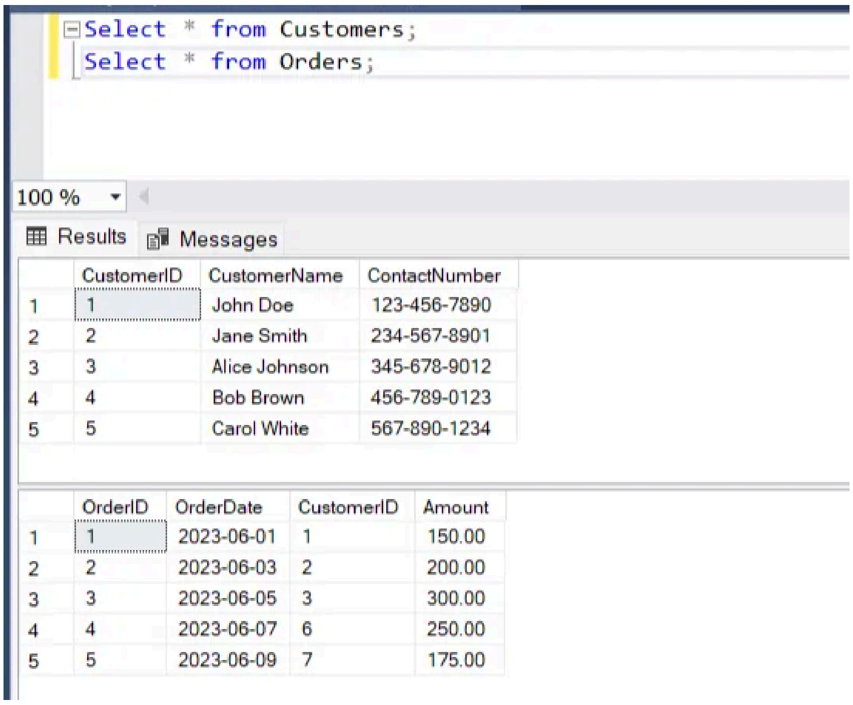
📘 Module 24: Subqueries

🧾 Reference Tables Used in This Module

The following two tables are used for examples in this module:

📸 Image Preview – `Customers` and `Orders` Tables



```
Select * from Customers;
Select * from Orders;
```

100 %

Results | Messages

	CustomerID	CustomerName	ContactNumber
1	1	John Doe	123-456-7890
2	2	Jane Smith	234-567-8901
3	3	Alice Johnson	345-678-9012
4	4	Bob Brown	456-789-0123
5	5	Carol White	567-890-1234

	OrderID	OrderDate	CustomerID	Amount
1	1	2023-06-01	1	150.00
2	2	2023-06-03	2	200.00
3	3	2023-06-05	3	300.00
4	4	2023-06-07	6	250.00
5	5	2023-06-09	7	175.00

🔷 What is a Subquery?

A **subquery** is a query that is written **inside another SQL statement**.

It is also known as a **nested query** and helps solve complex questions by breaking them into simpler steps.

🧩 Part 1 – Subquery in `WHERE` Clause

✅ Definition:

Used to **filter results** in the outer query using a condition based on a result from the inner query.

✅ Syntax:

SELECT column1
FROM table1
WHERE column2 IN (SELECT column2 FROM table2 WHERE condition);

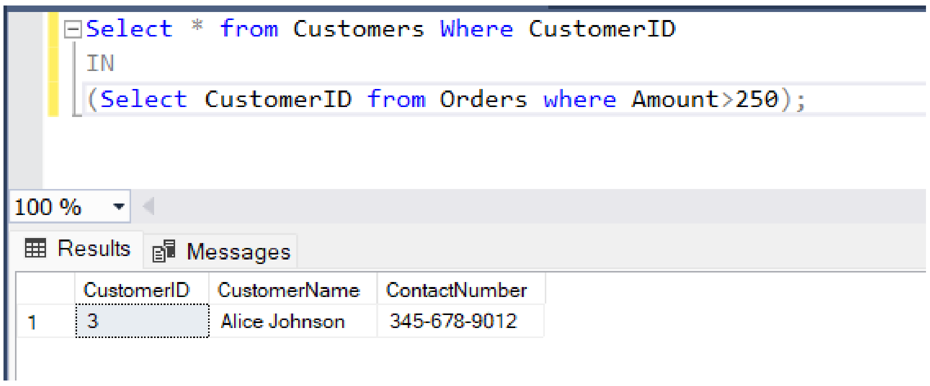
📌 Example:

SELECT * FROM Customers
WHERE CustomerID IN (
    SELECT CustomerID
    FROM Orders
    WHERE Amount > 250
);

🧾 Explanation:

- Inner query selects CustomerIDs with order amounts > 250.
- Outer query shows customer details **who placed such orders**.

