Big Data - Case Study:

Coffee sales Analysis

1. Introduction

In today's data-driven world, understanding customer preferences and sales trends is essential for business growth. This project focuses on Coffee Sales Analysis using MySQL, Hive, and Sqoop.
The objective is to analyze coffee sales performance, identify top-performing products, and gain insights into consumer buying patterns.

MySQL serves as the foundation for structured data management and querying. Hive, a Hadoop-based tool, enables large-scale data analysis, while Sqoop bridges data transfer between MySQL and Hadoop for scalable computation.

Through this case study, we aim to demonstrate the power of integrated data technologies to perform efficient, insightful, and data-driven decision-making in the retail coffee business.

2. Description of the Dataset

Dataset Name: Coffee_Sales.csv

This dataset records sales transactions for different coffee products across multiple dates and customer types.

Attributes:

1. **transaction_id** – Unique identifier for each transaction.
2. **date** – Date of the coffee sale.
3. **time** – Time of the transaction.
4. **day_of_week** – Day name (e.g., Monday, Tuesday).
5. **customer_type** – Indicates whether the customer is a member or non-member.
6. **coffee_name** – Type of coffee sold (e.g., Latte, Espresso, Cappuccino).
7. **size** – Coffee size (Small, Medium, Large).
8. **quantity** – Number of cups sold.
9. **unit_price** – Price per cup.
10. **money** – Total amount of the sale (quantity × unit_price).
11. **payment_mode** – Payment method used (e.g., Cash, Card, UPI).
12. **branch** – Store location or branch where the sale took place.

This structured dataset enables a detailed exploration of customer behavior, sales performance by coffee type, and revenue analysis across time periods.

3. Project Scope

The scope of this project is to conduct a comprehensive analysis of coffee sales data to uncover business insights.
Using **MySQL**, **Hive**, and **Sqoop**, we will integrate and process the dataset to understand:

- Daily and weekly sales trends
- Top-selling coffee types and branches
- Customer purchase behavior (members vs. non-members)
- Revenue contribution by size and payment mode

By combining SQL and big data tools, the project aims to build a robust analytical framework to help coffee businesses improve marketing, stock management, and overall sales strategy.

4. Goals

1. **Sales Trend Analysis:** Examine daily, weekly, and monthly sales patterns.
2. **Top Product Identification:** Identify best-selling coffee products and their revenue contribution.
3. **Customer Insights:** Compare member vs. non-member buying behaviors.
4. **Branch Performance:** Evaluate revenue performance across different branches.
5. **Payment Method Analysis:** Understand customer preferences in payment modes.
6. **Integration:** Implement data transfer between MySQL and Hadoop using Sqoop.
7. **Scalable Analytics:** Use Hive for efficient querying on large-scale datasets.
8. **Visualization:** Create visual dashboards for better insight communication.

5. Tools and Working Environment

1. MySQL

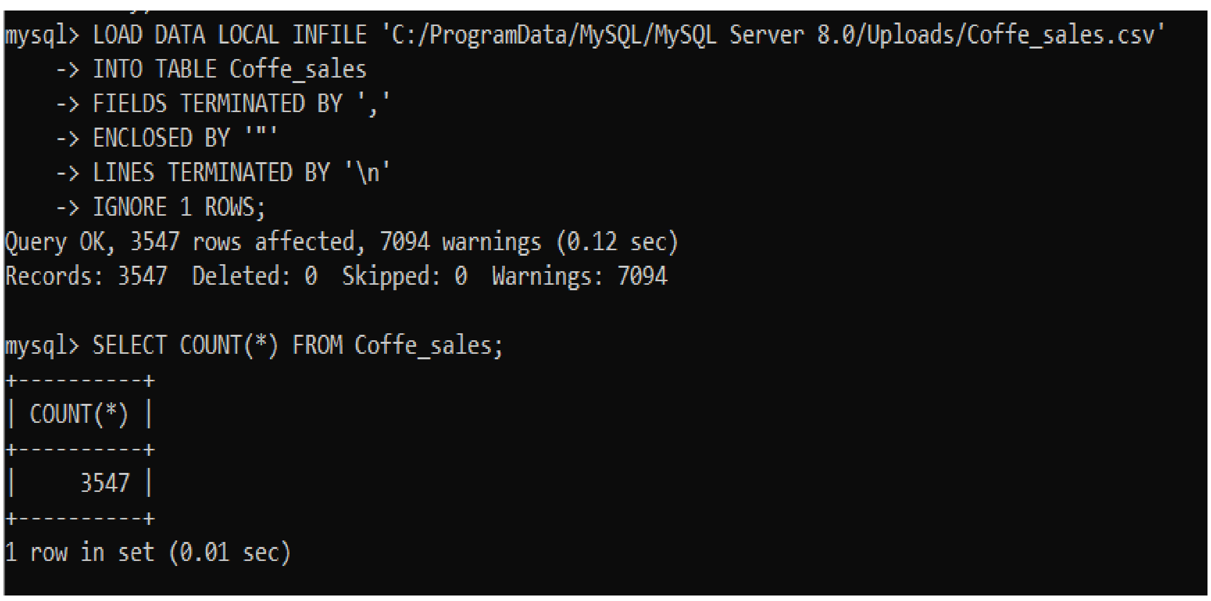
- **Description:**
  MySQL is an open-source relational database management system used to store and manage structured data efficiently.
- **Working Environment:**
  In this project, MySQL was used to import the Coffee_Sales.csv dataset, create

tables, and perform SQL queries for preliminary data analysis — such as total sales, popular coffee types, and weekday vs weekend trends.

