2025

NORTHWIND TRADERS ANALYSIS

VINCENT MUKOMBA

TABLE OF CONTENTS

Introduction ................................................................................................................................... 3

Objective of the Analysis ............................................................................................................... 3

Dataset .......................................................................................................................................... 3

**Problem Statement 1:** Second-Best Selling Product by Category ................................................. 4

Strategic Approach: ........................................................................................................ 4

Query: ............................................................................................................................. 5

Output: ........................................................................................................................... 6

Result analysis: ............................................................................................................... 6

Recommendation: .......................................................................................................... 6

**Problem Statement 2:** Top 3 Customers by Total Sales ................................................................... 7

Strategic Approach: .............................................................................................................. 7

Query: ................................................................................................................................... 8

Output Result Snapshot: ...................................................................................................... 9

Result analysis: ..................................................................................................................... 9

Recommendation: ............................................................................................................... 10

**Problem Statement 3:** Top Suppliers by Product Variety................................................................ 11

Strategic Approach: ............................................................................................................. 11

Query: .................................................................................................................................. 12

Output: ................................................................................................................................ 12

Result analysis: .................................................................................................................... 13

Recommendation: ............................................................................................................... 13

**Problem Statement 4:** Most Recent Order per Customer............................................................... 14

Strategic Approach: ............................................................................................................ 14

Query: ................................................................................................................................. 15

Output: ............................................................................................................................... 16

Result analysis: ................................................................................................................... 17

Recommendation: .............................................................................................................. 17

**Problem Statement 5:** Cumulative Sales by Month.......................................................................... 18

STRATEGIC APPROACH: ...................................................................................................... 18

Query: ................................................................................................................................. 19

Output: ............................................................................................................................... 20

Result analysis: ................................................................................................................... 20 Recommendation: .............................................................................................................. 21

**Problem Statement 6:** Days between Customer Orders.................................................................. 22

Strategic Approach: ............................................................................................................ 22

Query: ................................................................................................................................. 23

Output: ............................................................................................................................... 24

Result Analysis: ................................................................................................................... 24

RECOMMENDATION: .......................................................................................................... 24

**Problem Statement 7:** Next Order Date and Reorder Interval........................................................ 25

Strategic Approach: ............................................................................................................ 25

Query: ................................................................................................................................. 26

Output: ............................................................................................................................... 27

Result analysis: ................................................................................................................ 27

Recommendation: ........................................................................................................... 27

**Problem Statement 8:** Highest-Value Order and Its Salesperson.................................................. 28

Strategic Approach: ......................................................................................................... 28

Query: .............................................................................................................................. 29

Output: ............................................................................................................................ 30

Result analysis: ................................................................................................................ 30

Recommendation: ........................................................................................................... 31

# INTRODUCTION

Northwind Traders is a global gourmet food distributor specializing in high-quality products sourced from premium suppliers worldwide. The company manages an extensive catalog of products across multiple categories and serves customers ranging from small businesses to large enterprises.

As a **Data Analyst**, the role is to provide actionable insights that drive strategic decision-making. The company's leadership team relies on data-driven reports to optimize inventory, improve customer relationships, increase sales, and enhance operational efficiency.

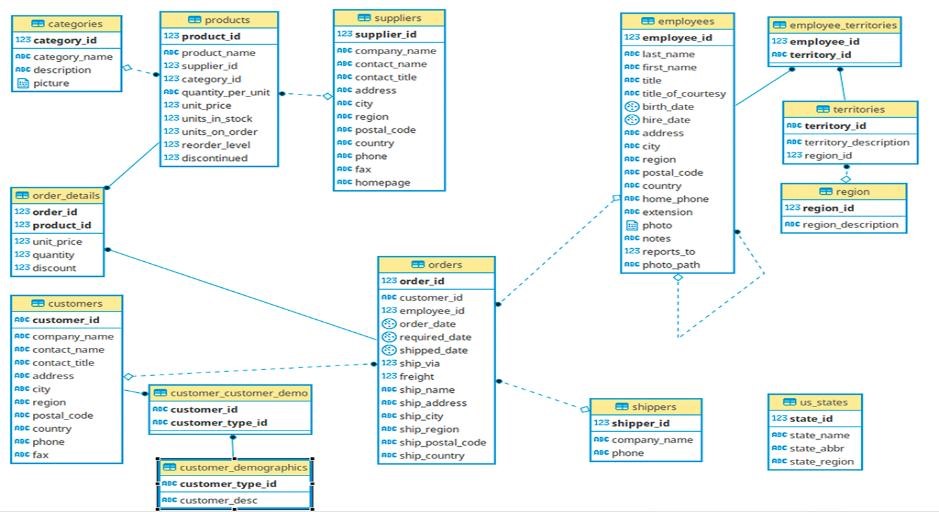
# OBJECTIVE OF THE ANALYSIS

The management has asked to explore and answer critical business questions using advanced SQL techniques. This analysis will focus on:

* Evaluating employee sales performance to recognize top performers.
* Understanding product sales trends to optimize inventory and marketing strategies.
* Identifying high-value customers for targeted promotions.
* Monitoring sales growth and forecasting trends to support company expansion.
* Analyzing supplier contributions to determine procurement efficiency.

