**Exploratory Data Analysis Project**

**Retail Sales**

**Project Title**  
Customer Segmentation (RFM) and Sales Performance Analysis for Strategic Growth

**Objective**

This project explores the relationship between customer behavior and sales trends using RFM segmentation integrated with sales performance analysis using MySQL. The goal is to uncover actionable insights that can drive smarter marketing strategies, optimize customer targeting, and ultimately enhance overall business performance.

**Business Context**  
Summarize the fictional or real-world scenario.  
*Who is this for? Why does it matter? What decisions depend on this?*

This project simulates a mid-sized retail company navigating a competitive and fast-evolving consumer market. With access to a growing pool of customer transactions, the organization wants to uncover deeper insights into purchasing behavior that can guide smarter business decisions.

The intended audience for this analysis includes marketing strategists, sales managers, and business analysts who are focused on optimizing customer engagement, improving campaign effectiveness, and identifying high-value customer segments.

Ultimately, decisions about promotional strategies, product placements, and retention efforts all depend on understanding who the customers are, how they shop, and what drives revenue.

**Dataset Description**

This project uses a fictional dataset representing a snapshot of a mid-sized retail operation, reflecting core elements of customer transactions and purchasing behavior.

* **Data source**: Kaggle
* **Dataset size**: 1,000 rows × 9 columns

**Key fields**:

* Transaction ID – unique identifier for each purchase
* Date – transaction date
* Customer ID – unique identifier for each customer
* Gender – customer gender
* Age – customer age
* Product Category – category of purchased product
* Quantity – number of items purchased
* Price per Unit – cost of a single item
* Total Amount – total sales

**Data Cleaning & Preparation**

Before analysis, the dataset was cleaned and preprocessed to ensure accuracy, consistency, and usability. Key preparation steps included:

* **Column renaming**: Replaced spaces with underscores and lower case all columns (e.g., price\_per\_unit → price\_per\_unit) for improved SQL readability.
* **Check for duplicates**: No duplicate found
* **Handling missing values**: Verified each column for nulls; none were found in critical fields like Customer ID, Transaction ID, or Total Amount.
* **Date formatting**: Date field: Changed data type from **text** to **date** for time-based analysis and RFM calculations.
* **Consistency checks**: Ensured uniform text casing in categorical variables (e.g., Gender, Product Category) to avoid grouping issues during segmentation.

Note: The dataset includes January 2024, but I focused only on the full year of 2023 to ensure complete and consistent analysis.

**Exploratory Data Analysis (EDA)**

1. **Sales performance and Customer Demographics Analysis**

SQL Queries

* 1. **KPIs**

**Purpose**: These KPIs provides instant clarity for decision-makers by highlighting key performance metrics by a glance, they can immediately understand how business is performing.

* + 1. **Total Sales**

SELECT SUM(total\_amount) AS total\_sales

FROM retail\_sales\_staging

WHERE YEAR(date) = 2023;



**Insight**: Shows the total amount earned from confirmed transactions.

* + 1. **Total Quantity Sold**

SELECT SUM (quantity) AS total\_quantity\_sold

FROM retail\_sales\_staging

WHERE Year(date) = 2023;



**Insight:** Reflect the total units sold over a year

* + 1. **Total Transaction**

SELECT COUNT (DISTINCT transaction\_id) AS total\_transaction

FROM retail\_sales\_staging

WHERE Year(date) = 2023;



**Observation:** I noticed that there is only one transaction per customer.

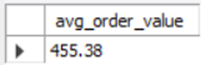
**Insight:** Shows the total unique transactions made by each customer.

