**MY SQL QUERIES**

**COFFEE SHOP SALES PROJECT**

**Creating the table coffee\_shop and loading the data in table from csv file**

create table coffee\_shop(

transaction\_id int primary key,

transaction\_date date,

transaction\_time time,

transaction\_qty float,

store\_id int,

store\_location varchar(50),

product\_id int,

unit\_price float,

product\_category varchar(50),

product\_type varchar(50),

product\_detail varchar(50)

);

copy coffee\_shop

from 'D:\Coffee Shop Analysis(SQL + Power BI)\Coffee Shop Sales.csv'

delimiter ','

csv header

**TOTAL SALES**

ALTER TABLE coffee\_shop ADD COLUMN Total\_Sales float;

update coffee\_shop

set Total\_sales=transaction\_qty \* unit\_price;

SELECT ROUND(SUM(Total\_Sales)) as Total\_Sales

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 5 -- for month of (CM-May)



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

SELECT

EXTRACT(MONTH FROM transaction\_date) AS Month,

round(SUM(Total\_Sales)) AS Total\_Sales,

(SUM(Total\_Sales) - LAG(SUM(Total\_Sales),1) OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date) asc)) /

LAG(SUM(Total\_Sales)) OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date)) \* 100 AS mom\_increase\_percentage

FROM

coffee\_shop

WHERE

EXTRACT(MONTH FROM transaction\_date) IN (4, 5)

GROUP BY

EXTRACT(MONTH FROM transaction\_date),

TO\_CHAR(transaction\_date, 'Month')

ORDER BY

EXTRACT(MONTH FROM transaction\_date);



**TOTAL ORDERS**

SELECT COUNT(transaction\_id) as Total\_Orders

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date)= 5 -- for month of (CM-May)



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

SELECT

EXTRACT(MONTH FROM transaction\_date) AS Month,

count(transaction\_id) AS Total\_Orders,

cast(((count(transaction\_id) - LAG(count(transaction\_id),1)

OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date)))\*1.0

/LAG(count(transaction\_id),1) OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date)))\* 100.00 as decimal(10,2)) AS mom\_increase\_percentage

FROM

coffee\_shop

WHERE

EXTRACT(MONTH FROM transaction\_date) IN (4, 5)

GROUP BY

EXTRACT(MONTH FROM transaction\_date),

TO\_CHAR(transaction\_date, 'Month')

ORDER BY

EXTRACT(MONTH FROM transaction\_date);



**TOTAL QUANTITY SOLD**

SELECT SUM(transaction\_qty) as Total\_Quantity\_Sold

FROM coffee\_shop\_sales

WHERE EXTRACT(MONTH FROM transaction\_date) = 5 -- for month of (CM-May)



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

SELECT

EXTRACT(MONTH FROM transaction\_date) AS month,

ROUND(SUM(transaction\_qty)) AS total\_quantity\_sold,

cast((SUM(transaction\_qty) - LAG(SUM(transaction\_qty), 1)

OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date)))\*1.0

/ LAG(SUM(transaction\_qty), 1) OVER (ORDER BY EXTRACT(MONTH FROM transaction\_date))) \* 100 as decimal(10,4))

AS mom\_increase\_percentage

FROM

coffee\_shop

WHERE

EXTRACT(MONTH FROM transaction\_date) IN (4, 5) -- for April and May

GROUP BY

EXTRACT(MONTH FROM transaction\_date)

ORDER BY

EXTRACT(MONTH FROM transaction\_date);



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

select \* from coffee\_shop;

select count(transaction\_id) as total\_orders,

sum(transaction\_qty) as total\_quantity,

sum(total\_sales) as total\_price

from coffee\_shop

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



***If you want to get exact Rounded off values then use below query to get the result:***

select concat(cast(count(transaction\_id)\*1.0/1000 as decimal(10,1)),'K') as total\_orders,

concat(cast(sum(transaction\_qty)/1000 as decimal(10,1)),'K') as total\_quantity,

concat(cast(sum(total\_sales)/1000 as decimal(10,1)),'K') as total\_price

from coffee\_shop

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



**SALES TREND OVER PERIOD**

select avg(totals) as average\_sales

from

(

select transaction\_date,sum(total\_sales) as totals

from coffee\_shop

where extract(month from transaction\_date)=5

group by transaction\_date

) as innerquery



**DAILY SALES FOR MONTH SELECTED**

SELECT

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

ROUND(SUM(total\_sales),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(total\_sales) 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(total\_sales),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(total\_sales) 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(total\_sales),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(total\_sales),1) as Total\_Sales

FROM coffee\_shop\_sales

WHERE

MONTH(transaction\_date) = 5

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)) 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);

