

2025

SQLPortfolio

Sales Analysis

Prepared by: Komal Mehra

Database Used: PostgreSQL

Sales Analysis

Objective

The objective of this project is to analyse sales data using SQL to gain insights into customer spending and overall sales performance. It aims to identify the top customers based on their spending, evaluate monthly sales trends, and generate insights that support informed business decisions.

Database schema

This project is based on a small, structured sales dataset created for analysis purposes. It contains information about customers, the products they purchase, the orders they place, and the quantities sold.

The data is organized into four tables, each serving a distinct purpose:

Table	Description	Key Columns
Customers	Stores basic customer information.	customer_id (PK), customer_name
Products	Contains details of items available for sale and their prices.	product_id (PK), product_name, price
Orders	Records each order placed by customers, including the date.	order_id (PK), customer_id (FK), order_date
Order Details	Breaks down every order into the products purchased and their quantities.	order_id (FK), product_id (FK), quantity

Analysis

➤ Top 3 Customers by Spending

This query identifies the three customers who have spent the most.

```
-- 3 Top 3 customers
SELECT
    c.customer_name,
    SUM(p.price * od.quantity) AS total_spent
FROM Customers c
JOIN Orders o      ON c.customer_id = o.customer_id
JOIN Order_Details od ON o.order_id = od.order_id
JOIN Products p    ON od.product_id = p.product_id
GROUP BY c.customer_name
ORDER BY total_spent DESC
LIMIT 3;
```

Result:-

	customer_name character varying (50) 🔒	total_spent numeric 🔒
1	Sita	51500.00
2	Rohan	51000.00
3	Aman	2500.00

Insight- Sita spent the most overall, followed by Rohan and Aman.

➤ Monthly Sales Trend

This query calculates the total sales amount for each month, helping us see whether sales are rising or falling over time.

```
-- 4 Monthly sales trend
SELECT
    DATE_TRUNC('month', o.order_date) AS month,
    SUM(p.price * od.quantity) AS monthly_sales
FROM Orders o
JOIN Order_Details od ON o.order_id = od.order_id
JOIN Products p      ON od.product_id = p.product_id
GROUP BY month
ORDER BY month;
```

Result:-

	month timestamp with time zone	monthly_sales numeric
1	2025-01-01 00:00:00+05:30	51000.00
2	2025-02-01 00:00:00+05:30	51500.00
3	2025-03-01 00:00:00+05:30	2500.00

Insight- Sales grew slightly from January to February, then dropped sharply in March. Focusing on March could improve performance.

➤ Product with Highest Revenue

This query identifies which product has generated the highest overall revenue.

```
--Product that generated the highest revenue
SELECT
    p.product_name,
    SUM(p.price * od.quantity) AS revenue
FROM Products p
JOIN Order_Details od ON p.product_id = od.product_id
GROUP BY p.product_name
ORDER BY revenue DESC
LIMIT 1;
```

Result:-

	product_name character varying (50)	revenue numeric
1	Laptop	100000.00

Insight-

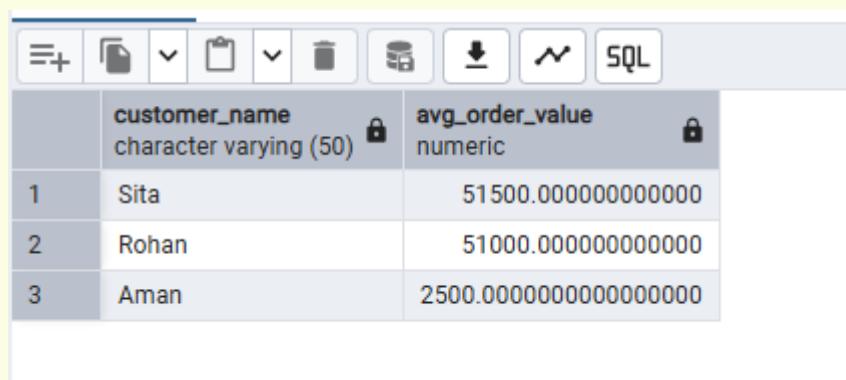
The product with the highest revenue is the best performer and should be kept in stock or promoted further.

