VALENTINE'S DAY DATA ANALYSIS

BY: Denis Kombe Karisa

ABSTRACT

This analysis delves into Valentine's Day sales trends, uncovering key insights into revenue performance, product popularity and peek shopping periods. Using SQL, we explore transactional data to identify top-selling items, high-spending customers and location-based sales patterns. By transforming raw data into actionable insights, this analysis provides a data-driven foundation for optimizing sales strategies and maximizing profitability during seasonal shopping spikes.

Date

14th February, 2025

Table of Contents

INTRODUCTION

In this analysis, I explore the Valentine Purchase Dataset to extract key insights into customer behavior, sales performance, product popularity and demographic trends. Understanding these insights is crucial for businesses, marketers and analysts to optimize sales strategies, improve product offerings and also enhance customer experience especially during this valentine's seasons.

By leveraging SQL queries in MySQL Workbench, I aim to extract meaningful patterns from the data to help businesses understand:

- Popular meetup locations.
- Top-selling items.
- Revenue distributions.
- Customer search trends and conversion rates.

This documentation provides a step-by-step breakdown of the dataset, queries and insights and combinations drawn from the analysis.

DATASET OVERVIEW

Dataset Structure

The dataset, *valentine_purchases.csv* contains transactional records related to Valentine's Day Purchases. Below is the structure of the dataset:

Field	Type	Null	Key	Default	Extra
purchase_id	int	YES		NULL	
date	date	YES		NULL	
item	text	YES		NULL	
price_usd	double	YES		NULL	
gender	text	YES		NULL	
age	int	YES		NULL	
searchItems	text	YES		NULL	
salesmade	int	YES		NULL	
totalsales	double	YES		NULL	
meetup_points	text	YES		NULL	

KEY OBJECTIVES

1. Analyze Revenue and Sales Performance.
   - Calculate the **total revenue generated** from sales
   - Identify the **most expensive and cheapest items** purchased.
   - Analyze the **total number of items sold** and determine the **average sales per item.**

2. Evaluate Customer Demographics and Buying Behavior.
   - Analyze **purchases by gender** to identify spending patterns.
   - Examine **age-based purchasing behavior** to understand customer preferences.
   - Segment customers into **age groups** and evaluate their purchasing trends.

3. Identify Shopping Trends and Best-Selling Products.
   - Determine the **most active shopping days** leading up to Valentine's Day.
   - Analyze **daily sales trends before Valentine's Day** to uncover peak purchase periods.
   - Identify the **best-selling** and **least-sellin**g items contributing to the revenue earned.

4. Assess Search Behavior and Conversion Rates.
   - Identify the **most common search items** to understand customer interest.
   - Evaluate **search item conversion rates** to measure how often search lead to purchases.

5. Analyze Geographical Performance Meetup Points.
   - Most visited meetup points.

6. Evaluate Customer Spending and Values.
   - Identify **high-spending** customers and their purchasing behavior.
   - Determine the **highest spending age group** to understand purchasing power.
   - Estimate **customer lifetime value** to access long-term revenue potential.

VALENTINE’S PURCHASES DATA ANALYSIS

SQL Queries

This section provides step-by-step analysis with comments for clarity. Each step is linked to the corresponding SQL query.

Data Cleaning and Preprocessing:

Step 1: Checking the total number of records.

Before analyzing the data, we first find the total number of records in the dataset.

```
-- Checking the total number of records
SELECT COUNT(*) AS total_records
FROM valentine_purchases;
```

Step 2: Checking for Duplicate records.

Duplicate records can affect analysis accuracy.

```
 -- Checking for duplicate records
SELECT purchase_id, COUNT(*)
FROM valentine_purchases
GROUP BY purchase_id
HAVING COUNT(*) > 1;
```

Step 3: Checking for Missing Values.

Missing values can indicate incomplete data.

```
 -- Checking for missing values
SELECT
     COUNT(*) - COUNT(date) AS missing_dates,
     COUNT(*) - COUNT(item) AS missing_items,
     COUNT(*) - COUNT(price_usd) AS missing_prices,
     COUNT(*) - COUNT(gender) AS missing_genders,
     COUNT(*) - COUNT(age) AS missing_ages,
     COUNT(*) - COUNT(searchItems) AS missing_searchItems,
     COUNT(*) - COUNT(salesmade) AS missing_salesmade,
     COUNT(*) - COUNT(totalsales) AS missing_totalsales,
     COUNT(*) - COUNT(meetup_points) AS missing_meetup_points
FROM valentine_purchases;
```

Step 4: Checking the Date Formats.