# DATASET

[Here](https://drive.google.com/drive/folders/1E0ZutoMBwuErNRdOBQBzXDHE2YxLPPAX?usp=sharing) is the dataset required for the analysis and the schema is shared below:



# PROBLEM STATEMENT 1: SECOND-BEST SELLING PRODUCT BY CATEGORY

Business Scenario: Category managers at Northwind want to promote products that are strong sellers but not the top in their category. They decide to look at the **second-highest grossing product in each category** by total sales revenue. This helps identify products that have high sales potential right behind the category leaders.

**Question:** *Which product is the second-best selling (by total revenue) in each product category, and how much revenue did it generate?*

**Answer:** Provide the category name, product name, and total sales for that product. Use a CTE to organize the calculation.

## STRATEGIC APPROACH:

1. **Schema:** 
   * To solve this, we need to join the following tables:
   * order\_details: has product\_id, unit\_price, quantity, discount (used to calculate revenue)
   * products: has product\_id, product\_name, category\_id
   * categories: has category\_id, category\_name
2. **CTE: Revenue by Product** 
   * Calculate total revenue per product (accounting for discounts):
   * Join order\_details and products
   * Group by product\_id
   * Revenue formula: SUM(unit\_price \* quantity \* (1 - discount))

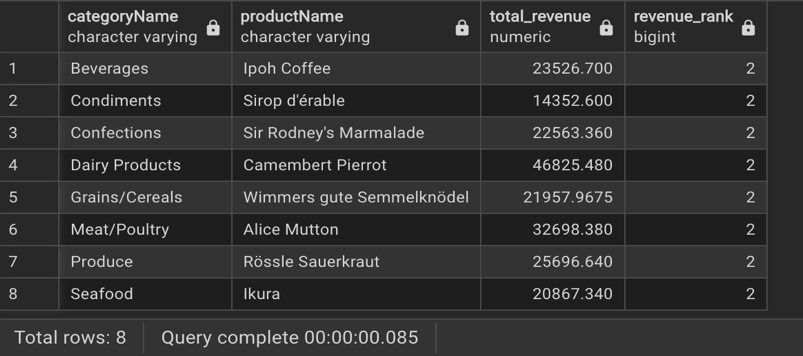
1. **CTE: Add Category Info** 
   * Join the revenue result with products and categories to get:
   * category\_id
   * category\_name
   * product\_name
   * Total\_revenue
2. **CTE: Rank Products Within Categories** 
   * Use ROW\_NUMBER() or DENSE\_RANK() window function partitioned by category\_id, ordered by revenue descending
   * This helps to assign ranks (1 = highest revenue, 2 = second-highest, etc.)

1. **Final Step: Filter for Rank = 2** 
   * Only select products where rank = 2 (i.e., second-highest revenue in that category)

## QUERY:

|  |
| --- |
| WITH product\_revenue AS (  SELECT  c."categoryName",  p."productName",  c."categoryID",  p."productID",  SUM(od."unitPrice" \* od."quantity" \* (1 - od."discount")) AS total\_revenue  --Total Revenue for Each Product FROM  northwind\_traders."order\_details" AS od  JOIN northwind\_traders."products" AS p ON od."productID" = p."productID"  JOIN northwind\_traders."categories" AS c ON p."categoryID" = c."categoryID"  GROUP BY  c."categoryName", c."categoryID", p."productID", p."productName" ),  ranked\_products AS (  SELECT  "categoryName", "productName", total\_revenue,  RANK() OVER (PARTITION BY "categoryID" ORDER BY total\_revenue DESC) AS revenue\_rank --Rank Products Within Each Category by Revenue FROM  product\_revenue  )  --Select Only the Second-Best  SELECT  "categoryName", "productName", total\_revenue,  revenue\_rank  FROM ranked\_products WHERE  revenue\_rank = 2 --Filter to revenue\_rank = 2  ORDER BY  "categoryName"; -- second-highest grossing products per category |

## OUTPUT:



## RESULT ANALYSIS:

1. Camembert Pierrot (Dairy) and Alice Mutton (Meat/Poultry) stand out with high second-place revenues, indicating these products are strong contenders for top positions.
2. Even the second-place products generate substantial revenue, suggesting the sales volume is not heavily skewed toward a single product in each category — this is a good sign of product diversity and demand.
3. Revenue gaps between first and second place (not shown here, but implied) can inform promotional prioritization.

## RECOMMENDATION:

1. **Promote Products**: Target second-best sellers with marketing campaigns to boost them toward the top.
2. **Ensure Stock Availability**: Maintain strong inventory for these high-performing items to meet demand.
3. **Use in Bundles**: Pair them with best-sellers for cross-sell opportunities and increased sales.

# PROBLEM STATEMENT 2: TOP 3 CUSTOMERS BY TOTAL SALES

Business Scenario: The sales team is planning a loyalty program and wants to reward the top 3 customers by purchase volume. If there is a tie for third place, all tied customers should be included. Knowing the biggest spenders will help tailor special offers to them.

Question: *Who are the top three customers in terms of total sales revenue?*

Show customer’s name, total spending, and their sales rank. Include any customers tied for third place by using a window function to rank the totals.

