Multiple Column Subqueries

Chapter 7

Objectives

After completing this lesson, you should be able to do the followving:

- Write a multiple-column subquery
- Describe and explain the behavior of subqueries when null values are retrieved
- Write a subquery in a FROM clause

Multiple-Column Subqueries

So far *you have* written single-row subqueries and mulliple-row subqueries where only one column was compared in the WHERE clause or HAVING clause of the SELECT statement. If you want compare two or more columns. you must write a compound WHERE clause using logical operators Multiple-column subqueries enable you to combine duplicate WHERE conditions into a single WHERE clause.

Using Multiple-Column Subqueries

Display the order number, product number, and quantity of any item in which the product number and quantity match both the product number and quantity of an item in ordid 365.

```
SELECT  ordid, prodid, qty
FROM item
WHERE       (prodid,   qty)   IN
                        (SELECT prodid,   qty
                         FROM       item
                         WHERE     ordid =   365)
AND   ordid   =   365 ;
```

ORDID	PRODID	QTY
365	84	22

Nonpairwise Comparison Subguery

Display the order number, product number, and quantity of any item in which the product number and quantity match any product number and any quantity of an item in order 605.

```
SELECT    ordid, prodid, qty
 FROM     item
 WHERE    prodid  IN  (SELECT        prodid
                       FROM   Item
                       WHERE        ordid  = 365)
 AND  qty  IN  (SELECT    qty
                FROM      item
                WHERE     ordid = 365)
 AND  ordid  =  365 ;
```

ORDID	PRODID	QTY
365	84	22

Null Values in a Subquery

```
SELECT  employee.ename
FROM    emp  employee
WHERE employee.empno   NOT   IN
                        (SELECT manager.mgr
                         FROM       emp, manager);
```

no rows   selected.

Returning Nulls in the Resulting Set of a Subquery

The SQL statement on the slide attempts to display all the employees who do not have any subordinates. Logically, this SQL statement should have returned single rows. Howevever, the SQL statement does not return any rows. One of the values returned by the inner query is a null value and hence the entire query returns no rows. The reason is that all conditions that compare a null value result in a null. So whenc \er null values all; likely to be part of the result and set of a subquery, do not use the NOT IN operator. The NOT IN operator is equivalent to != ALL

Notice that the null value as part of the resultant set of a subquery will not be a problem ıt'you Lire using the IN operator The IN operator is equivalent to = ANY. For example, to display the employees who have subordinate use the following SQL statement.

```
SELECTemployee.ename
FROM    emp employee
WHERE employee.empno IN
                        (SELECT manager.mgr
                        FROM emp manager);
```

ENAME
FORD
BLAKE
KING
JONES
SCOTT
CLARK

6 rows selected.

Using a Subquery in the FROM Clause

```
SELECT     a.ename,  a.sal,  a.deptno,  b.salavg
FROM emp  a,  (SELECT  deptno,  avg(sal)  salavg
                       FROM         emp
                       GROUP BY  deptno)  b
WHERE   a.deptno = b.deptno
AND
 a.sal  >  b.salavg;
```

ENAME	SAL	DEPTNO	SALAVG
ALLEN	1600	30	1566,66667
JONES	2975	20	2175
BLAKE	2850	30	1566,66667
SCOTT	3000	20	2175
KING	5000	10	2916,66667
FORD	3000	20	2175

6 rows selected.

Summary

- A multiple-column subquery returns more than one column.
- Column comparisons in multiple-column comparisons can be pairwise or nonpairwise.
- A multiple-column subguery can also be used in the FROM clause of a SELECT statement.

Exercices

1. Write a query to display the name, department number, and salary of any employee whose department number and salary match the department number and salary of any employee who earns a commission.

```
SELECT a.ename, a.deptno, a.sal
FROM emp a
WHERE (deptno, sal)  IN  (SELECT deptno, sal
                  FROM emp
                  WHERE comm IS NOT NULL ) ;
```

ENAME	DEPTNO	SAL
ALLEN	30	1600
MARTIN	30	1250
WARD	30	1250
TURNER	30	1500

Tis is equivalent to the script:

```
SELECT a.ename, a.deptno, a.sal
FROM emp a
WHERE comm IS NOT NULL  ;
```

2. Display the name, department name, and salary of any employee whose salary and commission match the salary and commission of any employee located in Dallas.

Solution by equijoin:

```
SELECT e.ename, d.dname, e.sal, d.loc
   FROM emp e, dept d
   WHERE e.deptno = d.deptno
   AND
   d.loc = 'DALLAS' ;
```

ENAME	DNAME	SAL	LOC
SMITH	RESEARCH	800	DALLAS
JONES	RESEARCH	2975	DALLAS
SCOTT	RESEARCH	3000	DALLAS
ADAMS	RESEARCH	1100	DALLAS
FORD	RESEARCH	3000	DALLAS

Solution by subquery:

```
SELECT ename, dname, sal, loc
   FROM emp e, dept d
   WHERE (sal , comm)  IN
                        ( SELECT sal, comm
                          FROM emp
                          WHERE d.loc = 'DALLAS');
```

ENAME	DNAME	SAL	LOC
ALLEN	RESEARCH	1600	DALLAS
WARD	RESEARCH	1250	DALLAS
MARTIN	RESEARCH	1250	DALLAS
TURNER	RESEARCH	1500	DALLAS

2. Create a query to display the name, hiredate, and salary of any employee who have both the same salary and commission as Scott.

```
SELECT ename, hiredate, sal
   FROM emp
   WHERE
   ename <> 'SCOTT'
AND
(sal, NVL(comm,0) )  IN
                ( SELECT sal, NVL(comm,0)
                  FROM emp
                  WHERE ename = 'SCOTT');
```

ENAME	HIREDATE	SAL
FORD	03/12/1981	3000

3. Create a query to display the employees that earn a salary that is higher than the salary of all of the clerks. Sort the results on salary from highest to lowest.

```
SELECT ename, job, sal
   FROM emp
   WHERE sal  >  ALL
                        ( SELECT sal
                          FROM emp
                          WHERE job = 'CLERK');
```

ENAME	JOB	SAL
ALLEN	SALESMAN	1600
JONES	MANAGER	2975
BLAKE	MANAGER	2850
CLARK	MANAGER	2450
SCOTT	ANALYST	3000
KING	PRESIDENT	5000
TURNER	SALESMAN	1500
FORD	ANALYST	3000

8 rows selected.