2. **Python (for Visualization)**
   - **Description:**
     Python, with libraries such as **Matplotlib** and **Seaborn**, was used to visualize key insights from the dataset.
   - **Working Environment:**
     Visualization scripts were executed in Jupyter Notebook to create bar charts and graphs for:
     - Top 3 best-selling coffees
     - Most profitable month
     - Peak sales hours
     - Day-wise and branch-wise revenue comparison

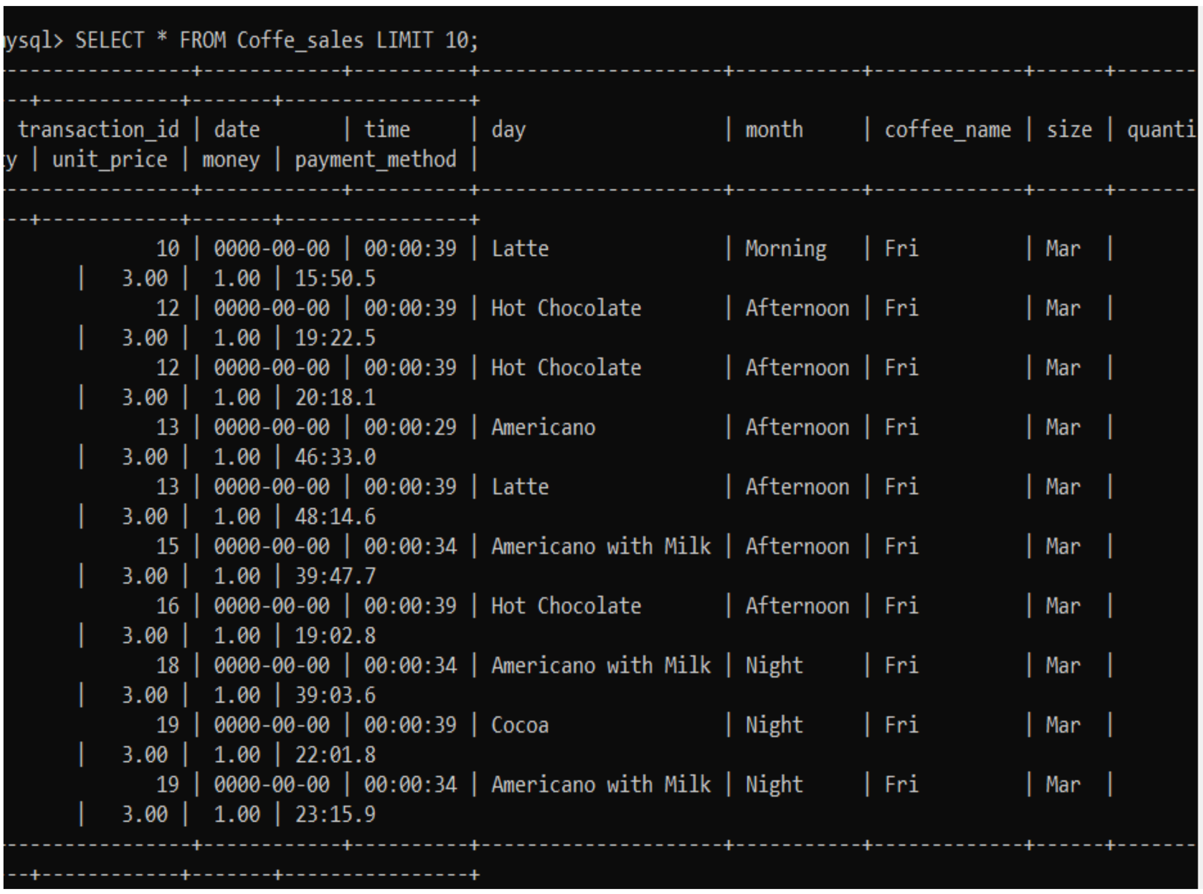
Performing Analysis on MySQL

To load data –



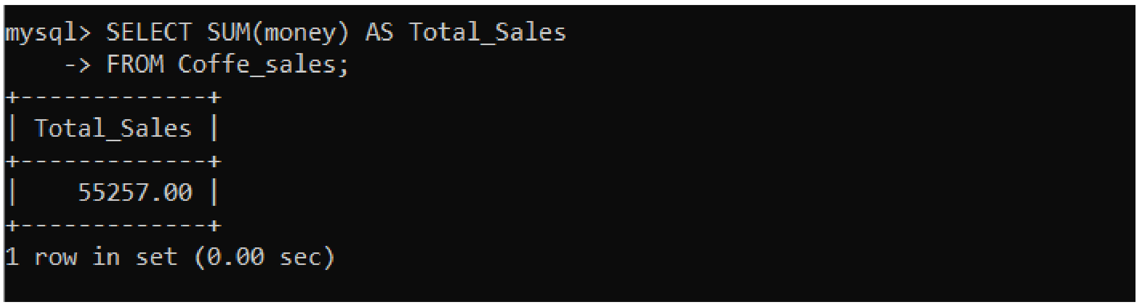
```
mysql> LOAD DATA LOCAL INFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/Coffe_sales.csv'
    -> INTO TABLE Coffe_sales
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY '"'
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 ROWS;
Query OK, 3547 rows affected, 7094 warnings (0.12 sec)
Records: 3547  Deleted: 0  Skipped: 0  Warnings: 7094

mysql> SELECT COUNT(*) FROM Coffe_sales;
+----------+
| COUNT(*) |
+----------+
|     3547 |
+----------+
1 row in set (0.01 sec)
```