## STRATEGIC APPROACH:

1. **Schema:**

To solve this, we need to join the following tables:

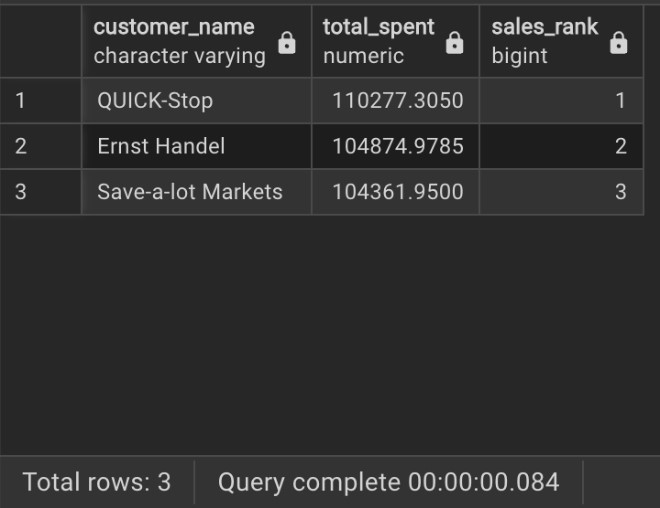
* + orders: contains order\_id, customer\_id
  + order\_details: contains order\_id, unit\_price, quantity, discount
  + customers: contains customer\_id, company\_name (customer name)

1. **CTE: Total Revenue per Order**
   * Join order\_details and orders
   * total\_revenue = unit\_price \* quantity \* (1 - discount)
   * Group by order\_id
2. **CTE: Total Revenue per Customer**
   * Join the revenue per order back with the orders table (to get customer\_id)
   * Sum the revenue **per customer**
   * Join with customers to get company\_name
3. **CTE: Rank Customers by Total Spending**
   * Use a window function like RANK() or DENSE\_RANK() ordered by total spending **descending**
   * RANK() if you want to leave gaps in case of ties
   * DENSE\_RANK() if you want contiguous ranks even if there are ties
4. **Final Step: Filter for Rank = 2**
   * Filter for customers where rank <= 3
   * This ensures:
   * You get the top 3 spenders
   * And you **include any ties** at rank 3

## QUERY:

|  |
| --- |
| WITH customer\_revenue AS (  SELECT  c."companyName" AS customer\_name, -- taking company name as a customer name bcz customer name is not given in the schema  o."customerID",  SUM(od."unitPrice" \* od."quantity" \* (1 - od."discount")) AS total\_spent --The portion of the price that is paid (1 - od."discount") FROM  northwind\_traders."order\_details" AS od  JOIN northwind\_traders."orders" AS o ON od."orderID" = o."orderID"  JOIN northwind\_traders."customers" AS c ON o."customerID" = c."customerID"  GROUP BY  o."customerID", c."companyName"  ),  --Rank customers using RANK() ranked\_customers AS ( SELECT  customer\_name, total\_spent,  DENSE\_RANK() OVER (ORDER BY total\_spent DESC) AS sales\_rank  FROM |
| customer\_revenue  )  --top 3 ranks including ties at 3rd SELECT  customer\_name, total\_spent, sales\_rank FROM  ranked\_customers WHERE  sales\_rank <= 3 ORDER BY sales\_rank; |
|  |
|  |

## OUTPUT RESULT SNAPSHOT:



## RESULT ANALYSIS:

* All three customers have spent over **$100K**, making them high-value clients.
* The spending gap between rank 2 and 3 is small (~$513), indicating a closely competitive top tier.

## RECOMMENDATION:

1. **Launch Loyalty Rewards**: Offer exclusive perks (discounts, early access) to retain these top customers.
2. **Personalized Engagement**: Assign account reps or create custom offers to strengthen relationships.
3. **Upsell Opportunities**: Target them with premium products or bulk-buy incentives to boost future sales.

# PROBLEM STATEMENT 3: TOP SUPPLIERS BY PRODUCT VARIETY

Business Scenario: The procurement department wants to evaluate supplier partnerships. They are interested in which suppliers offer the widest variety of products. Using a ranking that does not skip numbers (so ties share the same rank), they can list the top suppliers by product count to see who has a broad catalog.

Question: *Which suppliers provide the most products to Northwind, and how do they rank in terms of product count?*

List each top supplier’s name, the number of different products they supply, and their rank. Use DENSE\_RANK so that suppliers with equal product counts share the same rank.

## STRATEGIC APPROACH:

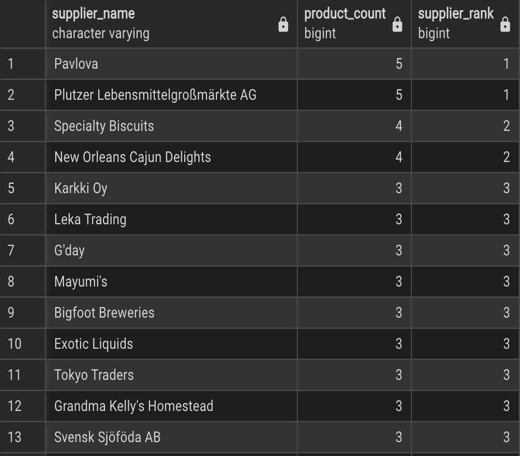
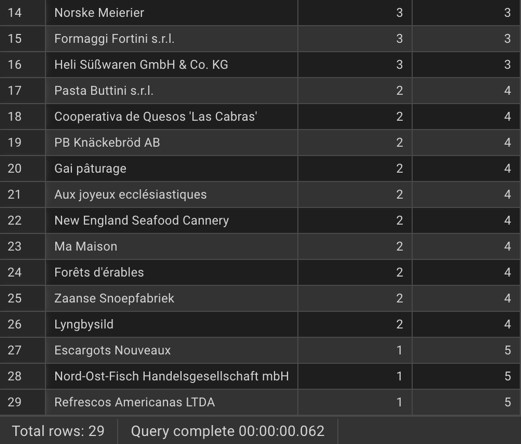
1. **Schema:** 
   * products: contains product\_id, supplier\_id, etc.
   * suppliers: contains supplier\_id, company\_name (or similar)
2. **CTE: Count of Products per Supplier** 
   * Group products by supplier\_id
   * Count the number of products (e.g., COUNT(\*))
   * Join with suppliers table to get the supplier’s name
3. **CTE: Rank Suppliers by Product Count** 
   * Use DENSE\_RANK() window function
   * Partitioning is not needed (rank globally)
   * Order by product count **descending**

