

Sub Query - SQL

By Dhandapani Yedappalli Krishnamurthi Sep 30, 2025

♦ What is a Subquery?

A **subquery** is a query inside another query.

It helps when you need results **derived from another query** before applying conditions or aggregations.

Think of it as:

- Main Query → Fetches results
- Sub Query → Supports the main query by providing intermediate data

♦ Types of Subqueries

1. Single-row subquery

Returns **only one value** (scalar result).

-- Find employees who earn more than the average salary

```
SELECT BusinessEntityID, JobTitle, Rate
FROM HumanResources.EmployeePayHistory
WHERE Rate > (
    SELECT AVG(Rate)
    FROM HumanResources.EmployeePayHistory
);
```

✓ Explanation:

- Inner query → AVG(Rate) returns one value (average salary).
- Outer query → Compares each employee's salary with the average.

2. Multi-row subquery

Returns **multiple values**.

Used with operators like IN, ANY, ALL.

-- Find customers who placed orders in 2013

```
SELECT CustomerID, PersonID
FROM Sales.Customer
WHERE CustomerID IN (
    SELECT DISTINCT CustomerID
    FROM Sales.SalesOrderHeader
    WHERE YEAR(OrderDate) = 2013
);
```

✓ Explanation:

- Inner query → Gets all customers from 2013.
- Outer query → Selects details of those customers.

3. Correlated subquery

The **inner query depends on the outer query** row by row.

-- Find sales orders where order qty is greater than the average order qty of that product

```
SELECT SalesOrderID, ProductID, OrderQty
FROM Sales.SalesOrderDetail sod
WHERE OrderQty > (
    SELECT AVG(OrderQty)
    FROM Sales.SalesOrderDetail
    WHERE ProductID = sod.ProductID
);
```

✓ Explanation:

- For each product in the outer query → inner query calculates the product's average quantity.
- Then compares row by row.

4. Subquery in SELECT clause

Used to **derive a computed column**.

-- For each order, show total amount along with average order amount

```
SELECT SalesOrderID,
    SUM(LineTotal) AS TotalOrderAmount,
    (SELECT AVG(LineTotal) FROM Sales.SalesOrderDetail) AS AvgOrderAmount
FROM Sales.SalesOrderDetail
GROUP BY SalesOrderID;
```

✓ Explanation:

- Subquery calculates overall **average order amount**.
- Appears as a column in the result.

5. Subquery in FROM clause (Inline view)

Treats subquery as a **temporary table**.

-- Find top 5 products with the highest sales

```
SELECT TOP 5 ProductID, TotalSales
FROM (
    SELECT ProductID, SUM(LineTotal) AS TotalSales
    FROM Sales.SalesOrderDetail
    GROUP BY ProductID
) AS ProductSales
ORDER BY TotalSales DESC;
```

✓ Explanation:

- Subquery calculates total sales by product.
- Outer query filters top 5.

6. Subquery with EXISTS

Checks if **rows exist** in subquery result.

-- Find customers who have placed at least one order

```
SELECT CustomerID, PersonID
FROM Sales.Customer c
WHERE EXISTS (
    SELECT 1
    FROM Sales.SalesOrderHeader soh
```

```
WHERE soh.CustomerID = c.CustomerID
);
```

✓ Explanation:

- Inner query checks if a customer has orders.
- If yes → outer query includes that customer.

♦ Business Scenario: AdventureWorks

Scenario: Management wants to analyze customers and sales.

Example 1: High-value customers

-- Find customers whose total purchase is greater than average purchase

```
SELECT CustomerID, SUM(TotalDue) AS TotalPurchase
FROM Sales.SalesOrderHeader
GROUP BY CustomerID
HAVING SUM(TotalDue) > (
    SELECT AVG(TotalDue)
    FROM Sales.SalesOrderHeader
);
```

Example 2: Best-selling products

-- Find products that sold more than the average quantity

```
SELECT ProductID, SUM(OrderQty) AS TotalQty
FROM Sales.SalesOrderDetail
GROUP BY ProductID
HAVING SUM(OrderQty) > (
    SELECT AVG(OrderQty)
    FROM Sales.SalesOrderDetail
);
```

Summary Table

Type	Usage	Operator	Example
Single-row	One value	=, >, <	Salary > AVG(Salary)
Multi-row	Multiple values	IN, ANY, ALL	Customers IN 2013
Correlated	Depends outer query	row by row	OrderQty > AVG for product
SELECT	Derived column	-	Show Avg salary column
FROM	Inline view	-	Top 5 products
EXISTS	Existence check	EXISTS	Customers with orders