Then Verify-



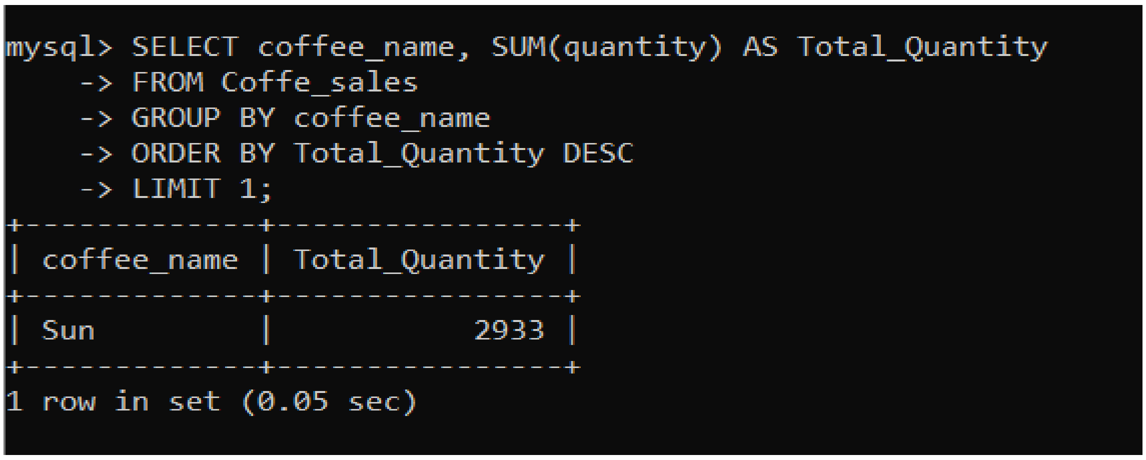
```
ysql> SELECT * FROM Coffe_sales LIMIT 10;
----------------+-----------+----------+--------------------+-----------+------------+------+-------
--+-----------+-------+---------------+
 transaction_id | date       | time     | day                | month     | coffee_name | size | quanti
y | unit_price | money | payment_method |
----------------+-----------+----------+--------------------+-----------+------------+------+-------
--+-----------+-------+---------------+
             10 | 0000-00-00 | 00:00:39 | Latte              | Morning   | Fri        | Mar  |
      |   3.00 |  1.00 | 15:50.5
             12 | 0000-00-00 | 00:00:39 | Hot Chocolate      | Afternoon | Fri        | Mar  |
      |   3.00 |  1.00 | 19:22.5
             12 | 0000-00-00 | 00:00:39 | Hot Chocolate      | Afternoon | Fri        | Mar  |
      |   3.00 |  1.00 | 20:18.1
             13 | 0000-00-00 | 00:00:29 | Americano          | Afternoon | Fri        | Mar  |
      |   3.00 |  1.00 | 46:33.0
             13 | 0000-00-00 | 00:00:39 | Latte              | Afternoon | Fri        | Mar  |
      |   3.00 |  1.00 | 48:14.6
             15 | 0000-00-00 | 00:00:34 | Americano with Milk | Afternoon | Fri        | Mar  |
      |   3.00 |  1.00 | 39:47.7
             16 | 0000-00-00 | 00:00:39 | Hot Chocolate      | Afternoon | Fri        | Mar  |
      |   3.00 |  1.00 | 19:02.8
             18 | 0000-00-00 | 00:00:34 | Americano with Milk | Night     | Fri        | Mar  |
      |   3.00 |  1.00 | 39:03.6
             19 | 0000-00-00 | 00:00:39 | Cocoa              | Night     | Fri        | Mar  |
      |   3.00 |  1.00 | 22:01.8
             19 | 0000-00-00 | 00:00:34 | Americano with Milk | Night     | Fri        | Mar  |
      |   3.00 |  1.00 | 23:15.9
----------------+-----------+----------+--------------------+-----------+------------+------+-------
--+-----------+-------+---------------+
```

Total Sales Revenue-



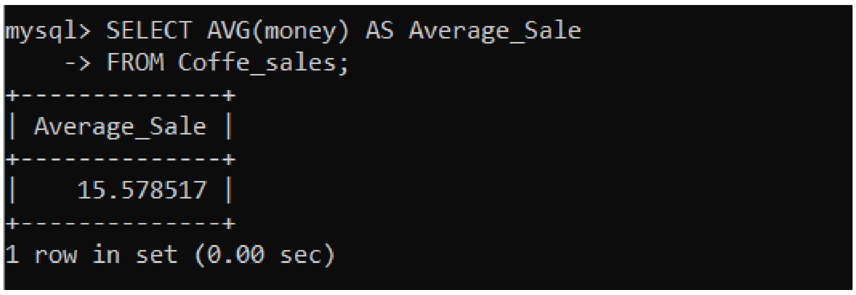
```
mysql> SELECT SUM(money) AS Total_Sales
    -> FROM Coffe_sales;
+-------------+
| Total_Sales |
+-------------+
|    55257.00 |
+-------------+
1 row in set (0.00 sec)
```

Most Popular Coffee-



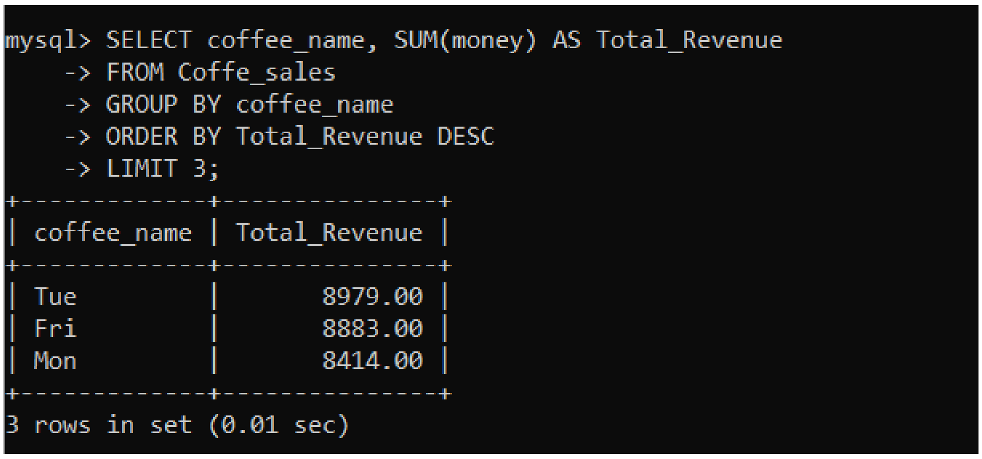
```
mysql> SELECT coffee_name, SUM(quantity) AS Total_Quantity
    -> FROM Coffe_sales
    -> GROUP BY coffee_name
    -> ORDER BY Total_Quantity DESC
    -> LIMIT 1;
+-------------+----------------+
| coffee_name | Total_Quantity |
+-------------+----------------+
| Sun         |           2933 |
+-------------+----------------+
1 row in set (0.05 sec)
```

