Experiment 8

Objectives: Write SQL Queries for Sub queries, Nested queries

Subqueries

Using a Subquery to Solve a Problem

"Who has a salary greater than Jones'?"

"Which employees have a salary greater than Jones' salary?"

"What is Jones' salary?"

Sub queries

SELECT select list FROM table

WHERE expr operator

(SELECT select_List FROM table);

- The subquery (inner query) executes once before the main query.
- The result of the subquery is used by the main query (outerquery).

A subquery is a SELECT statement that is embedded in a clause of another SELECT statement. You can build powerful statements out of simple ones by using subqueries. They can be very useful when you need to select rows from a table with a condition that depends on the data in the table itself. You can place the subquery in a number of SQL clauses

- WHERE clause
- HAVING clause
- FROM clause

ln the syntax:

operator includes a comparison operatör such as >, =, or IN

Note: Comparison operators fall into two classes: single-row operators

(>, =, >=, <, < >, <=) and multiple-row operators (IN, ANY, ALL).

Using a Sub query

SELECT ename

FROM EMP

WHERE sal >

(SELECT sal FROM emp WHERE empno=7566);

ENAME

FORD

SCOTT

KING

FORD

Using a Sub query

SELECT ename, sal, deptno, job FROM EMP WHERE job = (SELECT job FROM emp WHERE empno=7369);

```
ENAME               SAL        DEPTNO       JOB
ADAMS               1100          20          CLERK
JAMES                950          30         CLERK
MILLER              1300         10          CLERK
SMITH                800          20          CLERK
ADAMS               1100          20          CLERK
JAMES               950           30         CLERK
MILLER              1300         10          CLERK
7 rows selected.
```

```
SELECT enamel, sal, deptno
FROM EMP
WHERE sal IN
( SELECT MIN(sal)
FROM emp
GROUP BY deptno );
```

```
ENAME            SAL             DEPTNO
JAMES            950               30
SMITH            800               20
MILLER          1300              10
```

```
SELECT empno, ename, job
FROM emp
WHERE sal < ANY
( SELECT sal
FROM emp
WHERE job = 'CLERK' );
```

```
EMPNO                ENAME         JOB
7369                  SMITH         CLERK
7900                  JAMES         CLERK
7876                 ADAMS         CLERK
7521                 WARD          SALESMAN
7654                 MARTIN        SALESMAN
```

```
SELECT empno, ename, job FROM emp WHERE sal < ANY (SELECT sal FROM emp WHERE job = 'CLERK' )
AND job <> 'CLERK' ;
```

```
EMPNO            ENAME               JOB
7521              WARD                SALESMAN
7654              MARTIN              SALESMAN
```

SELECT empno, ename, job FROM emp WHERE sal > ALL (SELECT AVG(sal) FROM emp GROUP BY deptno) ;

```
EMPNO            ENAME                 JOB

7566               JONES              MANAGER

7788               SCOTT              ANALYST

7839                KING               PRESIDENT

7902               FORD                ANALYST
```

Guidelines for Using Subqueries

- Enclose subqueries in parentheses.
  Place subqueries on the right side of the comparison operator.
- Do not add an ORDER BY clause to a subquery.
- Use single-row operators with single row subqueries.
- Use multiple-row operators with multiple-row subqueries.

Types of Subqueries

- Single-row subquery
- Multiple-row subquery
- Multiple-column subquery

Types of Subqueries

Single-row subqueries: Queries that return only one row from the inner

SELECT statement

Multiple-row subqueries: QUERIES that return more than one rows from the inner SELECT statement

Multiple-column subqueries: QUERIES that return more than one column from the inner SELECT statement.

Using Group Functions in a Subquery

SELECT ename, sal, deptno FROM EMP WHERE sal IN (SELECT MIN(sal) FROM emp GROUP BY deptno) ;

```
ENAME          SAL                DEPTNO

SMITH          800                  20

JAMES          950                  30

MILLER        1300                  10
```

HAVING Clause with Subqueries

- The Oracle Server executes sub queries first. •The Oracle Server returns results into The HAVING clause of the main query.

