

9.1 서브쿼리 (Subquery)

■ 서브쿼리 정의

- : SELECT 문 (Main) 에 포함되어 있는 또 하나의 별도 SELECT 문 (Subquery)이다.
- : 여러 번의 SELECT문을 수행해야 얻을 수 있는 결과를, 하나의 중첩된 SELECT 문으로 쉽게 얻을 수 있도록 해준다.

■ 서브쿼리 예

- : 사원테이블에서 SCOTT 사원보다 많은 급여를 받는 사원의 이름을 검색하시오.

```
SQL> SELECT SAL  
  2  FROM EMP  
  3  WHERE ENAME = 'SCOTT';
```

```
      SAL  
-----  
      3000
```

```
SQL> SELECT ENAME  
  2  FROM EMP  
  3  WHERE SAL > 3000;
```

```
ENAME  
-----  
KING
```



```
SQL> SELECT ENAME  
  2  FROM EMP  
  3  WHERE SAL > ( SELECT SAL  
  4                  FROM EMP  
  5                  WHERE ENAME = 'SCOTT' );
```

```
ENAME  
-----  
KING
```

9.2 서브쿼리 (Subquery)

■ 서브쿼리 사용방법

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

- 바깥 쪽 쿼리를 Main query , 안쪽 쿼리를 Subquery 라고 한다.
- 서브쿼리가 먼저 실행되고, 그 결과가 메인 쿼리에 전달되어 실행된다.
- 서브쿼리는 WHERE 절 , HAVING , FROM 절에도 사용될 수 있다.
- operator 는 단일행 연산자 (> , = , >= , < , <= , !=)와 복수행 연산자 (IN , ANY , ALL)를 사용 할 수 있다.
- SQL 문장의 서브쿼리가 단일행 서브쿼리인 경우는 단일행 연산자를 사용하고 복수행 서브쿼리인 경우에는 복수행 연산자를 사용해야 한다.
- 서브쿼리에는 반드시 괄호를 사용한다.
- 서브쿼리에는 ORDER BY 절을 사용불가
(SELECT 문장에는 반드시 하나의 ORDER BY 를 사용할 수 있으며, 문장 끝에 사용한다)

9.3 서브쿼리 (Subquery)

■ 서브쿼리 종류

1. 단일행 서브쿼리

: 서브쿼리가 한 개의 행을 리턴.

: 반드시 단일행 연산자를 사용해야 한다. (= , > , < , >= , <= , !=)

```
SQL> SELECT EMPNO, ENAME, JOB
  2 FROM EMP
  3 WHERE JOB = (SELECT JOB
  4               FROM EMP
  5               WHERE EMPNO = 7844);
```

EMPNO	ENAME	JOB
7499	ALLEN	SALESMAN
7521	WARD	SALESMAN
7654	MARTIN	SALESMAN
7844	TURNER	SALESMAN

```
SQL> SELECT ENAME, JOB, SAL
  2 FROM EMP
  3 WHERE JOB = (SELECT JOB
  4               FROM EMP
  5               WHERE EMPNO = 7521)
  6 AND SAL > (SELECT SAL
  7             FROM EMP
  8             WHERE EMPNO = 7900);
```

ENAME	JOB	SAL
ALLEN	SALESMAN	1600
WARD	SALESMAN	1250
MARTIN	SALESMAN	1250
TURNER	SALESMAN	1500

```
SQL> SELECT EMPNO, ENAME, SAL
  2 FROM EMP
  3 WHERE SAL = (SELECT MIN(SAL)
  4               FROM EMP);
```

EMPNO	ENAME	SAL
7369	SMITH	800

```
SQL> SELECT DEPTNO, MIN(SAL)
  2 FROM EMP
  3 GROUP BY DEPTNO
  4 HAVING MIN(SAL) > (SELECT MIN(SAL)
  5                    FROM EMP
  6                    WHERE DEPTNO = 20);
```

DEPTNO	MIN(SAL)
10	1300
30	950

```
SQL> SELECT ENAME
  2 FROM EMP
  3 WHERE SAL = (SELECT MIN(SAL)
  4               FROM EMP
  5               GROUP BY DEPTNO);
WHERE SAL = (SELECT MIN(SAL)
```

*

3행에 오류:

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

9.4 서브쿼리 (Subquery)

■ 서브쿼리 종류

2. 복수행 서브쿼리

: 서브쿼리가 여러 개의 행을 리턴.

: 반드시 복수행 연산자를 사용해야 한다. (IN , ANY ,ALL)

IN 연산자

: WHERE 절에서 사용하는 일반 비교연산자와 동일하다.

: 메인쿼리의 비교 조건이 서브쿼리의 결과 중에서 하나라도 일치하면 검색가능.

```
SQL> SELECT EMPNO, ENAME, SAL, DEPTNO
2 FROM EMP
3 WHERE SAL IN ( SELECT MIN(SAL)
4                 FROM EMP
5                 GROUP BY DEPTNO );
```

EMPNO	ENAME	SAL	DEPTNO
7900	JAMES	950	30
7369	SMITH	800	20
7934	MILLER	1300	10

```
SQL> SELECT ENAME, HIREDATE, DEPTNO
2 FROM EMP
3 WHERE DEPTNO IN ( SELECT DEPTNO
4                     FROM EMP
5                     WHERE ENAME = 'SCOTT' );
```

ENAME	HIREDATE	DEPTNO
FORD	81/12/03	20
ADAMS	87/05/23	20
SCOTT	87/04/19	20
JONES	81/04/02	20
SMITH	80/12/17	20

9.5 서브쿼리 (Subquery)

ALL 연산자

- : 복수행 서브쿼리 결과가 메인 쿼리의 WHERE 절에서 부등호 조건으로 비교될 때 사용된다.
- : 서브쿼리에서 리턴된 모든 결과값이 WHERE 절에서 모두 비교된다.