## QUERY:

|  |
| --- |
| WITH supplier\_product\_count AS (  SELECT  s."companyName" AS supplier\_name, COUNT(p."productID") AS product\_count  FROM  northwind\_traders."products" AS p  JOIN northwind\_traders."suppliers" AS s ON p."supplierID" = s."supplierID"  GROUP BY  s."companyName". --Count of Products per Supplier  ),  --Rank Suppliers by Product Count  ranked\_suppliers AS ( SELECT supplier\_name, product\_count,  DENSE\_RANK() OVER (ORDER BY product\_count DESC) AS supplier\_rank  FROM  supplier\_product\_count  )  SELECT  supplier\_name, product\_count,  supplier\_rank  FROM  ranked\_suppliers  ORDER BY  supplier\_rank; |

OUTPUT

:



## RESULT ANALYSIS:

* Two suppliers tie for the top spot with **5 distinct products** each.
* A total of **29 suppliers** are involved, showcasing a diversified supplier base.
* Many suppliers provide **3 or more products**, reflecting stable procurement sources.

## RECOMMENDATION:

1. **Strengthen Partnerships** with top-ranked suppliers to ensure consistent supply and negotiate better terms.
2. **Consolidate Orders** with suppliers offering a broader catalog to reduce complexity and shipping costs
3. **Evaluate Low Variety Suppliers** for potential consolidation or secondary supplier status.

# PROBLEM STATEMENT 4: MOST RECENT ORDER PER CUSTOMER

Business Scenario: The customer relations team wants to improve engagement by contacting customers who haven’t ordered recently. They need to know the date of each customer’s most recent order. This information will be used to schedule follow-up calls or emails, focusing on those whose last orders were a while ago.

Question: *For each customer, what is the date of their latest order?* Provide the customer name and the date of their most recent order.

Use a window function to pick the most recent order for each customer.

## STRATEGIC APPROACH:

1. **Schema**

To solve this, we need to join the following tables:

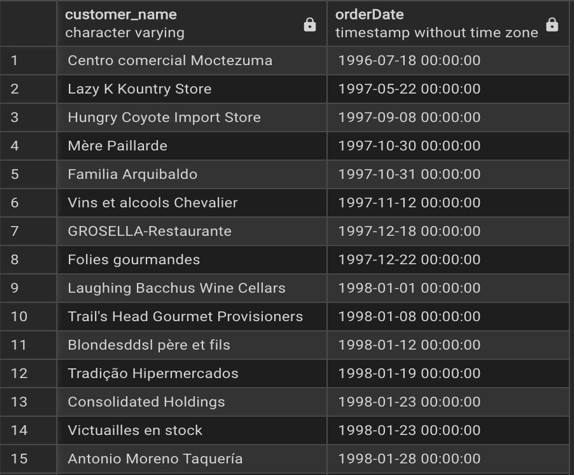
* + orders: contains order\_id, customer\_id, order\_date
  + customers: contains customer\_id, company\_name

1. **Join Customers with Orders** 
   * Join customers to orders using customer\_id
   * This gives access to both company\_name and their order\_dates
2. **Use a Window Function to Rank Orders per Customer** 
   * Use ROW\_NUMBER() or RANK() **partitioned by customer\_id**, ordered by order\_date DESC
   * This assigns a 1st rank to the most recent order for each customer
3. **Filter for the Most Recent Order Only** 
   * Wrap the above in a CTE or subquery
   * Filter for row\_number = 1 to get only the **latest order** for each customer

## QUERY:

|  |
| --- |
| --Join Customers with Orders  WITH ranked\_orders AS (  SELECT  c."companyName" AS customer\_name,  o."orderDate",  --Window Function to Rank Orders per Customer  RANK() OVER ( PARTITION BY o."customerID" ORDER BY o."orderDate" DESC ) AS order\_rank FROM  northwind\_traders."orders" AS o  JOIN northwind\_traders."customers" AS c  ON o."customerID" = c."customerID"  ) SELECT  customer\_name,  "orderDate" FROM  ranked\_orders WHERE  order\_rank = 1 --to get only the latest order for each customer ORDER BY  "orderDate" ASC; -- oldest recent orders first |

## OUTPUT:



## RESULT ANALYSIS:

* The data lists **each customer's most recent order date** using a window function.
* The **latest order date** in the dataset is **1998-04-27**.
* Many customers haven’t placed an order since **early 1998**, indicating a need for re-engagement.

