Subqueries in Oracle SQL

1. Introduction to Subqueries

Definition: A **subquery** (also called *inner query* or *nested query*) is a query inside another SQL statement. It is used to retrieve data that will be used by the **main query** (or *outer query*).

General Syntax:

```
SELECT column_list
FROM table_name
WHERE expression operator (SELECT column_list FROM table_name WHERE condition);
```

- The **inner query** executes first.
- Its result is passed to the **outer query**.
- Commonly used in `WHERE`, `HAVING`, `FROM`, and `SELECT` clauses.

---

2. Types of Subqueries

a) **Single-row Subquery**

Returns **only one row** from the inner query.

Example:

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

Finds employees whose salary is above the company's average salary.

---

b) **Multi-row Subquery**

Returns **multiple rows** from the inner query. Used with `IN`, `ANY`, `ALL`, `EXISTS`.

Example using `IN`:

```
SELECT ename, deptno
FROM emps
WHERE deptno IN (SELECT deptno FROM dept WHERE loc = 'NEW YORK');
```

Finds employees who work in departments located in NEW YORK.

---

c) **Multi-column Subquery**

Returns **multiple columns** from the inner query.

Example:

```
SELECT empno, ename, job, deptno
FROM emps
WHERE (job, deptno) IN
      (SELECT job, deptno FROM emps WHERE empno = 7839);
```

Finds employees having the same job and department as employee 7839.

---

d) **Correlated Subquery**

The subquery depends on values from the outer query and executes **once for each row** of the outer query.

Example:

```
SELECT e.ename, e.sal, e.deptno
FROM emps e
WHERE e.sal > (SELECT AVG(sal)
               FROM emps
               WHERE deptno = e.deptno);
```

Lists employees whose salary is higher than the average salary of their department.

---

3. Subquery Operators

Operator	Description	Example
`=`	Compares with a single value	`sal = (SELECT MAX(sal) FROM emps)`
`IN`	Compares with a list of values	`deptno IN (SELECT deptno FROM dept)`
`ANY`	Compares with any value	`sal > ANY (SELECT sal FROM emps WHERE job='SALESMAN')`
`ALL`	Compares with all values	`sal > ALL (SELECT sal FROM emps WHERE job='CLERK')`
`EXISTS`	Checks if subquery returns rows	`WHERE EXISTS (SELECT 1 FROM dept WHERE loc='DALLAS')`

---

4. Subqueries in Different Clauses

a) **In the WHERE Clause**

Used for filtering.

```
SELECT *
FROM emps
WHERE deptno IN (SELECT deptno FROM dept WHERE loc = 'CHICAGO');
```

---

b) **In the HAVING Clause**

Used with group functions.

```
SELECT deptno, AVG(sal)
FROM emps
GROUP BY deptno
HAVING AVG(sal) > (SELECT AVG(sal) FROM emps);
```

Shows departments where the average salary is greater than the overall average salary.

---

c) **In the SELECT Clause**

Used to display derived values.

Example:

```
SELECT e.ename,
       (SELECT dname FROM dept d WHERE d.deptno = e.deptno) AS department
FROM emps e;
```

Displays each employee's name along with their department name.

---

5. Rules and Guidelines

- Subqueries must be enclosed in **parentheses**.
- `ORDER BY` inside a subquery is allowed only with `FETCH FIRST` or `ROWNUM`.
- Subqueries can appear in DML statements (`INSERT`, `UPDATE`, `DELETE`).

Example (UPDATE):

```sql
UPDATE emps
SET sal = sal * 1.1
WHERE deptno = (SELECT deptno FROM dept WHERE dname = 'SALES');
```

Increases salaries by 10% for all employees in the SALES department.