* + 1. **Average Order Value (AOV)**

SELECT ROUND(SUM(total\_amount)/(COUNT(DISTINCT transaction\_id)),2) AS avg\_order\_value

FROM retail\_sales\_staging

WHERE Year(date) = 2023;



**Insight:** Shows amount of money a customer spends per transaction

* 1. **Sales Performance Analysis**
     1. **Monthly sales trends**

SELECT MONTH (date) AS month\_num, date\_format(date, '%M') AS month,

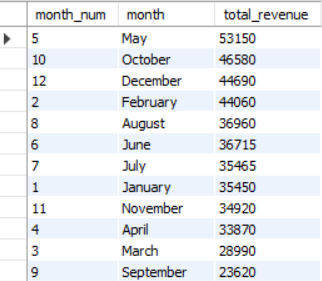
SUM (total\_amount) AS total\_revenue

FROM retail\_sales\_staging

WHERE Year(date) = 2023

GROUP BY month, month\_num

ORDER BY total\_revenue DESC;



* + 1. **Transaction frequency over months**

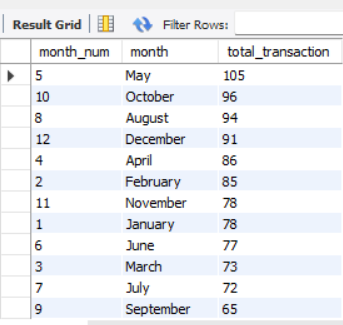
SELECT MONTH(date) AS month\_num, DATE\_FORMAT(date,'%M') AS month, COUNT(\*) AS total\_transaction

FROM retail\_sales\_staging

WHERE Year(date) = 2023

GROUP BY month, month\_num

ORDER BY total\_transaction DESC;



**Insights:**

**Monthly sales trends**

* Month of Mayshows highest sales which means it contributes most to the sales, more like due to seasonal demand or events.
* September has the lowest sales, potentially can improve or opportunity for growth.

**Recommended Action:** Introduce targeted promotionfor September and set measurable goals.

**Transaction frequency over months**

* May has the highest number of transactions, showing high customers activity.
* September has the lowest transactions; low customer activity means low sales.

**Recommended Action**: Re-engage with customers during September through strategic campaigns.

**Seasonal Trends**

* Spring, month of May, and the holiday season, month of December, are key sales periods.
* These seasonal surges likely reflect consumer behaviors tied to events like back-to-school, summer breaks, or holiday gift-giving.

**Recommended Action:** Use these insights to plan inventory, staffing, and campaigns effectively.

* + 1. **Product Category Distribution by sales**

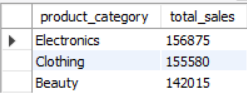
SELECT product\_category, SUM(total\_amount) AS total\_revenue

FROM retail\_sales\_staging

WHERE Year(date) = 2023

GROUP BY product\_category

ORDER BY total\_revenue DESC;



**Customer Demographics**

* + 1. **Gender Distribution by Product category**

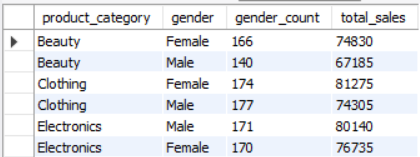
SELECT product\_category, gender, COUNT(\*) AS gender\_count, SUM(total\_amount) AS total\_sales

FROM retail\_sales\_staging

WHERE Year(date) = 2023

GROUP BY gender, product\_category

ORDER BY product\_category, total\_sales DESC;

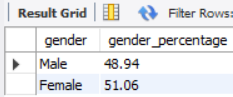


**% Sales by gender**

SELECT gender, ROUND(SUM(total\_amount) \* 100/ (SELECT SUM(total\_amount) FROM retail\_sales\_staging),2) AS gender\_percentage

FROM retail\_sales\_staging

GROUP BY gender;



Segmenting customers by age and name each group to enable targeted insights.

SELECT

CASE

WHEN age BETWEEN 0 AND 20 THEN 'Teens (0-20)'

WHEN age BETWEEN 21 AND 30 THEN 'Young (21-30)'

WHEN age BETWEEN 31 AND 45 THEN 'Adult (31-45)'

WHEN age BETWEEN 46 AND 59 THEN 'Middle Age (46-59)'

ELSE 'Old (60+)'

END AS age\_group

FROM retail\_sales\_staging;

Created view so I can use the above query for further analysis.