SELECT job, AVG (sal) FROM emp GROUP BY job HAVING AVG(sal) = (SELECT MIN(AVG(sal)) FROM emp GROUP BY job);

```
JOB        AVG(SAL)

CLERK       1037,5
```

Multiple-Row Subqueries

- Return more than one row
- Use multiple-row comparison operators

Operator Meaning

IN Equal to any member in the list

ANY Compare value to each value returned by the subquery

ALL Compare value to every value returned by the subquery

```
SELECT ename, sal, deptno FROM emp WHERE sal IN (SELECT MIN(sal) FROM emp GROUP BY deptno);
```

```
ENAME          SAL              DEPTNO

SMITH          800             20

AMES           950             30

MILLER         1300            10
```

Using ANY Operator in Multiple-Row Subqueries

```
SELECT ename, sal, job

FROM emp

WHERE sal < ANY

( SELECT sal

FROM emp

WHERE job = 'CLERK' )

AND

job <> 'CLERK' ;
```

```
        ENAME          SAL              JOB

        WARD           1250     SALESMAN

        MARTIN         1250     SALESMAN
```

Using ALL Operator in

Multiple-Row Subqueries

```
SELECT ename, sal, job

FROM emp

WHERE sal > ALL

(SELECT AVG(sal)

FROM emp

GROUP BY deptno );
```

```
        ENAME          SAL            JOB

        JONES          2975           MANAGER

        SCOTT          3000           ANALYST

        KING           5000           PRESIDENT

        FORD           3000           ANALYST
```

1. Write a query to display the employee name and hiredate for all employees in the same department as Blake. Exclude Blake.

   ```
   SELECT ename, hiredate FROM emp
   WHERE deptno =
   ```

```
(SELECT deptno
FROM emp
WHERE ename = 'BLAKE')
AND ename <> 'BLAKE';
```

2. Create a query to display the employee number and name for all employees who earn more than the average salary. Sort the results in descending order of salary.

   ```
   SELECT empno, ename
   FROM emp
   WHERE sal >
   (SELECT AVG(sal)
   FROM emp);
   ```

3. Write a query to display the employee number and name for all employees who work in a department with any employee whose name contains a T. Save your SQL statemant in a file called p6q3.sql .

   ```
   SELECT empno, ename
   FROM emp
   WHERE deptno IN
   (SELECT deptno
   FROM emp
   WHERE ename LIKE '%T%' );
   ```

Display the employee's name, department number, and job title for all employees whose department location is Dallas.

Solution with subquery:

```
SELECT ename, empno, job

FROM emp

WHERE deptno = (SELECT deptno

FROM dept

WHERE loc ='DALLAS');
```

Solution with equijoin:

```
SELECT ename, empno, job

FROM emp e, dept d

WHERE e.deptno = d.deptno

AND d.loc='DALLAS';
```

```
ENAME          EMPNO                    JOB

SMITH            7369                  CLERK

JONES            7566                  MANAGER

SCOTT            7788                  ANALYST

ADAMS            7876                  CLERK

FORD             7902                  ANALYST
```

4. Display the employee name and salary of all employees who report to King.
   Self join:

```
SELECT e.ename, e.sal

FROM emp e, emp d

WHERE e.mgr = d.empno
```

```
AND
d.ename ='KING';
```

Solution with subquery:

```
SELECT ename, sal
FROM emp
WHERE mgr = (SELECT empno
FROM emp
WHERE ename = 'KING' );
```

6. Display the department number, name,, and job for all employees in the Sales department.

```
SELECT e.deptno, e.ename, e.job , d.dname
FROM emp e, dept d
WHERE e.deptno = d.deptno
AND
d.dname = 'SALES'
```