

Retail Business Performance Analysis: A SQL Case Study

This presentation highlights key insights from a comprehensive SQL-based analysis of a retail company's sales, customer, and product data. The objective was to address challenges in product performance, customer segmentation, and behavior understanding.

1. Database Setup & Data Loading

The first step involved setting up the relational database schema and importing the cleaned datasets.

SQL Commands for Database & Table Creation

```
-- Create database if it doesn't exist
CREATE DATABASE IF NOT EXISTS sql_case_study;
USE sql_case_study;
```

```
-- Create customer_profiles table
CREATE TABLE customer_profiles (
  `CustomerID` DECIMAL(38, 0) NOT NULL,
  `Age` DECIMAL(38, 0) NOT NULL,
  `Gender` VARCHAR(6) NOT NULL,
  `Location` VARCHAR(7) NOT NULL,
  `JoinDate` DATE NOT NULL
);
```

```
-- Create product_inventory table
CREATE TABLE product_inventory (
  `ProductID` DECIMAL(38, 0) NOT NULL,
  `ProductName` VARCHAR(11) NOT NULL,
  `Category` VARCHAR(15) NOT NULL,
  `StockLevel` DECIMAL(38, 0) NOT NULL,
  `Price` DECIMAL(38, 2) NOT NULL
);
```

```
-- Create sales_transaction table
CREATE TABLE sales_transaction (
  `TransactionID` DECIMAL(38, 0) NOT NULL,
  `CustomerID` DECIMAL(38, 0) NOT NULL,
  `ProductID` DECIMAL(38, 0) NOT NULL,
```

```
`QuantityPurchased` DECIMAL(38, 0) NOT NULL,  
`TransactionDate` DATE NOT NULL,  
`Price` DECIMAL(38, 2) NOT NULL  
);
```

Data Loading Strategy (Example for MySQL)

Data was loaded into these tables from corresponding CSV files using the LOAD DATA LOCAL INFILE command.

```
-- Example for loading customer_profiles.csv data  
LOAD DATA LOCAL INFILE  
'/Users/kundankumar/mysql-files/cleandata/customer_profiles.csv'  
INTO TABLE customer_profiles  
FIELDS TERMINATED BY ','  
ENCLOSED BY ''''  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS; -- Ignores the header row
```

2. Product Performance Analysis

Understanding which products are performing well or poorly is crucial for inventory and marketing strategies.

Top 10 Products by Total Sales (Revenue)

SQL Query:

```
SELECT  
    p.ProductName,  
    SUM(st.QuantityPurchased * st.Price) AS TotalSales  
FROM  
    sales_transaction st  
JOIN  
    product_inventory p ON st.ProductID = p.ProductID  
GROUP BY  
    p.ProductName  
ORDER BY
```

```
TotalSales DESC
LIMIT 10;
```

Results:

```
'Product_17','9450.00'
'Product_87','7817.24'
'Product_179','7388.26'
'Product_96','7132.32'
'Product_54','7052.86'
'Product_187','6915.88'
'Product_156','6827.84'
'Product_57','6622.20'
'Product_200','6479.79'
'Product_127','6415.80'
```

Insight: Product_17 stands out as the highest revenue generator, followed by Product_87 and Product_179. These products are critical for the company's financial performance.

Bottom 10 Products by Total Sales (Revenue)

SQL Query:

```
SELECT
    p.ProductName,
    SUM(st.QuantityPurchased * st.Price) AS TotalSales
FROM
    sales_transaction st
JOIN
    product_inventory p ON st.ProductID = p.ProductID
GROUP BY
    p.ProductName
ORDER BY
    TotalSales ASC
LIMIT 10;
```

Results:

```
'Product_17','9450.00'  
'Product_87','7817.24'  
'Product_179','7388.26'  
'Product_96','7132.32'  
'Product_54','7052.86'  
'Product_187','6915.88'  
'Product_156','6827.84'  
'Product_57','6622.20'  
'Product_200','6479.79'  
'Product_127','6415.80'
```

Note: The provided results for the "Bottom 10 Products" appear to be identical to the "Top 10 Products". This suggests a possible discrepancy in the original result logging. In a real-world scenario, the bottom 10 products would show the lowest revenue figures.

Top 10 Products by Quantity Sold**SQL Query:**

```
SELECT  
    p.ProductName,  
    SUM(st.QuantityPurchased) AS TotalQ  
FROM  
    sales_transaction st  
JOIN  
    product_inventory p ON st.ProductID = p.ProductID  
GROUP BY  
    p.ProductName  
ORDER BY  
    TotalQ DESC  
LIMIT 10;
```

Results:

'Product_182','102' heighest quantities
'Product_17','100'
'Product_87','92'
'Product_71','88'
'Product_195','87'
'Product_54','86'
'Product_179','86'
'Product_177','86'
'Product_22','85'
'Product_13','85'

Insight: Product_182 is the most frequently purchased item by quantity. Product_17 and Product_87 also rank high in both quantity and revenue, indicating strong demand.

3. Sales Trends Analysis

Understanding sales trends over time helps in forecasting and strategic planning.