CREATE view age\_seg AS (

SELECT gender, age, total\_amount, transaction\_id, product\_category, quantity,

CASE

WHEN age BETWEEN 0 AND 20 THEN 'Teens (0-20)'

WHEN age BETWEEN 21 AND 30 THEN 'Young (21-30)'

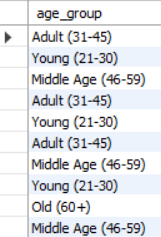
WHEN age BETWEEN 31 AND 45 THEN 'Adult (31-45)'

WHEN age BETWEEN 46 AND 59 THEN 'Middle Age (46-59)'

ELSE 'Old (60+)'

END AS age\_group

FROM retail\_sales\_staging);



* + 1. **Product categories purchased by age group**

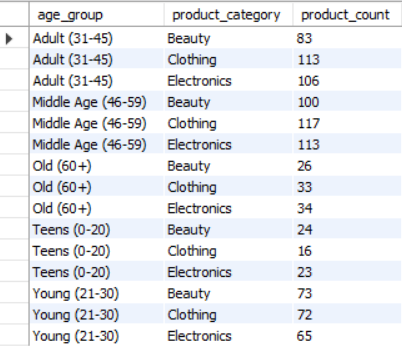
SELECT age\_group, product\_category, COUNT(\*) AS product\_count

FROM age\_seg

WHERE Year(date) = 2023

GROUP BY age\_group, product\_category

ORDER BY age\_group, product\_category, product\_count DESC;



**Insights:**

**Product Categories**

* Electronics contributes most to sales while beauty contributes least to sales.

**Recommended Action:** Ensure balanced inventory and tailored promotions across categories.

**Gender Trends**

* Female customers show a stronger preference for clothing, while clothing has balanced gender appeal.
* Male customers tend to favor electronics, indicating a focus on tech, gadgets, or utility-based products.

**Recommended Action:** Target beauty for females, electronics for males, and unisex campaigns for clothing.

**Product Categories by Age Group**:

* Electronics, clothing and beauty are most popular for middle-aged and adult customers. While young customers show interest to beauty products.

**Recommended Action:** Focus electronics ads on older groups, promote beauty to younger buyers, and maintain clothing inventory evenly.

**Sales and Customer Insight Summary**

Retail shows a good sales performance for the year 2023. After a strong peak in May, both sales transaction frequency declined in September before showing a clear recovery in October. Female customers among adult and middle-aged group have a strong preference for clothing. While male customers among adult and middle-aged group have a strong interest for electronics product. Beauty products are most popular to young female customers.

1. **Customer Segmentation**

**Brief description of Recency, Frequency and Monetary**

* **Recency** – Days since the last purchase (lower score is better)
* **Frequency** – Total number of purchases (higher score is better)
* **Monetary** – Total amount spent (higher score is better)

In order to categorize customers, let's compute first for recency, frequency and monetary

SELECT customer\_id,

datediff(‘2023-12-31’, MAX (date)) AS recency\_days,

COUNT (transaction\_id) AS frequency,

SUM (total\_amount) AS monetary

FROM retail\_sales\_staging

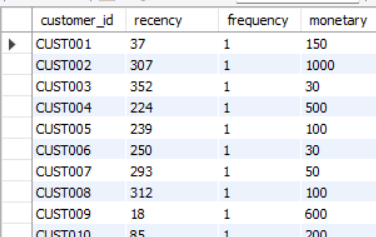
WHERE Year(date) = 2023

GROUP BY customer\_id

ORDER BY frequency desc;

This tells how many days it's been since each customer’s last purchase *as of the end of 2023*. In that way, my analysis stays consistent and accurate for that historical window.

When calculating **Recency** for RFM analysis, it’s all about determining how recently each customer made a purchase **relative to your dataset’s most recent date**, *not* today’s real-world date.



