# RETAIL SALES ANALYSIS REPORT

SALES ANALYSIS


## OVER VIEW

This project focuses on analyzing retail sales and invoice transaction data using SQL to extract meaningful business insights. The primary objective is to understand sales performance, product demand, customer purchasing behavior, return patterns, and the impact of discounts and payment methods on overall revenue. By converting raw transactional data into a well-structured, normalized relational database, this project enables accurate and efficient analysis of large volumes of sales data.

The insights derived from this analysis help the business identify high-performing and underperforming products, monitor revenue trends over time, assess financial losses due to product returns, and evaluate the effectiveness of pricing and promotional strategies. This data-driven approach supports better decision-making in areas such as inventory planning, sales strategy, revenue optimization, and customer experience management.

## *Retail Data Insights Using Power Query + SQL*

**Prepared By: Renin Rappai**

# PROBLEM STATEMENT

Retail businesses generate large volumes of sales data every day, but the raw dataset is often inconsistent, unstructured, and difficult to analyze. Without proper cleaning and modeling, organizations struggle to understand product performance, customer buying behavior, return patterns, discount effectiveness, and payment method trends.

This project solves these challenges by cleaning, normalizing, and analyzing retail sales data to extract insights that support operational and strategic decision-making.

## SALES PERFOMANCE ANALYSIS

# I. Current Situation

- The business currently stores its retail transaction data in a raw, unstructured format that makes detailed analysis challenging.
- There is limited clarity on important performance areas such as overall sales growth, product demand patterns, return behavior, and customer purchasing trends.
- Valuable insights related to high-performing products, revenue contribution, and seasonal sales fluctuations remain hidden within scattered transactional records.
- Many sales and inventory-related decisions are made based on intuition rather than on reliable, data-supported evidence.

# II. Solution:

- Prepare and normalize the retail sales dataset into well-structured relational tables including **invoices, products, and invoice line items**.
- Apply advanced SQL methods such as **joins, subqueries, common table expressions (CTEs), and aggregations** to uncover meaningful business insights.
- Address key business questions such as:
    - Which products contribute the most to overall revenue?
    - Which items experience the highest number of returns?
    - During which months does sales activity peak?
    - How do discounts influence customer purchases?
    - Which modes of payment generate the highest revenue?

- Enable business leaders to make informed decisions related to pricing strategy, inventory management, promotional planning, and operational efficiency by converting raw sales transactions into actionable intelligence.

## Data Description

**Title:** Retail Sales and Invoice Transaction Dataset

- **Sample Size:** ~1300–1400 Rows, 9 Columns
- **Data Types:** Mixed — Numeric, Categorical, Text, Date

