**COFFEE\_SHOP\_SALES PROJECT**

**OBJECTIVES: Import, Clean, Analyze and Visualize Dataset**

**QUERIES:**

SELECT \*

FROM coffeeshop\_sales;-- After checking, I saw I needed to do some data cleaning

**DATA CLEANING**

DESCRIBE coffeeshop\_sales;-- From the result , I had the datatype of the transaction date & time column as TEXT instead of DATE and TIME



UPDATE coffeeshop\_sales

SET transaction\_date = str\_to\_date(transaction\_date, '%m/%d/%Y');

ALTER TABLE coffeeshop\_sales

MODIFY COLUMN transaction\_date DATE;

UPDATE coffeeshop\_sales

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

ALTER TABLE coffeeshop\_sales

MODIFY COLUMN transaction\_time TIME;

ALTER TABLE coffeeshop\_sales

CHANGE COLUMN ï»¿transaction\_id transaction\_id INT;-- DATA CLEANING DONE!!!

**KPIs**

**-- 1. TOTAL SALES**

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

FROM coffeeshop\_sales;

/\* WHERE

MONTH(transaction\_date) = 5;(Add the where clause to filter by months)\*/



**-- 2.MONTH ON MONTH DIFFERENCE AND MONTH ON MONTH GROWTH IN RESPECT TO SALES**

SELECT

MONTH(transaction\_date) AS month,-- Number of Month

ROUND(SUM(unit\_price \* transaction\_qty)) AS total\_sales,-- Total Sales Column

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

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

FROM

coffeeshop\_sales

WHERE

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

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);



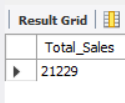
**-- 3.TOTAL ORDERS PER MONTH**

SELECT COUNT(transaction\_id) Total\_Sales

FROM coffeeshop\_sales

WHERE

MONTH(transaction\_date) = 3;-- (Add the where clause to filter by months)



**-- 4.MONTH ON MONTH DIFFERENCE AND MONTH ON MONTH GROWTH IN RESPECT TO ORDERS**

SELECT

MONTH(transaction\_date) AS month,

ROUND(SUM(transaction\_qty)) AS total\_quantity\_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\_increase\_percentage

FROM

coffeeshop\_sales

WHERE

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

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);



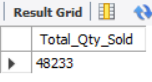
**-- 5.TOTAL QUANTITY SOLD PER MONTH**

SELECT SUM(transaction\_qty) Total\_Qty\_Sold

FROM coffeeshop\_sales

WHERE

MONTH(transaction\_date) = 5;-- (Add the where clause to filter by months)



**-- 6.MONTH ON MONTH DIFFERENCE AND MONTH ON MONTH GROWTH IN RESPECT TO QUANTITIES SOLD**

SELECT

MONTH(transaction\_date) AS month,

ROUND(SUM(transaction\_qty)) AS total\_quantity\_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\_increase\_percentage

FROM

coffeeshop\_sales

WHERE

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

GROUP BY

MONTH(transaction\_date)

ORDER BY

MONTH(transaction\_date);



**-- 7.CALENDAR TABLE: DAILY SALES,QUANTITY SOLD AND ORDERS**

SELECT

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

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

COUNT(transaction\_id) AS total\_orders

FROM

coffee\_shop\_sales

WHERE

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



***To get exact Rounded off values then:***

SELECT

CONCAT(ROUND(SUM(unit\_price \* transaction\_qty) / 1000, 1),'K') AS total\_sales,

CONCAT(ROUND(COUNT(transaction\_id) / 1000, 1),'K') AS total\_orders,

CONCAT(ROUND(SUM(transaction\_qty) / 1000, 1),'K') AS total\_quantity\_sold

FROM

coffee\_shop\_sales

WHERE

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



**-- 8.SALES OVER WEEKDAYS AND WEEKENDS**

SELECT

CASE

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

ELSE 'Weekdays'

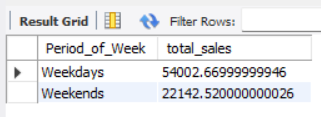
END AS Period\_of\_Week,

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

FROM coffeeshop\_sales

WHERE MONTH(transaction\_date) = 2 -- FEBRUARY

GROUP BY Period\_of\_Week;



**-- 9.SALES BY STORE LOCATION**

SELECT

store\_location,

CONCAT(ROUND(SUM(unit\_price \* transaction\_qty)/1000,1),'K') as Total\_Sales

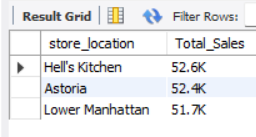
FROM coffeeshop\_sales

WHERE

MONTH(transaction\_date) = 5

GROUP BY store\_location

ORDER BY Total\_Sales DESC;



**-- 10.DAILY AVERAGE SAE PERFORMANCE**

SELECT AVG(total\_sales) AS avg\_sales

FROM (

SELECT

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

FROM

coffeeshop\_sales

WHERE

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

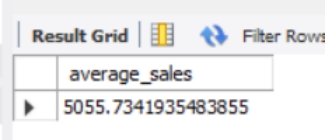
GROUP BY

transaction\_date

) AS AVGSales;

***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.



-- 11.DAILY SALES PER MONTH

SELECT

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

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

FROM

coffeeshop\_sales

WHERE

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

GROUP BY day\_of\_month

ORDER BY day\_of\_month;



-- 12.DAILY SALES VS 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

coffeeshop\_sales

WHERE

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

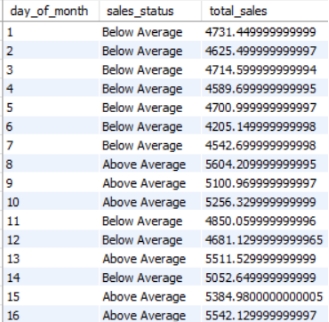
GROUP BY

DAY(transaction\_date)

) AS sales\_data

ORDER BY

day\_of\_month;

**-- 13.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;



**-- 13.SALES BY PRODUCT TYPE**

SELECT

product\_type,

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

FROM coffeeshop\_sales

WHERE

MONTH(transaction\_date) = 5

GROUP BY product\_type

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

LIMIT 10;



**-- 14.SALES BY HOUR**

SELECT

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

SUM(transaction\_qty) AS Total\_Quantity,

COUNT(\*) AS Total\_Orders

FROM

coffeeshop\_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)



**-- 15.TOTAL SALES BY WEEKDAYS**

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

coffeeshop\_sales

WHERE

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

GROUP BY Day\_of\_Week;



**-- 16.SALES FOR ALL HOURS FOR A SELECTED MONTH**

SELECT

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

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

FROM

coffeeshop\_sales

WHERE

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

GROUP BY

HOUR(transaction\_time)

ORDER BY

HOUR(transaction\_time);