-Added RFM columns

ALTER TABLE retail\_sales\_staging

ADD COLUMN recency int,

ADD COLUMN frequency int,

ADD COLUMN frequency int;

-Inserted the data to each column

UPDATE retail\_sales\_staging AS r

JOIN (

SELECT customer\_id,

MAX(date) AS last\_purchase

FROM retail\_sales\_staging

WHERE Year(date) = 2023

GROUP BY customer\_id

) AS recent ON r.customer\_id = recent.customer\_id

SET r.recency = DATEDIFF(‘2023-12-31’, recent.last\_purchase);

UPDATE retail\_sales\_staging AS f

JOIN (

SELECT

customer\_id,

COUNT (transaction\_id) AS frequency

FROM retail\_sales\_staging

WHERE Year(date) = 2023

GROUP BY customer\_id

) AS freq ON f.customer\_id = freq.customer\_id

SET f.frequency = freq.frequency;

UPDATE retail\_sales\_staging AS m

JOIN (SELECT

customer\_id,

SUM (total\_amount) AS monetary

FROM retail\_sales\_staging

WHERE Year(date) = 2023

GROUP BY customer\_id

) AS mon ON m.customer\_id = mon.customer\_id

SET m.monetary = mon.monetary;

-Based on the computed RFM, I assigned RFM scores for each customer.

SELECT customer\_id,

-- The more recent the transaction (lower number of days), the higher the recency score.

CASE

WHEN recency <= 30 THEN 5

WHEN recency <= 100 THEN 4

WHEN recency <= 190 THEN 3

WHEN recency <= 380 THEN 2

ELSE 1

END AS r\_score,

-- The higher the value of frequency and monetary, the higher the score.

CASE

WHEN frequency >= 10 THEN 5

WHEN frequency >= 7 THEN 4

WHEN frequency >= 5 THEN 3

WHEN frequency >= 3 THEN 2

ELSE 1

END AS f\_score,

CASE

WHEN monetary >= 1800 THEN 5

WHEN monetary >= 1350 THEN 4

WHEN monetary >= 900 THEN 3

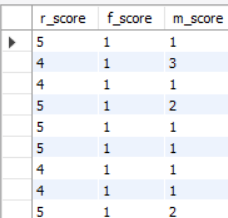
WHEN monetary >= 450 THEN 2

ELSE 1

END AS m\_score

FROM retail\_sales\_staging

WHERE Year(date) = 2023;



-Adding RFM score columns

ALTER TABLE retail\_sales\_staging

ADD column r\_score int,

ADD column f\_score int,

ADD column m\_score int;

-Inserting data to each column

UPDATE retail\_sales\_staging

SET r\_score = (SELECT

CASE

WHEN recency <= 30 THEN 5

WHEN recency <= 100 THEN 4

WHEN recency <= 190 THEN 3

WHEN recency <= 380 THEN 2

ELSE 1

END AS r\_score);

UPDATE retail\_sales\_staging

SET f\_score = (SELECT

CASE

WHEN frequency >= 10 THEN 5

WHEN frequency >= 7 THEN 4

WHEN frequency >= 5 THEN 3

WHEN frequency >= 3 THEN 2

ELSE 1

END AS f\_score);

UPDATE retail\_sales\_staging

SET m\_score = (SELECT

CASE

WHEN monetary >= 1800 THEN 5

WHEN monetary >= 1350 THEN 4

WHEN monetary >= 900 THEN 3

WHEN monetary >= 450 THEN 2

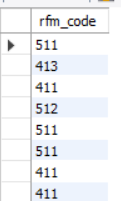
ELSE 1

END AS m\_score);

-Combining the scores to create a RFM code

SELECT CONCAT (r\_score, f\_score, m\_score) AS rfm\_code

FROM retail\_sales\_staging;



-Adding column and insert rfm codes

-- At first, I stored the data type as int, after searching, it should be a string (varchar or char) to use the rfm\_code for labeling the segments.

-- Make sure your rfm\_code is stored as a string (like this '532') so you can use LIKE patterns to classify customer types flexibly.

ALTER TABLE retail\_sales\_staging

ADD column rfm\_code int;

UPDATE retail\_sales\_staging

SET rfm\_code = CONCAT (r\_score, f\_score, m\_score);

**Grouping customers into descriptive segments based on recency, frequency and monetary definitions and scores:**

1. Champions: High Recency, Frequency, and Monetary
2. Loyal Customers: Frequent buyers with consistent value
3. At Risk: Haven’t purchased in a while
4. Big Spenders: High spenders
5. Low engagement: Low activity across the board, long-term nurturing required.

