

COFFEE SHOP SALES ANALYSIS



Project Introduction

This presentation explores the **Coffee Shop Sales Dashboard** using **MySQL** and **Power BI**, developed to analyze and visualize sales data from **January to June 2023**. The dashboard offers key insights into sales performance, product trends, customer preferences, and operational efficiency across various store locations.

Objectives: Leverage MySQL for data extraction and transformation, and Power BI for visualization to understand overall sales figures and compare them to previous periods.

Identify Key Trends: Pinpoint peak periods and popular products through detailed data analysis.

Optimize Strategies: Use data-driven insights to guide marketing, product offerings, and operational decisions.

By the end of this presentation, we'll gain a clear understanding of the coffee shop's performance over the past six months and the essential insights to drive future success

Project Overview

In this project, I conducted an in-depth analysis of **COFFEE SHOP SALES** data using **MySQL** and **Power BI**. The objective was to uncover key insights into sales trends, customer behavior, and product performance.

MySQL was utilized for data extraction, cleaning, and transformation. I wrote complex queries to structure and organize the raw data, ensuring it was ready for analysis.

Power BI was employed to create interactive dashboards and visualizations that provide a clear and insightful representation of the data. These visual tools allow stakeholders to easily understand sales patterns, identify top-performing products, and make data-driven decisions.

This project not only demonstrates my technical proficiency in data analysis tools but also showcases my ability to translate raw data into actionable business insights.

Problem Statements

1. Total Sales Analysis

- => Calculate the total sales for February Month.
- => Determine the month-on-month increase or decrease in sales.
- => Calculate the difference in sales between the selected month and the previous month.

2. Total Orders Analysis

- => Calculate the total orders for April Month.
- => Determine the month-on-month increase or decrease in number or orders .
- => Calculate the difference in number of orders between the selected month and the previous month.

3. Total Quantity Sold Analysis

- => Calculate the total quantity sold for June Month.
- => Determine the month-on-month increase or decrease in the total quantity sold .
- => Calculate the difference in the total quantity sold between the selected month and the previous month.

4. Sales Analysis by Weekdays and WeekEnds

=> Calculate all the metrics (i.e., Sales, Orders, Quantity) for a specific day.

=> Calculate total sales on both weekdays and weekends of any Month.

5. Sales Analysis by Store Location

=> Calculate the total sales for different store location for any one month.

6. Daily Sales Analysis with Average line

=> Daily sales analysis for the selected month with line chart.

7. Sales Analysis with Product Category

=> Analyse sales performance across different product category.

Problem Statements

8. Top 10 Products by Sales

=> Identify and display top 10 products based on sales volume.

9. Sales Analysis by Days and Hours

=> Calculate the total sales patterns by Days and Hours for April Month

=> Detailed metrics(Sales, Orders, Quantity) over a specific day-hour for April Month.

=> Day wise total sales for a particular month.

Total Sales Analysis

=> Calculate the total sales for February Month.

```
1 • select concat(round(sum(unit_price * transaction_qty))/1000,'K') as Total_Sales
2   from coffee
3   where month(transaction_date)=2;
```

Result Grid	
	Total_Sales
▶	76.145K


=> Determine the month-on-month increase or decrease in sales.

```
1 • SELECT
2     MONTH(transaction_date)AS Month,
3     ROUND(SUM(transaction_qty * unit_price)) AS Total_Sales,
4     (SUM(transaction_qty * unit_price)- LAG(SUM(transaction_qty * unit_price),1)
5     OVER(ORDER BY MONTH(transaction_date)))/ LAG(SUM(transaction_qty * unit_price),1)
6     OVER(ORDER BY MONTH(transaction_date)) * 100 AS M_O_M_percentage
7 FROM coffee
8 WHERE MONTH(transaction_date) IN (1,2)
9 GROUP BY MONTH(transaction_date)
10 ORDER BY MONTH(transaction_date);
```

Result Grid			
	Month	Total_Sales	M_O_M_percentage
▶	1	81678	NULL
	2	76145	-6.773632571126168

