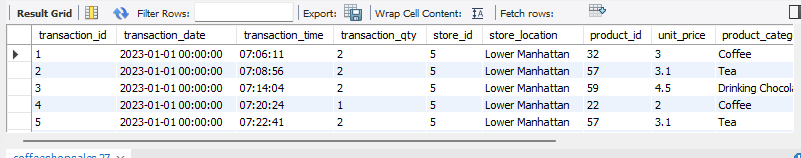
**MY SQL QUERIES**

**COFFEE SHOP SALES PROJECT**

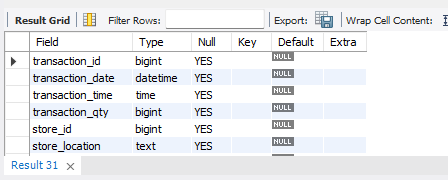
**DATA given**

SELECT \* FROM coffeeshopsales;



**DATA TYPES OF DIFFERENT COLUMNS**

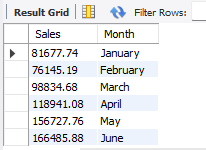
describe coffeeshopsales;



**TOTAL SALES KPI - MOM DIFFERENCE AND MOM GROWTH**

select round(sum(unit\_price\* transaction\_qty),2) as 'Sales', monthname(transaction\_date) as 'Month' from coffeeshopsales

group by monthname(transaction\_date);



**TOTAL SALES KPI - MOM DIFFERENCE AND MOM GROWTH**

with TBL AS (

with tbl as (

select round(sum(unit\_price\* transaction\_qty),2) as 'Sales',

monthname(transaction\_date) as 'Month'

from coffeeshopsales

group by monthname(transaction\_date)

)

select Month, Sales,

LAG(Sales) over() AS previous\_value,

CASE

WHEN Sales > LAG(Sales) over() THEN 'greater'

WHEN Sales = LAG(Sales) over() THEN 'equal'

ELSE 'less'

END AS comparison\_result

from tbl )

SELECT Month,

Sales,

previous\_value,

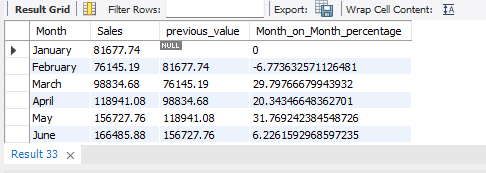
CASE

WHEN Sales is not null and previous\_value is not null THEN ((Sales-previous\_value)/previous\_value)\*100

ELSE 0

END as "Month\_on\_Month\_percentage"

FROM TBL;



**TOTAL ORDERS**

with TBL AS (

with tbl as (

select monthname(transaction\_date) as 'Month',count(transaction\_id) as 'Total\_no\_Orders' from coffeeshopsales

group by monthname(transaction\_date)

)

select Month, Total\_no\_Orders,

LAG(Total\_no\_Orders) over() AS previous\_value,

CASE

WHEN Total\_no\_Orders > LAG(Total\_no\_Orders) over() THEN 'greater'

WHEN Total\_no\_Orders = LAG(Total\_no\_Orders) over() THEN 'equal'

ELSE 'less'

END AS comparison\_result

from tbl )

SELECT Month,

Total\_no\_Orders,

previous\_value,

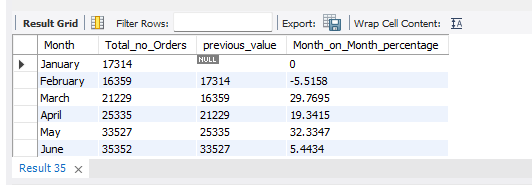
CASE

WHEN Total\_no\_Orders is not null and previous\_value is not null THEN ((Total\_no\_Orders-previous\_value)/previous\_value)\*100

ELSE 0

END as "Month\_on\_Month\_percentage"

FROM TBL;



**TOTAL QUANTITY SOLD KPI - MOM DIFFERENCE AND MOM GROWTH**

with TBL AS (

with tbl as (

select monthname(transaction\_date) as 'Month',sum(transaction\_qty) as 'Total\_no\_qty' from coffeeshopsales

group by monthname(transaction\_date)

)

select Month, Total\_no\_qty,

LAG(Total\_no\_qty) over() AS previous\_value,

CASE

WHEN Total\_no\_qty > LAG(Total\_no\_qty) over() THEN 'greater'

WHEN Total\_no\_qty = LAG(Total\_no\_qty) over() THEN 'equal'

ELSE 'less'

END AS comparison\_result

from tbl )

SELECT Month,

Total\_no\_qty,

previous\_value,

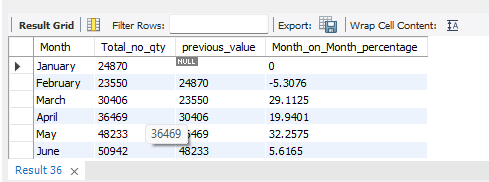
CASE

WHEN Total\_no\_qty is not null and previous\_value is not null THEN ((Total\_no\_qty-previous\_value)/previous\_value)\*100