## RECOMMENDATION:

1. **Prioritize Outreach** to customers who haven't ordered since early 1998 to revive engagement.
2. **Segment Customers** by recency tiers (e.g., 3+ months inactive) for targeted follow-up campaigns.
3. **Automate Reminders** to proactively nudge customers before long gaps in ordering occur.

# PROBLEM STATEMENT 5: CUMULATIVE SALES BY MONTH

Business Scenario: The finance department is tracking sales trends over time. They want a month-by- month sales report for 1997 that includes the running total revenue up to the end of each month. This cumulative total helps visualize growth and determine if sales targets are being met as the year progresses.

Question: *How can we calculate the cumulative year-to-date sales total at the end of each month in 1997?*

List each month of 1997, the sales for that month, and the running total of sales through that month.

## STRATEGIC APPROACH:

1. **Schema:**

To solve this, we need to join the following tables:

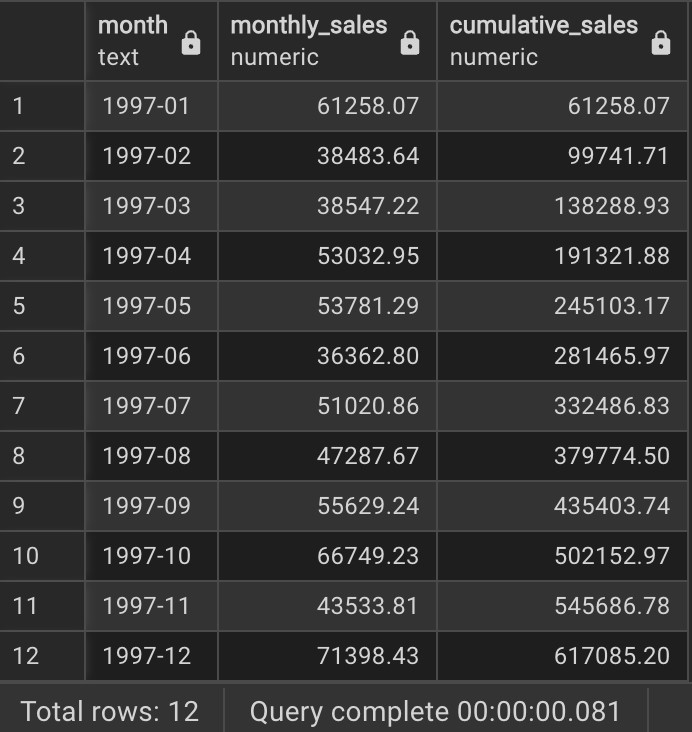
* + orders: contains order\_date, order\_id
  + order\_details: contains unit\_price, quantity, discount, order\_id

1. **Join Orders with Order Details**
   * Link orders and order\_details using order\_id
   * This will calculate revenue per order line
2. **Filter to Only Orders from 1997**
   * Use a filter like WHERE EXTRACT(YEAR FROM order\_date) = 1997
3. **Group by Month**
   * Extract month from order\_date
   * Group by month and sum revenue using: unit\_price \* quantity \* (1 - discount)
   * This will give **monthly sales**
4. **Use a Window Function for the Running Total**
   * Used SUM(monthly\_sales) OVER (ORDER BY month) to get the cumulative sum • This provides the year-to-date total as each month progresses

## QUERY:

|  |
| --- |
| WITH monthly\_sales AS (  SELECT  DATE\_TRUNC('month', o."orderDate") AS month,  --monthly sales  SUM(od."unitPrice" \* od."quantity" \* (1 - od."discount")) AS monthly\_revenue FROM  northwind\_traders."orders" AS o  JOIN northwind\_traders."order\_details" AS od  ON o."orderID" = od."orderID"  WHERE  EXTRACT(YEAR FROM o."orderDate") = 1997  GROUP BY  DATE\_TRUNC('month', o."orderDate")  )  SELECT  TO\_CHAR(month, 'YYYY-MM') AS month,  ROUND(monthly\_revenue, 2) AS monthly\_sales,  --to get the cumulative sum  ROUND(SUM(monthly\_revenue) OVER (ORDER BY month), 2) AS cumulative\_sales FROM  monthly\_sales ORDER BY month; |

## OUTPUT:



## RESULT ANALYSIS:

* **Total sales in 1997** reached **617,085.20** units of currency.
* The **highest sales month** was **December (71,398.43)**, suggesting strong end-of-year performance—possibly due to holiday demand.
* **Lowest sales** occurred in **June (36,362.80)**, which could indicate a seasonal dip.
* Sales grew steadily through the year, with **notable growth from September to December**, where cumulative sales rose by over **180K** in just four months.
* The growth trend is **non-linear**, showing volatility—especially between February, March, and April, followed by a recovery mid-year.

## RECOMMENDATION:

1. **Capitalize on Year-End Momentum**: Increase marketing spend and promotions in Q4 to amplify already strong sales trends.
2. **Investigate June Dip**: Explore internal and external factors behind the June sales drop (e.g., fewer promotions, supply chain issues).
3. **Set Quarterly Benchmarks**: Use cumulative targets per quarter to better align sales efforts and track against goals.

# PROBLEM STATEMENT 6: DAYS BETWEEN CUSTOMER ORDERS

**Business Scenario:** Marketing analysts are studying **customer reorder patterns**. For each order a customer makes (after their first), they want to know how many days have passed since that customer’s previous order. This helps identify purchasing frequency—whether customers order weekly, monthly, etc.—to tailor marketing communications.

