**COFFEE SHOP SALES ANALYSIS**

* **CREATE DATABASE**

Create database if not exists coffee\_sales;

* **SEEING TABLES**

select \* from coffeesales;

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

update coffeesales

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

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

alter TABLE coffeesales

modify COLUMN transaction\_date DATE;

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

update coffeesales

set transaction\_time = str\_to\_date(transaction\_time , '%H:%i:%s');

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

alter table coffeesales

modify column transaction\_time time;

* **DATA TYPES OF DIFFERENT COLUMNS**

desc coffeesales;

****

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

alter table coffeesales

change column ï»¿transaction\_id transaction\_id int;

* **TOTAL SALES Analysis**

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

from coffeesales

where

month(transaction\_date) = 5;



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

SELECT

MONTH(transaction\_date) AS month,

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

(SUM(unit\_price \* transaction\_qty) - LAG(SUM(unit\_price \* transaction\_qty), 1)

OVER (ORDER BY MONTH(transaction\_date))) / LAG(SUM(unit\_price \* transaction\_qty), 1)

OVER (ORDER BY MONTH(transaction\_date)) \* 100 AS mom\_increase\_percentage

FROM coffeesales

WHERE

MONTH(transaction\_date) IN (4, 5) -- for months of April and May

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);



* **Total Orders Analysis**

select count(transaction\_id) as Total\_Orders

from coffeesales

where month(transaction\_date) = 5;



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

SELECT

MONTH(transaction\_date) AS month,

ROUND(count(transaction\_id)) AS total\_orders,

(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 AS mom\_increase\_percentage

FROM

coffeesales

WHERE

MONTH(transaction\_date) IN (4, 5) -- for months of April and May

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);



* **Total Quantity sold**

select sum(transaction\_qty) as Total\_Quantity\_Sold

from coffeesales

where month(transaction\_date) = 5;



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

SELECT

MONTH(transaction\_date) AS month,

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

(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\_increase\_percentage

FROM

coffeesales

WHERE

MONTH(transaction\_date) IN (4, 5) -- for months of April and May

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);



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

SELECT

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

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

COUNT(transaction\_id) AS total\_orders

from coffeesales

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,

round(sum(unit\_price \* transaction\_qty),2)as total\_sales

from coffeesales

where month(transaction\_date) = 5

group by day\_type;



* **SALES BY STORE LOCATION**

select , store\_location ,

concat(round(sum(unit\_price \* transaction\_qty)/1000 , 1),"k") as Total\_sales

from coffeesales

where month(transaction\_date) = 5

group by store\_location

order by Total\_sales desc;



* **SALES TREND OVER PERIOD**

SELECT concat(round(AVG(Total\_Sales)/1000,1),"k") AS average\_sales

FROM (

SELECT

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

FROM

coffeesales

WHERE

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

GROUP BY

transaction\_date

) AS internal\_query;



* **DAILY SALES FOR MONTH SELECTED**

SELECT

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

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

FROM

coffeesales

WHERE

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

GROUP BY

day\_of\_month

ORDER BY

day\_of\_month;

 

* ***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

coffeesales

WHERE

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

GROUP BY

DAY(transaction\_date)

) AS sales\_data

ORDER BY

day\_of\_month;

 

* **SALES BY PRODUCT CATEGORY**

SELECT

product\_category,

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

FROM coffeesales

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 coffeesales

WHERE

MONTH(transaction\_date) = 5

GROUP BY product\_type

ORDER BY Total\_Sales DESC

LIMIT 10;



* ***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

coffeesales

WHERE

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

WHERE

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

GROUP BY Day\_of\_Week;

