SELECT TOP (1000) [transactions\_id]

,[sale\_date]

,[sale\_time]

,[customer\_id]

,[gender]

,[age]

,[category]

,[quantiy]

,[price\_per\_unit]

,[cogs]

,[total\_sale]

FROM [Retail\_Sales].[dbo].[Retail\_Sales\_Data]

SELECT COUNT(\*) Total\_Records FROM Retail\_Sales\_Data

-- Data Cleaning

SELECT \* FROM Retail\_Sales\_Data

WHERE transactions\_id IS NULL

SELECT \* FROM Retail\_Sales\_Data

WHERE [sale\_date] IS NULL

SELECT \* FROM Retail\_Sales\_Data

WHERE [sale\_time] IS NULL

-- We can Check the Nulls by this basic way or by Using OR operator, Also we can use CASE WHEN method to do so

-- Detecting Nulls using OR operator

SELECT \* FROM Retail\_Sales\_Data

WHERE [customer\_id] IS NULL

OR [gender] IS NULL

OR [age] IS NULL

OR [category] IS NULL

OR [quantiy] IS NULL

OR [price\_per\_unit] IS NULL

OR [cogs] IS NULL

OR [total\_sale] IS NULL

-- Detecting Nulls using Case when function

SELECT \*

FROM Retail\_Sales\_Data

WHERE

CASE

WHEN [customer\_id] IS NULL THEN 1

WHEN [gender] IS NULL THEN 1

WHEN [age] IS NULL THEN 1

WHEN [category] IS NULL THEN 1

WHEN [quantiy] IS NULL THEN 1

WHEN [price\_per\_unit] IS NULL THEN 1

WHEN [cogs] IS NULL THEN 1

WHEN [total\_sale] IS NULL THEN 1

ELSE 0

END = 1;

-- Now We can delete those Rows having Nulls using Delete function

DELETE FROM Retail\_Sales\_Data

WHERE

[customer\_id] IS NULL

OR [gender] IS NULL

OR [age] IS NULL

OR [category] IS NULL

OR [quantiy] IS NULL

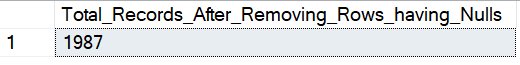
OR [price\_per\_unit] IS NULL

OR [cogs] IS NULL

OR [total\_sale] IS NULL;

-- Now the Count of Records is

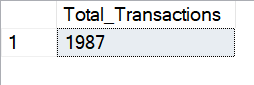
SELECT COUNT(\*) Total\_Records\_After\_Removing\_Rows\_having\_Nulls FROM Retail\_Sales\_Data



--Data Exploration

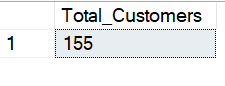
-- How many Transaction we have?

Select COUNT(transactions\_id) Total\_Transactions from Retail\_Sales\_Data



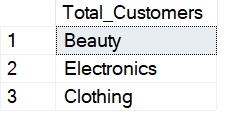
-- How many Unique Customers we have?

Select COUNT(DISTINCT customer\_id) Total\_Customers from Retail\_Sales\_Data



-- What are the Unique Categories?

SELECT DISTINCT category Total\_Customers from Retail\_Sales\_Data

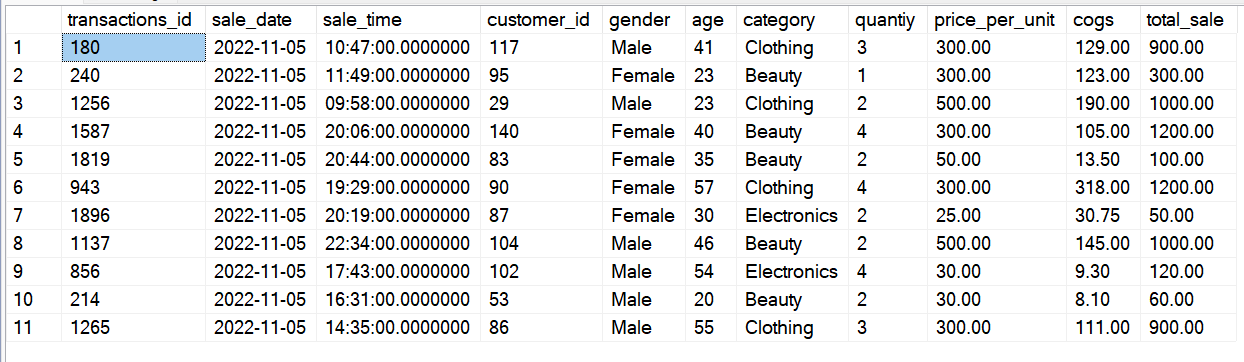


-- Data analysis and Solution of Business Problems

-- Q.1 Write a SQL query to retrieve all columns for sales made on '2022-11-05

SELECT \* FROM Retail\_Sales\_Data

WHERE sale\_date = '2022-11-05'

