In SQL, **subqueries** are queries nested inside other queries. They can be classified as **correlated** or **non-correlated** based on their relationship with the outer query.

**1. Non-Correlated Subqueries**

* **Definition**: A **non-correlated subquery** is independent of the outer query. It is executed only once and returns a result that is then used by the outer query.
* **Characteristics**:
  + Executed before the outer query.
  + Does not rely on any columns or data from the outer query.
  + Can often be executed as a standalone query.
* **Example**:
* SELECT employee\_id, employee\_name
* FROM employees
* WHERE department\_id = (
* SELECT department\_id
* FROM departments
* WHERE department\_name = 'Sales'
* );

Here, the subquery retrieves the department\_id for the Sales department. The outer query then uses this value to filter employees.

**2. Correlated Subqueries**

* **Definition**: A **correlated subquery** depends on the outer query for its execution. It is evaluated once for each row processed by the outer query.
* **Characteristics**:
  + Executed repeatedly, once for each row of the outer query.
  + Relies on a value or column from the outer query.
  + Slower than non-correlated subqueries due to multiple executions.
* **Example**:
* SELECT employee\_id, employee\_name
* FROM employees e
* WHERE salary > (
* SELECT AVG(salary)
* FROM employees
* WHERE department\_id = e.department\_id
* );

Here, the subquery calculates the average salary for the department of the current employee in the outer query. The e.department\_id in the subquery is provided by the outer query.

**Key Differences**

| **Feature** | **Non-Correlated Subquery** | **Correlated Subquery** |
| --- | --- | --- |
| **Execution** | Executed once. | Executed multiple times. |
| **Dependency** | Independent of the outer query. | Depends on the outer query. |
| **Performance** | Faster. | Slower due to repeated execution. |
| **Use Case** | Aggregation, filtering, or returning single/multiple values. | Row-by-row comparisons or dependent conditions. |

**Summary**

* Use **non-correlated subqueries** when the inner query is static and independent of the outer query.
* Use **correlated subqueries** when the inner query must dynamically depend on the outer query's rows.

For better performance, correlated subqueries are often rewritten using **JOINs** or **WITH (Common Table Expressions)** in modern SQL practices.