

서브쿼리 (Subquery)

서브쿼리 (Subquery)



서브쿼리 (Subquery)

1. 서브쿼리(Subquery) 개요

1.1 서브쿼리 개념

- ☐ 서브쿼리는 다른 **SELECT** 문장의 절에 **내장된 SELECT 문**을 의미
- ☐ 서브쿼리는 메인 쿼리 이전에 한번만 실행
- ☐ 서브쿼리의 결과는 메인 쿼리에 의해 사용
- ☐ **WHERE, HAVING, FROM** 절에 서브쿼리 작성

```
SELECT    select_list
FROM      table
WHERE     expr operator
          (SELECT    select_list
           FROM      table);
```

서브쿼리 (Subquery)

1.2 서브쿼리 사용 지침

- ❑ 서브쿼리는 괄호로 둘러싸야 함
- ❑ 서브쿼리는 비교 연산자의 다음에 위치
- ❑ 서브쿼리에 **ORDER BY** 절을 포함할 수 없음 (**Top-N** 분석 제외)
- ❑ 단일 행 (**Single-row**) 서브쿼리에는 단일 행 비교 연산자 사용
- ❑ 다중 행 (**Multiple-row**) 서브쿼리에는 다중 행 비교 연산자 사용



서브쿼리 (Subquery)

1.3 서브쿼리 종류

☐ 단일 행 (Single-row) 서브쿼리

메인쿼리 (Main query)

서브쿼리

return

KING

☐ 다중 행 (Multiple-row) 서브쿼리

메인쿼리 (Main query)

서브쿼리

return

KING
SCOTT

☐ 다중 열 (Multiple-column) 서브쿼리

메인쿼리 (Main query)

서브쿼리

return

KING 5000
SCOTT 3000

서브쿼리 (Subquery)

2. 단일 행(Single-row) 서브쿼리

2.1 단일 행 서브쿼리 개념

- ☐ 하나의 행만 반환
- ☐ 단일 행 비교 연산자 사용


연산자	의 미
=	같다
>	보다 크다
>=	보다 크거나 같다
<	보다 작다
<=	보다 작거나 같다
<> (!=)	같지 않다

서브쿼리 (Subquery)

2.2 단일 행 서브쿼리 활용

☐ 서브쿼리에서 단일 값 ('MANAGER') 을 반환하므로 단일 행 서브쿼리

```
SQL> SELECT ename, job
2  FROM emp
3  WHERE job =
4          (SELECT job
5             FROM emp
6             WHERE empno = 7782);
```




ENAME	JOB
JONES	MANAGER
BLAKE	MANAGER
CLARK	MANAGER

서브쿼리 (Subquery)

2.2 단일 행 서브쿼리 활용 (계속)

☐ 메인쿼리와 서브쿼리는 다른 테이블의 데이터를 가져올 수 있음

```
SQL> SELECT ename, job, deptno
2 FROM emp
3 WHERE deptno =
4 (SELECT deptno
5 FROM dept
6 WHERE dname = 'ACCOUNTING') ;
```




ENAME	JOB	DEPTNO
KING	PRESIDENT	10
CLARK	MANAGER	10
MILLER	CLERK	10

서브쿼리 (Subquery)

2.2 단일 행 서브쿼리 활용 (계속)

☐ 단일 행을 반환하는 그룹 함수를 서브쿼리에 사용

```
SQL> SELECT ename, job, sal
2 FROM emp
3 WHERE sal =
4         (SELECT MIN(sal)
5          FROM emp);
```



ENAME	JOB	SAL
SMITH	CLERK	800



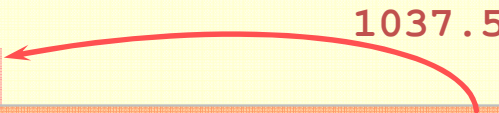
서브쿼리 (Subquery)

2.2 단일 행 서브쿼리 활용 (계속)

- ☐ WHERE 절 뿐만 아니라 HAVING 절에도 서브쿼리를 사용할 수 있음
- ☐ 오라클 서버는 서브쿼리를 실행하여 결과를 메인쿼리의 HAVING 절에 반환

```
SQL> SELECT      job, AVG(sal)
2  FROM          emp
3  GROUP BY      job
4  HAVING         AVG(sal) >
5
6
7                (SELECT      MIN(AVG(sal))
                  FROM          emp
                  GROUP BY      job);
```

1037.5

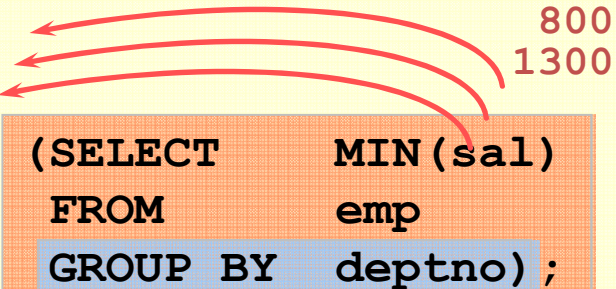


서브쿼리 (Subquery)

2.3 단일 행 서브쿼리 오류

☐ 서브쿼리의 일반적인 오류는 단일 행 서브쿼리에서 여러 행이 반환되는 것

```
SQL> SELECT empno, ename          950  
2 FROM emp          800  
3 WHERE sal =  
4 (SELECT MIN(sal)  
5 FROM emp  
6 GROUP BY deptno); 1300
```



4행에 오류:

ORA-01427: 단일 행 하위 질의에 2개 이상의 행이 리턴되었습니다.



