**Retail Sales Analysis**

**Objective –**

The primary objective of this project is to set up and populate a retail sales database using the provided sales data. This involves creating the necessary database structure and ensuring accurate storage of all relevant sales data. Additionally, the project focuses on conducting exploratory data analysis (EDA) to gain deeper insights into the dataset, identifying key patterns and trends. The final goal is to perform business analysis by leveraging SQL queries in Python to visualize the data, answer specific business questions, and uncover insights that can guide strategic decision-making.

**Project Structure –**

* **Importing librabries**

import pandas as pd

import numpy as np

import matplotlib as plt

import seaborn as sns

import mysql.connector

from datetime import datetime

import warnings

warnings.filterwarnings('ignore')

* **Database Connection**

import mysql.connector

conn = mysql.connector.connect(

host="localhost",

user="root",

password="Suryakant@3",

database="retail\_sales"

)

* **Check Tables**

tables = pd.read\_sql\_query("SHOW TABLES;",conn)

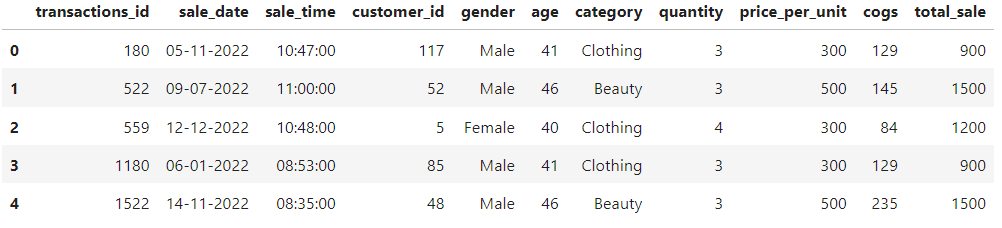
print(tables)



print(tables.columns)

for table in tables [tables.columns[0]]:

display(pd.read\_sql\_query(f"select \* from {table} limit 5", conn))



**Data Analysis**

* **Find Total Number of Sales**

pd.read\_sql\_query("select count(\*) as total\_sale

from retail\_sales", conn)



* **Find different customer id’s or total customers**

pd.read\_sql\_query("Select count(distinct customer\_id) as total\_customers

from retail\_sales", conn)



* **Find Average Age of customers**

pd.read\_sql\_query("Select Round(avg(age), 2) as Avg\_age from retail\_sales", conn)



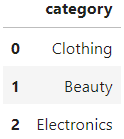
* **Find all transactions where the total sale is greater than 1000**

pd.read\_sql\_query("Select\* from retail\_sales where total\_sale > 1000", conn)



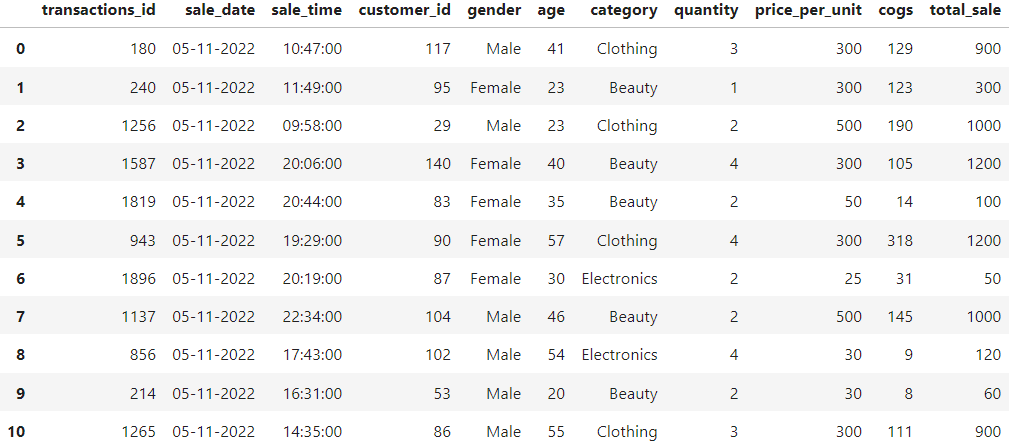
* **Find the number of unique customers who purchased items from each category.**

pd.read\_sql\_query("Select distinct category from retail\_sales", conn)



* **Find all columns for sales made on '2022-11-05.**

pd.read\_sql\_query("select \* from retail\_sales where sale\_date = '05-11-2022'", conn)



* **Create a pie chart for Net Sales by Category.**

pd.read\_sql\_query("SELECT category, SUM(total\_sale) AS net\_sales FROM retail\_sales GROUP BY category", conn)

# Create the pie chart

labels = df['category']

sizes = df['net\_sales']

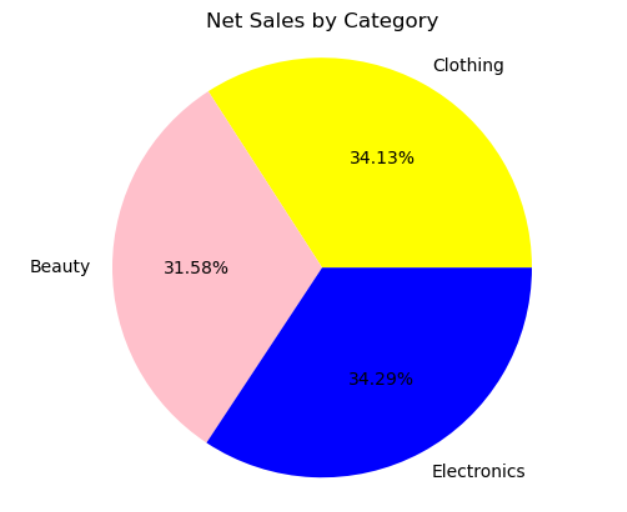
colors = ["yellow", "pink", "blue"]

plt.pie(sizes, colors=colors, labels=labels, autopct="%.2f%%")

plt.title("Net Sales by Category")

plt.axis('equal')

plt.show()

