COFFEE SHOP SALES ANALYSIS

BY KESAVAN

INTRODUCTION

PROJECT OVERVIEW

This project aims to provide actionable insights into the sales performance of our coffee shop by analyzing sales data . We will explore various aspects of the sales data , including trends,patterns and key performance indicators to enhance our understanding and drive strategic decisions.

Objective

The primary objectives of this analysis are:
1.Understand Sales Performance : Identify peak sales periods , Top-selling products , and overall sales trends.
2.Optimize Operations : Discover Insights that can help in optimizing inventory,staffing, and promotional strategies.

Scope of Analysis

Data Dimensions: The Analysis will cover various dimensions such as:

1.Sales by hour of the day.

2.Sales by day of the week.

3.Product sales performance.

4.Sales distribution by store location

COFFEE SALES DATA OVERVIEW

Table size and structure

Total Rows : 1,49,456

Total Columns : 11

Columns Overview

- Transaction ID
- Transaction Date
- Transaction Time
- Transaction Quantity
- Store ID
- Store Location
- Product ID
- Unit Price
- Product Category
- Product Type
- Product Details

Before:

Column Name	Data Type (Initial)
Transaction Date	TEXT
Transaction Time	TEXT
Store Location	TEXT
Product Category	TEXT
Product Type	TEXT
Product Details	TEXT

After:

Column Name	Data Type (Updated)
Transaction Date	DATE
Transaction Time	TIME
Store Location	VARCHAR(255)
Product Category	VARCHAR(255)
Product Type	VARCHAR(255)
[illegible]	VARCHAR(255)

DATA TYPE TRANSFORMATION OVERVIEW

Purpose of Transformation

- IMPROVED DATA INTEGRITY : Ensures accurate DATE and TIME calculations
- ENHANCED QUERY PERFORMANCE : Optimizes data retrieval and analysis
- DATA CONSISTENCY : Facilities correct sorting and grouping of data

QUERIES USED FOR DATA TYPE TRANSFORMATION

```sql
-- Change Text to Date Format
UPDATE coffee_sales
SET Transaction_date=str_to_date(Transaction_date,'%m/%d/%y');

-- Change Text to Time Format
UPDATE coffee_sales
SET Transaction_time=str_to_date(Transaction_time,'%r');

ALTER TABLE coffee_sales
MODIFY COLUMN Transaction_date DATE,
MODIFY COLUMN Transaction_time TIME,
MODIFY COLUMN Store_location VARCHAR(100),
MODIFY COLUMN Product_category VARCHAR(100),
MODIFY COLUMN Product_type VARCHAR(100),
MODIFY COLUMN Product_detail VARCHAR(100);
```

QUESTIONS AND SQL QUERY ANSWERS

1.Write a query to find sales for may month?

QUERY

```
SELECT round(sum(transaction_qty * unit_price),2)AS Total_sales
FROM coffee_sales
WHERE transaction_date BETWEEN '2023-05-01' and '2023-05-31';
```

OUTPUT

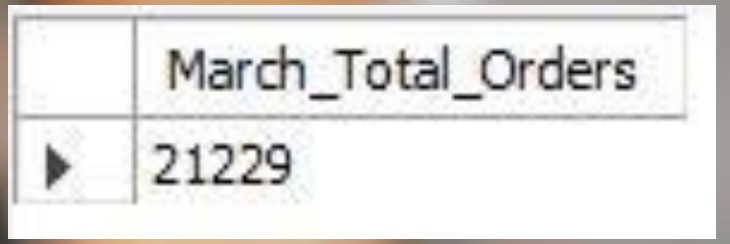
	Total_sales
▶	156727.76

2.Calculate total number of order for each march month?

QUERY

```
SELECT count(ï»¿transaction_id) AS March_Total_Orders
FROM coffee_sales
WHERE
 transaction_date BETWEEN '2023-03-01' and '2023-03-31';
```

OUTPUT



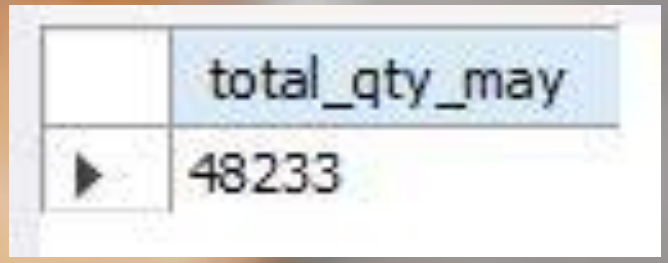
	March_Total_Orders
▶	21229

3.Calculate total quantity for may month?

