department_id
) ;
```

예: CASE 수식에서의 스칼라 서브 쿼리

스칼라 서브 쿼리는 DECODE 및 CASE 수식에서도 쓰일 수 있습니다.

```
SELECT employee_id, last_name,  
       (CASE  
         WHEN department_id IN  
           (SELECT department_id FROM departments  
            WHERE location_id = 1800)  
         THEN 'Canada' ELSE 'other'  
       END) location  
FROM employees;
```