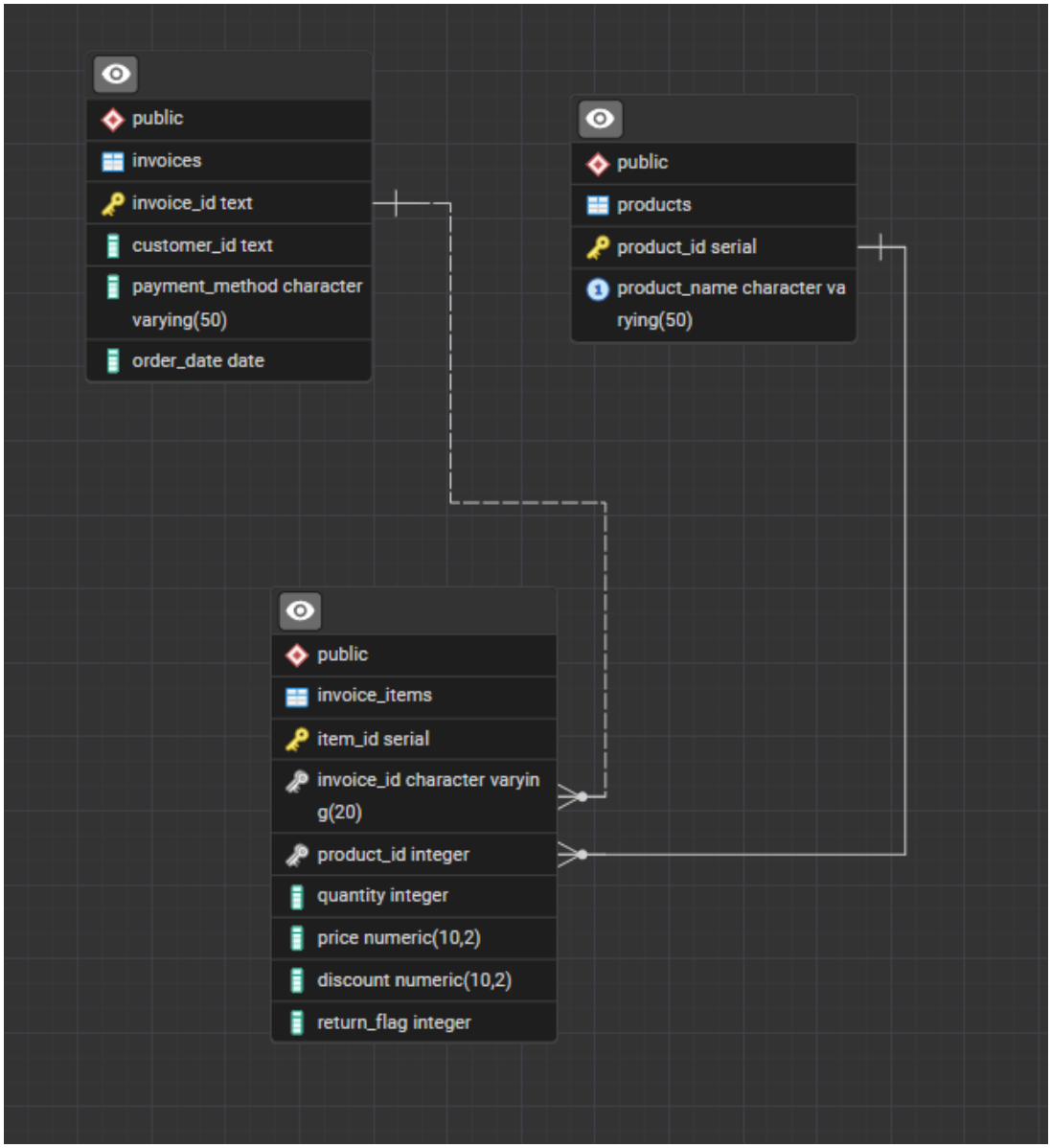
**Key Fields:**

- **Invoice_id, Customer_id, Product**
- **Quantity, Price, Discount**
- **Payment_Method, Order_Date**
- **Return_Flag**

---

## Data Cleaning Steps

- Identified and handled missing or null values in key transactional columns such as **quantity, price, discount, and return_flag**.
- Standardized inconsistent text formats (e.g., "cash" → "Cash", "upi" → "UPI", "credit card" → "Credit Card").
- Corrected improper data types (e.g., **quantity as INT, price as NUMERIC, order_date as DATE**).
- Removed duplicate transaction records to ensure invoice and product accuracy.
- Fixed mixed upper and lower case text values by applying consistent capitalization across product names and payment methods.
- Normalized the dataset into multiple related SQL tables:
    - **invoices** (invoice header information)
    - **products** (unique product master data)
    - **invoice_items** (detailed transaction-level data)




- The data model follows a **one-to-many relationship structure**, where each master table (**invoices** and **products**) is linked to multiple rows in the **invoice_items** table. The foreign keys in the invoice_items table reference the primary keys from the invoices and products tables. This design simplifies complex analytical queries, improves query performance, and ensures strong data integrity by preventing orphan or inconsistent transaction records.

# Category 1: Revenue & Sales Performance

---

1. **What is the total revenue generated from all sales?**

```sql
SELECT
    SUM(quantity * price) AS total_revenue
FROM invoice_items;
```

| | total_revenue<br>numeric |
|---|---|
| 1 | 8486500.00 |

## Insight

- Shows overall business performance and total earnings.
- Helps evaluate yearly/quarter financial success.
- Useful for annual reporting & budgeting decisions.

**2. Which products generate the highest revenue?**

```sql
SELECT p.product_name,
       SUM(i.quantity * i.price) AS revenue
FROM invoice_items i
JOIN products p ON i.product_id = p.product_id
GROUP BY p.product_name
ORDER BY revenue DESC
LIMIT 10;
```

| | product_name character varying (50) | revenue numeric |
|---|---|---|
| 1 | Charger | 995000.00 |
| 2 | Laptop | 942500.00 |
| 3 | Webcam | 915500.00 |
| 4 | Mouse | 866500.00 |
| 5 | Keyboard | 855500.00 |
| 6 | Speaker | 831000.00 |
| 7 | Headphones | 814500.00 |
| 8 | Usb Cable | 781000.00 |
| 9 | Smartphone | 744000.00 |
| 10 | Monitor | 741000.00 |

## Insight

- Identifies best-performing products.
- Helps prioritize stock, marketing and pricing.

- Ideal for planning promotions for top-selling items.