Average Sale Value-



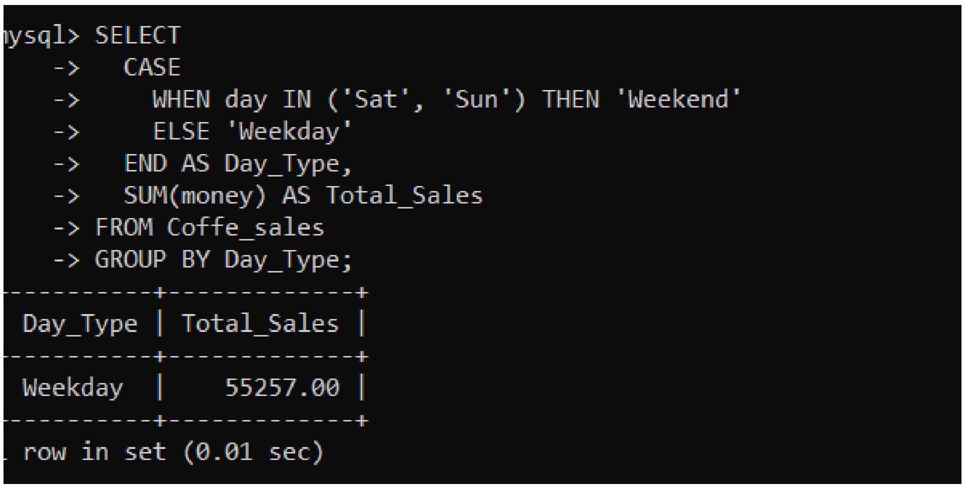
```
mysql> SELECT AVG(money) AS Average_Sale
    -> FROM Coffe_sales;
+--------------+
| Average_Sale |
+--------------+
|    15.578517 |
+--------------+
1 row in set (0.00 sec)
```

Top 3 Best-Selling Coffee Types-



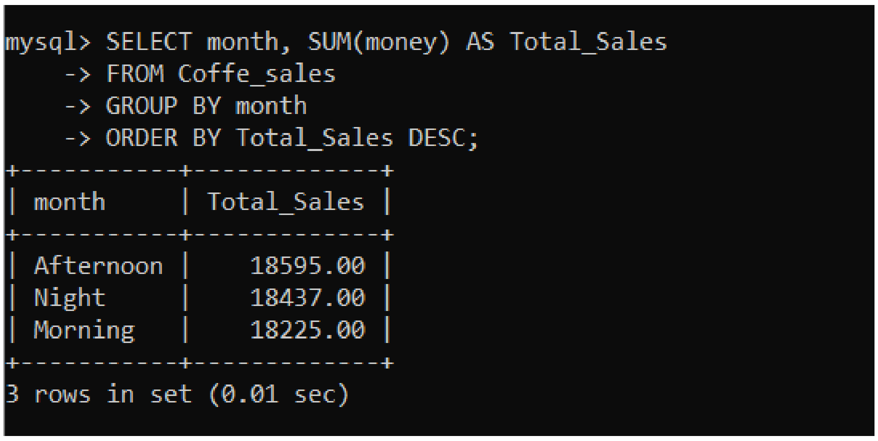
```
mysql> SELECT coffee_name, SUM(money) AS Total_Revenue
    -> FROM Coffe_sales
    -> GROUP BY coffee_name
    -> ORDER BY Total_Revenue DESC
    -> LIMIT 3;
+-------------+---------------+
| coffee_name | Total_Revenue |
+-------------+---------------+
| Tue         |       8979.00 |
| Fri         |       8883.00 |
| Mon         |       8414.00 |
+-------------+---------------+
3 rows in set (0.01 sec)
```

Weekend vs Weekday Sales-



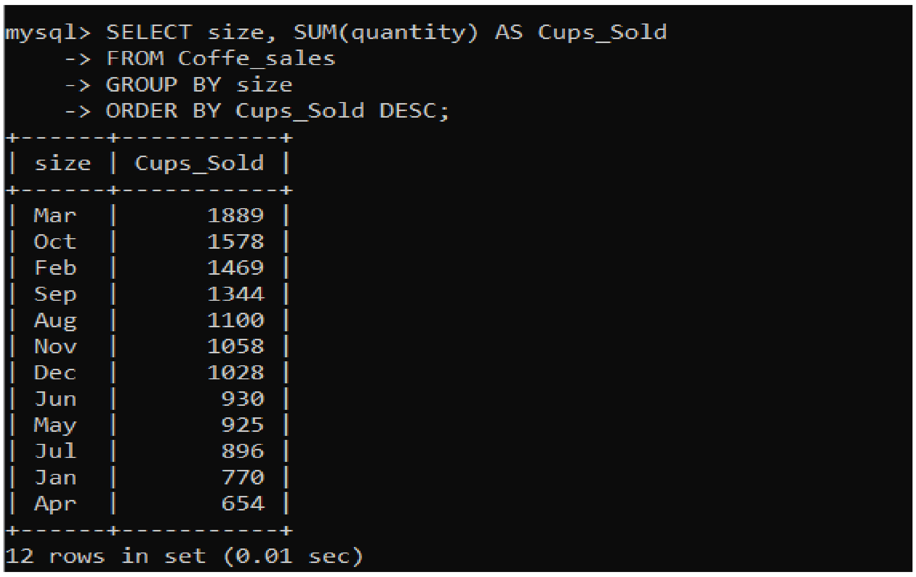
```
ysql> SELECT
    ->   CASE
    ->     WHEN day IN ('Sat', 'Sun') THEN 'Weekend'
    ->     ELSE 'Weekday'
    ->   END AS Day_Type,
    ->   SUM(money) AS Total_Sales
    -> FROM Coffe_sales
    -> GROUP BY Day_Type;
----------+-------------+
 Day_Type | Total_Sales |
----------+-------------+
 Weekday  |    55257.00 |
----------+-------------+
 row in set (0.01 sec)
```