➤ Average Order Value per Customer

This query calculates the average value of each customer's orders, showing which customers spend the most per order.

```
SELECT
    c.customer_name,
    AVG(order_total) AS avg_order_value
FROM (
    SELECT
        o.order_id,
        o.customer_id,
        SUM(p.price * od.quantity) AS order_total
    FROM Orders o
    JOIN Order_Details od ON o.order_id = od.order_id
    JOIN Products p ON p.product_id = od.product_id
    GROUP BY o.order_id, o.customer_id
) sub
JOIN Customers c ON c.customer_id = sub.customer_id
GROUP BY c.customer_name
ORDER BY avg_order_value DESC;
```

Result:-



	customer_name character varying (50)	avg_order_value numeric
1	Sita	51500.000000000000
2	Rohan	51000.000000000000
3	Aman	2500.00000000000000

Insight: The results show how much, on average, each customer spends per order. This helps identify customers with high-value orders, who may be good targets for loyalty rewards.

➤ Total Quantity Sold per Product

This query shows how many units of each product were sold overall. It helps to identify best-selling items.

```
--Total quantity sold per product
SELECT
    p.product_name,
    SUM(od.quantity) AS qty_sold
FROM Products p
JOIN Order_Details od ON p.product_id = od.product_id
GROUP BY p.product_name
ORDER BY qty_sold DESC;
```

Result:-

	product_name character varying (50)	qty_sold bigint
1	Mouse	7
2	Laptop	2
3	Keyboard	1

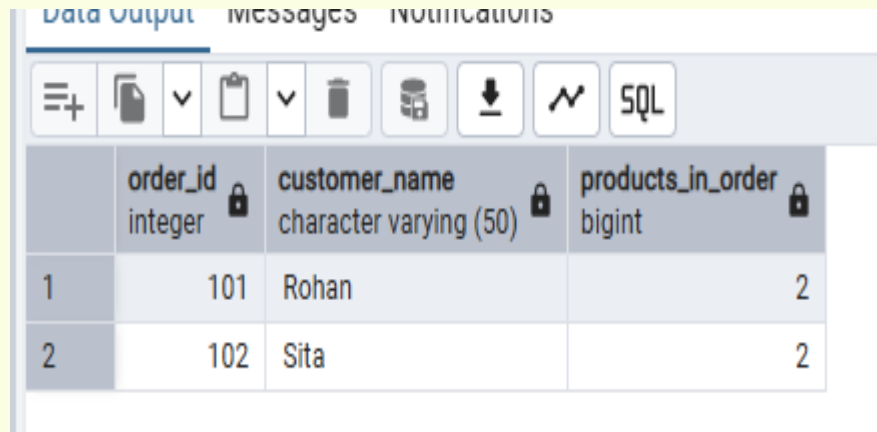
Insight: The output highlights which products are most popular, helping in stock management and marketing focus.

➤ Customers who bought more than one product in the same order

This query lists all customers (and their order IDs) where an order contained more than one product. It's useful for spotting buyers who prefer bundles or large baskets.

```
--Customers who bought more than one product in the same order
SELECT
    o.order_id,
    c.customer_name,
    COUNT(od.product_id) AS products_in_order
FROM Orders o
JOIN Order_Details od ON o.order_id = od.order_id
JOIN Customers c ON c.customer_id = o.customer_id
GROUP BY o.order_id, c.customer_name
HAVING COUNT(od.product_id) > 1
ORDER BY o.order_id;
```

Result:-

A screenshot of a data table interface. At the top, there are tabs labeled 'Data Output', 'Messages', and 'Notifications'. Below the tabs is a toolbar with icons for adding, deleting, and other actions, along with an 'SQL' button. The table itself has four columns: an index column, 'order_id' (integer), 'customer_name' (character varying (50)), and 'products_in_order' (bigint). The data rows are as follows:

	order_id integer	customer_name character varying (50)	products_in_order bigint
1	101	Rohan	2
2	102	Sita	2

Insight: The result highlights customers who often purchase multiple items per order. These are great candidates for upselling, combo offers, or loyalty programs.

Conclusion & Recommendations

The analysis of sales data reveals:

- **Best-selling items** (from total quantity sold) help identify which products should always stay in stock.
- Customers who purchase **multiple products in one order** are valuable for cross-selling and bundle offers.
- Monitoring **monthly trends** and **average order value** allows management to plan seasonal promotions and control inventory more effectively.

Recommendation:

Focus marketing on the top-selling items, encourage bulk purchases with discounts or combos, and keep track of monthly sales fluctuations to align stock levels and campaigns.