**MY SQL QUERIES**

**COFFEE SHOP SALES PROJECT**

CONVERT DATE (transaction\_date) COLUMN TO PROPER DATE FORMAT

UPDATE coffee\_shop\_sales

SET transaction\_date = STR\_TO\_DATE(transaction\_date, &#39;%d-%m-%Y&#39;);

ALTER DATE (transaction\_date) COLUMN TO DATE DATA TYPE

ALTER TABLE coffee\_shop\_sales

MODIFY COLUMN transaction\_date DATE;

CONVERT TIME (transaction\_time) COLUMN TO PROPER DATE FORMAT

UPDATE coffee\_shop\_sales

SET transaction\_time = STR\_TO\_DATE(transaction\_time, &#39;%H:%i:%s&#39;);

ALTER TIME (transaction\_time) COLUMN TO DATE DATA TYPE

ALTER TABLE coffee\_shop\_sales

MODIFY COLUMN transaction\_time TIME;

DATA TYPES OF DIFFERENT COLUMNS

DESCRIBE coffee\_shop\_sales;

CHANGE COLUMN NAME `ï»¿transaction\_id` to transaction\_id

ALTER TABLE coffee\_shop\_sales

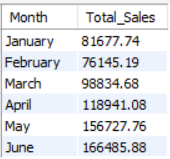
CHANGE COLUMN `ï»¿transaction\_id` transaction\_id INT;

TOTAL SALES

Select MonthName(transaction\_date) as 'Month' , Round(Sum(transaction\_qty \* unit\_price),2) as Total\_Sales

From coffee\_shop

Group by MonthName(transaction\_date);



TOTAL SALES KPI - MOM DIFFERENCE AND MOM GROWTH

Select Month(transaction\_date) as 'Month' ,

Round(Sum(transaction\_qty \* unit\_price)) as Total\_Sales ,

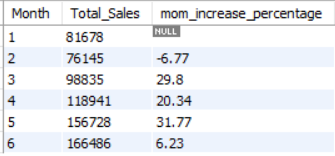
Round((Sum(transaction\_qty \* unit\_price) - Lag(Sum(transaction\_qty \* unit\_price),1)

Over(Order By Month(transaction\_date))) / Lag(Sum(transaction\_qty \* unit\_price),1)

Over(Order By Month(transaction\_date)) \* 100,2) as mom\_increase\_percentage

From coffee\_shop

Group by Month(transaction\_date);

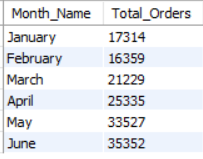


TOTAL ORDERS

Select MonthName(transaction\_date) , Count(transaction\_id) as Total\_Orders

From coffee\_shop

Group by MonthName(transaction\_date);



TOTAL ORDERS KPI - MOM DIFFERENCE AND MOM GROWTH

Select Month(transaction\_date) as 'Month' , Count(transaction\_id) as Total\_Orders ,

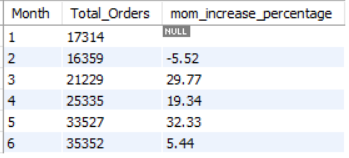
Round((Count(transaction\_id) - Lag(Count(transaction\_id),1)

Over( Order By Month(transaction\_date))) / Lag(Count(transaction\_id),1)

Over( Order By Month(transaction\_date)) \* 100,2) as mom\_increase\_percentage

From coffee\_shop

Group by 1;



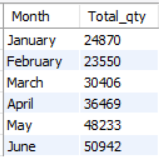
TOTAL QUANTITY SOLD

Select MonthName(transaction\_date) as 'Month' ,

Sum(transaction\_qty) as Total\_qty

From coffee\_shop

Group by MonthName(transaction\_date);



TOTAL QUANTITY SOLD KPI - MOM DIFFERENCE AND MOM GROWTH

Select Month(transaction\_date) as 'Month' , Sum(transaction\_qty) as Total\_qty ,

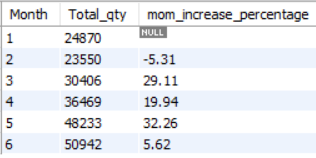
Round((Sum(transaction\_qty) - Lag(Sum(transaction\_qty),1)

