Retail Sales Analysis using PostgreSQL and Power BI

General Objective

To design and implement a complete retail sales analysis pipeline using PostgreSQL for data storage, cleaning, and querying, complemented by Power BI for interactive visualization, to uncover key business insights and support data-driven decision making.

Specific Objectives

- To import, clean, and structure raw retail sales data from multiple Excel sheets into PostgreSQL, ensuring data consistency and accuracy.
- To perform SQL-based exploratory data analysis (EDA), addressing questions related to product performance, customer behavior, regional sales, and discount strategies.
- To calculate key business metrics such as total revenue, profit, average order value, and customer lifetime value using SQL queries.
- To identify sales trends over time (monthly, quarterly, yearly) and evaluate how customer demographics influence purchasing behavior.
- To build a Power BI dashboard that visually summarizes the insights obtained from SQL queries, enabling clear communication of findings to stakeholders.
- To document the full workflow (data ingestion → SQL analysis → visualization) as a complete portfolio project demonstrating data analytics skills.

Business Questions You Can Answer

- Which products and categories drive the most sales?
- What is the trend of sales over time (monthly/quarterly/yearly)?
- Which stores or regions perform best?
- How do customer demographics (age, gender, city) influence sales?
- What is the average order value and customer lifetime value?
- How do discounts affect revenue?
- What is the profit Margin by Product?

Dataset Description

This dataset simulates a retail business environment, capturing customer purchases across multiple stores and product categories. It was obtained from Kaggle – Retail Sales Dataset and is realistic enough to demonstrate business insights.

- 4 Tables with realistic business relationships (star schema).
- 5,000+ transactions for meaningful trend analysis.
- Rich enough to showcase calculated columns (Age, Tenure, Profit) and measures (Sales, Profit, AOV, YOY Growth).
- Suitable for building interactive dashboards in Power BI.

Tables Overview

1. Customers

Contains demographic and membership details of customers.

- Key Field: CustomerID
- Columns: First name, last name, gender, birth date, city, join date.
- Use Cases:
    - Calculate customer age and tenure.
    - Segment customers into age groups.
    - Analyse sales by gender, city, or customer loyalty.

2. Products

Provides product-level information, pricing, and categories.

- Key Field: ProductID
- Columns: Product name, category, subcategory, unit price, cost price.
- Use Cases:
    - Calculate profit per unit.
    - Compare sales and profitability across categories.
    - Identify top-performing products.

3. Stores

Represents retail store locations and their regions.

- Key Field: StoreID
- Columns: Store name, city, region.
- Use Cases:
    - Compare store sales and performance.
    - Analyse regional trends.
    - Identify top and underperforming stores.

4. Transactions

Captures each sales transaction. This is the fact table in the data model.

- Key Field: TransactionID
- Columns: Date, customer ID, product ID, store ID, quantity, discount, payment method.
- Use Cases:
    - Calculate total sales and profit.
    - Track sales over time (monthly, yearly).
    - Evaluate discount impact.
    - Analyse customer purchase behaviour.

Data Collection and Schema

- The tables were created in PostgreSQL using SQL CREATE TABLE statements. Example schema:

```
CREATE TABLE Customers (
   CustomerID VARCHAR(10) PRIMARY KEY,
   FirstName VARCHAR(50),
   LastName VARCHAR(50),
   Gender CHAR(1),
   BirthDate DATE,
   City VARCHAR(50),
   JoinDate DATE
);
```

- Since the original Excel file had multiple sheets, each sheet was saved as a separate CSV**.** Data was imported into PostgreSQL using the COPY command:

```
COPY Customers
FROM 'File_path.CSV'
DELIMITER ','
CSV HEADER;
```

Exploratory Data Analysis (EDA) and Data Cleaning

- **Identify and handle missing values**.

```
SELECT
   COUNT(*) FILTER (WHERE CustomerID IS NULL) AS missing_CustomerID,
   COUNT(*) FILTER (WHERE FirstName IS NULL) AS missing_FirstName,
   COUNT(*) FILTER (WHERE LastName IS NULL) AS missing_LastName,
   COUNT(*) FILTER (WHERE Gender IS NULL) AS missing_Gender,
   COUNT(*) FILTER (WHERE BirthDate IS NULL) AS missing_BirthDate,
   COUNT(*) FILTER (WHERE City IS NULL) AS missing_City,
   COUNT(*) FILTER (WHERE JoinDate IS NULL) AS missing_JoinDate
FROM Customers;
```