=> Calculate the difference in sales between the selected month and the previous month.



```
2 • with cte as (  
3     select month(transaction_date) as Month,  
4           sum(transaction_qty * unit_price) as Total_Sales  
5     from coffee  
6     where month(transaction_date) in (1,2)  
7     group by month(transaction_date)  
8 )  
9  
10    SELECT  
11        Month,  
12        Total_Sales,  
13        ROUND(Total_Sales - LAG(total_sales) OVER (ORDER BY month)) AS Sales_Difference  
14    FROM cte  
15    ORDER BY month;
```

Result Grid  Filter Rows: <input type="text"/> Exp			
	Month	Total_Sales	Sales_Difference
▶	1	81677.739999999928	NULL
	2	76145.189999999958	-5533

Total Orders Analysis



=> Calculate the total orders for April Month.

```
1 • select count(*) as Number_of_orders from coffee
2   where month(transaction_date)=4;
```

Result Grid			
	Number_of_orders		
▶	25335		

=> Determine the month-on-month increase or decrease in number of orders.

```
1 • select month(transaction_date) as Month,
2       count(transaction_id) as Total_Orders,
3       (count(transaction_id)-lag(count(transaction_id),1)
4       OVER(ORDER BY MONTH(transaction_date)))/ LAG(COUNT(transaction_id),1)
5       OVER(ORDER BY MONTH(transaction_date)) * 100 AS M_O_M_percentage
6
7   from coffee
8  where month(transaction_date) in (3,4)
9  group by month(transaction_date)
10 order by month(transaction_date);
```

Result Grid				Filter Rows:
	Month	Total_Orders	M_O_M_percentage	
▶	3	21229	NULL	
	4	25335	19.3415	

=> Calculate the difference in number of orders between the selected month and the previous month.

```
2 • with cte as(  
3     select month(transaction_date) as Months,  
4     count(transaction_id) as Total_Orders  
5     from coffee  
6     where month(transaction_date) in (4,5)  
7     group by month(transaction_date)  
8 )  
9     select months, Total_Orders,  
10         Total_Orders - lag(Total_Orders) over(order by Months) as Orders_Diff  
11 from cte;
```

Result Grid Filter Rows:			
	months	Total_Orders	Orders_Diff
▶	4	25335	NULL
	5	33527	8192

Total Quantity Sold Analysis

=> Calculate the total quantity sold for June Month.

```
2 • select sum(transaction_qty) as Total_Quantity_Sold  
3     from coffee  
4     where month(transaction_date)=6;
```

Result Grid Filter Rows:	
	Total_Quantity_Sold
▶	50942

=> Determine the month-on-month increase or decrease in the total quantity sold .

```
1 • select month(transaction_date) as Month,  
2       sum(transaction_qty) as Total_Quantity,  
3       (sum(transaction_qty)-lag(sum(transaction_qty),1)  
4       over (order by month(transaction_date)))/lag(sum(transaction_qty),1)  
5       over (order by month(transaction_date)) * 100 AS M_O_M_Percentage  
6 from coffee  
7 where month(transaction_date) in (5,6)  
8 group by month(transaction_date);
```

Result Grid			
Filter Rows:			
	Month	Total_Quantity	M_O_M_Percentage
▶	5	48233	NULL
	6	50942	5.6165

=> Calculate the difference in the total quantity sold between the selected month and the previous month.

```
1 • with cte as(  
2   select month(transaction_date) as Month,  
3         sum(transaction_qty) as Total_Quantity  
4   from coffee  
5   where month(transaction_date) in (5,6)  
6   group by month(transaction_date))  
7   select Month, Total_Quantity,  
8         Total_Quantity-lag(Total_Quantity) over(order by Month)  
9         AS Total_Quantity_Difference  
10  from cte;
```

Result Grid			
Filter Rows:			
Export:			
	Month	Total_Quantity	Total_Quantity_Difference
▶	5	48233	NULL
	6	50942	2709