Since incorrect date formats cause errors, it important to inspect the date column format.

```
-- Checking the date format
DESC valentine_purchases;
```

Step 5: Creating a New Date Column.

To maintain a structured dataset, create a correctly formatted date column.

```
 -- new date column
ALTER TABLE valentine_purchases
ADD COLUMN new_date DATE;
```

Step 6: Converting the Date Format (MM/DD/YYYY)

```
 -- Coverting the date format
UPDATE valentine_purchases
SET new_date = STR_TO_DATE(date, '%m/%d/%Y')
WHERE date IS NOT NULL;
```

Step 7: Dropping the old date column.

```
-- Dropping the old date column
ALTER TABLE valentine_purchases DROP COLUMN date;
```

Exploratory Data Analysis (EDA)

Sales Performance Analysis:

Step 8: Calculating the total revenue.

Sum up all sales to determine the total revenue generated from purchases.

```
-- Total revenue generated
SELECT SUM(totalsales) AS total_revenue_usd
FROM valentine_purchases;
```

Step 9: Identifying most expensive and cheapest items.

```
-- Most expensive and cheapest items
SELECT item, price_usd FROM valentine_purchases
ORDER BY price_usd ASC LIMIT 1;

SELECT item, price_usd FROM valentine_purchases
ORDER BY price_usd DESC LIMIT 1;
```

Step 10: Total items sold and average sales per item.

Total quantity sold and the average sales per product.

```
-- Total items sold and average sales per item
 SELECT
      SUM(salesmade) AS total_items_sold,
      ROUND(AVG(salesmade), 2) AS avg_sales_per_item
FROM valentine_purchases;
```

Customer Demographics Analysis:

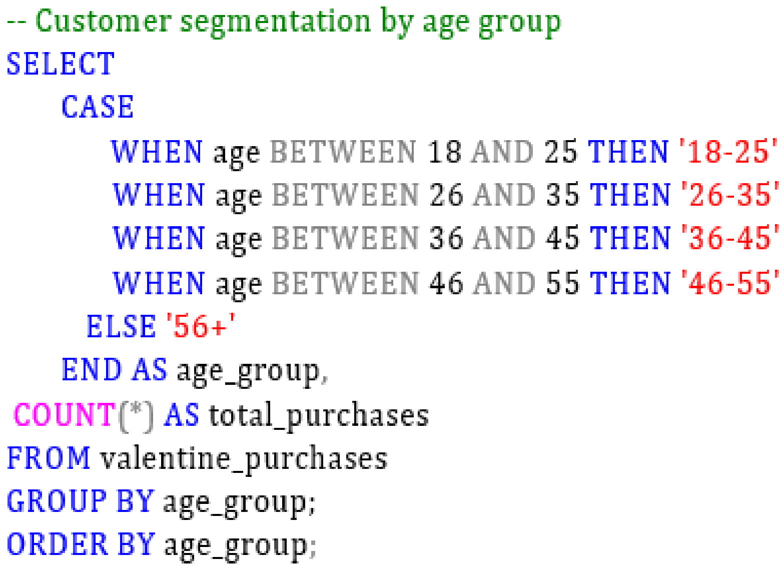
Step 11: Purchases by gender.

```
-- Purchases by gender
SELECT
      gender,
      COUNT(*) AS Total_purchases
FROM valentine_purchases
GROUP BY gender;
```

Step 12: Age-Based purchasing behavior.
  How different age groups contribute to sales.

```
-- Age-Based Purchasing Behavior
SELECT
      age,
      COUNT(*) AS purchases
FROM valentine_purchases
GROUP BY age
ORDER BY age;
```

Step 13: Customer segmentation by age group.



```
-- Customer segmentation by age group
SELECT
      CASE
            WHEN age BETWEEN 18 AND 25 THEN '18-25'
            WHEN age BETWEEN 26 AND 35 THEN '26-35'
            WHEN age BETWEEN 36 AND 45 THEN '36-45'
            WHEN age BETWEEN 46 AND 55 THEN '46-55'
         ELSE '56+'
      END AS age_group,
 COUNT(*) AS total_purchases
FROM valentine_purchases
GROUP BY age_group;
ORDER BY age_group;
```

Purchase Trends and Time-Based Analysis:

Step 14: Identifying the most active Shopping days.

```
-- Most active shopping days
SELECT
    DAYNAME(date) AS day_of_week,
    COUNT(*) AS purchases
 FROM valentine_purchases
GROUP BY  day_of_week
ORDER BY purchases DESC;
```

Step 15: Daily Sales Trends Before Valentine’s Day.