Data Output | Messages | Notifications

	missing_customerid bigint	missing_firstname bigint	missing_lastname bigint	missing_gender bigint	missing_birthdate bigint	missing_city bigint	missing_joindate bigint
1	0	0	0	0	0	0	0

- **Check for duplicate records**.

```
SELECT CustomerID, BirthDate, JoinDate, COUNT(*)
FROM Customers
GROUP BY 1,2,3
HAVING COUNT(*) > 1;
```

- **Validate date formats**.

```
SELECT MIN(Date) AS first_transaction, MAX(Date) AS last_transaction
FROM Transactions;
```

Data Output | Messages | Notifications

	first_transaction date	last_transaction date
1	2023-09-10	2025-09-09

- **Look for invalid or future dates**

```
SELECT *
FROM transactions
WHERE transaction_date > CURRENT_DATE;
```

- **Check unit price and cost price in Products**

```
SELECT
   MIN(UnitPrice) AS min_unit_price,
   MAX(UnitPrice) AS max_unit_price,
   AVG(UnitPrice) AS avg_unit_price,
   MIN(CostPrice) AS min_cost_price,
   MAX(CostPrice) AS max_cost_price,
   AVG(CostPrice) AS avg_cost_price
FROM Products;
```

Data Output | Messages | Notifications

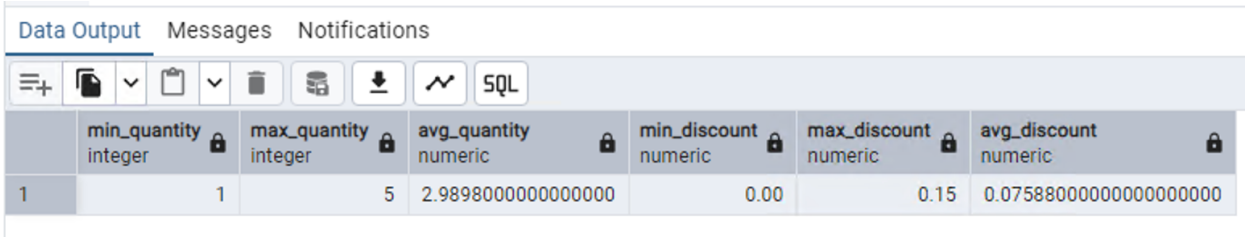
	min_unit_price numeric	max_unit_price numeric	avg_unit_price numeric	min_cost_price numeric	max_cost_price numeric	avg_cost_price numeric
1	25.57	1952.65	1028.2488000000000000	15.28	1451.27	696.3930000000000000

- **Detect negative or zero prices**

```
SELECT *
FROM Products
WHERE UnitPrice <= 0 OR CostPrice <= 0;
```

- **Check quantity and discount in Transactions**

```
SELECT
   MIN(Quantity) AS min_quantity,
   MAX(Quantity) AS max_quantity,
   AVG(Quantity) AS avg_quantity,
   MIN(Discount) AS min_discount,
   MAX(Discount) AS max_discount,
   AVG(Discount) AS avg_discount
FROM Transactions;
```



	min_quantity integer	max_quantity integer	avg_quantity numeric	min_discount numeric	max_discount numeric	avg_discount numeric
1	1	5	2.9898000000000000	0.00	0.15	0.07588000000000000000

- **Detect invalid values**

```
SELECT *
FROM Transactions
WHERE Quantity <= 0 OR Discount < 0 OR Discount > 1;
```

