DBMS Lab 7.0

Sub Query

Subquery is a SELECT statement that is nested within another SELECT statement, which returns intermediate results

It is useful when you need to select rows from a table with a condition that depends on the data in another table. Here, inner query evaluates first, generates values that are tested in the condition of the outer query. Subqueries are of two types:

- ***Single-row subquery***: It is a subquery that returns only one row of data
- ***Multiple-row subquery***: It is a subquery that returns more than one row of data

The subtypes of subqueries are:

- Multiple column Subqueries
- Correlated subqueries

Single-row Subquery

Single-row subquery returns zero or one row to the outer SQL statement. The general syntax of this is:
SELECT columns FROM table1 WHERE column operator (SELECT column FROM table2 WHERE condition);

Let, *Student (roll,name,age, deptno)* and *Department (deptno, dname, campus)* be two schemas. If the user wants to find the roll and name of students belongs to 'CSE' department, the query is:

SELECT roll, name FROM Student WHERE deptno=(SELECT deptno FROM Department WHERE dname='CSE');

Single-row Subquery used in HAVING clause

Subquery can be used in place of the value in the HAVING clause

Q. Find the department and the departmental average age which are less than the maximum departmental age

SELECT deptno, AVG(age) FROM Student GROUP BY deptno HAVING AVG(age)<(SELECT MAX(AVG(age)) FROM Student GROUP BY deptno);

Single-row Subquery used in FROM clause

The inner SELECT statement in the FROM clause is used as the data source for the outer SELECT statement. This is also called as Inline View

SELECT age FROM (SELECT age FROM Student WHERE age<20);

A subquery may not use the ORDER BY clause, but an inline view may

SELECT age FROM (SELECT age FROM Student ORDER BY age DESC) WHERE ROWNUM<=2;

Multiple-row Subquery

Multiple-row subquery returns more than one row. It uses either IN, ALL or ANY operator

```
SELECT roll, name FROM Student WHERE age<
ANY(SELECT age FROM Customer);

SELECT roll, name FROM Student WHERE age< IN(SELECT
age FROM Customer);

SELECT roll, name FROM Student WHERE age<
ALL(SELECT age FROM Customer);
```

Multiple-column Subquery

Multiple-column subquery returns more than one column as result

```
SELECT roll,deptno,age FROM Student WHERE (deptno,age)
IN (SELECT deptno,MIN(age) FROM Student Group BY
deptno);
```

Correlated Subquery

In a correlated subquery, the inner query can reference columns from the outer query. The inner query is executed once for each row in the outer query

```
SELECT roll, deptno, name, age FROM Student s1 WHERE
age>(SELECT AVG(age) FROM Student s2 WHERE
s1.deptno = s2.deptno);

SELECT * FROM Catalog WHERE Catalog.bookid IN(SELECT
bookid FROM Order_Details WHERE Catalog.bookid=
Order_Details.bookid);
```

Nested Subquery

You can nest subqueries to a depth of 255

```
SELECT empid, empname, salary, hiredate FROM Employee
WHERE empid IN(SELECT empid FROM Employee WHERE
mgrid = (SELECT empid FROM Employee WHERE empname
= 'Paresh'));

SELECT deptno,AVG(age) FROM Student GROUP BY deptno
HAVING AVG(age)<( SELECT MAX(AVG(age)) FROM Student
WHERE deptno IN (SELECT deptno FROM Staff WHERE
qty>10) GROUP BY deptno);
```

UPDATE with Subquery

The column values can be updated with the help of a subquery statement. The general syntax is:
UPDATE tablename SET columnname = value WHERE columnnane operator (SELECT subquery);

The value in the update statement can also be calculated by using a subquery. The general syntax for this is:
UPDATE tablename SET columnname = (SELECT subquery) [WHERE condition];

Update the age of the student having roll no. 4 to the average age of all the students

UPDATE Student SET age=(SELECT AVG(age) FROM Student) WHERE roll=4;

DELETE with Subquery

A row or rows from a table can be deleted based on a value returned by a subquery. The general syntax is:
DELETE FROM tablename WHERE columnname = (SELECT subquery);

Delete all the student records whose ages are greater than the average age of the students

DELETE FROM Student WHERE age>(SELECT AVG(age) FROM Student);

END