서브쿼리 (Subquery)

다중 행, 다중 열 서브쿼리



서브쿼리 (Subquery)

1. 다중 행(Multiple-row) 서브쿼리

1.1 다중 행 서브쿼리 개요

- 여러 행을 반환
- 단일 행 연산자 대신 다음과 같은 다중 행 비교연산자를 사용

연산자	의 미
IN	목록에 있는 임의의 구성원과 동일
ANY	값을 서브쿼리에 의해 반환된 각 값과 비교
ALL	값을 서브쿼리에 의해 반환된 모든 값과 비교
SOME	ANY 연산자와 동일

서브쿼리 (Subquery)

1.2 다중 행 서브쿼리 활용

- ☐ 서브쿼리가 먼저 실행되어 950, 800, 1300의 3행을 포함하는 질의 결과를 생성

```

SQL> SELECT empno, ename, sal
2  FROM emp
3  WHERE sal IN
4      (SELECT MIN(sal)
5       FROM emp
6       GROUP BY deptno);

```

EMPNO	ENAME	SAL
7900	JAMES	950
7369	SMITH	800
7934	MILLER	1300

서브쿼리 (Subquery)

1.2 다중 행 서브쿼리 활용 (계속)

☐ **ANY** 연산자는 값을 서브쿼리에 의해 반환되는 각각의 값과 비교

- **<ANY** : 최대값보다 작음을 의미
- **>ANY** : 최소값보다 많음을 의미
- **=ANY** : IN과 동일

```
SQL> SELECT empno, ename, job                2975
      2 FROM emp                             2850
      3 WHERE sal < ANY                       2450
      4 (SELECT sal
      5 FROM emp
      6 WHERE job = 'MANAGER' )
      7 AND job <> 'MANAGER' ;
```

서브쿼리 (Subquery)

1.2 다중 행 서브쿼리 활용 (계속)

☐ ALL 연산자는 값을 서브쿼리에 의해 반환되는 모든 값과 비교

- <ALL : 최소값보다 작음을 의미
- >ALL : 최대값보다 많음을 의미

```
SQL> SELECT empno, ename, job
2 FROM emp
3 WHERE sal > ALL
4 (SELECT avg(sal)
5 FROM emp
6 GROUP BY deptno);
```

1566.66667
2175
2916.66667

서브쿼리 (Subquery)

2. 다중 열(Multiple-column) 서브쿼리

2.1 다중 열 서브쿼리 개념

- ☐ 두 개 이상의 열 (column) 을 비교하려면 , 논리 연산자를 사용하여 혼합 WHERE 절로 결합 (Non-pairwise)
- ☐ 다중 열 서브쿼리를 사용하면 , 중복된 WHERE 조건을 단일 WHERE 절로 결합 가능 (Pairwise)

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



서브쿼리 (Subquery)

2.2 Pairwise와 Nonpairwise

- 다중 열 서브쿼리에서의 열 비교는 **Pairwise**(쌍) 비교
- Pairwise** 비교는 "SAL"과 "COMM"이 일치
- Non-pairwise** 비교는 특정 "SAL"에 대해 다양한 "COMM"과 비교

Pairwise

SAL	COMM
-----	-----
5000	500
1250	1400
1300	0

Non-pairwise

SAL	COMM
-----	-----
5000	500
1250	1400
1300	0

서브쿼리 (Subquery)



2.3 다중 열 서브쿼리 활용

```
SQL> UPDATE emp  
2 SET      sal = 1250, comm = 1400  
3 WHERE    empno = 7782;
```

```
SQL> UPDATE emp  
2 SET      comm = 500  
3 WHERE    empno = 7839;
```



서브쿼리 (Subquery)

2.3 다중 열 서브쿼리 활용 (계속)

Pairwise (쌍) 비교

```

SQL> SELECT  ename, sal, comm, deptno
2  FROM      emp
3  WHERE     (sal, NVL(comm,-1)) IN
4
5              (SELECT sal, NVL(comm,-1)
6                 FROM   emp
7                 WHERE  deptno=10)
7  AND       deptno <> 10;

```

ENAME	SAL	COMM	DEPTNO
MARTIN	1250	1400	30

서브쿼리 (Subquery)

2.3 다중 열 서브쿼리 활용 (계속)

Non-pairwise (비쌍) 비교

```

SQL> SELECT  ename, sal, comm, deptno
2  FROM      emp
3  WHERE     sal          IN (SELECT  sal
4                               FROM    emp
5                               WHERE   deptno=10)
6  AND       NVL(comm, -1) IN (SELECT  NVL(comm, -1)
7                               FROM    emp
8                               WHERE   deptno=10)
9  AND       deptno <> 10;

```

Diagram illustrating the execution of the SQL query. Red arrows show the flow of data from the subqueries to the main query conditions. The first subquery (SELECT sal FROM emp WHERE deptno=10) returns values 5000, 1250, and 1300. The second subquery (SELECT NVL(comm, -1) FROM emp WHERE deptno=10) returns values 500, 1400, and -1.

ENAME	SAL	COMM	DEPTNO
MARTIN	1250	1400	30
WARD	1250	500	30

서브쿼리 (Subquery)

3. FROM 절에 서브쿼리 사용

- SELECT 문의 FROM 절에 서브쿼리 사용 가능
- FROM 절에 있는 서브쿼리는 특정 SELECT 문에 대한 데이터 소스를 정의

```
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;
```