### 3.**Which products show above-average revenue performance?**

```sql
WITH product_revenue AS (
    SELECT
        p.product_name,
        SUM(i.quantity * i.price) AS revenue
    FROM invoice_items i
    JOIN products p ON i.product_id = p.product_id
    GROUP BY p.product_name
),
avg_revenue AS (
    SELECT AVG(revenue) AS avg_rev
    FROM product_revenue
)
SELECT pr.product_name,
       pr.revenue,
       round((SELECT avg_rev FROM avg_revenue),2)AS average_revenue
FROM product_revenue pr
WHERE pr.revenue > (SELECT avg_rev FROM avg_revenue)
ORDER BY pr.revenue DESC;
```

| | product_name character varying (50) | revenue numeric | average_revenue numeric |
|---|---|---|---|
| 1 | Charger | 995000.00 | 848650.00 |
| 2 | Laptop | 942500.00 | 848650.00 |
| 3 | Webcam | 915500.00 | 848650.00 |
| 4 | Mouse | 866500.00 | 848650.00 |
| 5 | Keyboard | 855500.00 | 848650.00 |

## Insight

- Identifies products performing **above the overall average revenue**.
- Useful for deciding which products deserve **more marketing and stock allocation**.
- Helps discover hidden high-revenue items that aren't top sellers but generate strong profits.

## Category 2: Product Quality & Return Behavior

---

### 4. Which products have the highest return count?

```sql
SELECT p.product_name,
       COUNT(*) AS return_count
FROM invoice_items i
JOIN products p ON i.product_id = p.product_id
WHERE return_flag = 1
GROUP BY p.product_name
ORDER BY return_count DESC;
```

| | product_name character varying (50) | return_count bigint |
|---|---|---|
| 1 | Charger | 80 |
| 2 | Laptop | 79 |
| 3 | Headphones | 76 |
| 4 | Speaker | 76 |
| 5 | Mouse | 70 |
| 6 | Webcam | 69 |
| 7 | Keyboard | 68 |
| 8 | Monitor | 65 |
| 9 | Usb Cable | 59 |
| 10 | Smartphone | 56 |

## Insight

- High return counts may signal quality or delivery problems.
- Helps reduce losses by improving product or vendor quality.
- Indicates where warranty/return policies need revision.

### 5. Which customers are most likely to return products?

```sql
SELECT
    customer_id,
    COUNT(*) AS returned_items
FROM invoices i
JOIN invoice_items it ON i.invoice_id = it.invoice_id
WHERE it.return_flag = 1
GROUP BY customer_id
ORDER BY returned_items DESC
LIMIT 10;
```

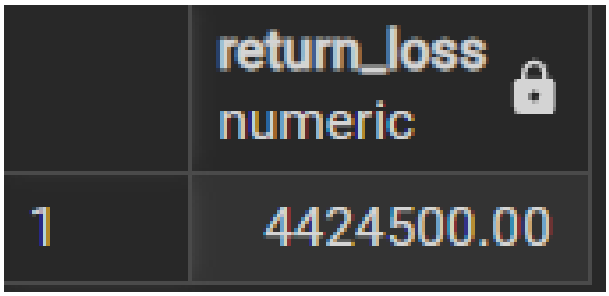
| | customer_id text | returned_items bigint |
|---|---|---|
| 1 | C-347 | 8 |
| 2 | C-488 | 8 |
| 3 | C-4 | 7 |
| 4 | C-233 | 7 |
| 5 | C-72 | 7 |
| 6 | C-182 | 7 |
| 7 | C-183 | 6 |
| 8 | C-439 | 6 |
| 9 | C-203 | 6 |
| 10 | C-241 | 6 |

## Insight

- Helps Detect habitual returners.
- Understand customer issues or fraudulent behavior.
- Useful for return-policy improvement.

### 6. How much total revenue was lost due to product returns?

```sql
SELECT SUM(quantity * price) AS return_loss
FROM invoice_items
WHERE return_flag = 1;
```



| | return_loss<br>numeric |
|---|---|
| 1 | 4424500.00 |

## Insight

- Measures financial impact of returned goods.
- Helps evaluate refund/return cost burden.
- High value = immediate need to improve product quality/logistics.