QUERY

```
SELECT sum(transaction_qty) AS total_qty_may
FROM coffee_sales
WHERE transaction_date BETWEEN '2023-05-01' and '2023-05-31';
```

OUTPUT



	total_qty_may
▶	48233

4.Calculate total_sales , total_qty – sold , total order for particular date 5/18/2023?

QUERY

```
SELECT round(sum(transaction_qty * unit_price) ,2) AS Total_sales,
sum(transaction_qty) AS Total_Qty_sold,
count(ï»¿transaction_id) AS Total_Order
FROM coffee_sales
WHERE
transaction_date='2023-05-18';
```

OUTPUT

	Total_sales	Total_Qty_sold	Total_Order
▶	5583.47	1659	1192

5.Calculate total sale for weekdays and weekend in may month?

QUERY

```
SELECT
  CASE
    WHEN DAYOFWEEK(Transaction_date) IN (2,3,4,5,6) THEN 'WEEKDAYS'
    WHEN dayofweek(Transaction_date) IN (1,7) THEN 'WEEKEND'
    END AS DAY_type,
    round(sum(transaction_qty * unit_price),2) AS Total_Sales_May
  FROM
    coffee_sales
  WHERE
    Transaction_date  BETWEEN '2023-05-01' and '2023-05-31'
  GROUP BY
    DAY_type;
```

OUTPUT

	DAY_type	Total_Sales_May
▶	WEEKDAYS	116627.84
	WEEKEND	40099.92

6.Calculate total sale by location for may month?

QUERY

```
SELECT round(sum(transaction_qty * unit_price ) ,2 ) AS Total,
store_location
FROM coffee_sales
WHERE
transaction_date BETWEEN '2023-05-01' and '2023-05-31'
GROUP BY
store_location
ORDER BY
sum(transaction_qty * unit_price) DESC;
```

OUTPUT

	Total	store_location
▶	52598.93	Hell's Kitchen
	52428.76	Astoria
	51700.07	Lower Manhattan

7.Calculate total sale by product_category for may month?

QUERY

```
SELECT round(sum(transaction_qty * unit_price ) ,2 ) AS Total,
product_category
FROM coffee_sales
WHERE
 transaction_date BETWEEN '2023-05-01' and '2023-05-31'
GROUP BY
 product_category
ORDER BY
 sum(transaction_qty * unit_price) DESC;
```

OUTPUT

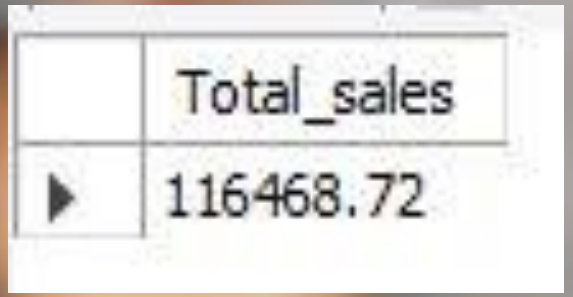
Total	product_category
60362.85	Coffee
44539.85	Tea
18565.52	Bakery
16319.75	Drinking Chocolate
8768.95	Coffee beans
2889	Branded
2395.15	Loose Tea
1905.6	Flavours
981.09	Packaged Chocolate

8.Find average monthly sale for may month?

QUERY

```
SELECT round(sum(transaction_qty * unit_price /6),2)AS Total_sales
FROM coffee_sales
```

OUTPUT



	Total_sales
▶	116468.72

9.Find average daily sale for may month?

QUERY

```
SELECT round(sum(transaction_qty * unit_price /31),2)AS Average_daily_sale
FROM coffee_sales
WHERE
transaction_date BETWEEN '2023/05/01'and '2023/05/31';
```

OUTPUT

	Average_daily_sale
▶	5055.73

10.Determine top 10 product for may month?