🧾 Expected Result:



```
Select * from Customers Where CustomerID
IN
(Select CustomerID from Orders where Amount>250);
```

100 %

Results | Messages

	CustomerID	CustomerName	ContactNumber
1	3	Alice Johnson	345-678-9012

🧩 Part 2 – Subquery in `FROM` Clause

✅ Definition:

Used to **create a temporary result set** that can be used as a table by the main query.

✅ Syntax:

SELECT column1
FROM (SELECT ... FROM table WHERE ...) AS temp_table;

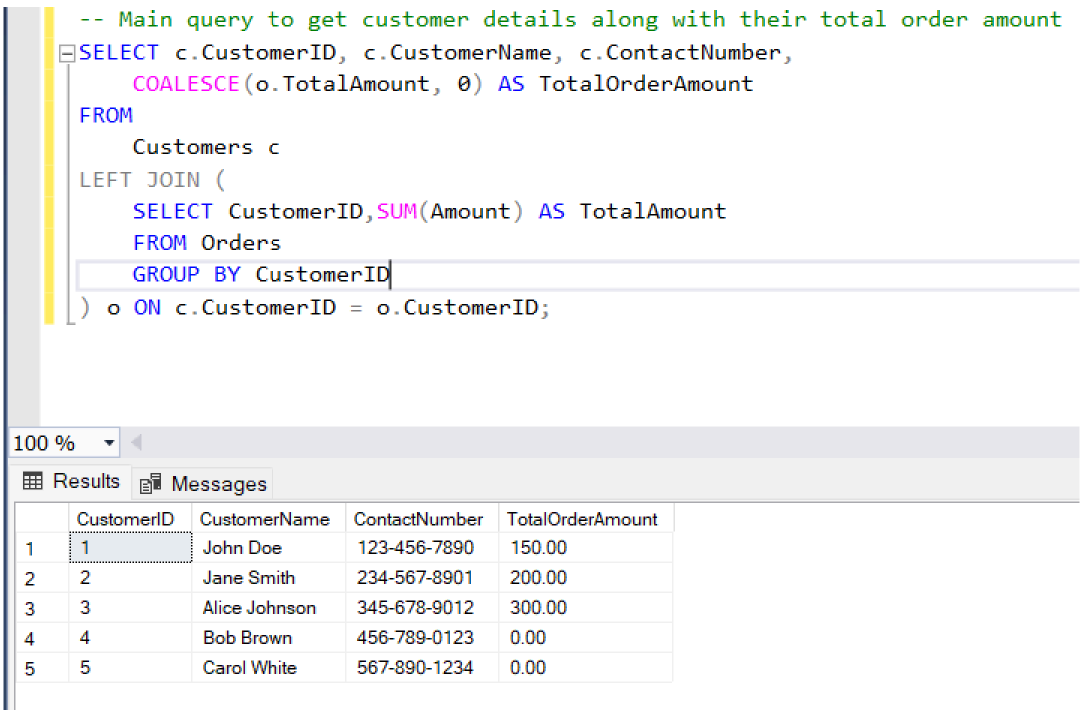
📌 Example:

SELECT c.CustomerID, c.CustomerName, c.ContactNumber,
       COALESCE(o.TotalAmount, 0) AS TotalOrderAmount
FROM Customers c
LEFT JOIN (
    SELECT CustomerID, SUM(Amount) AS TotalAmount
    FROM Orders
    GROUP BY CustomerID
) o ON c.CustomerID = o.CustomerID;

🧾 Explanation:

- The **subquery** (`o`) calculates total amount per customer.
- The **outer query** joins it with `Customers` to show each customer and their total order amount.
- `COALESCE(..., 0)` returns 0 if no orders exist for that customer.

🧾 Expected Result:



```
-- Main query to get customer details along with their total order amount
SELECT c.CustomerID, c.CustomerName, c.ContactNumber,
    COALESCE(o.TotalAmount, 0) AS TotalOrderAmount
FROM
    Customers c
LEFT JOIN (
    SELECT CustomerID,SUM(Amount) AS TotalAmount
    FROM Orders
    GROUP BY CustomerID
) o ON c.CustomerID = o.CustomerID;
```

100 %

Results | Messages

	CustomerID	CustomerName	ContactNumber	TotalOrderAmount
1	1	John Doe	123-456-7890	150.00
2	2	Jane Smith	234-567-8901	200.00
3	3	Alice Johnson	345-678-9012	300.00
4	4	Bob Brown	456-789-0123	0.00
5	5	Carol White	567-890-1234	0.00

🧩 Part 3 – Subquery in `SELECT` Clause

✅ Definition:

Used to **return a value** in a column by running a subquery **for each row** in the outer query.