Over(Order by Month(transaction\_date))) / lag(Sum(transaction\_qty),1)

Over(Order by Month(transaction\_date)) \* 100,2) as mom\_increase\_percentage

From coffee\_shop

Group by Month(transaction\_date);



CALENDAR TABLE – TOTAL SALES, TOTAL QUANTITY and TOTAL ORDERS

Select Round(Sum(transaction\_qty \* unit\_price),2) as Total\_Sales ,

Sum(transaction\_qty) as Total\_qty\_Sold ,

Count(transaction\_id) as Total\_Order

From coffee\_shop

Where transaction\_date = '2023-05-18'; -- For 18 May 2023



SALES BY WEEKDAY / WEEKEND

Select

Case When DayOfWeek(transaction\_date) IN (1,7) Then 'WeekEnds'

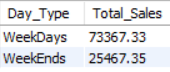
Else 'WeekDays'

End As Day\_Type , Round(Sum(transaction\_qty \* unit\_price),2) as Total\_Sales

From coffee\_shop

Where MonthName(transaction\_date) = 'March' -- March Month Sale

Group By Day\_Type;



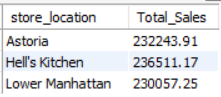
SALES BY STORE LOCATION

Select store\_location , Round(Sum(transaction\_qty \* unit\_price),2) as Total\_Sales

From coffee\_shop

Group by store\_location

Order by store\_location asc;



SALES TREND OVER PERIOD

SELECT AVG(total\_sales) AS average\_sales

FROM (

SELECT

SUM(unit\_price \* transaction\_qty) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5 -- Filter for May

GROUP BY

transaction\_date

) AS internal\_query;

Query Explanation:

 This inner subquery calculates the total sales (unit\_price \* transaction\_qty) for each date in

May. It filters the data to include only transactions that occurred in May by using the

MONTH() function to extract the month from the transaction\_date column and filtering for

May (month number 5).

 The GROUP BY clause groups the data by transaction\_date, ensuring that the total sales are

aggregated for each individual date in May.

 The outer query calculates the average of the total sales over all dates in May. It references

the result of the inner subquery as a derived table named internal\_query.

 The AVG() function calculates the average of the total\_sales column from the derived table,

giving us the average sales for May.



DAILY SALES FOR MONTH SELECTED

SELECT

DAY(transaction\_date) AS day\_of\_month,

ROUND(SUM(unit\_price \* transaction\_qty),1) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

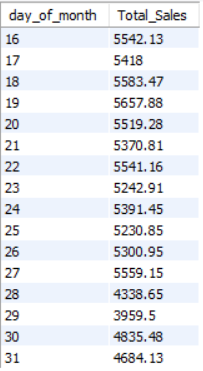
MONTH(transaction\_date) = 5 -- Filter for May

GROUP BY

DAY(transaction\_date)

ORDER BY

DAY(transaction\_date);

COMPARING DAILY SALES WITH AVERAGE SALES – IF GREATER THAN “ABOVE AVERAGE” and

LESSER THAN “BELOW AVERAGE”

Select day\_of\_month , Total\_Sales , Avg\_Sales ,

Case

When Total\_sales > Avg\_sales Then 'Above Average'

When Total\_sales < Avg\_sales Then 'Below Average'

Else 'Average'

End as Sales\_Type

From (

Select Day(transaction\_date) as day\_of\_month ,

Round(Sum(transaction\_qty \* unit\_price),2) as Total\_Sales ,

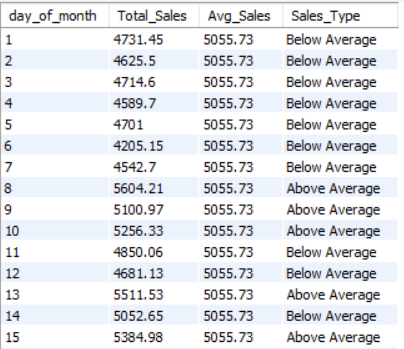
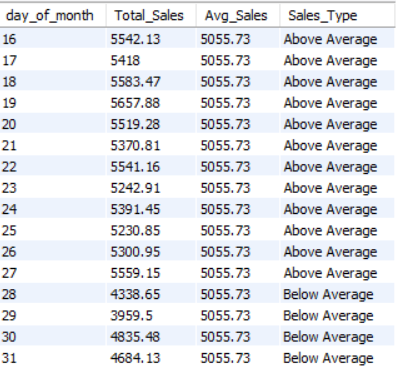
Round(AVG(SUM(unit\_price \* transaction\_qty)) OVER (),2) AS avg\_sales