QUERY

```
SELECT
    product_id ,
    product_detail,
    sum(transaction_qty * unit_price) AS Total_Sales_May
FROM
    coffee_sales
WHERE
    transaction_date BETWEEN '2023-05-01' and '2023-05-31'
GROUP BY
    product_id ,
    product_detail
ORDER BY
    Total_Sales_May DESC
    LIMIT 10;
```

OUTPUT

product_id	product_detail	Total_Sales_May
61	Sustainably Grown Organic Lg	4921
59	Dark chocolate Lg	4617
39	Latte Rg	4330.75
55	Morning Sunrise Chai Lg	3960
41	Cappuccino Lg	3935.5
38	Latte	3888.75
36	Jamaican Coffee River Lg	3693.75
60	Sustainably Grown Organic Rg	3656.25
40	Cappuccino	3633.75
27	Brazilian Lg	3353

11.Total_sale,Total_qty_sold,Total_Order . When Month = May,Day = Monday,Hour = 8?

QUERY

```
SELECT
    round(sum(transaction_qty * unit_price),2) AS Total_sales_May,
    sum(transaction_qty) AS Total_qty_Sold,
    count(ï»¿transaction_id) AS Total_Orders
FROM
    coffee_sales
WHERE
    MONTH(transaction_date) = 5 AND
    HOUR(transaction_time) = 8  AND
    DAYOFWEEK(transaction_date) = 2;
```

OUTPUT

	Total_sales_May	Total_qty_Sols	Total_Orders
▶	2697.03	819	572

12.Write query to get sales for all hours for month of may?

QUERY

```
SELECT
  HOUR(transaction_time) AS HOUR,
  sum(transaction_qty * unit_price) AS Total_Sales_by_Hours
FROM
  coffee_sales
WHERE
  transaction_date BETWEEN "5/1/2023" and "5/31/2023"
GROUP BY
  HOUR(transaction_time)
ORDER BY
  HOUR;
```

OUTPUT

HOUR	Total_Sales_by_Hours
1	7406.399999999997
2	7104.110000000006
3	7377.210000000003
4	7170.849999999999
5	7001.45
6	10539.88000000002
7	17069.320000000054
8	16548.740000000063
9	16311.190000000037
10	16518.89000000004
11	8325.490000000009
12	6610.499999999995

13.Write query to get sales from monday to sunday for month of may?

QUERY

```
    SELECT
    YEAR(Transaction_date) AS Year,
    MONTH(Transaction_date) AS Month,
    WEEK(Transaction_date, '1') AS Week_number,
    ROUND(SUM(transaction_qty * unit_price),2) AS Total_sales
FROM
    coffee_sales
WHERE
    Transaction_date BETWEEN "2023-05-01" AND "2023-05-31"
GROUP BY
    YEAR(Transaction_date),
    MONTH(Transaction_date),
    WEEK(Transaction_date, '1')
ORDER BY
    YEAR(Transaction_date),
    MONTH(Transaction_date),
    WEEK(Transaction_date, '1');
```

OUTPUT

	Year	Month	Week_number	Total_sales
▶	2023	5	18	32110.1
	2023	5	19	36056.88
	2023	5	20	38476.55
	2023	5	21	36605.12
	2023	5	22	13479.11

14.Determine the month on month increase or decrease in total number of order?

QUERY

```
SELECT
MONTH(Transaction_date) AS Total_Month,
ROUND(COUNT(ï»¿transaction_id)) AS Total_Orders,
COUNT(ï»¿transaction_id) - LAG(COUNT(ï»¿transaction_id),1)
OVER(ORDER BY MONTH(Transaction_date)) / LAG(COUNT(ï»¿transaction_id),1)
OVER(ORDER BY MONTH(Transaction_date)) * 100 AS Month_on_Month_Percentage_Change
FROM
coffee_Sales
WHERE
MONTH(transaction_date) in (1,2,3,4,5)
GROUP BY
MONTH(transaction_date)
ORDER BY
MONTH(transaction_date);
```

OUTPUT

	Total_Month	Total_Orders	Month_on_Month_Percentage_Change
▶	1	17314	NULL
	2	16359	16259.0000
	3	21229	21129.0000
	4	25335	25235.0000
	5	33527	33427.0000

15.Determine the month on month increase or decrease in sale?

