End-to-End Sales Analysis and Customer Segmentation in MySQL

Author: Chintada Bargav Sree Naidu

August 24, 2025

Project Repository Link: GitHub

Abstract

This report documents a production-ready analytics pipeline built entirely within MySQL. The project transforms raw sales data into actionable insights by implementing a comprehensive workflow: data cleaning, exploratory analysis, and advanced RFM customer segmentation. Automation is achieved via Stored Procedures, and performance is optimized with strategic indexing. This scalable approach enables data-driven decision-making, supporting targeted retention, marketing, and resource allocation in a business environment.

Contents

1 Introduction **2**

1.1 Problem Context 2

1.2 Project Objectives 2

2 Methodology **2**

2.1 Dataset and Environment 2

2.2 Data Cleaning and Transformation 2

2.3 Modeling and Analysis 3

2.3.1 Exploratory Data Analysis (EDA) 3

2.3.2 RFM Segmentation Model 3

2.4 Automation and Optimization 3

3 Results and Discussion **4**

3.1 Key Findings 4

3.2 Discussion 4

4 Conclusion **5**

5 Appendix: Code Snippet **5**

1 Introduction

1.1 Problem Context

Quantity of data alone does not yield business value. In retail and sales, meaningful impact is only achieved by cleaning, modeling, and analyzing raw transactional information. The challenge addressed by this project is the conversion of unstructured sales data into strategic insights to guide retention, resource allocation, and marketing.

1.2 Project Objectives

Major goals include:

- Building an end-to-end workflow in MySQL, from data ingest to actionable results.
- Addressing data quality issues to enable reliable analysis.
- Conducting thorough EDA to surface key business drivers.
- Designing and iteratively improving an RFM customer segmentation model.
- Delivering production-level automation and query optimization using procedures and indexing.

2 Methodology

2.1 Dataset and Environment

The project uses a real-world sales dataset (`sales.csv`) with transaction records. All analysis and modeling were performed in **MySQL Workbench**.

2.2 Data Cleaning and Transformation

Key preparatory steps:

- **Schema Creation:** A new database (`sales_db`) and table (`sales`) designed to match CSV structure.
- **Data Loading:** Imported with the MySQL Workbench Table Data Import Wizard.
- **Date Correction:** The `Order Date` was integer-encoded as days offset from 1899-12-30 (Excel default). Created a `Formated_Order_Date` column using `DATE_ADD()`.
- **Data Integrity:** Applied `NOT NULL` constraints and standardized data types for key columns (`Order ID`, `Customer ID`, `Sales`).

2.3 Modeling and Analysis

2.3.1 Exploratory Data Analysis (EDA)

EDA was performed to determine business performance and patterns. Topics included:

- **Overall Metrics:** Total revenue, average sale, and unique customer counts.
- **Performance Analysis:** Best/worst customers, product trends, and sales by region.
- **Temporal Trends:** Yearly and monthly analysis to observe seasonal shifts.

2.3.2 RFM Segmentation Model

RFM (Recency, Frequency, Monetary) is the core segmentation strategy.

- **Raw Values:** Used CTEs to compute recency, frequency, and monetary values for every customer.
- **Scoring:** Applied `NTILE(5)` window function to score RFM metrics (1–5).
- **Segmentation:** Three models created using `CASE` statements. Segment definitions (Champion Customers, Loyal Customers, At Risk, etc.) were refined iteratively, and final logic was encapsulated in a SQL `VIEW`.

2.4 Automation and Optimization

- **Stored Procedures:**
  1. `GetCustomerHistory(IN customerId INT)`: One-command access to purchase history.
  2. `RunRFMSegmentation(IN analysisDate DATE)`: Automates the full RFM workflow for any cutoff date.
- **Indexing:** Strategic indexes added for `Customer ID`, `Order ID`, and `Formated_Order_Date`. Query plans examined with `EXPLAIN` before/after indexing to ensure efficiency.

3 Results and Discussion

3.1 Key Findings

The EDA identified the 'Central' region as top in sales and seasonal peaks in Q4. The final RFM segmentation separates the customer base into actionable groups, as shown in Table 1.

Table 1: Customer Distribution Across Key RFM Segments (Model 3)

Customer Segment	**Number of Customers**	**Avg. Monetary Value ($)**
Champion Customers	56	6,450
Loyal Customers	77	4,010
Potential Loyalists	65	2,550
At Risk	58	980
Lost Customers - Low Value	45	350

3.2 Discussion

The results highlight a classic business scenario: a small cohort of **Champion Customers** (56 individuals) are disproportionately valuable, with an average spend that far exceeds any other group. These customers are prime candidates for loyalty programs, exclusive offers, and early access to new products to maintain their high level of engagement.

Conversely, the **At Risk** segment represents a significant opportunity for proactive intervention. These customers have demonstrated value in the past but have not purchased recently. Targeted re-engagement campaigns, such as personalized discounts or "we miss you" emails, could prevent them from churning. Between these two extremes lie the **Loyal Customers** and **Potential Loyalists**, who form the pipeline for future champions and should be nurtured with consistent communication and upselling opportunities.

The implementation of Stored Procedures and Indexing transforms this analysis from a static report into a dynamic and robust business tool. The `RunRFMSegmentation` procedure allows the business to track customer migration between segments on a recurring basis (e.g., monthly or quarterly). The performance optimizations ensure that this analysis remains fast and efficient, even as the dataset grows over time, making the segmentation logic both scalable and immediately actionable.

4 Conclusion

This project demonstrates the full power of MySQL for not only routine queries but advanced analytics. Starting from raw sales data, it produces an efficient, repeatable pipeline for EDA, segmentation, and reporting. Iterative model improvements and robust database engineering together yield a solution that can continue delivering insights as the business and volume of data grows.

5 Appendix: Code Snippet

```sql
-- This CTE-based query forms the core of the RunRFMSegmentation procedure.
WITH CUSTOMER_AGGREGATED_DATA AS (
    SELECT
        ‘Customer ID‘, ‘Customer Name‘,
        DATEDIFF(analysisDate, MAX(Formated_Order_Date)) AS RECENCY_VALUE,
        COUNT(DISTINCT ‘Order ID‘) AS FREQUENCY_VALUE,
        ROUND(SUM(Sales)) AS MONETARY_VALUE
    FROM SALES
    WHERE Formated_Order_Date <= analysisDate
    GROUP BY ‘Customer ID‘, ‘Customer Name‘
),
RFM_SCORE AS (
    SELECT
        CAD.*,
        NTILE(5) OVER (ORDER BY RECENCY_VALUE DESC) AS R_SCORE,
        NTILE(5) OVER (ORDER BY FREQUENCY_VALUE ASC) AS F_SCORE,
        NTILE(5) OVER (ORDER BY MONETARY_VALUE ASC) AS M_SCORE
    FROM CUSTOMER_AGGREGATED_DATA AS CAD
)
SELECT
    RS.*,
    CONCAT_WS(’’, R_SCORE, F_SCORE, M_SCORE) AS RFM_SCORE_COMBINATION,
    CASE
        WHEN CONCAT_WS(’’, R_SCORE, F_SCORE, M_SCORE) IN (’555’, ’554’, ’553’) THEN ’Champion Customers’
        -- ... other segmentation cases
        WHEN CONCAT_WS(’’, R_SCORE, F_SCORE, M_SCORE) IN (’113’, ’112’, ’111’) THEN ’Lost Customers - Low Value’
        ELSE ’Other’
    END AS CUSTOMER_SEGMENT
FROM RFM_SCORE AS RS;
```

Listing 1: Core RFM Segmentation Logic within Stored Procedure