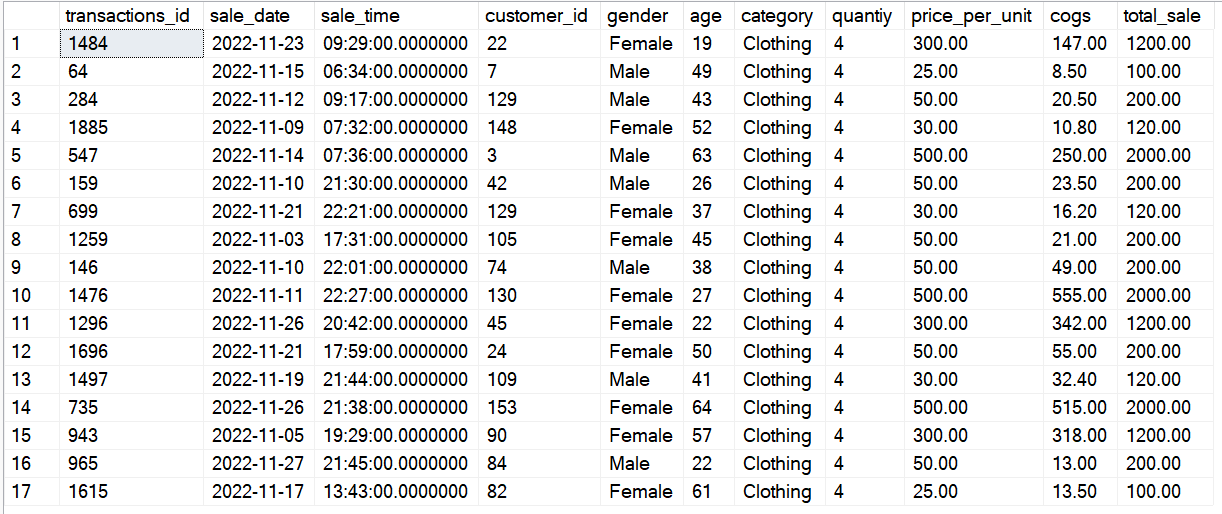


-- Q.2 Write a SQL query to retrieve all transactions where the category is 'Clothing'

-- and the quantity sold is more than 10 in the month of Nov-2022

SELECT \* FROM Retail\_Sales\_Data

WHERE category = 'Clothing' AND FORMAT(sale\_date,'MMM yyyy') = 'Nov 2022' and quantiy >= 4



-- Q.3 Write a SQL query to calculate the total sales (total\_sale) for each category.

SELECT category,

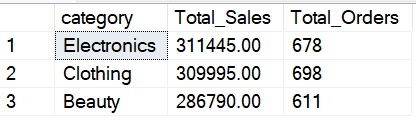
SUM(total\_sale) as Total\_Sales,

COUNT(\*) Total\_Orders

FROM Retail\_Sales\_Data

GROUP BY category

ORDER BY SUM(total\_sale) DESC



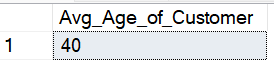
-- Q.4 Write a SQL query to find the average age of customers who purchased items from the 'Beauty' category.

SELECT

AVG(age) as Avg\_Age\_of\_Customer

FROM Retail\_Sales\_Data

WHERE category = 'Beauty'