Business Analysis

- **Top products by total sales and percentage of units sold.**
  Shows which products drive the most revenue for the business, can guides marketing, inventory, and strategic focus.

```
WITH total_units AS (
   SELECT SUM(Quantity) AS total_units
   FROM Transactions
),
product_summary AS (
   SELECT
      p.ProductID,
      p.ProductName,
      p.category,
      (SELECT SUM(t.Quantity)
       FROM Transactions t
```

```
        WHERE t.ProductID = p.ProductID) AS total_units_sold,
       (SELECT ROUND(SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)),2)
        FROM Transactions t
        WHERE t.ProductID = p.ProductID) AS total_sales,
       (SELECT COUNT(*)
        FROM Transactions t
        WHERE t.ProductID = p.ProductID) AS num_transactions
   FROM Products p
)
SELECT
   ps.ProductID,
   ps.ProductName,
   ps.category,
   COALESCE(ps.total_units_sold,0) AS total_units_sold,
   COALESCE(ROUND(ps.total_sales,2),0) AS total_sales,
   COALESCE(ps.num_transactions,0) AS num_transactions,
   ROUND((COALESCE(ps.total_units_sold,0)::numeric / NULLIF(tu.total_units,0)) * 100,2)
AS pct_of_total_units
FROM product_summary ps
CROSS JOIN total_units tu
ORDER BY ps.total_sales DESC
LIMIT 10;
```

Showing rows

	productid [PK] character varying (10)	productname character varying (100)	category character varying (50)	total_units_sold bigint	total_sales numeric	num_transactions bigint	pct_of_total_units numeric
1	P037	Book Television	Electronics	337	605028.60	115	2.25
2	P023	And Footwear	Fashion	336	585613.02	115	2.25
3	P045	Set Dairy	Groceries	330	573777.08	111	2.21
4	P019	Traditional Laptop	Electronics	317	521031.01	106	2.12
5	P033	Beat Accessories	Fashion	299	498224.74	103	2.00
6	P040	Piece Headphones	Electronics	306	497149.73	101	2.05
7	P001	Like Camera	Electronics	314	482608.51	102	2.10
8	P029	Road Clothing	Fashion	346	479596.90	111	2.31
9	P041	Present Television	Electronics	317	477913.42	101	2.12
10	P004	Four Accessories	Fashion	275	471228.33	98	1.84

- **Top categories by total sales and profit**

```
SELECT
   p.Category,
   SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)) AS total_sales,
   SUM(t.Quantity * (p.UnitPrice * (1 - t.Discount) - p.CostPrice)) AS total_profit
FROM Transactions t
JOIN Products p ON t.Productid = p.Productid
GROUP BY 1
ORDER BY total_sales DESC;
```

Data Output | Messages | Notifications

	category character varying (50)	total_sales numeric	total_profit numeric
1	Electronics	6316917.3150	1632228.9550
2	Fashion	6232279.7455	1660525.1855
3	Groceries	1752706.0870	533560.5270

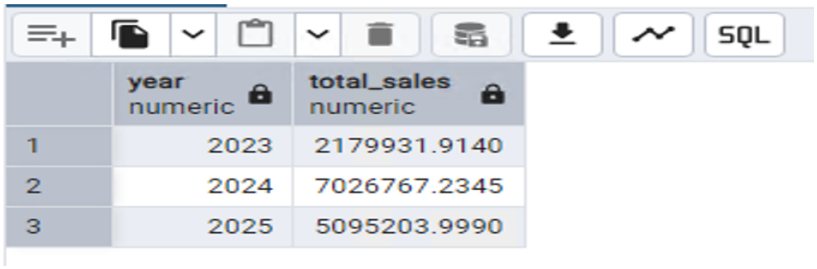
- **Trend of sales Monthly**

```
SELECT
        EXTRACT(YEAR FROM t.Date) AS Year,
        EXTRACT(MONTH FROM t.Date) AS Month,
        SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)) AS total_sales
FROM Transactions t
JOIN Products p ON t.productid = p.Productid
GROUP BY 1,2
ORDER BY Year, Month;
```

	year numeric	month numeric	total_sales numeric
1	2023	9	380992.2780
2	2023	10	561231.1620
3	2023	11	584407.2790
4	2023	12	653301.1950
5	2024	1	584040.1460
6	2024	2	541610.5095
7	2024	3	553927.1820
8	2024	4	590578.4035
9	2024	5	565966.3105
10	2024	6	634753.7710
11	2024	7	555082.5825
12	2024	8	518784.3440
13	2024	9	589517.5645
14	2024	10	661780.3440
15	2024	11	567454.7275
16	2024	12	663271.3495
17	2025	1	628306.7690
18	2025	2	516524.2395
19	2025	3	585784.7300
20	2025	4	708233.4800
21	2025	5	632784.8700
22	2025	6	679817.8090
23	2025	7	610693.2775
24	2025	8	587098.4135
25	2025	9	145960.4105