✅ Syntax:

```
SELECT column1,
    (SELECT ... FROM table2 WHERE table2.id = table1.id) AS alias
FROM table1;
```

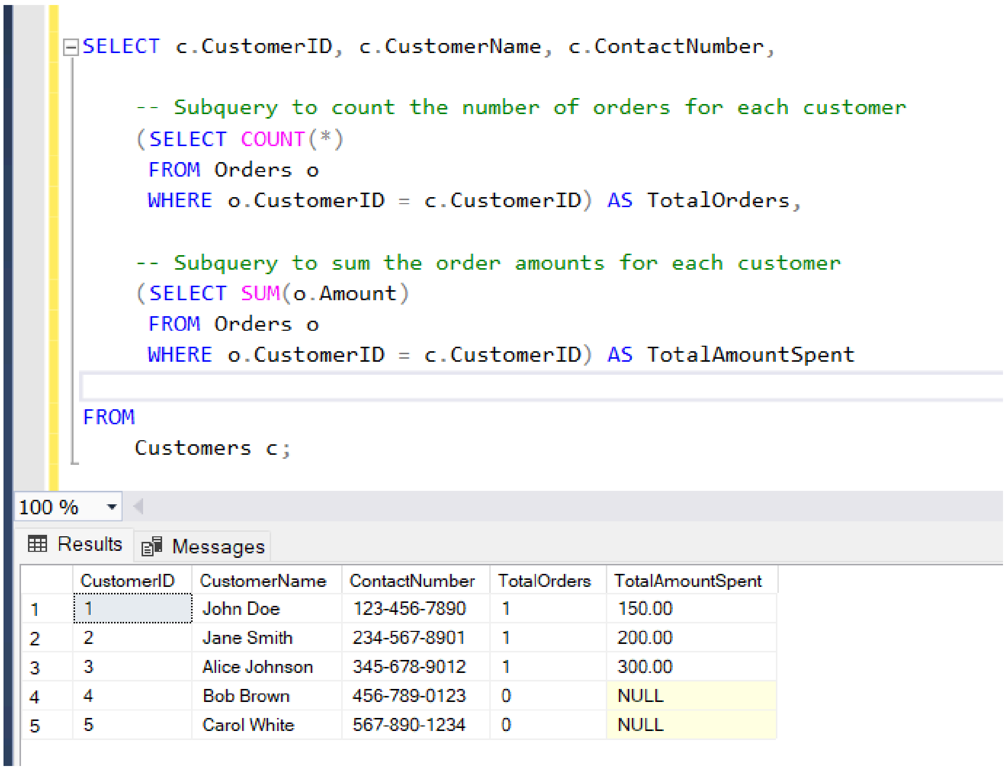
📌 Example:

```
SELECT c.CustomerID, c.CustomerName, c.ContactNumber,

    (SELECT COUNT(*)
     FROM Orders o
     WHERE o.CustomerID = c.CustomerID) AS TotalOrders,

    (SELECT SUM(o.Amount)
     FROM Orders o
     WHERE o.CustomerID = c.CustomerID) AS TotalAmountSpent

FROM Customers c;
```

🧾 Explanation:

- Two subqueries run for each customer:
    - One to count how many orders they placed.
    - Another way to sum how much they spent.
- All values are shown in the result for each customer.

🧾 Expected Result:



```
SELECT c.CustomerID, c.CustomerName, c.ContactNumber,

    -- Subquery to count the number of orders for each customer
    (SELECT COUNT(*)
     FROM Orders o
     WHERE o.CustomerID = c.CustomerID) AS TotalOrders,

    -- Subquery to sum the order amounts for each customer
    (SELECT SUM(o.Amount)
     FROM Orders o
     WHERE o.CustomerID = c.CustomerID) AS TotalAmountSpent

FROM
    Customers c;
```

	CustomerID	CustomerName	ContactNumber	TotalOrders	TotalAmountSpent
1	1	John Doe	123-456-7890	1	150.00
2	2	Jane Smith	234-567-8901	1	200.00
3	3	Alice Johnson	345-678-9012	1	300.00
4	4	Bob Brown	456-789-0123	0	NULL
5	5	Carol White	567-890-1234	0	NULL

📝 Key Points to Remember

✔️ A **subquery returns data** that is used by the outer/main query.

✔️ You can use subqueries in:

- `WHERE` clause (to filter rows)
- `FROM` clause (as a virtual table)
- `SELECT` clause (to calculate values per row)

✔️ Subqueries can return:

- A **single value**
- A **list of values**
- A **table**

✔️ Always **alias** subqueries in the `FROM` clause.

✔️ Use `COALESCE()` to handle NULLs when using subqueries with outer joins.

✔️ Subqueries can be **correlated** (use values from outer query) or **independent**.