Sales Analysis by Weekdays and Weekends

=> Calculate all the metrics (i.e., Sales, Orders, Quantity) for a specific day.

```
1 • select concat(round(sum(transaction_qty * unit_price)/1000,1),'K') as Total_Sales,  
2         concat(round(count(transaction_id)/1000,1),'K') as Total_Orders,  
3         concat(round(sum(transaction_qty)/1000,1),'K') as Total_quantity_Sold  
4 from coffee  
5 where transaction_date='2023-03-27';
```

Result Grid				Filter Rows:	
	Total_Sales	Total_Orders	Total_quantity_Sold		
▶	3.7K	0.8K	1.2K		

=> Calculate total sales on both weekdays and weekends of any Month.

```
2 • select  
3     case when dayofweek(transaction_date) in (1,7) then 'WeekEnds'  
4     else 'WeekDays'  
5     end as Day_Type,  
6     concat(round(sum(unit_price * transaction_qty)/1000,1), 'K') as Total_Sales  
7 from coffee  
8 where month(transaction_date)=2  
9 group by case when dayofweek(transaction_date) in (1,7) then 'WeekEnds'  
10          else 'WeekDays'  
11          end;
```

Result Grid			Filter R	
	Day_Type	Total_Sales		
▶	WeekDays	54K		
	WeekEnds	22.1K		

Sales Analysis by Store Location

=> Calculate the total sales for different store location for any one month.

```
2 • select store_location, concat(round(sum(transaction_qty * unit_price)/1000,2),'K') as Total_Sales
3   from coffee
4   where month(transaction_date)=6
5   group by store_location
6   order by sum(transaction_qty * unit_price) desc;
```

Result Grid		Filter Rows:
	store_location	Total_Sales
	Hell's Kitchen	56.96K
	Astoria	55.08K
	Lower Manhattan	54.45K

Daily Sales Analysis with Average Line

=> Daily sales analysis for the selected month with line chart.

```
1 • select concat(round(avg(Total_Sales)/1000,1),'K') as AVG_Sales
2   from
3   (select sum(transaction_qty * unit_price)as Total_Sales
4    from coffee
5    where month(transaction_date)=5
6    group by transaction_date) as sub;
7
8 • select day(transaction_date) as Day, sum(unit_price * transaction_qty) as Total_Sales
9   from coffee
10  where month(transaction_date)=5
11  group by day(transaction_date);
12
```

```
13 • select Day,
14   (Case when Total_Sales > AVG_Sales then 'Above Average'
15    when Total_Sales <AVG_Sales then 'Below Average'
16    else 'Equal to Average'
17    End as Sales_Status, Total_Sales
18  from ( select day(transaction_date) as Day,
19   sum(unit_price * transaction_qty) as Total_Sales,
20   avg(sum(unit_price * transaction_qty)) over () AVG_Sales
21
22   from coffee
23   where month(transaction_date)=5
24   group by day(transaction_date)
25  ) as Sales_Data
26  order by Day;
```


Result Grid			
Filter Rows:			
	Day	Sales_Status	Total_Sales
▶	1	Below Average	4731.4499999999999
	2	Below Average	4625.4999999999997
	3	Below Average	4714.5999999999994
	4	Below Average	4589.6999999999995
	5	Below Average	4700.9999999999997
	6	Below Average	4205.1499999999998
	7	Below Average	4542.6999999999998
	8	Above Average	5604.2099999999995
	9	Above Average	5100.9699999999997
	10	Above Average	5256.3299999999999
	11	Below Average	4850.0599999999996
	12	Below Average	4681.12999999999965
	13	Above Average	5511.5299999999999
	14	Below Average	5052.6499999999999
	15	Above Average	5384.98000000000005

	16	Above Average	5542.1299999999997
	17	Above Average	5418.0000000000001
	18	Above Average	5583.4700000000001
	19	Above Average	5657.8800000000005
	20	Above Average	5519.2800000000003
	21	Above Average	5370.8100000000003
	22	Above Average	5541.16
	23	Above Average	5242.9100000000001
	24	Above Average	5391.45
	25	Above Average	5230.8499999999985
	26	Above Average	5300.9499999999998
	27	Above Average	5559.1500000000015
	28	Below Average	4338.6499999999998
	29	Below Average	3959.4999999999998
	30	Below Average	4835.4799999999997
	31	Below Average	4684.1299999999993