- **Trend of sales Yearly**

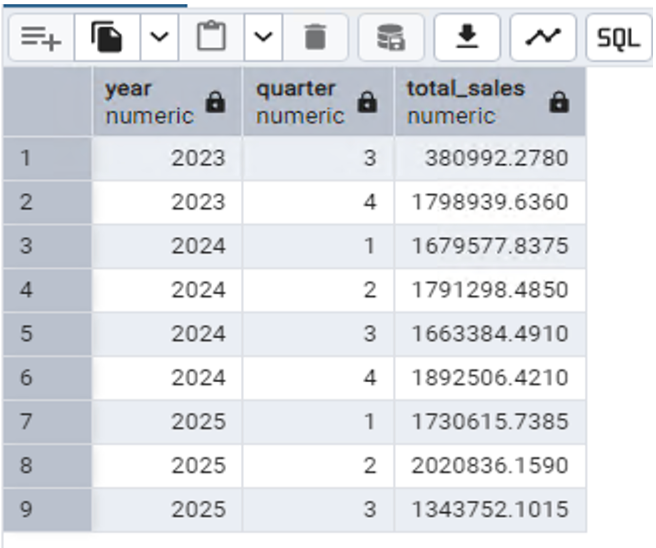
```
SELECT
        EXTRACT (YEAR FROM t.Date) AS Year,
        SUM (t.Quantity * p.UnitPrice * (1 - t.Discount)) AS total_sales
FROM Transactions t
JOIN Products p ON t.Productid = p.Productid
GROUP BY Year
ORDER BY Year;
```



	year numeric	total_sales numeric
1	2023	2179931.9140
2	2024	7026767.2345
3	2025	5095203.9990

- **Trend of sales by Quarterly**

```
SELECT
        EXTRACT (YEAR FROM t.Date) AS Year,
        EXTRACT (QUARTER FROM t.Date) AS Quarter,
        SUM (t.Quantity * p.UnitPrice * (1 - t.Discount)) AS total_sales
FROM Transactions t
JOIN Products p ON t.Productid = p.Productid
GROUP BY 1,2
ORDER BY Year, Quarter;
```



	year numeric	quarter numeric	total_sales numeric
1	2023	3	380992.2780
2	2023	4	1798939.6360
3	2024	1	1679577.8375
4	2024	2	1791298.4850
5	2024	3	1663384.4910
6	2024	4	1892506.4210
7	2025	1	1730615.7385
8	2025	2	2020836.1590
9	2025	3	1343752.1015

- **Best performing stores & cities (Using Windows functions)**

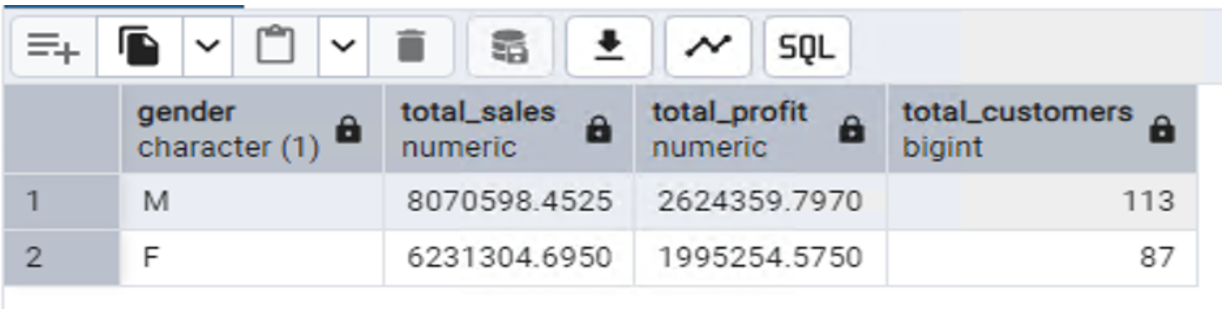
```
SELECT
   s.StoreID,
   s.Region,
   s.City,
   SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)) AS total_sales,
   SUM((p.UnitPrice - p.CostPrice) * t.Quantity * (1 - t.Discount)) AS total_profit,
   RANK() OVER (ORDER BY SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)) DESC)
   AS sales_rank,
   RANK() OVER (ORDER BY SUM((p.UnitPrice - p.CostPrice) * t.Quantity * (1 - t.Discount))
DESC) AS profit_rank
FROM Transactions t
JOIN Products p ON t.ProductID = p.ProductID
JOIN Stores s ON t.StoreID = s.StoreID
GROUP BY 1,2,3
ORDER BY sales_rank;
```