**Question:** *For each customer order (except their first), how many days elapsed since the customer’s prior order?*

Show the customer name, the order date, and the number of days since that customer’s previous order.

Use the LAG window function to access the date of the prior order.

## STRATEGIC APPROACH:

1. **Schema:**

To solve this, we need to join the following tables:

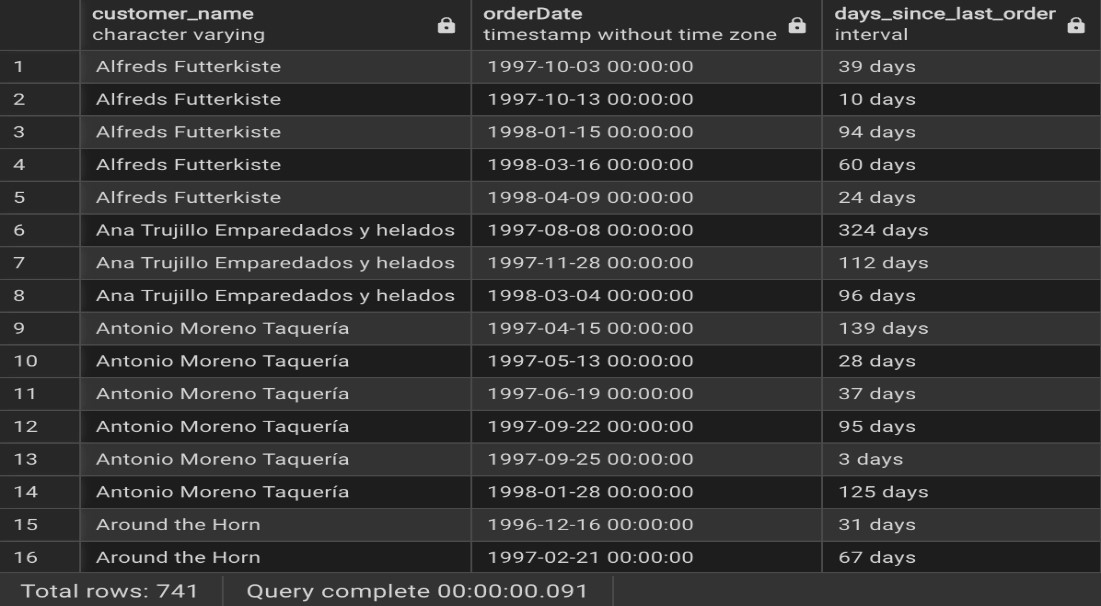
* + orders: contains order\_id, order\_date, customer\_id
  + customers: contains customer\_id, company\_name

1. **Join Orders with Customers**
   * Join orders with customers on customer\_id
   * This will give access to both order\_date and company\_name
2. **Use the LAG() Function**
   * Use LAG(order\_date) to get the **previous order’s date** for the same customer
   * Use PARTITION BY customer\_id so the lag only considers **that customer's history**
   * Use ORDER BY order\_date to ensure the order sequence is correct
3. **Calculate Days Between Orders**
   * Subtract: order\_date - previous\_order\_date
   * This gives the **number of days since the last order**

## QUERY:

|  |
| --- |
| WITH customer\_orders AS (  SELECT  c."companyName" AS customer\_name,  o."orderDate",  --get the previous order's date for the same customer  LAG(o."orderDate") OVER (PARTITION BY o."customerID" ORDER BY  o."orderDate") AS previous\_order\_date FROM  northwind\_traders."orders" AS o  JOIN northwind\_traders."customers" AS c  ON o."customerID" = c."customerID"  ) SELECT  customer\_name,  "orderDate",  --This gives the number of days since the last order  ("orderDate" - previous\_order\_date) AS days\_since\_last\_order FROM  customer\_orders WHERE  previous\_order\_date IS NOT NULL ORDER BY  customer\_name, "orderDate"; |

## OUTPUT:



## RESULT ANALYSIS:

* **Wide variability in reordering behavior**:
  + **Alfreds Futterkiste** shows both short gaps (10 days) and longer ones (94 days).
  + **Ana Trujillo Emperadados y helados** had a **324-day gap**, indicating potential churn risk or seasonality.
* Some customers like **Antonio Moreno Taquería** had highly irregular patterns—ranging from **3 days to 139 days**.

**Around the Horn** had orders spaced **31 and 67 days apart**, suggesting a **monthly cadence**.

## RECOMMENDATION:

1. **Segment by Frequency**: Group customers by reorder intervals (e.g., <30 days = frequent, 30–90 = moderate, >90 = infrequent) to tailor campaigns.
2. **Re-engage Inactive Customers**: Target those with >90-day gaps using win-back promotions or personalized outreach.
3. **Predictive Reminders**: Set automated reminders or discounts **just before the average reorder window** to nudge repeat purchases.

# PROBLEM STATEMENT 7: NEXT ORDER DATE AND REORDER INTERVAL

**Business Scenario:** Continuing the analysis of customer ordering habits, the team now looks forward. After each order, they want to know **when the next order from the same customer occurred** and the gap in days between the two orders. This forward-looking gap (until the next order) can indicate how quickly customers come back to buy again.