QUERY

```
SELECT
MONTH(Transaction_date) AS Total_Month,
ROUND(SUM(unit_price * Transaction_qty)) AS Total_Sale,
SUM(unit_price * Transaction_qty) - LAG(SUM(unit_price * Transaction_qty),1)
OVER(ORDER BY MONTH(Transaction_date)) / LAG(SUM(unit_price * Transaction_qty),1)
OVER(ORDER BY MONTH(Transaction_date)) * 100 AS Month_on_Month_Percentage_Change
FROM
coffee_Sales
WHERE
MONTH(transaction_date) IN (1,2,3,4,5)
GROUP BY
MONTH(transaction_date)
ORDER BY
MONTH(transaction_date);
```

OUTPUT

Total_Month	Total_Sale	Month_on_Month_Percentage_Change
1	81678	NULL
2	76145	76045.18999999958
3	98835	98734.68000000001
4	118941	118841.08000000106
5	156728	156627.7600000045

16.Determine month on month increase or decrease in total quantity sold

QUERY

```
SELECT
MONTH(Transaction_date) AS Total_Month,
ROUND(SUM(Transaction_qty)) AS Total_quantity_order,
SUM(Transaction_qty) - LAG(SUM(Transaction_qty),1)
OVER(ORDER BY MONTH(Transaction_date)) / LAG(SUM(Transaction_qty),1)
OVER(ORDER BY MONTH(Transaction_date)) * 100 AS Month_on_Month_Percentage_Change
FROM
coffee_Sales
WHERE
MONTH(transaction_date) IN (1,2,3,4,5)
GROUP BY
MONTH(transaction_date)
ORDER BY
MONTH(transaction_date);
```

OUTPUT

Total_Month	Total_quantity_order	Month_on_Month_Percentage_Change
1	24870	NULL
2	23550	23450.0000
3	30406	30306.0000
4	36469	36369.0000
5	48233	48133.0000

17.Compare each day of may month weather they are below average?

QUERY

```
SELECT
    Transaction_date , ROUND(SUM(transaction_qty * unit_price ),2)AS Below_Average_daily_sale
FROM
    coffee_sales
WHERE
    MONTH(transaction_date)='5'
GROUP BY
    transaction_date
HAVING
    ROUND(SUM(transaction_qty * unit_price ),2) < 5055;
```

OUTPUT

Transaction_date	Below_Average_daily_sale
2023-05-01	4731.45
2023-05-02	4625.5
2023-05-03	4714.6
2023-05-04	4589.7
2023-05-05	4701
2023-05-06	4205.15
2023-05-07	4542.7
2023-05-11	4850.06
2023-05-12	4681.13
2023-05-14	5052.65
2023-05-28	4338.65
2023-05-29	3959.5
2023-05-30	4835.48
2023-05-31	4684.13

18.Compare each day of may month weather they are above average?

QUERY

```
SELECT
   transaction_date ,ROUND(SUM(transaction_qty * unit_price ),2 ) AS Above_Average_daily_sales
FROM
   coffee_sales
WHERE
   MONTH(transaction_date)='5'
GROUP BY
   Transaction_date
HAVING
   ROUND(SUM(transaction_qty * unit_price ),2 ) > 5055;
```

OUTPUT

transaction_date	Above_Average_daily_sales
2023-05-08	5604.21
2023-05-09	5100.97
2023-05-10	5256.33
2023-05-13	5511.53
2023-05-15	5384.98
2023-05-16	5542.13
2023-05-17	5418
2023-05-18	5583.47
2023-05-19	5657.88
2023-05-20	5519.28
2023-05-21	5370.81
2023-05-22	5541.16
2023-05-23	5242.91
2023-05-24	5391.45
2023-05-25	5230.85
2023-05-26	5300.95
2023-05-27	5559.15

CONCLUSION

The Coffee Sales analysis provided valuable insights into sales performance across various dimensions such as Time , Location and Product Categories by transforming raw sales data into actionable information . We identified peak sales periods , Top selling products and patterns that can help optimize inventory management , Improve promotional strategies and Enhance overall customer experience . This Analysis equips decision-makers with Data-Driven insights to boost operationa; efficiency and drive sales growth

KEY TAKEAWAYS :

- PEAK SALES PERIODS : Identified busiest times to optimize staffing and inventory.
- TOP PRODUCTS : Highlighted Best – Selling items for focused marketing.
- LOCATION INSIGHTS : Provided sales performance by location to tailor offerings.


THANK YOU