Sales Analysis with Product Category

=> Analyse sales performance across different product category for a particular month.

```
1 • select product_category, concat(round(sum(unit_price * transaction_qty)/1000,1),'K') as Total_Sales
2   from coffee
3  where month(transaction_date)=2
4  group by product_category
5  order by sum(unit_price * transaction_qty) desc;
```

Result Grid			Filter Rows:
	product_category	Total_Sales	
▶	Coffee	29.3K	
	Tea	21.7K	
	Bakery	9K	
	Drinking Chocolate	8.1K	
	Coffee beans	4.1K	
	Loose Tea	1.3K	
	Branded	1.2K	
	Flavours	0.9K	
	Packaged Chocolate	0.5K	

Top 10 Products by Sales

=> Identify and display top 10 products based on sales volume for a particular month.

```

1 • select product_type, concat(round(sum(unit_price * transaction_qty)/1000,1),'K') as Total_Sales
2   from coffee
3  where month(transaction_date)=1
4  group by product_type
5  order by sum(unit_price * transaction_qty) desc
6  limit 10;

```

Result Grid			Filter Rows:
	product_type	Total_Sales	
▶	Barista Espresso	10.5K	
	Brewed Chai tea	8.8K	
	Hot chocolate	8.3K	
	Gourmet brewed coffee	8.1K	
	Brewed Black tea	5.5K	
	Brewed herbal tea	5.4K	
	Organic brewed coffee	4.6K	
	Premium brewed coffee	4.5K	
	Scone	4.3K	
	Drip coffee	3.6K	



Sales Analysis by Days and Hours

=> Calculate the total sales patterns by Days and Hours for April month.

```

1 • select concat(round(sum(unit_price * transaction_qty)/1000,1),'K') as Total_Sales,
2         count(*) as Total_Orders,
3         sum(transaction_qty) as Total_Quantity_Sold
4 from coffee
5 where month(transaction_date)=4
6 and dayofweek(transaction_date)=2
7 and hour(transaction_time)=8

```



Result Grid   Filter Rows:			
	Total_Sales	Total_Orders	Total_Quantity_Sold
▶	1.6K	353	501

=> Detailed metrics(Sales, Orders, Quantity) over a specific day-hour for April Month.

```



1 • select hour(transaction_time) as Hours,
2         concat(round(sum(unit_price * transaction_qty)/100,2),'K') as Total_Sales
3 from coffee
4 where month(transaction_date)=4
5 group by hour(transaction_time)
6 order by hour(transaction_time);

```

Result Grid   Filter Rows:		
	Hours	Total_Sales
▶	6	37.72K
	7	105.01K
	8	137.23K
	9	146.09K
	10	154.51K
	11	82.17K
	12	69.02K
	13	65.53K
	14	69.33K
	15	71.45K
	16	70.65K
	17	70.13K
	18	57.63K
	19	48.24K
	20	4.69K