Helps in understanding of shopping behavior in the lead-up to February 14th.

```
-- Daily sales trends before valentine's day
 SELECT
        date,
        DAYNAME(date) as day_of_week,
        SUM(totalsales) AS total_sales
FROM valentine_purchases
WHERE date BETWEEN '2025-02-01' AND '2025-02-14'
GROUP BY date ORDER BY date;
```

Best-Selling Performance Analysis:

Step 16: Best-Selling and Least-Selling Items.

```
-- Best selling and least selling items
SELECT
     item,
     SUM(salesmade) AS total_sold
FROM valentine_purchases
GROUP BY item ORDER BY total_sold DESC;
```

Step 17: Top-Earning Items.

Items contributing to the most revenue.

```
-- Top earning items
SELECT
      item,
      SUM(totalsales) AS revenue
FROM valentine_purchases
GROUP BY item ORDER BY revenue DESC;
```

Customer Search Behavior and Conversation:

Step 18: Most Common Search Items.

Helps in understanding customer interests.

```
-- Most common search terms
SELECT
    searchItems,
    COUNT(*) AS search_frequency
FROM valentine_purchases
GROUP BY searchItems
ORDER BY search_frequency DESC;
```

Step 19: Search Item Conversion Rates.

Checking how often searched items resulted in actual purchases.

```
-- Search Term conversion rates
SELECT
    searchItems,
    SUM(salesmade) AS total_sold
FROM valentine_purchases
GROUP BY searchItems
ORDER BY total_sold DESC;
```

Location-Based Insights:

Step 20: Most and Least Profitable Meetup Points.

Which meetup points were visited most generating high revenue.

```
-- Most and least profitable meetup points
SELECT
    meetup_points,
    FORMAT(SUM(totalsales), 2) AS revenue
FROM valentine_purchases
GROUP BY meetup_points
ORDER BY revenue DESC;
```

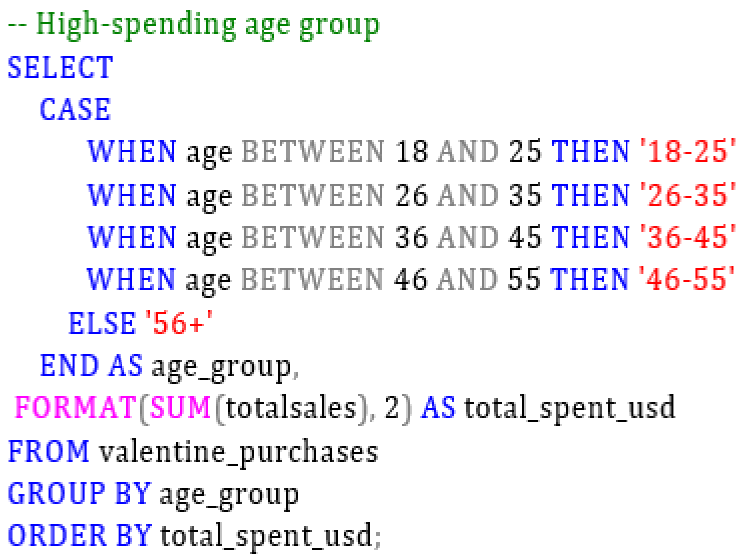
Customer Spending Behavior:

Step 21: High-Spending Customers.

Finding customers with the highest spending amounts.

```
-- High-spending customers
SELECT
      age,
      gender,
      FORMAT(SUM(totalsales), 2) AS total_spent_usd
FROM valentine_purchases
GROUP BY age, gender
ORDER BY total_spent_usd DESC;
```

Step 22: High-Spending Age Group.



```
-- High-spending age group
SELECT
   CASE
        WHEN age BETWEEN 18 AND 25 THEN '18-25'
        WHEN age BETWEEN 26 AND 35 THEN '26-35'
        WHEN age BETWEEN 36 AND 45 THEN '36-45'
        WHEN age BETWEEN 46 AND 55 THEN '46-55'
      ELSE '56+'
   END AS age_group,
 FORMAT(SUM(totalsales), 2) AS total_spent_usd
FROM valentine_purchases
GROUP BY age_group
ORDER BY total_spent_usd;
```

Step 23: Customer Lifetime Value Estimation.

To predict long-term revenue.

```
-- Customer lifetime value estimation
SELECT
       age,
        gender,
 COUNT(*) AS purchases,
 ROUND(SUM(totalsales), 2) AS total_spent
FROM valentine_purchases
GROUP BY age, gender
ORDER BY total_spent DESC;
```