Monthly Sales Volume

SQL Query:

```
SELECT
    DATE_FORMAT(TransactionDate, '%c-%Y') AS SalesMonth,
    SUM(QuantityPurchased * Price) AS MonthlySales
FROM
    sales_transaction
GROUP BY
    SalesMonth
ORDER BY
    SalesMonth;
```

Results :

'Jan-2023','86668.84'

'Feb-2023','76453.48'
'Mar-2023','88551.05'
'Apr-2023','83071.58'
'May-2023','87054.61'
'Jun-2023','85680.91'
'Jul-2023','74022.27'
'Aug-2023','24520.26'
'Sep-2023','24024.06'
'Oct-2023','23189.72'
'Nov-2023','23894.37'
'Dec-2023','24872.47'

Insight: Sales were strong in the first half of 2023 (Jan-Jul), peaking in March. A significant drop in revenue is observed from August onwards, indicating a potential seasonal downturn or other contributing factors needing further investigation.

4. Customer Segmentation

Segmenting customers enables targeted marketing and improved customer satisfaction.

Customer Segmentation by Total Number of Orders

SQL Query:

```
SELECT
    cp.CustomerID,
    COUNT(st.TransactionID) AS TotalOrders,
    CASE
        WHEN COUNT(st.TransactionID) = 0 THEN 'No orders'
        WHEN COUNT(st.TransactionID) BETWEEN 1 AND 10 THEN 'Low'
        WHEN COUNT(st.TransactionID) BETWEEN 11 AND 30 THEN 'Mid'
        WHEN COUNT(st.TransactionID) > 30 THEN 'High Value'
        ELSE 'Unknown'
    END AS CustomerSegment_ByOrders
FROM
    customer_profiles cp
LEFT JOIN
    sales_transaction st ON cp.CustomerID = st.CustomerID
```

```
GROUP BY
  cp.CustomerID
ORDER BY
  TotalOrders DESC;
```

Results Excerpt:

```
'664','14','Mid'
'39','12','Mid'
'99','12','Mid'
'113','12','Mid'
'670','12','Mid'
...
'12','10','Low'
'84','10','Low'
'161','10','Low'
...
'52','0','No orders'
'71','0','No orders'
...
```

Insight:

- **Mid-Value Customers:** A segment of customers (e.g., CustomerID 664, 39, 99) frequently place orders (11-14 orders). These customers are valuable and could be targeted for loyalty programs or exclusive offers.
- **Low-Frequency Customers:** The majority of customers fall into the 'Low' segment (1-10 orders). Strategies to encourage repeat purchases could be beneficial for this group.
- **No Orders:** There are customers (e.g., CustomerID 52, 71) who have registered but not placed any orders. Re-engagement campaigns could be designed for them.

Customer Segmentation by Total Spending

SQL Query:

```
WITH CustomerSpending AS (
```

```

SELECT
    CustomerID,
    SUM(QuantityPurchased * Price) AS TotalSpending
FROM
    sales_transaction
GROUP BY
    CustomerID
)
SELECT
    cp.CustomerID,
    COALESCE(cs.TotalSpending, 0) AS TotalSpending,
    CASE
        WHEN COALESCE(cs.TotalSpending, 0) = 0 THEN 'No Spending'
        WHEN COALESCE(cs.TotalSpending, 0) < 500 THEN 'Low Spender'
        WHEN COALESCE(cs.TotalSpending, 0) BETWEEN 500 AND 2000 THEN 'Mid
Spender'
        WHEN COALESCE(cs.TotalSpending, 0) > 2000 THEN 'High Spender'
        ELSE 'Unknown'
    END AS CustomerSegment_BySpending
FROM
    customer_profiles cp
LEFT JOIN
    CustomerSpending cs ON cp.CustomerID = cs.CustomerID
ORDER BY
    TotalSpending DESC;

```

Results Excerpt:

```

'936','2834.47','High Spender'
'664','2519.04','High Spender'
'670','2432.15','High Spender'
'39','2221.29','High Spender'
'435','2158.98','High Spender'
'958','2104.71','High Spender'
...
'364','1885.40','Mid Spender'
...

```


'158','499.99','Low Spender'

...

'52','0.00','No Spending'

...

Insight:

- **High Spenders:** A small but significant group of customers (e.g., CustomerID 936, 664) contribute substantially to overall revenue. These "High Spenders" warrant premium service and exclusive offers.
- **Mid Spenders:** The largest segment, these customers provide consistent revenue. Marketing efforts should focus on increasing their average transaction value or purchase frequency.
- **Low Spenders/No Spending:** These segments represent opportunities for re-engagement or conversion strategies.

5. Conclusion & Recommendations

This analysis provides actionable insights into product performance and customer behavior, which can inform strategic decisions.

Key Takeaways:

- **Top Products Identified:** Focus marketing and inventory on high-performing products like Product_17 and Product_182.
- **Seasonal Sales Dip:** Investigate the reasons behind the significant sales decline in the latter half of the year to formulate mitigation strategies.
- **Customer Segmentation:** Leverage the identified customer segments for more effective and personalized marketing campaigns (e.g., loyalty programs for "Mid" and "High Spender" customers, re-engagement for "No orders/spending" customers).