## What is Subquery?State different Types of. Subqueries

A **subquery** is a query nested inside another query, often used to retrieve data that will be used in the main query as a condition to further restrict the data to be retrieved. Subqueries are commonly found in SELECT, INSERT, UPDATE, and DELETE statements, as well as in clauses like WHERE, HAVING, and FROM.

**Types of Subqueries**

1. **Single-row Subquery**: Returns only one row of results. It is often used with comparison operators like =, >, <, etc.

**Example**:

SELECT name

FROM employees

WHERE department_id = (SELECT department_id FROM departments WHERE name = 'Sales');

2. **Multiple-row Subquery**: Returns multiple rows of results. Operators like IN, ANY, or ALL are typically used to handle multiple rows.

**Example**:

SELECT name

FROM employees

WHERE department_id IN (SELECT department_id FROM departments WHERE location_id = 1700);

3. **Correlated Subquery**: This subquery references a column from the outer query, and it's evaluated once for each row processed by the outer query.

**Example**:

SELECT name, salary

FROM employees e1

WHERE salary > (SELECT AVG(salary) FROM employees e2 WHERE e1.department_id = e2.department_id);

4. **Nested Subquery**: A subquery within another subquery, where you can have multiple layers of subqueries.

**Example**:

```
SELECT name
FROM employees
WHERE department_id = (SELECT department_id
                       FROM departments
                       WHERE location_id = (SELECT location_id
                                            FROM locations
                                            WHERE city = 'New York'));
```

5. **Inline View**: A subquery in the FROM clause, which acts as a temporary table within the main query.

**Example**:

```
SELECT dept.name, avg_emp_salary
FROM (SELECT department_id, AVG(salary) AS avg_emp_salary
      FROM employees
      GROUP BY department_id) AS dept
JOIN departments d ON dept.department_id = d.department_id;
```

Each type of subquery serves different use cases and adds flexibility to SQL by enabling more complex and conditional data retrieval.

## Can we update or delete record using Subquery? Explain with Example.

**1. Update with Subquery**

A subquery can be used in an UPDATE statement to modify records based on data from another table.

**Example**:

UPDATE employees

SET salary = salary * 1.10

WHERE department_id = (SELECT department_id FROM departments WHERE name = 'Sales');

*Explanation*: This query increases the salary of employees in the "Sales" department by 10%.

**2. Delete with Subquery**

A subquery can also specify which records to delete based on another table's data.

**Example**:

DELETE FROM employees

WHERE department_id = (SELECT department_id FROM departments WHERE name = 'Marketing');

*Explanation*: This query deletes all employees in the "Marketing" department.

These types of subqueries make it possible to dynamically filter the records to update or delete.

## What are the limitations of Subquery?

Subqueries have some limitations, including:

1. **Performance Issues**: Subqueries can be slower, especially for large datasets, as they may execute multiple times.
2. **Limited Use with Certain Clauses**: Some databases restrict subqueries in certain clauses like ORDER BY or GROUP BY within subqueries.
3. **Single-column Returns**: In WHERE or HAVING clauses, subqueries must typically return a single column. Multiple columns can lead to errors.
4. **Complexity**: Nested subqueries can make the query hard to read and maintain.
5. **Scalability Issues**: Correlated subqueries, which execute once per row, can be inefficient and may not scale well with larger tables.