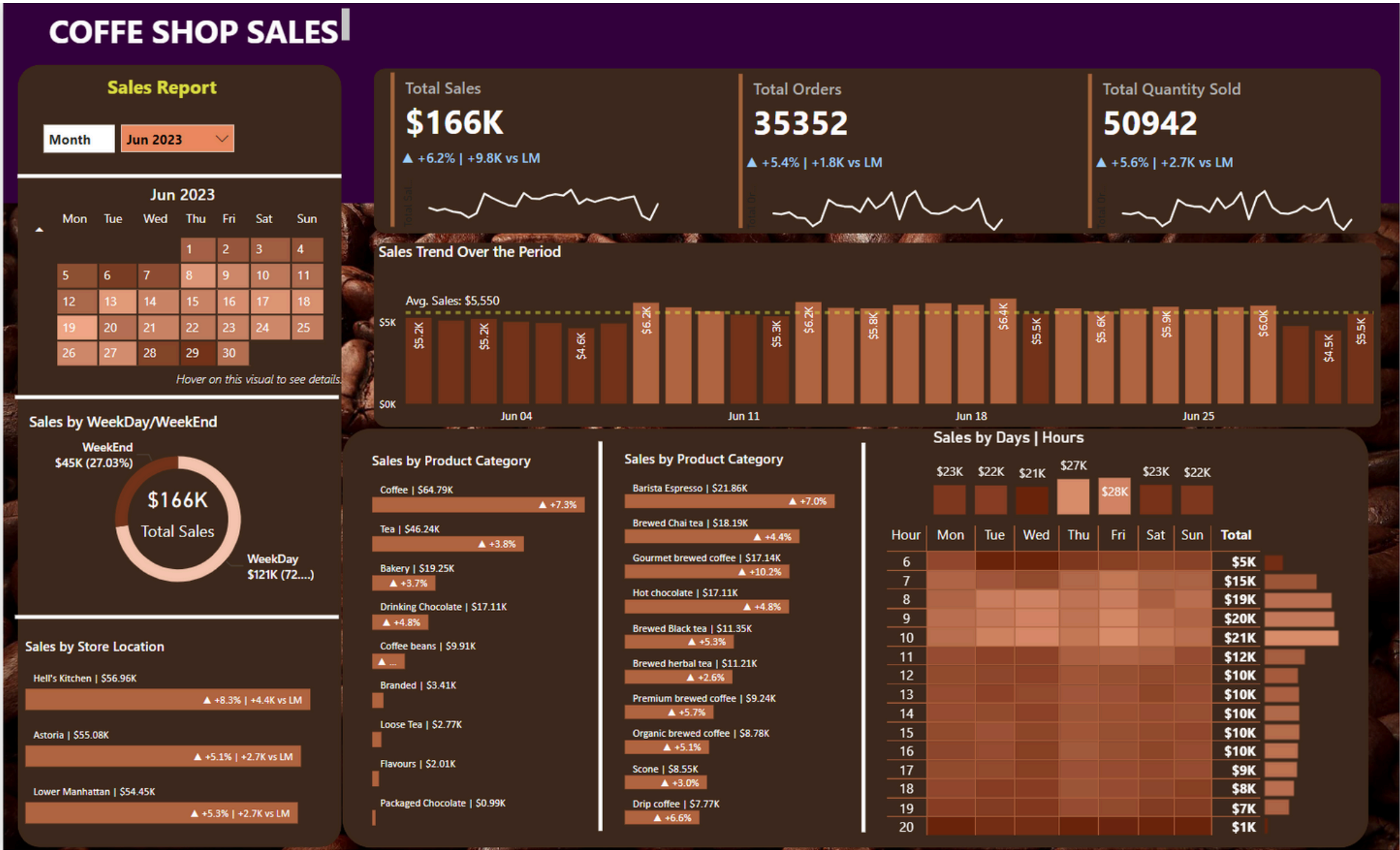
=> Day wise total sales for a particular month.

```
1 • SELECT
2   CASE
3     WHEN DAYOFWEEK(transaction_date) = 2 THEN 'Monday'
4     WHEN DAYOFWEEK(transaction_date) = 3 THEN 'Tuesday'
5     WHEN DAYOFWEEK(transaction_date) = 4 THEN 'Wednesday'
6     WHEN DAYOFWEEK(transaction_date) = 5 THEN 'Thursday'
7     WHEN DAYOFWEEK(transaction_date) = 6 THEN 'Friday'
8     WHEN DAYOFWEEK(transaction_date) = 7 THEN 'Saturday'
9     ELSE 'Sunday'
10  END AS Day_of_Week,
11  ROUND(SUM(unit_price * transaction_qty)) AS Total_Sales
12 FROM
13   coffee
14 WHERE
15   MONTH(transaction_date) = 5 -- Filter for May (month number 5)
16 GROUP BY
17   CASE
18     WHEN DAYOFWEEK(transaction_date) = 2 THEN 'Monday'
19     WHEN DAYOFWEEK(transaction_date) = 3 THEN 'Tuesday'
20     WHEN DAYOFWEEK(transaction_date) = 4 THEN 'Wednesday'
21     WHEN DAYOFWEEK(transaction_date) = 5 THEN 'Thursday'
22     WHEN DAYOFWEEK(transaction_date) = 6 THEN 'Friday'
23     WHEN DAYOFWEEK(transaction_date) = 7 THEN 'Saturday'
24     ELSE 'Sunday'
25  END;
```