Showing rows

	storeid [PK] character varying (10)	region character varying (50)	city character varying (50)	total_sales numeric	total_profit numeric	total_customers bigint	sales_rank bigint	profit_rank bigint
1	S003	West	New Michele	2920406.9300	961701.0385	198	1	1
2	S004	North	Brianahaven	2900701.0780	933773.2035	199	2	2
3	S001	South	Jimenezborough	2880114.7260	918829.6785	198	3	4
4	S005	East	Johnmouth	2847047.8030	926300.6210	196	4	3
5	S002	East	Peckmouth	2753632.6105	879009.8305	198	5	5

- **Sales by Gender**

```
SELECT
        c.gender,
        SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)) AS total_sales,
        SUM((p.UnitPrice - p.CostPrice) * t.Quantity * (1 - t.Discount)) AS total_profit,
        COUNT(DISTINCT c.customerid) AS total_customers
FROM Transactions t
JOIN Products p ON t.ProductID = p.ProductID
JOIN Customers c ON c.customerid = t.customerid
GROUP BY 1
ORDER BY total_sales DESC;
```

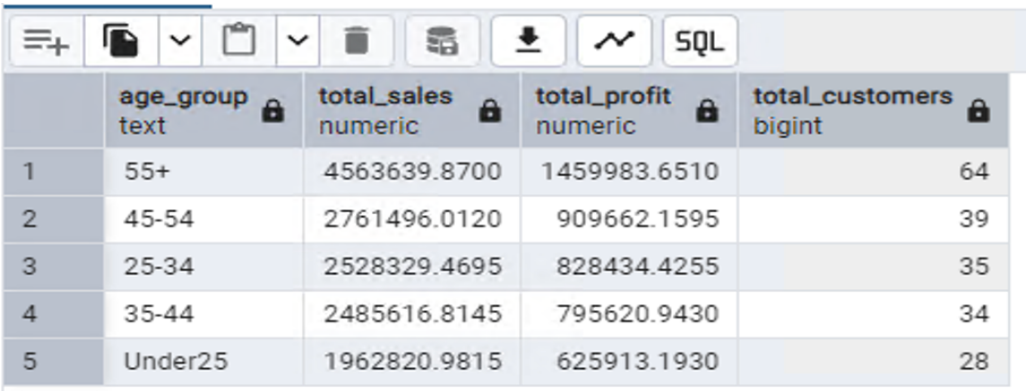


	gender character (1)	total_sales numeric	total_profit numeric	total_customers bigint
1	M	8070598.4525	2624359.7970	113
2	F	6231304.6950	1995254.5750	87