**Question:** *For each customer order (except their last), what was the date of the next order by that same customer, and how many days later did it occur?*

Show the customer name, the current order date, the next order date, and the interval in days. Use the LEAD function to get the next order’s date.

## STRATEGIC APPROACH:

1. **Schema:**

To solve this, we need to join the following tables:

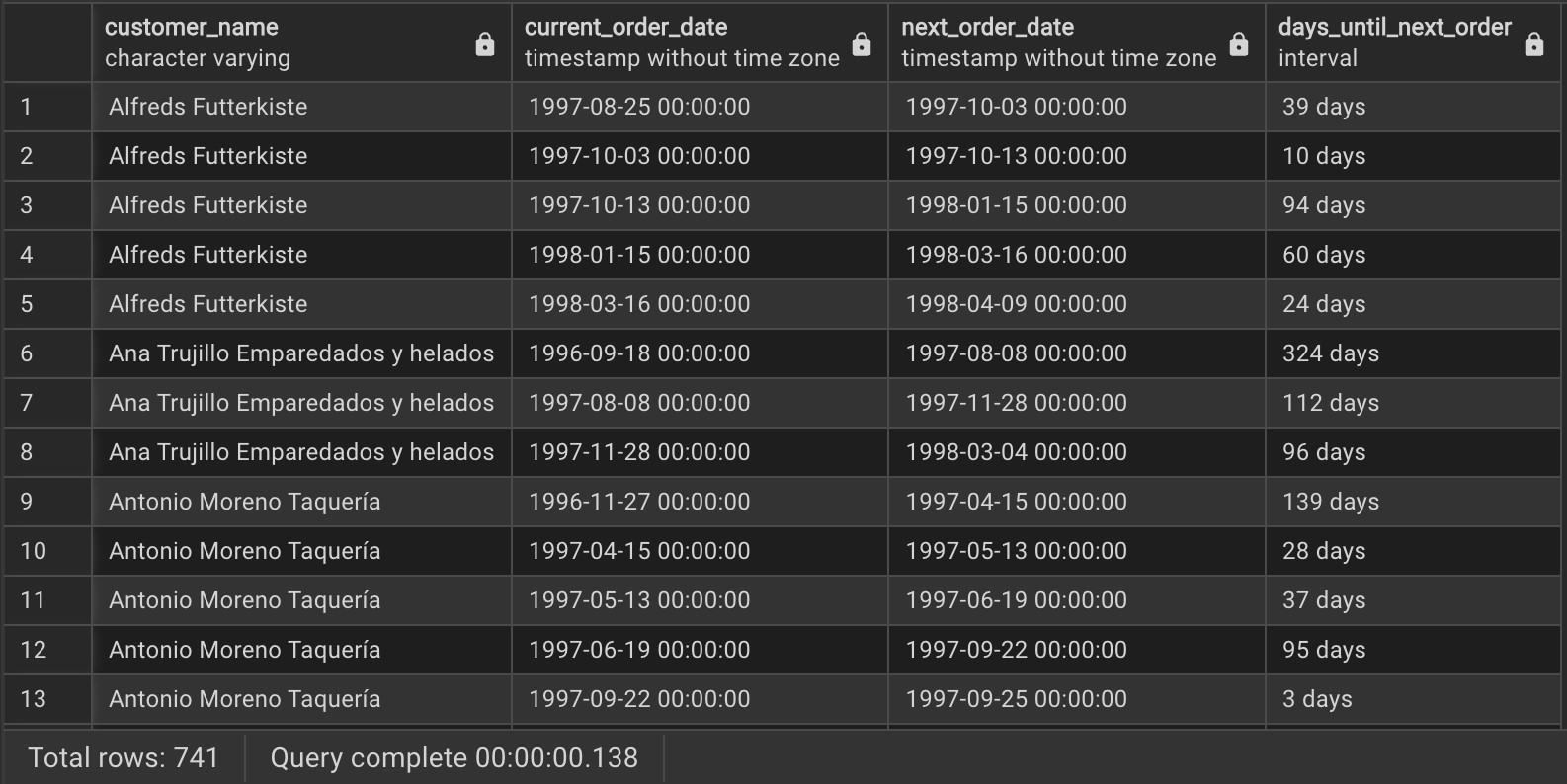
* + orders: contains order\_id, order\_date, customer\_id
  + customers: contains customer\_id, company\_name

1. **Join Orders with Customers**
   * Join orders with customers on customer\_id
   * This will give both the customer name and their order date
2. **Use the LEAD() Window Function**
   * Use LEAD(order\_date) to get the **next order date**
   * Use PARTITION BY customer\_id, only look at orders within the **same customer**
   * Use ORDER BY order\_date to ensure order sequence is chronological
3. **Calculate the Reorder Interval**
   * Subtract: next\_order\_date - current\_order\_date
   * This gives the **number of days** until the next order
4. **Use a Window Function for the Running Total**
   * Used SUM(monthly\_sales) OVER (ORDER BY month) to get the cumulative sum
   * This provides the year-to-date total as each month progresses.