Most Profitable Month-



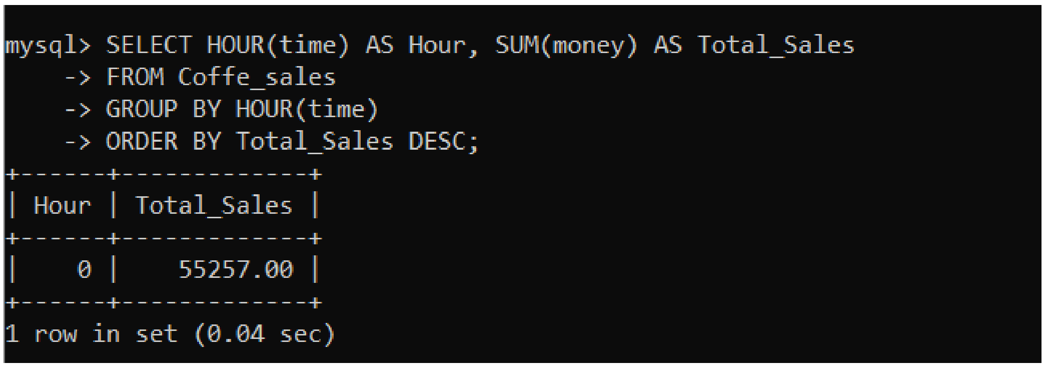
```
mysql> SELECT month, SUM(money) AS Total_Sales
    -> FROM Coffe_sales
    -> GROUP BY month
    -> ORDER BY Total_Sales DESC;
+-----------+-------------+
| month     | Total_Sales |
+-----------+-------------+
| Afternoon |    18595.00 |
| Night     |    18437.00 |
| Morning   |    18225.00 |
+-----------+-------------+
3 rows in set (0.01 sec)
```

Most Popular Cup Size-



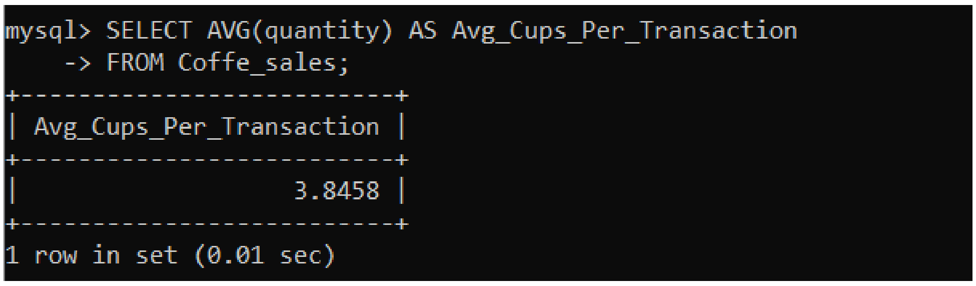
```
mysql> SELECT size, SUM(quantity) AS Cups_Sold
    -> FROM Coffe_sales
    -> GROUP BY size
    -> ORDER BY Cups_Sold DESC;
+------+-----------+
| size | Cups_Sold |
+------+-----------+
| Mar  |      1889 |
| Oct  |      1578 |
| Feb  |      1469 |
| Sep  |      1344 |
| Aug  |      1100 |
| Nov  |      1058 |
| Dec  |      1028 |
| Jun  |       930 |
| May  |       925 |
| Jul  |       896 |
| Jan  |       770 |
| Apr  |       654 |
+------+-----------+
12 rows in set (0.01 sec)
```

Peak Hour of Sales-



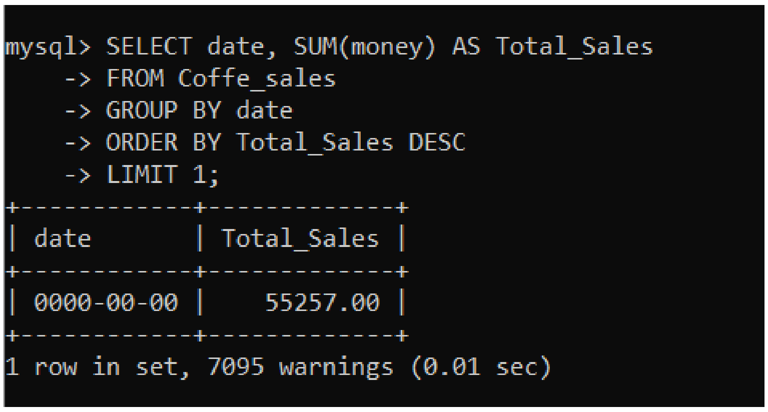
```
mysql> SELECT HOUR(time) AS Hour, SUM(money) AS Total_Sales
    -> FROM Coffe_sales
    -> GROUP BY HOUR(time)
    -> ORDER BY Total_Sales DESC;
+------+-------------+
| Hour | Total_Sales |
+------+-------------+
|    0 |    55257.00 |
+------+-------------+
1 row in set (0.04 sec)
```

Average Quantity Sold per Transaction-



```
mysql> SELECT AVG(quantity) AS Avg_Cups_Per_Transaction
    -> FROM Coffe_sales;
+--------------------------+
| Avg_Cups_Per_Transaction |
+--------------------------+
|                   3.8458 |
+--------------------------+
1 row in set (0.01 sec)
```