- **Sales by Age Group (Using CTE table)**

```
WITH customer_age as(
	SELECT
		c.customerid,
		DATE_PART('year', AGE(CURRENT_DATE, c.birthdate)) AS age
	FROM Customers c
)

SELECT
	CASE
		WHEN ca.age<25 THEN 'Under25'
		WHEN ca.age BETWEEN 25 AND 34 THEN '25-34'
		WHEN ca.age BETWEEN 35 AND 44 THEN '35-44'
		WHEN ca.age BETWEEN 45 AND 54 THEN '45-54'
		ELSE '55+'
	END AS age_group,
	SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)) AS total_sales,
	SUM((p.UnitPrice - p.CostPrice) * t.Quantity * (1 - t.Discount)) AS total_profit,
	COUNT(DISTINCT ca.customerid) AS total_customers
FROM Transactions t
JOIN Products p ON t.ProductID = p.ProductID
JOIN Customer_age ca ON ca.customerid = t.customerid
GROUP BY age_group
ORDER BY total_sales DESC;
```



	age_group text	total_sales numeric	total_profit numeric	total_customers bigint
1	55+	4563639.8700	1459983.6510	64
2	45-54	2761496.0120	909662.1595	39
3	25-34	2528329.4695	828434.4255	35
4	35-44	2485616.8145	795620.9430	34
5	Under25	1962820.9815	625913.1930	28

- **Customer Value Metrics**

1. Average Order Value (AOV)
- Definition: The average revenue generated per order.
- Formula: AOV = Total Sales / Number of Orders

2. Purchase Frequency (per day)
- Definition: How often a customer makes purchases within their active lifespan.
- Formula: Purchase Frequency = Number of Orders / Lifespan (days)

3. Historical Customer Lifetime Value (CLV)
- Definition: The total amount of revenue a customer has generated so far.
- Formula: Historical CLV = Total Sales (to date)

4. Estimated Annual CLV
- Definition: A projection of a customer's potential annual value, based on past behavior.
- Formula: Estimated Annual CLV = Total Sales Lifespan (days)×365

Relationship:
- AOV shows how much a customer spends per order.
- Purchase Frequency shows how often they order.
- CLV combines both to measure the overall value of a customer.

Query:

```
WITH customer_data AS (
   SELECT
      t.customerid,
      COUNT(DISTINCT t.transactionid) AS number_of_orders,
      ROUND(SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)),2) AS total_sales,
      MIN(t.date) AS first_order_date,
      MAX(t.date) AS last_order_date,
      (MAX(t.date) - MIN(t.date)) AS lifespan_days
   FROM Transactions t
         JOIN Products p ON t.ProductID = p.ProductID
   GROUP BY 1
)
SELECT
   customerid,
   number_of_orders,
   total_sales AS historical_clv,
   lifespan_days,
   ROUND(total_sales / NULLIF(number_of_orders, 0),2) AS AOV,
   ROUND(number_of_orders::NUMERIC/NULLIF(lifespan_days,0),2)
   AS purchase_frequency_per_day,
   ROUND((total_sales / NULLIF(lifespan_days,0)) * 365,2) AS estimated_annual_clv
FROM
   customer_data
ORDER BY
   historical_clv DESC
   LIMIT 10;
```

Data Output | Messages | Notifications

Showir

	customerid character varying (10)	number_of_orders bigint	historical_clv numeric	lifespan_days integer	aov numeric	purchase_frequency_per_day numeric	estimated_annual_clv numeric
1	C012	32	117085.76	707	3658.93	0.05	60447.39
2	C110	31	112255.43	704	3621.14	0.04	58200.61
3	C085	34	111867.56	704	3290.22	0.05	57999.52
4	C077	32	102378.93	682	3199.34	0.05	54792.24
5	C042	26	102068.05	570	3925.69	0.05	65359.37
6	C168	28	101478.73	683	3624.24	0.04	54230.95
7	C186	36	101239.40	683	2812.21	0.05	54103.05
8	C029	30	100719.71	701	3357.32	0.04	52443.22
9	C102	34	99498.95	702	2926.44	0.05	51733.78
10	C033	28	97397.94	715	3478.50	0.04	49720.63

