**MY SQL QUERIES**

**COFFEE SHOP SALES PROJECT**

* Creating Database

create database coffee\_shop\_sales\_db;

* Using the Database :

use coffee\_shop\_sales\_db;

* DATA TYPES OF DIFFERENT COLUMNS :

describe coffee\_shop\_sales;

****

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

UPDATE coffee\_shop\_sales

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

* ALTERING TIME (transaction\_time) COLUMN TO DATE DATA TYPE :

alter table coffee\_shop\_sales

modify column transaction\_time time;

* CONVERTING DATE (transaction\_date) COLUMN TO PROPER DATE FORMAT :

UPDATE coffee\_shop\_sales

SET transaction\_date = STR\_TO\_DATE(transaction\_date, '%d:%m:%Y');

* ALTERING DATE (transaction\_date) COLUMN TO DATE DATA TYPE

alter table coffee\_shop\_sales

modify column transaction\_date date;

* TOTAL SALES FOR MAY MONTH

select round(sum(unit\_price\*transaction\_qty),1) as Total\_Sales

from coffee\_shop\_sales



* TOTAL SALES KPI - MOM DIFFERENCE AND MOM GROWTH

select month(transaction\_date) as month, -- number of month

round(sum(unit\_price \* transaction\_qty)) as Total\_Sales, -- total sales

(sum(unit\_price \* transaction\_qty) - lag(sum(unit\_price \* transaction\_qty),1) -- Month difference Sales

over(order by month(transaction\_date))) / lag(sum(unit\_price\*transaction\_qty),1) -- division by previous month sales

over(order by month(transaction\_date))\* 100 as mom\_increase\_percentage -- percentage mom=Month on Month

from coffee\_shop\_sales

where month(transaction\_date) in (4,5) -- for month of April(Previous Month) and May(Current Month)

group by month(transaction\_date)

order by month(transaction\_date);



* TOTAL ORDERS

select count(transaction\_id) as Total\_Sales from coffee\_shop\_sales

where month(transaction\_date)=5; -- for March month



* TOTAL ORDERS KPI - MOM DIFFERENCE AND MOM GROWTH

select month(transaction\_date) as month, -- number of month

round(count(transaction\_id)) as Total\_Order, -- total sales

(count(transaction\_id) - lag(count(transaction\_id),1) -- Month difference Sales

over(order by month(transaction\_date))) / lag(count(transaction\_id),1) -- division by previous month sales

over(order by month(transaction\_date))\* 100 as mom\_increase\_percentage -- percentage mom=Month on Month

from coffee\_shop\_sales

where month(transaction\_date) in (4,5) -- for month of April(Previous Month) and May(Current Month)

group by month(transaction\_date)

order by month(transaction\_date);



* TOTAL QUANTITY SOLD

select sum(transaction\_qty) as Total\_Quantity from coffee\_shop\_sales

where month(transaction\_date)=5; -- MAY month



* TOTAL QUANTITY SOLD KPI - MOM DIFFERENCE AND MOM GROWTH

select month(transaction\_date) as month,

round(sum(transaction\_qty)) as total\_qty\_sold,

(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 mom\_increased\_percentage

from coffee\_shop\_sales

where month(transaction\_date) in (4,5)

group by month(transaction\_date)

order by month(transaction\_date);



* CALENDAR TABLE – DAILY SALES, QUANTITY and TOTAL ORDERS WITH Rounded off values

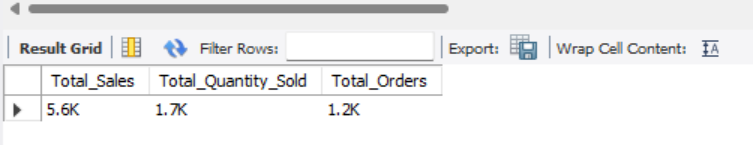
select concat(round(sum(unit\_price \* transaction\_qty)/1000,1),'K') as Total\_Sales,

concat(round(sum(transaction\_qty)/1000,1),'K') as Total\_Quantity\_Sold,

concat(round(count(transaction\_id)/1000,1),'K') as Total\_Orders

from coffee\_shop\_sales

where transaction\_date = '2023-05-18';



* SALES BY WEEKDAY / WEEKEND:

select

case when dayofweek(transaction\_date) in (1,7) then 'Weekends'

else 'Weekdays'

end as 'Day\_type' ,

concat(round(sum(unit\_price \* transaction\_qty)/1000,1),'K') as Total\_Sales

from coffee\_shop\_sales

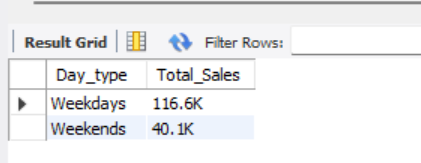
where month(transaction\_date)=5 -- MAY Month

group by

case when dayofweek(transaction\_date) in (1,7) then 'Weekends'

else 'Weekdays'

end;