ELSE 0

END as "Month\_on\_Month\_percentage"

FROM TBL;



**CALENDAR TABLE – DAILY SALES, QUANTITY and TOTAL ORDERS**

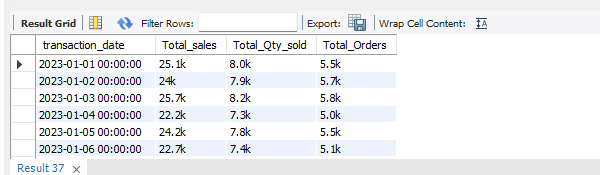
select transaction\_date,concat(round(sum(unit\_price \* transaction\_qty)/100,1), "k") as "Total\_sales",

concat(round(sum(transaction\_qty)/100,1), "k") as "Total\_Qty\_sold",

concat(round(count(transaction\_id)/100,1), "k") as "Total\_Orders"

from coffeeshopsales

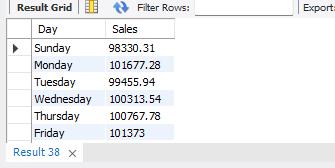
group by transaction\_date;



**Sales made over the weekdays and weekends**

select distinct(dayname(transaction\_date)) as 'Day', round(sum((unit\_price \* transaction\_qty)),2) as 'Sales' from coffeeshopsales

group by Day;



**Weekends sales**

with tbl as (

with tbl as (

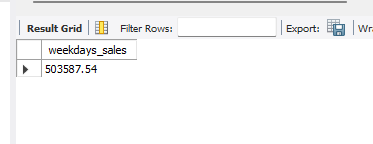
select distinct(dayname(transaction\_date)) as 'Day', round(sum((unit\_price \* transaction\_qty)),2) as 'Sales' from coffeeshopsales

group by Day)

select \* from tbl

where Day not in ('Sunday','Saturday'))

select round(sum(Sales),2) as 'weekdays\_sales' from tbl;



**Weekends sales per month**

with tbl as (

with tbl as (

select distinct(dayname(transaction\_date)) as 'Day',monthname(transaction\_date) as "Months", round(sum((unit\_price \* transaction\_qty)),2) as 'Sales' from coffeeshopsales

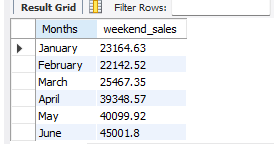
group by Day, monthname(transaction\_date))

select \* from tbl

where Day in ('Sunday','Saturday'))

select Months,round(sum(Sales),2) as 'weekend\_sales' from tbl

group by Months;



**weekdays sales per month**

with tbl as (

with tbl as (

select distinct(dayname(transaction\_date)) as 'Day',monthname(transaction\_date) as "Months", round(sum((unit\_price \* transaction\_qty)),2) as 'Sales' from coffeeshopsales

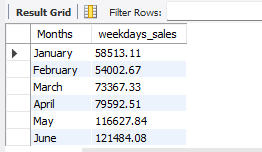
group by Day, monthname(transaction\_date))

select \* from tbl

where Day not in ('Sunday','Saturday'))

select Months,round(sum(Sales),2) as 'weekdays\_sales' from tbl

group by Months;



**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,

CASE

WHEN total\_sales > avg\_sales THEN 'Above Average'

WHEN total\_sales < avg\_sales THEN 'Below Average'

ELSE 'Average'

END AS sales\_status,

total\_sales

FROM (

SELECT

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

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

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

FROM

coffee\_shop\_sales

WHERE

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

GROUP BY

DAY(transaction\_date)

) AS sales\_data

ORDER BY

day\_of\_month;

**SALES BY WEEKDAY / WEEKEND:**

SELECT

CASE

WHEN DAYOFWEEK(transaction\_date) IN (1, 7) THEN 'Weekends'

ELSE 'Weekdays'

END AS day\_type,

ROUND(SUM(unit\_price \* transaction\_qty),2) AS total\_sales

FROM

coffee\_shop\_sales

WHERE

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

GROUP BY

CASE

WHEN DAYOFWEEK(transaction\_date) IN (1, 7) THEN 'Weekends'

ELSE 'Weekdays'

END;



**SALES BY STORE LOCATION**

SELECT

store\_location,

SUM(unit\_price \* transaction\_qty) as Total\_Sales

FROM coffee\_shop\_sales

WHERE

MONTH(transaction\_date) =5

GROUP BY store\_location

ORDER BY SUM(unit\_price \* transaction\_qty) DESC



**SALES BY PRODUCT CATEGORY**

SELECT

product\_category,

ROUND(SUM(unit\_price \* transaction\_qty),1) as Total\_Sales

FROM coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5

GROUP BY product\_category

ORDER BY SUM(unit\_price \* transaction\_qty) DESC



**SALES BY PRODUCTS (TOP 10)**

SELECT

product\_type,

ROUND(SUM(unit\_price \* transaction\_qty),1) as Total\_Sales

FROM coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5

GROUP BY product\_type

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

FROM

coffee\_shop\_sales

WHERE

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



***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);