## QUERY:

|  |
| --- |
| WITH customer\_orders AS (  SELECT  c."companyName" AS customer\_name,  o."orderDate" AS current\_order\_date,  --to get the next order date  LEAD(o."orderDate") OVER (  PARTITION BY o."customerID"  ORDER BY o."orderDate"  ) AS next\_order\_date FROM  northwind\_traders."orders" o  JOIN northwind\_traders."customers" c  ON o."customerID" = c."customerID"  ) SELECT  customer\_name, current\_order\_date, next\_order\_date,  -- Calculate the gap in days number of days until the next order next\_order\_date - current\_order\_date AS days\_until\_next\_order FROM  customer\_orders WHERE  next\_order\_date IS NOT NULL ORDER BY customer\_name, current\_order\_date; |

## OUTPUT:



## RESULT ANALYSIS:

* **Alfreds Futterkiste** maintains a mostly regular reordering cycle (10–94 days) with **an average of ~45 days** between orders.
* **Ana Trujillo** has **long intervals (up to 324 days)**, suggesting either seasonal buying patterns or reduced engagement.
* **Antonio Moreno Taquería** appears very active over short bursts, with orders just **3 days apart**, but also has gaps over **130 days**.

## RECOMMENDATION:

1. **Use LEAD-based reorder gaps for forecasting**: Predict when a customer is likely to reorder again and proactively send reminders.
2. **Implement smart nudges**: If the reorder gap exceeds their historical average, trigger a promo email or check-in campaign.
3. **Create customer lifecycle stages**: Classify customers by expected return window (e.g., “likely to reorder in 30 days”) for CRM automation.

# PROBLEM STATEMENT 8: HIGHEST-VALUE ORDER AND ITS SALESPERSON

Business Scenario: Senior management wants to highlight the single largest order in terms of revenue, and recognize the employee who handled it. Knowing which order brought in the most money (and who was responsible for it) can be useful for awards or case studies on successful sales.

Question: *Which order had the highest total value, and which employee handled that order?*

Provide the order ID, the total order amount, and the full name of the employee who handled it. Use an aggregate subquery or CTE to identify the maximum order total.

## STRATEGIC APPROACH:

1. **Schema:**

To solve this, we need to join the following tables:

* + orders: contains order\_id, employee\_id
  + order\_details: orderID, unitPrice, quantity, discount
  + Employees: employee\_id, firstName, lastName

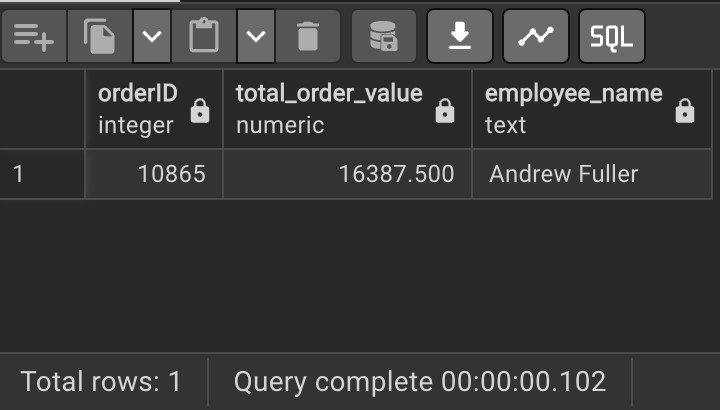
1. **Calculate Total Value per Order**
   * Use the order\_details table to compute:
   * total\_order\_value = unit\_price × quantity × (1 - discount)
   * Group this by orderID to get **total revenue per order**.
2. **Join with Employee Info**
   * **Use the orders table to:**
   * **Link each orderID to the employeeID.**
   * Join with the employees table to get the **employee's full name**.
3. **Find the Maximum Order Value**
   * Use either:
   * A **CTE** or A **subquery**
   * **Find the maximum of all total\_order\_values.**
4. **Filter for the Highest Order**
   * Compare the total\_order\_value from each order with the **maximum** found above.
   * Return the **order ID**, **total value**, and the **employee name** for that highest-value order.

## QUERY:

|  |
| --- |
| WITH order\_totals AS ( --calculating the total revenue per order. SELECT  o."orderID",  e."firstName" || ' ' || e."lastName" AS employee\_name, SUM(od."unitPrice" \* od."quantity" \* (1 - od."discount")) AS total\_order\_value FROM  northwind\_traders."orders" o  JOIN northwind\_traders."order\_details" od ON o."orderID" = od."orderID"  JOIN northwind\_traders."employees" e ON o."employeeID" = e."employeeID"  GROUP BY  o."orderID", e."firstName", e."lastName" -- to get totals per order.  ),  --This part finds the maximum order value across all orders from the previous step. max\_order AS (  SELECT  MAX(total\_order\_value) AS max\_value  FROM |
| order\_totals  )  --join the order\_totals CTE with the max\_order CTE. SELECT  ot."orderID", ot.total\_order\_value, ot.employee\_name FROM  order\_totals ot  JOIN max\_order mo ON ot.total\_order\_value = mo.max\_value; |

OUTPUT

:



## RESULT ANALYSIS:

* **Order ID**: 10865
* **Total Order Value**: **16,387.50**
* **Employee**: **Andrew Fuller**
* Andrew Fuller handled the **highest-value order** in the dataset, making him a **top-performing salesperson** for high-revenue deals. This order likely represents a significant client or bulk purchase and can be studied as a **model case for successful sales**.

## RECOMMENDATION:

1. **Recognize & Reward**: Highlight Andrew Fuller for this high-value achievement to boost motivation across the team.
2. **Analyze this deal further**: Understand the customer, product mix, and discount structure for replicable strategies.
3. **Train using top performance cases**: Use this order as a case study in sales training programs.