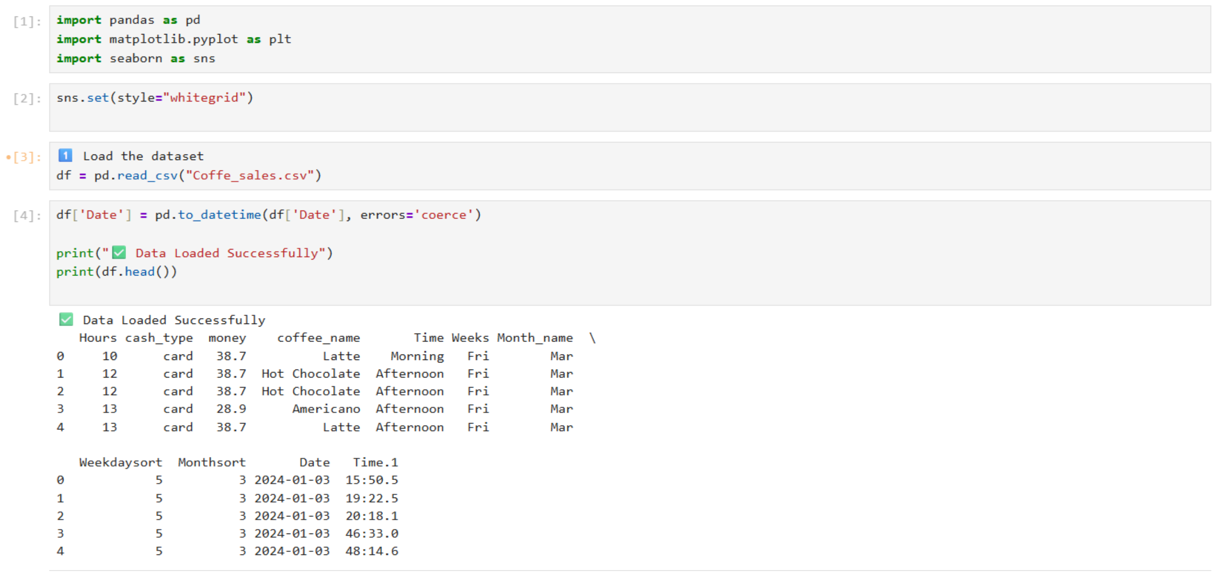
Day with Highest Total Sales-



```
mysql> SELECT date, SUM(money) AS Total_Sales
    -> FROM Coffe_sales
    -> GROUP BY date
    -> ORDER BY Total_Sales DESC
    -> LIMIT 1;
+------------+-------------+
| date       | Total_Sales |
+------------+-------------+
| 0000-00-00 |    55257.00 |
+------------+-------------+
1 row in set, 7095 warnings (0.01 sec)
```

Data Visulization

Step 1: Import Libraries



```
[1]: import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
```

```
[2]: sns.set(style="whitegrid")
```

```
*[3]: 1️⃣ Load the dataset
      df = pd.read_csv("Coffe_sales.csv")
```

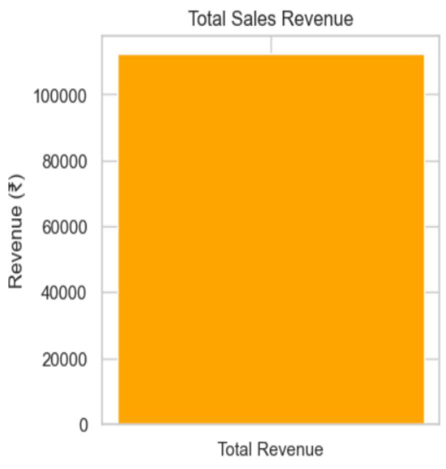
```
[4]: df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

     print("✅ Data Loaded Successfully")
     print(df.head())
```

```
✅ Data Loaded Successfully
   Hours cash_type  money    coffee_name       Time Weeks Month_name  \
0     10      card   38.7          Latte    Morning   Fri        Mar
1     12      card   38.7  Hot Chocolate  Afternoon   Fri        Mar
2     12      card   38.7  Hot Chocolate  Afternoon   Fri        Mar
3     13      card   28.9      Americano  Afternoon   Fri        Mar
4     13      card   38.7          Latte  Afternoon   Fri        Mar

   Weekdaysort  Monthsort       Date   Time.1
0            5          3 2024-01-03  15:50.5
1            5          3 2024-01-03  19:22.5
2            5          3 2024-01-03  20:18.1
3            5          3 2024-01-03  46:33.0
4            5          3 2024-01-03  48:14.6
```

Total Sales Revenue

```
# Total Sales Revenue
total_sales = df['money'].sum()
print("💰 Total Sales Revenue:", round(total_sales, 2))

plt.figure(figsize=(4,4))
plt.bar(['Total Revenue'], [total_sales], color='orange')
plt.title('Total Sales Revenue')
plt.ylabel('Revenue (₹)')
plt.show()
```

```
💰 Total Sales Revenue: 112245.58
```

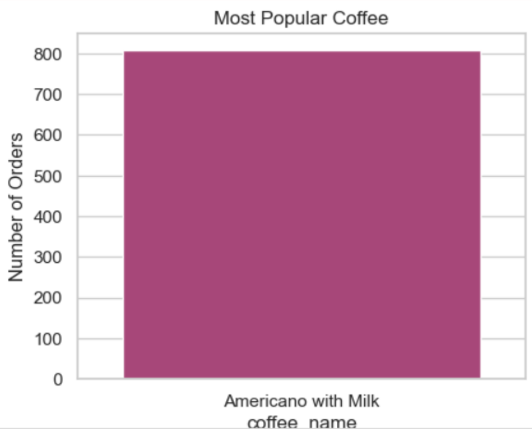


Most Popular Coffee

```
[10]: # Most Popular Coffee
popular_coffee = df['coffee_name'].value_counts().head(1)
print("☕ Most Popular Coffee:\n", popular_coffee)

plt.figure(figsize=(5,4))
sns.barplot(x=popular_coffee.index, y=popular_coffee.values, palette='magma')
plt.title('Most Popular Coffee')
plt.ylabel('Number of Orders')
plt.show()
```

```
☕ Most Popular Coffee:
 coffee_name
Americano with Milk    809
Name: count, dtype: int64
```