From coffee\_shop

Where Month(transaction\_date) = 5

Group by Day(transaction\_date) ) x

Order by day\_of\_month ;

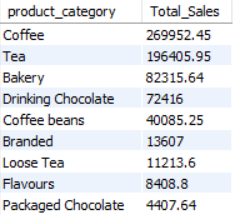
SALES BY PRODUCT CATEGORY

Select product\_category , Round(Sum(transaction\_qty \* unit\_price),2) as Total\_Sales

From coffee\_shop

Group by product\_category

Order by Total\_Sales Desc;



SALES BY PRODUCTS (TOP 10)

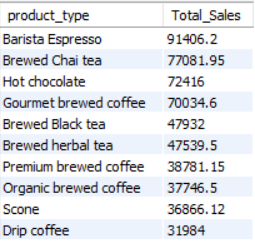
Select product\_type , Round(Sum(transaction\_qty \* unit\_price),2) as Total\_Sales

From coffee\_shop

Group by product\_type

Order by Total\_Sales Desc

Limit 10 ;



SALES BY DAY | HOUR

SELECT

ROUND(SUM(unit\_price \* transaction\_qty)) AS Total\_Sales,

SUM(transaction\_qty) AS Total\_Quantity,

COUNT(\*) AS Total\_Orders

FROM

coffee\_shop\_sales

WHERE

DAYOFWEEK(transaction\_date) = 3 -- Filter for Tuesday (1 is Sunday, 2 is Monday, ..., 7 is Saturday)

AND HOUR(transaction\_time) = 8 -- Filter for hour number 8

AND MONTH(transaction\_date) = 5; -- Filter for May (month number 5)



TO GET SALES FROM MONDAY TO SUNDAY FOR MONTH OF MAY

SELECT

CASE

WHEN DAYOFWEEK(transaction\_date) = 2 THEN 'Monday'

WHEN DAYOFWEEK(transaction\_date) = 3 THEN 'Tuesday'

WHEN DAYOFWEEK(transaction\_date) = 4 THEN 'Wednesday'

WHEN DAYOFWEEK(transaction\_date) = 5 THEN 'Thursday'

WHEN DAYOFWEEK(transaction\_date) = 6 THEN 'Friday'

WHEN DAYOFWEEK(transaction\_date) = 7 THEN 'Saturday'

ELSE 'Sunday'

END AS Day\_of\_Week,

ROUND(SUM(unit\_price \* transaction\_qty), 2) AS Total\_Sales

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5

GROUP BY

CASE

WHEN DAYOFWEEK(transaction\_date) = 2 THEN 'Monday'

WHEN DAYOFWEEK(transaction\_date) = 3 THEN 'Tuesday'

WHEN DAYOFWEEK(transaction\_date) = 4 THEN 'Wednesday'

WHEN DAYOFWEEK(transaction\_date) = 5 THEN 'Thursday'

WHEN DAYOFWEEK(transaction\_date) = 6 THEN 'Friday'

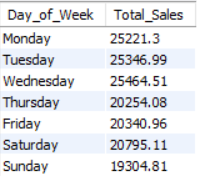
WHEN DAYOFWEEK(transaction\_date) = 7 THEN 'Saturday'

ELSE 'Sunday'

END

ORDER BY

FIELD(Day\_of\_Week, 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday');



TO GET SALES FOR ALL HOURS FOR MONTH OF MAY

SELECT

HOUR(transaction\_time) AS Hour\_of\_Day,

ROUND(SUM(unit\_price \* transaction\_qty)) AS Total\_Sales

FROM

coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5 -- Filter for May (month number 5)

GROUP BY

HOUR(transaction\_time)

ORDER BY

HOUR(transaction\_time);