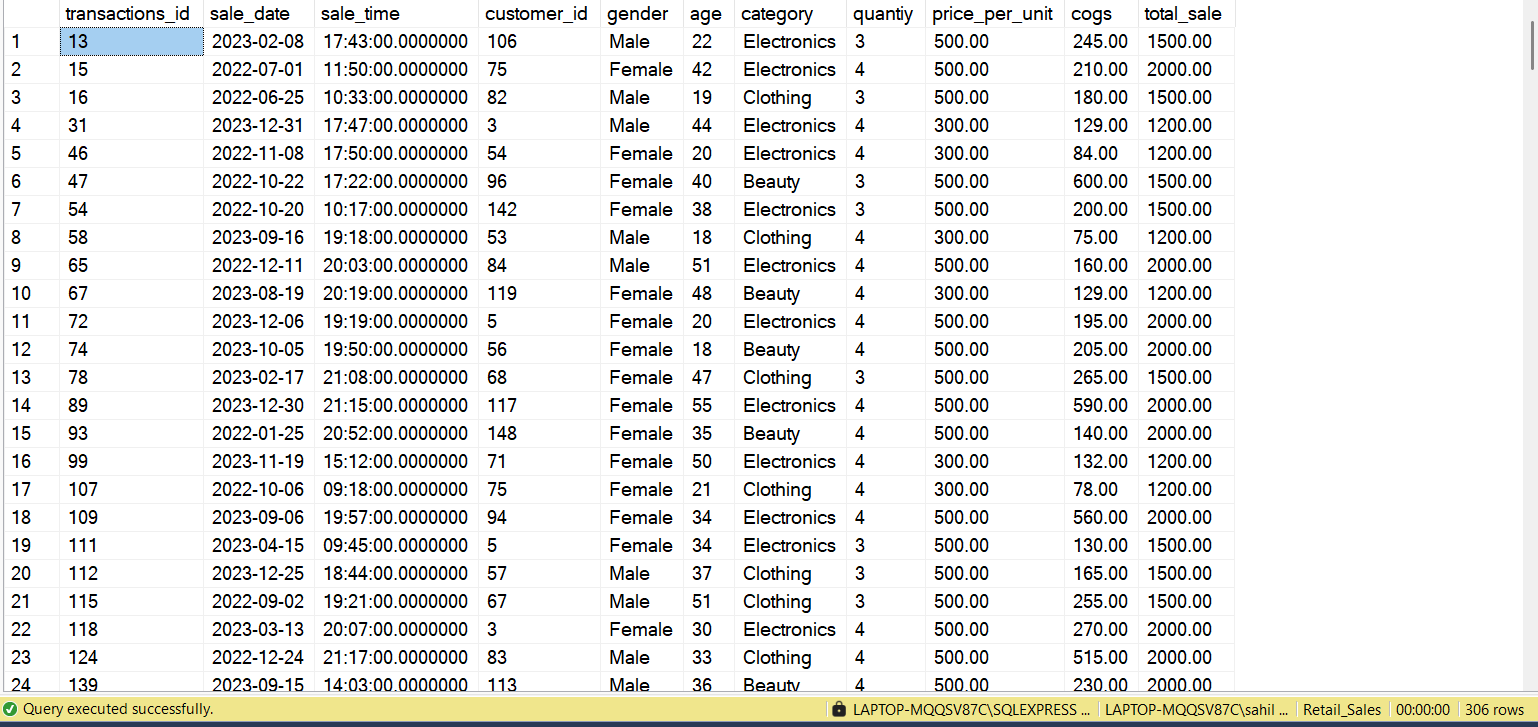


-- Q.5 Write a SQL query to find all transactions where the total\_sale is greater than 1000.

SELECT \* FROM Retail\_Sales\_Data

WHERE total\_sale > 1000

ORDER BY transactions\_id



-- Q.6 Write a SQL query to find the total number of transactions (transaction\_id) made by each gender in each category.

SELECT

category,

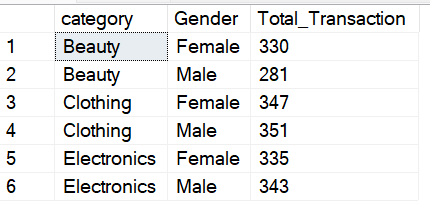
gender as Gender,

COUNT(\*) Total\_Transaction

FROM Retail\_Sales\_Data

GROUP BY category, gender

ORDER BY 1



-- Q.7 Write a SQL query to calculate the average sale for each month. Find out best selling month in each year

