

SQL SELECT QUERIES OUTPUT

1 Orders Count by Country

Query Summary

This query shows how many orders were placed from each country. It connects the Customers table with the Orders table and counts the number of orders for every country.

The screenshot shows the SSMS interface with the following details:

- Navigator:** Shows the database schema. Under the **ecommerce_db** schema, there are tables, views, stored procedures, and functions. Under the **ecommerce_store** schema, there are tables for account, account_type, address, and address_type.
- SQL File 4***: The current query window title.
- ecommerce_dataset***: The current query window title.
- Code Area:** The query is as follows:

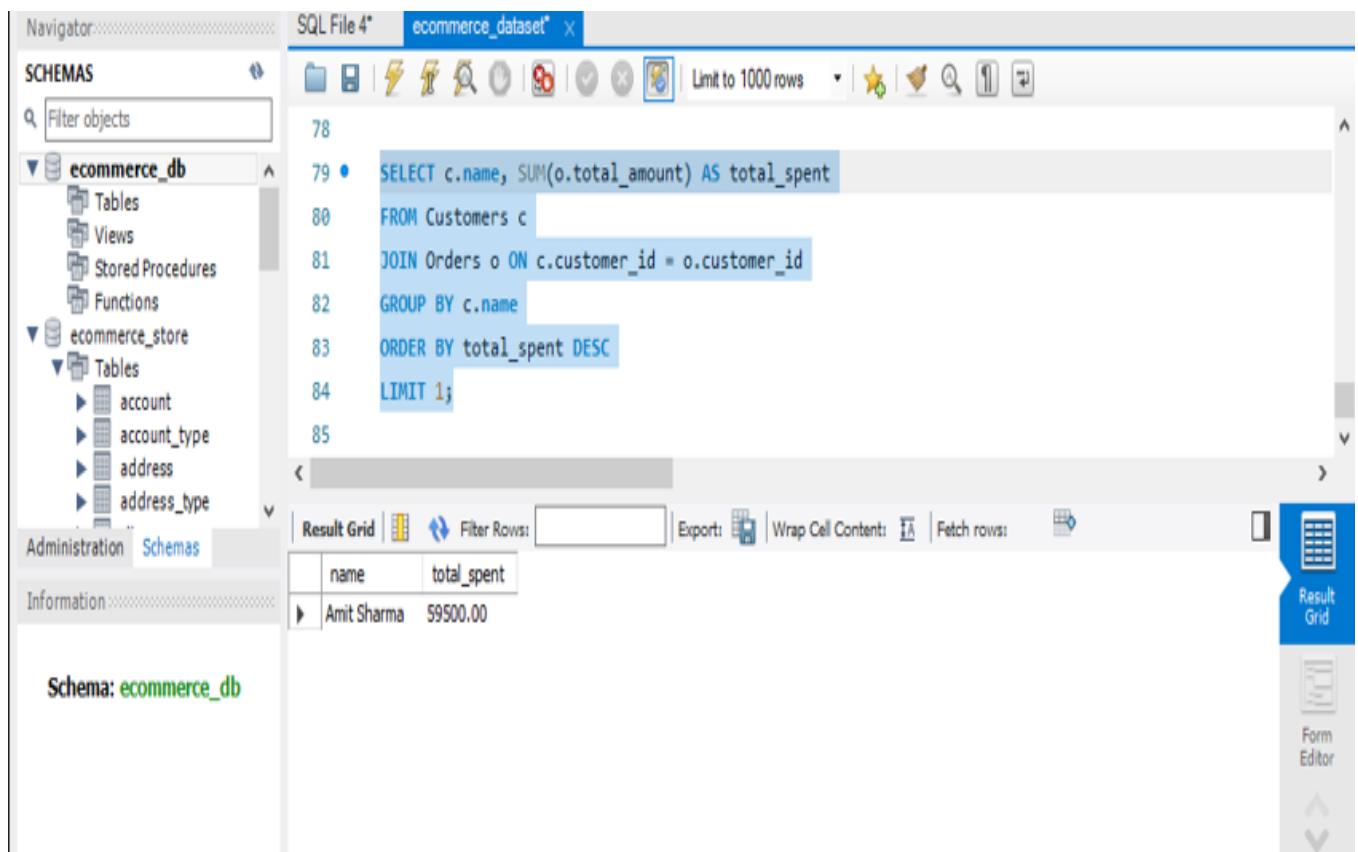
```
71
72 • SHOW TABLES;
73
74 • SELECT c.country, COUNT(o.order_id) AS total_orders
75   FROM Customers c
76   LEFT JOIN Orders o ON c.customer_id = o.customer_id
77   GROUP BY c.country;
```
- Result Grid:** The results of the query are displayed in a grid format. The columns are **country** and **total_orders**. The data is:

country	total_orders
India	3
USA	1
Korea	1
Canada	0
- Information:** Shows the schema is **ecommerce_db**.
- Toolbars and Buttons:** Standard SSMS toolbar buttons for file, edit, search, and navigation.

2 Highest Spending Customer

Query Summary

This query calculates how much each customer has spent in total and then identifies the customer who spent the most.
It uses **SUM(total_amount)** on the Orders table.



The screenshot shows the SQL Server Management Studio interface. The left pane displays the Navigator with Schemas and Tables listed under ecommerce_db and ecommerce_store. The main pane shows a query window with the following SQL code:

```
SELECT c.name, SUM(o.total_amount) AS total_spent
FROM Customers c
JOIN Orders o ON c.customer_id = o.customer_id
GROUP BY c.name
ORDER BY total_spent DESC
LIMIT 1;
```

The result grid shows one row of data:

name	total_spent
Amit Sharma	59500.00

3 Product with the Highest Revenue

Query Summary

This query finds out which product generated the highest revenue. It sums the subtotal from the OrderItems table for each product.

The screenshot shows a SQL editor window titled "SQL File 4*" with the database "ecommerce_dataset". The query is:

```
86 •  SELECT p.product_name, SUM(oi.subtotal) AS revenue
87  FROM Products p
88  JOIN OrderItems oi ON p.product_id = oi.product_id
89  GROUP BY p.product_name
90  ORDER BY revenue DESC
91  LIMIT 1;
92
93
```

The result grid shows one row:

	product_name	revenue
▶	Laptop	55000.00

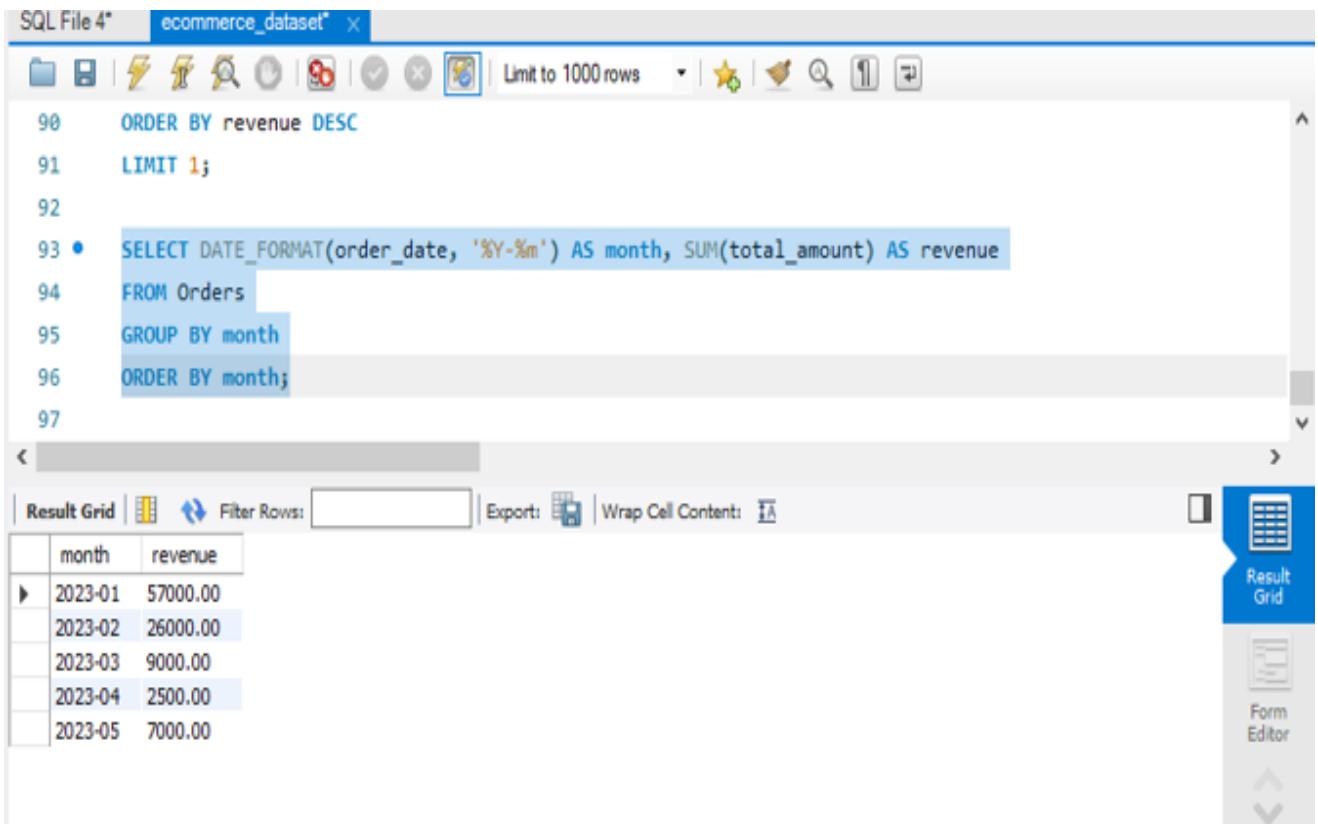
The right sidebar has tabs for "Result Grid" (selected), "Form Editor", and "Table Editor".

4 Monthly Revenue Report

Query Summary

This query calculates total monthly revenue by formatting the order date into a YEAR-MONTH format.

Then it groups orders by month to get monthly earnings.



The screenshot shows a SQL editor interface with the following details:

- Title Bar:** SQL File 4* ecommerce dataset*
- Toolbar:** Includes icons for file operations, search, and database management.
- Query Editor:** Displays the following SQL code:

```
90     ORDER BY revenue DESC
91     LIMIT 1;
92
93 • SELECT DATE_FORMAT(order_date, '%Y-%m') AS month, SUM(total_amount) AS revenue
94     FROM Orders
95     GROUP BY month
96     ORDER BY month;
97
```
- Result Grid:** Shows the output of the query in a tabular format:

month	revenue
2023-01	57000.00
2023-02	26000.00
2023-03	9000.00
2023-04	2500.00
2023-05	7000.00
- Result Grid Panel:** A sidebar on the right contains icons for "Result Grid" (selected), "Form Editor", and navigation arrows.

5 Customers with More Than One Order

Query Summary

This query finds customers who placed more than one order.
It groups orders by customer and filters using the HAVING clause.

The screenshot shows a SQL editor interface with the following details:

- Title Bar:** SQL File 4* ecommerce_dataset*
- Toolbar:** Includes icons for file operations, search, and database management.
- Query Editor:** Displays the following SQL code:

```
97
98 •  SELECT c.name, COUNT(o.order_id) AS orders_count
99   FROM Customers c
100  JOIN Orders o ON c.customer_id = o.customer_id
101 GROUP BY c.name
102 HAVING orders_count > 1;
103
104
```
- Result Grid:** Shows the output of the query in a tabular format:

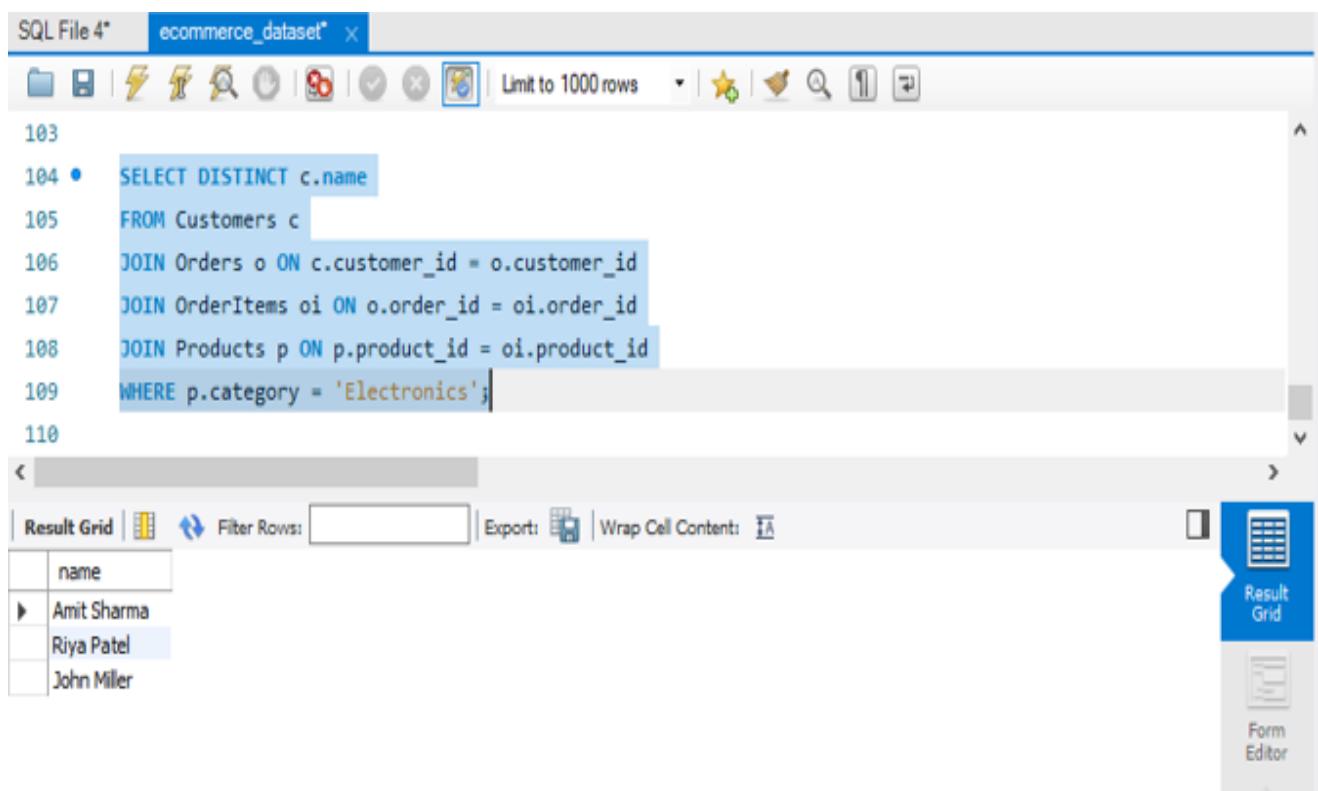
name	orders_count
Amit Sharma	2
- Right Panel:** Contains buttons for "Result Grid" and "Form Editor".

6 Customers Who Bought Electronics

Query Summary

This query finds all customers who purchased items from the Electronics category.

It joins Customers → Orders → OrderItems → Products to trace category-wise purchases.



The screenshot shows a SQL query window in SSMS. The query retrieves distinct customer names who have purchased items from the 'Electronics' category. The results are displayed in a grid, showing three names: Amit Sharma, Riya Patel, and John Miller.

```
SQL File 4* ecommerce_dataset* x
103
104 • SELECT DISTINCT c.name
105   FROM Customers c
106   JOIN Orders o ON c.customer_id = o.customer_id
107   JOIN OrderItems oi ON o.order_id = oi.order_id
108   JOIN Products p ON p.product_id = oi.product_id
109   WHERE p.category = 'Electronics';
110
```

name
Amit Sharma
Riya Patel
John Miller