- **Revenue vs Discount**

```
SELECT
   Discount,
   COUNT(*) AS num_transactions,
   ROUND(SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)),2) AS total_revenue,
   ROUND(AVG(t.Quantity * p.UnitPrice * (1 - t.Discount)),2) AS avg_revenue_per_transaction
   ROUND(SUM((p.UnitPrice - p.CostPrice) * t.Quantity * (1 - t.Discount)),2) AS total_profit
FROM Transactions t
JOIN Products p ON t.ProductID = p.ProductID
GROUP BY Discount
ORDER BY total_revenue DESC;
```

	discount numeric (3,2)	num_transactions bigint	total_revenue numeric	avg_revenue_per_transaction numeric	total_profit numeric
1	0.00	1243	3846472.43	3094.51	1237899.91
2	0.05	1250	3651418.60	2921.13	1177542.63
3	0.15	1324	3443835.69	2601.08	1118199.87
4	0.10	1183	3360176.42	2840.39	1085971.96

- **Profit Margin by Product (TOP 10)**

```
WITH total_revenue AS(
        SELECT
                t.productid,
                p.productname,
                SUM(t.Quantity) AS total_units_sold,
                SUM(t.Quantity * p.UnitPrice * (1 - t.Discount)) AS total_sales,
                SUM(t.Quantity * p.CostPrice) AS total_cost
        FROM Transactions t
```

```
        JOIN Products p ON t.ProductID = p.ProductID
        GROUP BY 1,2
)
SELECT
        r.productid,
        r.productname,
        r.total_units_sold,
        r.total_sales,
        r.total_cost,
        ROUND(((r.total_sales - r.total_cost)/ r.total_sales)*100,2) AS Profit_margin
FROM total_revenue r
ORDER BY profit_margin DESC
LIMIT 10;
```

	productid character varying (10)	productname character varying (100)	total_units_sold bigint	total_sales numeric	total_cost numeric	profit_margin numeric
1	P003	Here Footwear	277	86180.0575	46821.31	45.67
2	P049	Possible Watches	317	114206.6790	63178.10	44.68
3	P007	Understand Camera	245	333519.6500	187184.90	43.88
4	P030	National Watches	306	8327.5520	4675.68	43.85
5	P043	Soon Accessories	256	132187.1640	75253.76	43.07
6	P045	Set Dairy	330	573777.0780	331504.80	42.22
7	P023	And Footwear	336	585613.0225	341009.76	41.77
8	P028	Deal Smartphone	303	188321.5100	110088.99	41.54
9	P032	Difficult Vegetables	302	83932.3695	49531.02	40.99
10	P024	Him Smartphone	318	431274.5295	258556.26	40.05

Summary of insights.

1. Top Products.

- Book Television(P037), Footwear (P023), and Set Dairy (P045) generate the highest total sales, making them the most critical products for business revenue, accounting for 2.25% of all units sold
- Products like Beat Accessories (P033), Piece Headphones (P040), and Like Camera (P001) form the mid-tier and around 2% of units.
- Interestingly, Road Clothing (P029) sold the highest number of units (346) among the group, though its total revenue (479K) is slightly lower, showing it's a high-volume but lower-priced product.
- By categories, Electronics generate the highest total sales, but Fashion the highest total profit.
- Each top product contributes only about 2% of units sold, showing a diverse sales spread across many products rather than dominance by one or two items.

2. Sales Trends.

- Despite monthly fluctuations, the company has demonstrated solid growth when comparing the same months across different years.
- The data shows a very predictable repeating pattern: sales consistently peak in December (Q4), making it the strongest sales month of the year. October and November also show strong numbers, building up to the December peak. Following the December peak, there is a significant drop in January and February. This is a common "holiday hangover" effect where consumer spending slows down after the festive period.
- Sales generally begin to recover from March onwards (Q2 & Q3)
- This pattern (Low Q1 -> Recovery -> High Q4) is clearly visible in both 2023-2024 and 2024-2025 cycles.
- The business shows a strong and healthy growth pattern with clear seasonality. The data for September 2025 is a critical outlier that requires immediate validation due to an unexplainable drop.

3. Top Stores and Cities by Sales and Profitability.

- Store S003 ("New Michele") Located in the West region, is the undisputed leader, ranking #1 in both Total Sales and Total Profit.
- Store S005 ("Johnmouth"): Ranks #4 in Sales but #3 in Profit. This indicates that while its sales volume is lower than the top three, it operates with higher margins or better cost control, making it more profitable than Store S001.
- Store S001 ("Jimenezborough"): Ranks #3 in Sales but drops to #4 in Profit. This suggests that despite high sales volume, its profitability is comparatively lower, possibly due to higher costs or lower-margin products.
- The top three stores are each from a different region (West, North, South), indicating that high performance is not confined to a single geographic area.