* SALES BY STORE LOCATION

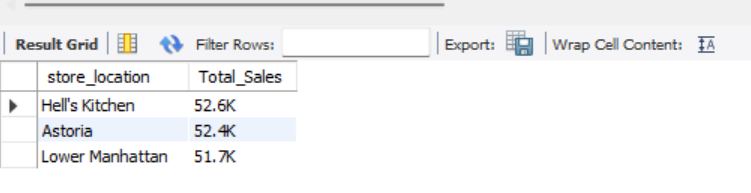
select store\_location,concat(round(sum(unit\_price \* transaction\_qty)/1000,1),'K') as Total\_Sales

from coffee\_shop\_sales

where month(transaction\_date) = 5 -- May Month

group by store\_location

order by sum(unit\_price \* transaction\_qty) desc;



* SALES TREND OVER PERIOD

select concat(round(avg(total\_sales)/1000,1),'K') as Avg\_Sales

from(

select sum(unit\_price \* transaction\_qty) as total\_sales

from coffee\_shop\_sales

where month(transaction\_date) = 5 -- May Month

group by transaction\_date

) as Internal\_query;



* DAILY SALES FOR MONTH SELECTED

select day(transaction\_date) as Day\_of\_Month,

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

from coffee\_shop\_sales

where month(transaction\_date) =5 – MAY Month

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 -- May Month

group by day(transaction\_date)

order by day(transaction\_date)

) as sales\_data

order by day\_of\_month;





* SALES BY PRODUCT CATEGORY

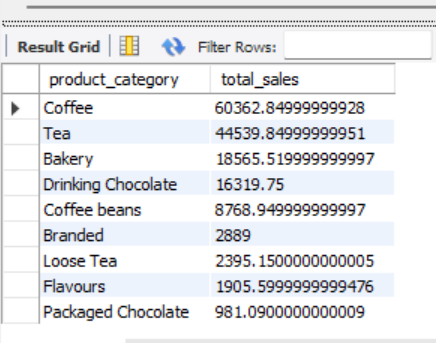
select product\_category, sum(unit\_price \* transaction\_qty) as total\_sales

from coffee\_shop\_sales

where month(transaction\_date)=5

group by product\_category

order by total\_sales desc;



* SALES BY PRODUCTS (TOP 10)

select product\_type, sum(unit\_price \* transaction\_qty) as total\_sales

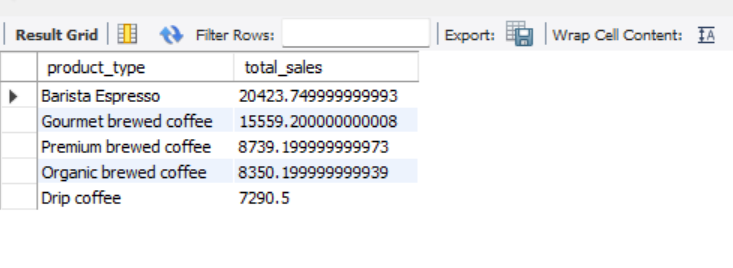
from coffee\_shop\_sales

where month(transaction\_date)=5 and product\_category='Coffee'

group by product\_type

order by total\_sales desc

limit 10;



* SALES BY DAY | HOUR

select sum(unit\_price \* transaction\_qty) as total\_sales,

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

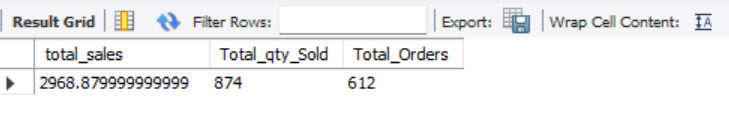
count(\*) as Total\_Orders

from coffee\_shop\_sales

where month(transaction\_date) = 5 -- May month

and dayofweek(transaction\_date) = 3 -- Monday

and hour(transaction\_time) = 8; -- hour no 8



* TO GET SALES FOR ALL HOURS FOR MONTH OF MAY

select hour(transaction\_time) as hours\_of\_day,

sum(unit\_price \* transaction\_qty) as total\_sales

from coffee\_shop\_sales

where month(transaction\_date) = 5

group by hour(transaction\_time)

order by hour(transaction\_time);



* 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;