Then, assign scores from 1 (lowest) to 5 (highest) for each metric. Combine the three scores into a single 3-digit number (e.g., **555** is a top customer, **111** is the least engaged).

**RFM Segment Customer Label**

555 Champions

5xx Loyal Customers

x5x Big Spenders

2xx At Risk

Others Low Engagement

-- Grouping customers into descriptive segments

SELECT customer\_id,

CASE

WHEN rfm\_code = '555' THEN 'Champions'

WHEN rfm\_code LIKE '5\_\_' AND rfm\_code != '555' THEN 'Loyal Customers'

WHEN rfm\_code LIKE '\_\_5' THEN 'Big Spenders'

WHEN rfm\_code LIKE '2\_\_' THEN 'At Risk'

ELSE 'Low Engagement'

END AS segment\_label

FROM retail\_sales\_staging;

-- Adding customer segment into a new column

ALTER TABLE retail\_sales\_staging

ADD column customer\_segment VARCHAR (50);

UPDATE retail\_sales\_staging

SET customer\_segment = (SELECT

CASE

WHEN rfm\_code = '555' THEN 'Champions'

WHEN rfm\_code LIKE '5\_\_' AND rfm\_code != '555' THEN 'Loyal Customers'

WHEN rfm\_code LIKE '\_\_5' THEN 'Big Spenders'

WHEN rfm\_code LIKE '2\_\_' THEN 'At Risk'

ELSE 'Low Engagement'

END AS customer\_segment);

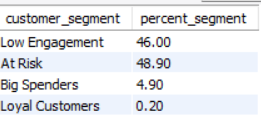


* + 1. **Customers Segment (as percent)**

SELECT customer\_segment, ROUND (COUNT (customer\_id) \* 100 / (SELECT COUNT (customer\_id) FROM retail\_sales\_staging),2) AS percent\_segment

FROM retail\_sales\_staging

GROUP BY customer\_segment;



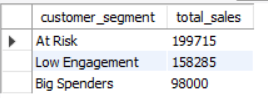
* + 1. **Total Sales by Segments**

SELECT DISTINCT (customer\_segment), SUM (total\_amount) AS total\_sales

FROM retail\_sales\_staging

GROUP BY customer\_segment

ORDER BY total\_sales DESC;



* + 1. **Average RFM Scores by Segment**

SELECT

customer\_segment,

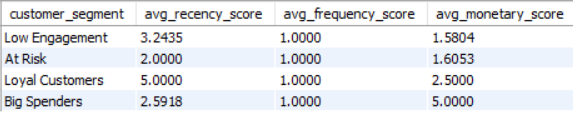
AVG(r\_score) AS avg\_recency\_score,

AVG(f\_score) AS avg\_frequency\_score,

AVG(m\_score) AS avg\_monetary\_score

FROM retail\_sales\_staging

GROUP BY customer\_segment;



**Insights from RFM Analysis**

1. **Customer Segments**:
   * The distribution indicates higher count of at-risk and low engagement, while lower count of loyal customers.
   * The at-risk segment contributes more to sales despite of showing signs of declining engagement.
   * Low RFM scores signal opportunities to improve through targeted reactivation efforts.

**Recommended Actions:**

* **Retention Focus**: Prioritize reactivation campaigns for at-risk customers, especially those high monetary value from the past purchases.
* **Value Growth**: Introduce loyalty programs, exclusive discounts, or upsell strategies for low-engagement customers. The goal is to increase their frequency and spend, moving them toward more valuable segments.
* **Customer Experience**: Identify potential reasons for inactivity through feedback mechanisms like short surveys or post-exit polls.

**RFM Analysis Summary**

Most customers fall into the at-risk and low engagement groups, while loyal customers are fewer. Even though at-risk customers aren’t purchase as much, they still make up a big part of the sales, so it's worth focusing on keeping them. To grow value, we can re-engage inactive customers with discounts, rewards, or surveys to understand what’s stopping them from buying again.

**Note**: Exported the data into excel for later visualization

**Kindly check my tableau project for this dataset. Thank you!**