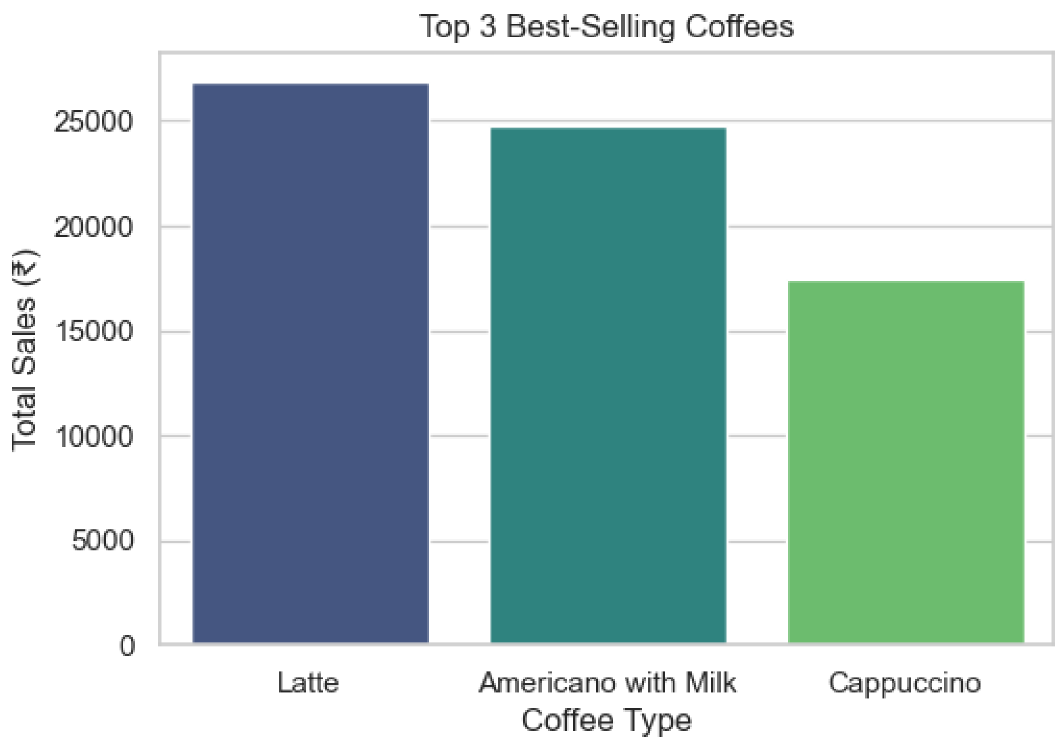
```
C:\Users\anubh\AppData\Local\Temp\ipykernel_12984\3615374467.py:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=popular_coffee.index, y=popular_coffee.values, palette='magma')
```



Top 3 Best-Selling Coffee Types

[13]:
```
# Top 3 Best-Selling Coffee Types
top3 = df.groupby('coffee_name')['money'].sum().sort_values(ascending=False).head(3)
print("🏆 Top 3 Best-Selling Coffees:\n", top3)

plt.figure(figsize=(6,4))
sns.barplot(x=top3.index, y=top3.values, palette='viridis')
plt.title('Top 3 Best-Selling Coffees')
plt.ylabel('Total Sales (₹)')
plt.xlabel('Coffee Type')
plt.show()
```

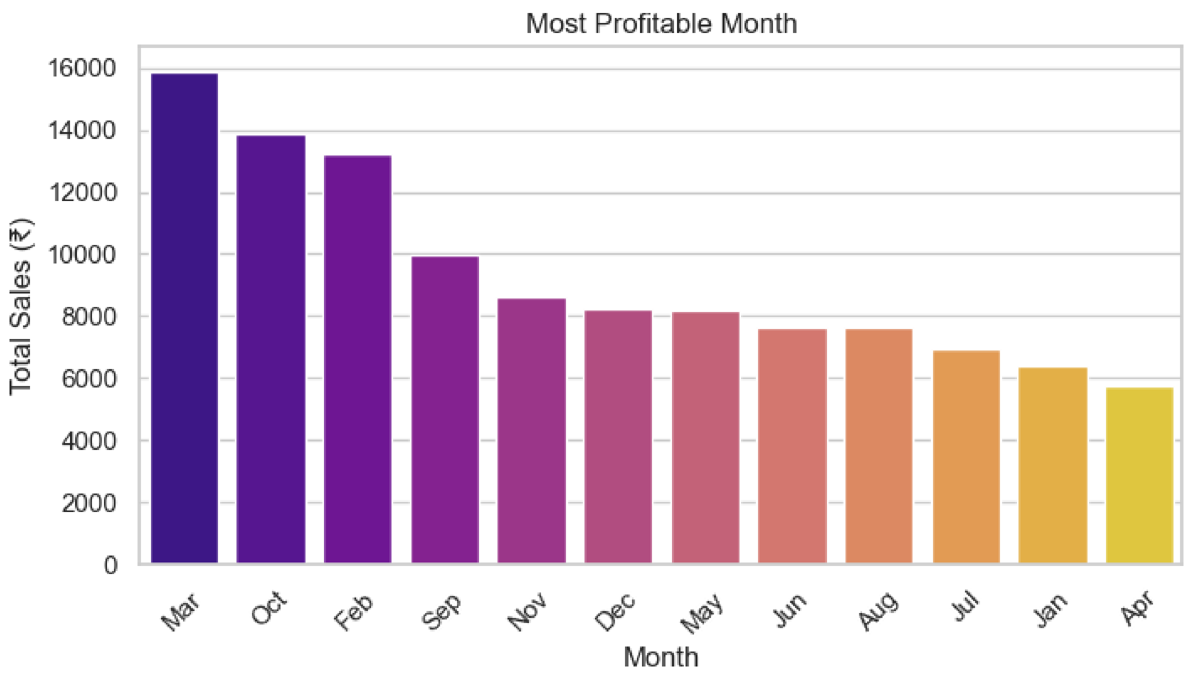
```
🏆 Top 3 Best-Selling Coffees:
 coffee_name
Latte                 26875.30
Americano with Milk   24751.12
Cappuccino            17439.14
Name: money, dtype: float64
```



Most Profitable Month

```
•[16]: # Most Profitable Month
month_sales = df.groupby('Month_name')['money'].sum().sort_values(ascending=False)
print("🗓️ Monthly Sales:\n", month_sales)

plt.figure(figsize=(8,4))
sns.barplot(x=month_sales.index, y=month_sales.values, palette='plasma')
plt.title('Most Profitable Month')
plt.ylabel('Total Sales (₹)')
plt.xlabel('Month')
plt.xticks(rotation=45)
plt.show()
```