4. Customer demographics influence sales.

- The data shows a clear dominance in sales from male customers. The pattern holds for profit, with male customers generating 56.8% of total profit.
- The customer base is split 56.5% Male and 43.5% Female (113 vs. 87 customers). This indicates that the higher sales from male customers are not just due to a larger customer base, but also to a higher Average Order Value (AOV) or more frequent purchases.
- Purchasing power increases with age, The 55+ age group is the most significant, contributing 31.9% of total sales, despite representing only about 32% of the customer base (64 out of 200 customers).
- The oldest group also contributes the highest profit ($1,459,984), making them the most valuable customer segment.
- There is a strong positive correlation between age and spending (except for the 35-44 group slightly underperforming the 25-34 group).

5. Top Customers Analysis

- Historical CLV represents the total revenue generated by the customer to date.
- Estimated Annual CLV helps prioritize marketing campaigns to maximize future revenue.
- Customer C042 have higher projected annual CLV, indicating recent high activity.
- Customers with high AOV but fewer orders (C042) are premium customers making large purchases occasionally.
- Customers with lower AOV but many orders (C186) demonstrate frequent purchasing of smaller amounts.
- Purchase Frequency per Day are around 0.04–0.05 orders per day, reflecting roughly one order every 20–25 days.
- Number of Orders varies between 26 and 36 among the top 10.
- High CLV can come from either high AOV, high frequency, or both.

6. Revenue vs Discount

- Higher discounts do not consistently increase transaction counts.
- Revenue decreases as discounts increase.
- Maximum revenue occurs at 0% discount (3,846,472), showing that high discounts reduce total income despite a small increase in transactions at 15%.
- Higher discounts reduce the revenue generated per sale.
- Profit also decreases with higher discounts, reflecting lower margins.
- Higher discounts do not necessarily increase total revenue or profit.
- No discount or low discount levels appear more effective for maximizing income and profitability.

7. Profit Margins

- The top products maintain healthy margins (40–46%), indicating effective pricing strategies across different categories.
- Understand Camera (P007), Set Dairy (P045), and Footwear (P023) generate the highest revenue, making them the most impactful products in total profit.
- Smaller products like National Watches (P030) have high margin but lower total sales, showing niche profitability.
- Some products (e.g., Here Footwear) have slightly higher margins, but fewer units sold compared to higher-volume items (Set Dairy, And Footwear), highlighting a balance between profit per unit and overall contribution to revenue.

Conclusion

The analysis reveals a company with strong overall growth and clear seasonal patterns. Performance is driven by top-tier stores in specific cities and a customer base predominantly consisting of older male shoppers. The most critical finding is the significant anomaly in September 2025, which requires immediate attention.

Key Strategic Recommendations

- Address the September 2025 Anomaly: Priority #1. Verify if the data for September 2025 is complete. This outlier must be explained before it impacts strategic planning.
- Leverage Top-Performing Stores: Use the leading stores (S003 in New Michele, S004 in Brianahaven) as benchmarks to identify and share best practices across the network to elevate overall performance.
- Diversify the Customer Base: Develop targeted strategies to better engage female customers and the "Under 25" age group. This is crucial for unlocking sustainable long-term growth and reducing reliance on a single demographic.
- Formalize Seasonal Planning: Institutionalize strategies around the predictable Q4 peak and Q1 dip to optimize inventory, marketing, and cash flow throughout the year.
- Focus marketing, promotions, and inventory on high-revenue products to maximize overall profit.
- Maintain attention on high-margin items for potential growth through pricing strategies or targeted campaigns.

Acknowledgment & Closing Note

Completing this end-to-end analysis has been an extremely valuable project, significantly enhancing my skills in SQL and data storytelling. Thank you for the opportunity to contribute these insights, which I am confident will guide our future strategy effectively.

Sincerely,

Mayerlin Diaz
Data Analyst