# Category 3: Customer & Ordering Behavior

---

### 7. Monthly Sales Trend — Which months perform best?

```sql
SELECT DATE_TRUNC('month', order_date) AS month,
       SUM(quantity * price) AS monthly_sales
FROM invoices inv
JOIN invoice_items it ON inv.invoice_id = it.invoice_id
GROUP BY month
ORDER BY month;
```

| | month<br>timestamp with time zone | monthly_sales<br>numeric |
|---|---|---|
| 1 | 2023-01-01 00:00:00+05:30 | 264500.00 |
| 2 | 2023-02-01 00:00:00+05:30 | 165500.00 |
| 3 | 2023-03-01 00:00:00+05:30 | 290500.00 |
| 4 | 2023-04-01 00:00:00+05:30 | 315500.00 |
| 5 | 2023-05-01 00:00:00+05:30 | 395500.00 |
| 6 | 2023-06-01 00:00:00+05:30 | 527500.00 |
| 7 | 2023-07-01 00:00:00+05:30 | 216000.00 |
| 8 | 2023-08-01 00:00:00+05:30 | 213000.00 |
| 9 | 2023-09-01 00:00:00+05:30 | 327000.00 |
| 10 | 2023-10-01 00:00:00+05:30 | 318000.00 |
| 11 | 2023-11-01 00:00:00+05:30 | 412000.00 |
| 12 | 2023-12-01 00:00:00+05:30 | 428000.00 |

**Insight**

- **Shows demand pattern and seasonality.**
- **Helps plan inventory and promotional campaigns.**
- **Peak months indicate high-profit sales windows.**

**8. Payment Method Performance — Which payment mode brings more revenue?**

```sql
SELECT payment_method,
       SUM(quantity * price) AS revenue
FROM invoices i
JOIN invoice_items it ON i.invoice_id = it.invoice_id
GROUP BY payment_method
ORDER BY revenue DESC;
```

| | payment_method character varying (50) | revenue numeric |
|---|---|---|
| 1 | Card | 3134000.00 |
| 2 | Cash | 2394000.00 |
| 3 | Upi | 2029000.00 |
| 4 | Netbanking | 557500.00 |
| 5 | Debit Card | 372000.00 |

**Insight**

- **Shows customer payment preference.**
- **Helps negotiate partnerships with banks/payment apps.**
- **Useful for designing cashback/discount schemes.**

# Category 1: Revenue & Sales Performance

## Insights:

- Products with higher revenue contribution clearly indicate the company's primary profit drivers.
- A small group of top-selling items generates a large share of total revenue, showing revenue concentration.
- Several products perform well above the average revenue level, highlighting strong customer demand and brand preference.
- Inconsistent revenue distribution across products suggests opportunities for better pricing and promotion strategies.

## Recommendations:

- Focus marketing and promotional efforts on high-revenue products to maximize profit.
- Ensure consistent stock availability for top-selling items to avoid missed sales opportunities.
- Review pricing strategies for underperforming products to improve their revenue contribution.
- Use high-performing products as part of bundled offers to increase overall sales value.

## Category 2: Product Quality & Return Behavior

## Insights:

- Certain products show significantly higher return counts, which may indicate quality defects, mismatched customer expectations, or delivery-related issues.
- A small group of customers is responsible for a large portion of returned items, suggesting possible habitual return behavior.
- Revenue loss due to product returns has a noticeable financial impact on overall business profitability.
- High return trends highlight weaknesses in quality control and post-sale customer experience.

## Recommendations:

- Investigate frequently returned products to identify quality, packaging, or supplier issues.
- Strengthen return policy monitoring for customers with repeated return behavior.
- Improve product descriptions and images to reduce expectation gaps.
- Introduce stricter quality checks and better logistics handling to minimize return-related losses.

## Category 3: Customer & Ordering Behavior

## Insights:

- Monthly sales trends reveal clear seasonality, with certain months consistently outperforming others.
- Customers show strong preferences for specific payment methods, which directly influence revenue flow.
- Some products experience significantly higher sales only when discounts are applied, indicating price sensitivity.
- Average items per invoice suggest moderate basket sizes with potential for increasing cross-selling.

### Recommendations:

- Use peak-month sales data to plan inventory, staffing, and marketing activities in advance.
- Promote top-performing payment methods with exclusive offers and cashback incentives.
- Apply discounts strategically only to price-sensitive products to protect overall profit margins.
- Introduce product bundles and cross-sell strategies to increase average order value per customer.

# Conclusion

This retail sales analytics project, powered by structured SQL queries and a normalized relational data model, has helped uncover critical patterns across sales performance, product demand, return behavior, discount impact, and customer purchasing trends. These insights provide a data-driven foundation for business leaders to design targeted strategies that improve profitability, optimize inventory management, reduce return-related losses, and enhance overall customer experience.

By transforming raw transactional data into actionable business intelligence, this project demonstrates the real-world value of SQL and data analysis in driving smarter retail decision-making and supporting sustainable business growth.