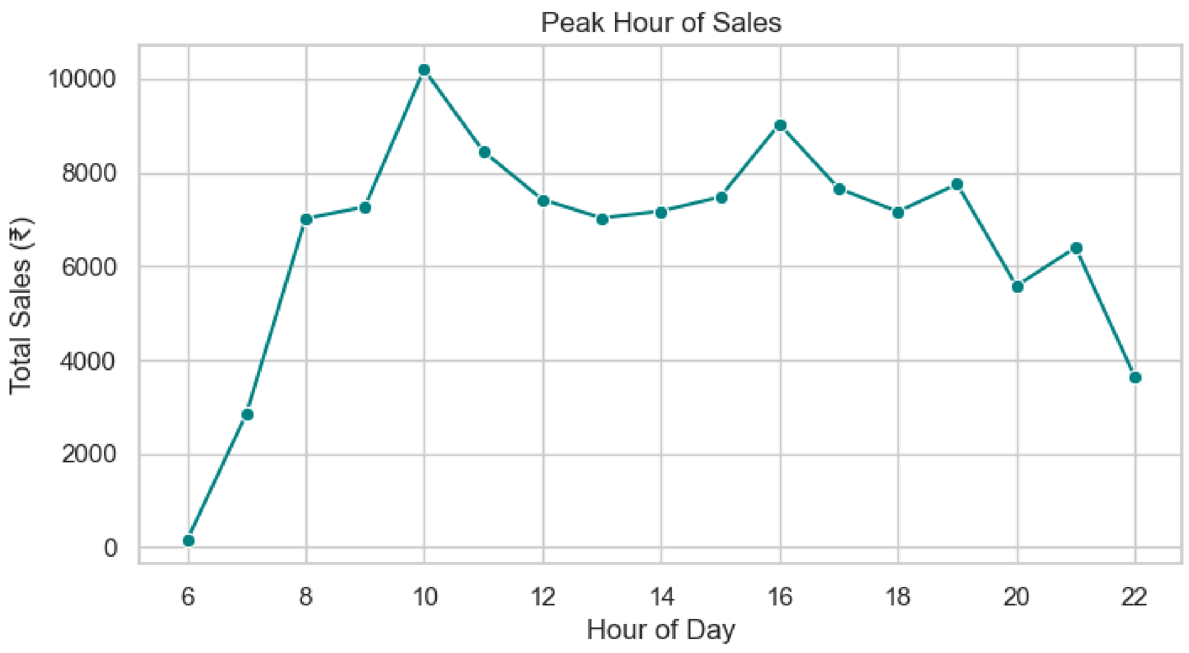
```
🗓️ Monthly Sales:
 Month_name
Mar    15891.64
Oct    13891.16
Feb    13215.48
Sep     9988.64
Nov     8590.54
Dec     8237.74
May     8164.42
Jun     7617.76
Aug     7613.84
Jul     6915.94
Jan     6398.86
Apr     5719.56
Name: money, dtype: float64
```



Peak Hour of Sales

```
[18]: hour_sales = df.groupby('Hours')['money'].sum().sort_values(ascending=False)
      print("⏰ Hourly Sales:\n", hour_sales)

      plt.figure(figsize=(8,4))
      sns.lineplot(x=hour_sales.index, y=hour_sales.values, marker='o', color='teal')
      plt.title('Peak Hour of Sales')
      plt.xlabel('Hour of Day')
      plt.ylabel('Total Sales (₹)')
      plt.show()
```

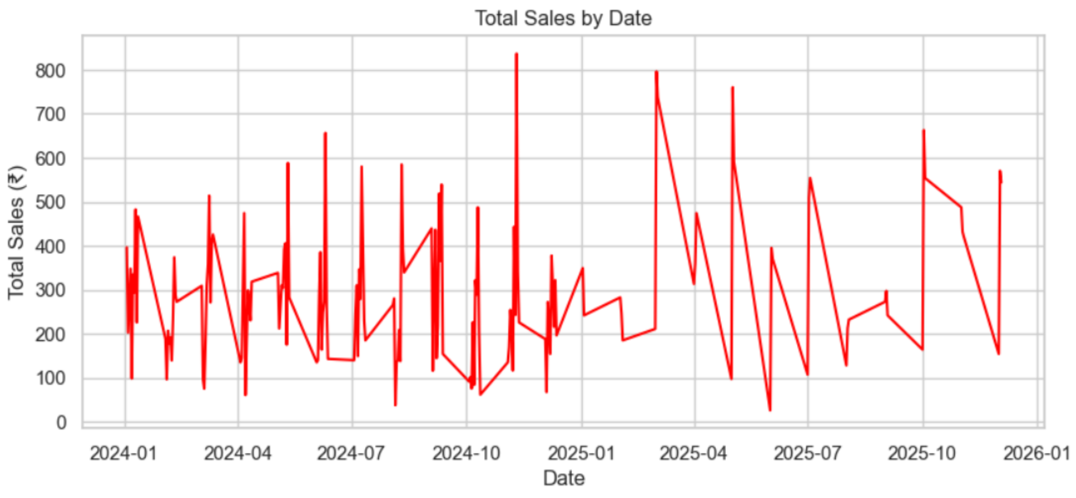
```
⏰ Hourly Sales:
 Hours
10    10198.52
16     9031.84
11     8453.10
19     7751.96
17     7659.76
15     7476.02
12     7419.62
9      7264.28
14     7173.80
18     7162.60
13     7028.76
8      7017.88
21     6397.94
20     5578.92
22     3635.16
7      2846.02
6       149.40
Name: money, dtype: float64
```



Day with Highest Total Sales

```
daily_sales = df.groupby('Date')['money'].sum().sort_values(ascending=False)
top_day = daily_sales.head(1)
print("🌟 Day with Highest Total Sales:\n", top_day)

plt.figure(figsize=(10,4))
sns.lineplot(x=daily_sales.index, y=daily_sales.values, color='red')
plt.title('Total Sales by Date')
plt.xlabel('Date')
plt.ylabel('Total Sales (₹)')
plt.show()

print("✅ Visualization Completed Successfully!")
```

```
🌟 Day with Highest Total Sales:
 Date
2024-11-10    836.66
Name: money, dtype: float64
```


Total Sales by Date
Total Sales (₹)
Date