사용 예

- : 사원 테이블에서 업무가 MANAGER인 사원의 급여보다 적은 급여를 받는 사원들의 이름 검색?

```
SQL> SELECT SAL
2 FROM EMP
3 WHERE JOB = 'MANAGER';
```

SAL
2975
2850
2450

```
SQL> SELECT EMPNO, ENAME, SAL
2 FROM EMP
3 WHERE SAL < 2450;
```

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7654	MARTIN	1250
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7934	MILLER	1300

8 개의 행이 선택되었습니다.

```
SQL> SELECT EMPNO, ENAME, SAL
2 FROM EMP
3 WHERE SAL < ALL (SELECT SAL
4 FROM EMP
5 WHERE JOB = 'MANAGER' );
```

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7654	MARTIN	1250
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7934	MILLER	1300

8 개의 행이 선택되었습니다.

9.6 서브쿼리 (Subquery)

ANY 연산자

- : 복수행 서브쿼리 결과가 메인 쿼리의 WHERE 절에서 부등호 조건으로 비교될 때 사용된다.
- : 서브쿼리에서 리턴된 각각의 결과값이 WHERE 절에서 비교된다.

사용 예

- : 사원 테이블에서 업무가 MANAGER인 사원의 급여보다 많은 급여를 받는 사원들의 이름 검색?

```
SQL> SELECT SAL  
2 FROM EMP  
3 WHERE JOB = 'MANAGER';
```

SAL
2975
2850
2450

```
SQL> SELECT EMPNO, ENAME , SAL  
2 FROM EMP  
3 WHERE SAL > 2450;
```

EMPNO	ENAME	SAL
7566	JONES	2975
7698	BLAKE	2850
7788	SCOTT	3000
7839	KING	5000
7902	FORD	3000

```
SQL> SELECT EMPNO, ENAME , SAL  
2 FROM EMP  
3 WHERE SAL > ANY ( SELECT SAL  
4 FROM EMP  
5 WHERE JOB = 'MANAGER');
```

EMPNO	ENAME	SAL
7839	KING	5000
7902	FORD	3000
7788	SCOTT	3000
7566	JONES	2975
7698	BLAKE	2850

9.7 서브쿼리 (Subquery)

```
SQL> SELECT EMPNO, ENAME , SAL
2 FROM EMP
3 WHERE SAL < ALL ( SELECT SAL
4 FROM EMP
5 WHERE JOB = 'MANAGER');
```

EMPNO	ENAME	SAL
7369	SMITH	800
7499	ALLEN	1600
7521	WARD	1250
7654	MARTIN	1250
7844	TURNER	1500
7876	ADAMS	1100
7900	JAMES	950
7934	MILLER	1300

8 개의 행이 선택되었습니다.

```
SQL> SELECT EMPNO, ENAME , SAL
2 FROM EMP
3 WHERE SAL > ALL ( SELECT SAL
4 FROM EMP
5 WHERE JOB = 'MANAGER');
```

EMPNO	ENAME	SAL
7788	SCOTT	3000
7839	KING	5000
7902	FORD	3000

```
SQL> SELECT SAL
2 FROM EMP
3 WHERE JOB = 'MANAGER';
```

SAL
2975
2850
2450

```
SQL> SELECT EMPNO, ENAME , SAL
2 FROM EMP
3 WHERE SAL < ANY ( SELECT SAL
4 FROM EMP
5 WHERE JOB = 'MANAGER');
```

EMPNO	ENAME	SAL
7369	SMITH	800
7900	JAMES	950
7876	ADAMS	1100
7521	WARD	1250
7654	MARTIN	1250
7934	MILLER	1300
7844	TURNER	1500
7499	ALLEN	1600
7782	CLARK	2450
7698	BLAKE	2850

```
SQL> SELECT EMPNO, ENAME , SAL
2 FROM EMP
3 WHERE SAL > ANY ( SELECT SAL
4 FROM EMP
5 WHERE JOB = 'MANAGER');
```

EMPNO	ENAME	SAL
7839	KING	5000
7902	FORD	3000
7788	SCOTT	3000
7566	JONES	2975
7698	BLAKE	2850

10 개의 행이 선택되었습니다.

9.8 서브쿼리 (Subquery)

1. 사원 테이블에서 BLAKE 보다 급여가 많은 사원들의 사번, 이름 , 급여를 검색하시오.
2. 사원 테이블에서 MILLER 보다 늦게 입사한 사원의 사번, 이름, 입사일을 검색하시오.
3. 사원 테이블에서 사원 전체 평균 급여보다 급여가 많은 사원들의 사번, 이름 , 급여를 검색하시오.
4. 사원 테이블에서 CLARK와 같은 부서이며, 사번이 7698인 직원의 급여보다 많은 급여를 받는 사원들의 사번, 이름, 급여를 검색하시오.
5. 사원 테이블에서 부서별 최대 급여를 받는 사원들의 사번, 이름, 부서코드, 급여를 검색하시오.