SELECT

FORMAT(sale\_date,'yyyy') as Years,

FORMAT(sale\_date,'MMMM') Month\_Name,

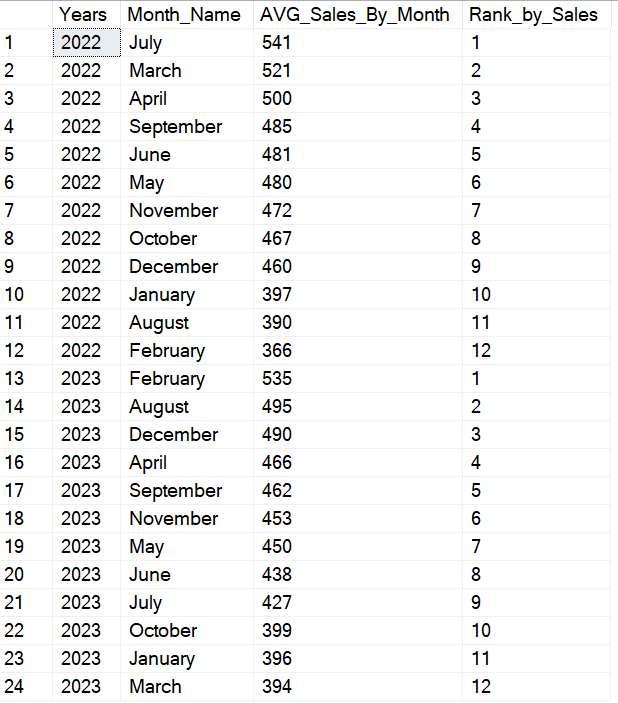
AVG(CAST(total\_sale AS INT)) AS AVG\_Sales\_By\_Month,

RANK() OVER(PARTITION BY FORMAT(sale\_date,'yyyy') ORDER BY AVG(CAST(total\_sale AS INT)) DESC) AS Rank\_by\_Sales

FROM Retail\_Sales\_Data

GROUP BY FORMAT(sale\_date,'yyyy'),

FORMAT(sale\_date,'MMMM')



-- Best Selling Month in Each Year

SELECT

Years,

Month\_Name,

AVG\_Sales\_By\_Month

FROM (SELECT

FORMAT(sale\_date,'yyyy') as Years,

FORMAT(sale\_date,'MMMM') Month\_Name,

AVG(CAST(total\_sale AS INT)) AS AVG\_Sales\_By\_Month,

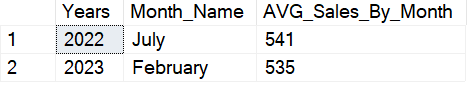
RANK() OVER(PARTITION BY FORMAT(sale\_date,'yyyy') ORDER BY AVG(CAST(total\_sale AS INT)) DESC) AS Rank\_by\_Avg\_Sales

FROM Retail\_Sales\_Data

GROUP BY FORMAT(sale\_date,'yyyy'),

FORMAT(sale\_date,'MMMM') )t

WHERE Rank\_by\_Avg\_Sales = 1



-- Q.8 Write a SQL query to find the top 5 customers based on the highest total sales

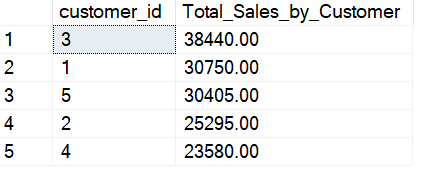
SELECT TOP 5 customer\_id,

SUM(total\_sale) as Total\_Sales\_by\_Customer

FROM Retail\_Sales\_Data

GROUP BY customer\_id

ORDER BY 2 DESC



-- Q.9 Write a SQL query to find the number of unique customers who purchased items from each category.

SELECT

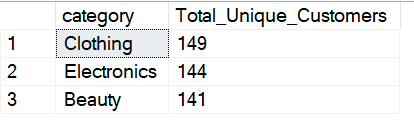
category,

COUNT(DISTINCT customer\_id) Total\_Unique\_Customers

FROM Retail\_Sales\_Data

GROUP BY category

ORDER BY 2 DESC



-- Q.10 Write a SQL query to create each shift and number of orders (Example Morning <=12, Afternoon Between 12 & 17, Evening >17 and rest is of Night)

SELECT

CASE

WHEN DATEPART(HH,sale\_time) < 12 THEN 'Morning'

WHEN DATEPART(HH,sale\_time) >= 12 AND DATEPART(HH,sale\_time) < 17 THEN 'Afternoon'

WHEN DATEPART(HH,sale\_time) >= 17 AND DATEPART(HH,sale\_time) < 21 THEN 'Evening'

ELSE 'Night'

END AS Time\_Shift, COUNT(\*) Total\_Transaction\_Orders

FROM Retail\_Sales\_Data

Group by

CASE

WHEN DATEPART(HH,sale\_time) < 12 THEN 'Morning'

WHEN DATEPART(HH,sale\_time) >= 12 AND DATEPART(HH,sale\_time) < 17 THEN 'Afternoon'

WHEN DATEPART(HH,sale\_time) >= 17 AND DATEPART(HH,sale\_time) < 21 THEN 'Evening'

ELSE 'Night'

END

ORDER BY 2 DESC