Result Grid   Filter Rows:		
	Day_of_Week	Total_Sales
▶	Monday	25221
	Tuesday	25347
	Wednesday	25465
	Thursday	20254
	Friday	20341
	Saturday	20795
	Sunday	19305

Coffee Shop Sales Analysis

Dashboard Link: <https://app.powerbi.com/groups/me/reports/1ecf75b6-a662-4697-8854-d5f35ac7103d/3c161ca44113ed51426f?redirectedFromSignup=1&experience=power-bi>



Power BI DAX Queries

Previous Month Sales

```
PM Sales = CALCULATE([CM Sales], DATEADD('Date_Table'[Date],-1,MONTH))
```

Current Month Sales

```
CM Sales = VAR selected_month= SELECTEDVALUE(Date_Table[Month])
RETURN
TOTALMTD(CALCULATE([Total Sales],'Date_Table'[Month]=
selected_month),'Date_Table'[Date])
```

Month-on-Month Growth and Diff. Sales

```
MOM Growth & Diff Sales = VAR month_diff= [CM Sales] - [PM Sales]
VAR mom= ([CM Sales] - [PM Sales])/[PM Sales]
VAR _sign= IF(month_diff > 0, "+","")
VAR _sign_trend= IF(month_diff > 0,
" ▲ ", " ▼ ")
RETURN
_sign_trend & " " & _sign & FORMAT(mom, "#0.0%" & " | " & _sign & FORMAT(month_diff/1000,
"0.0K")) & " " & "vs LM"
```

Key Insights

TOTAL SALES:- June 2023 saw the highest Sales i.e., \$166K and lowest in February at \$76.

TOTAL ORDERS:- June 2023 had the highest number of orders i.e., 35,352 and lowest in February at 16,359.

TOTAL QUANTITY SOLD:- June 2023 had the highest quantity sold i.e., 50,942.

WEEKDAYS VS WEEKENDS:- Weekdays outperformed Weekends in sales.

TOP SELLING PRODUCT:- Barista Espresso and Brewed Chai Tea are the top selling product types, reflecting strong customer preferences for these items.

PEAK HOURS SALES :- Peak Sales were from 7 AM to 10AM. Also, Monday to Wednesday, are the peak days for sales.

STORE LOCATION PERFORMANCE :- Sales were balanced across Hell's Kitchen, Astoria and Lower Manhattan.

Suggestions for Improvement

Boost Sales During Low Periods

1. Promotions in February: Since February had the lowest sales, consider introducing promotions or discounts to attract more customers during this slow period.
2. Daily Sales Consistency: Address the variability in daily sales by offering daily specials or loyalty programs to encourage consistent visits.

Weekend Sales Enhancement

1. Weekend Brunch or Special Events: Introduce special weekend brunch menus or events like live music or trivia nights to draw in more customers.
2. Targeted Marketing: Use social media and email marketing to highlight weekend-only deals and promotions to boost foot traffic.

Product Optimization

1. Expand Popular Products: Increase inventory of top sellers like Barista Espresso, Brewed Chai Tea, and Hot Chocolate, and consider introducing new variations or sizes.
- 2 Blended Chocolate Rebranding: Since blended chocolate lags in sales, consider rebranding it or creating new recipes that align with current customer tastes.

Peak Hours Focus

1. Morning Specials: Capitalize on peak hours (7:00 to 10:00 AM) by offering morning combos or discounts on popular breakfast items.
2. Staffing Adjustments: Ensure adequate staffing during peak hours to maintain quick service and customer satisfaction.

Implementing these strategies can help the coffee shop enhance its customer experience, increase sales, and sustain growth in the months following June 2023.



Thank You