****

* **Create a pie chart for Net Sales by Gender**

pd.read\_sql\_query(

"SELECT gender, SUM(total\_sale) as net\_sales "

"FROM retail\_sales "

"GROUP BY gender",

conn

)

# Create Pie chart

labels = df['gender']

sizes = df['net\_sales']

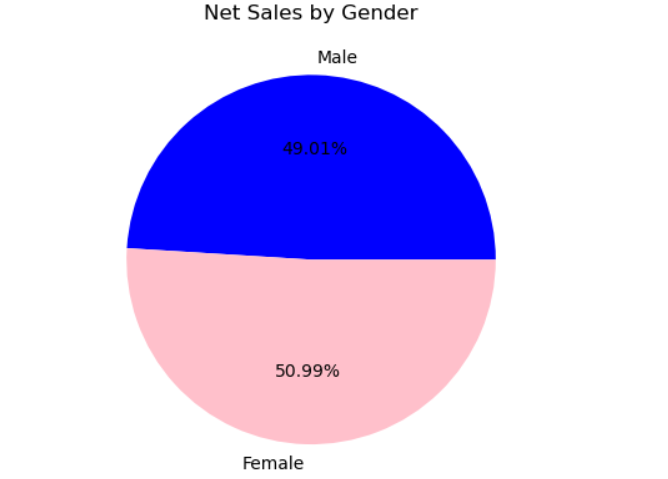
colors = ["blue", "pink"]

plt.axis('equal')

plt.title("Net Sales by Gender")

plt.pie(sizes, colors=colors, labels=labels, autopct="%.2f%%")

plt.show()

****

* **Create bar chart total sales in each category by gender.**

pd.read\_sql\_query("SELECT category, gender, COUNT(\*) AS total\_trans FROM retail\_sales GROUP BY category, gender ORDER BY category", conn)

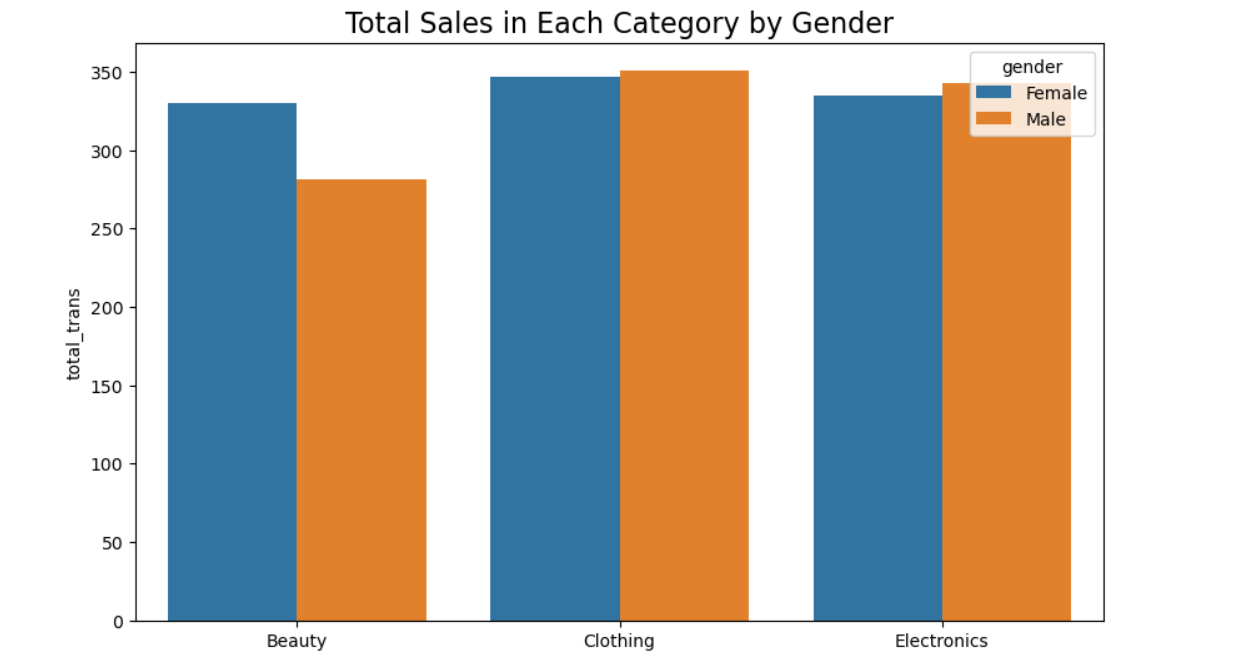
# Create the bar plot

plt.figure(figsize=(10, 6)) # Ensure plt.figure is correctly called

sns.barplot(data=df, x='category', y='total\_trans', hue='gender')

plt.title('Total Sales in Each Category by Gender', fontsize=16)

plt.show()



* **Create bar chart of total sales by age group**

query = """

WITH age\_group\_sales AS (

SELECT \*,

CASE

WHEN age BETWEEN 18 AND 25 THEN '18-25'

WHEN age BETWEEN 26 AND 35 THEN '26-35'

WHEN age BETWEEN 36 AND 45 THEN '36-45'

WHEN age BETWEEN 46 AND 55 THEN '46-55'

WHEN age > 55 THEN '55+'

ELSE 'Unknown'

END AS age\_group

FROM retail\_sales

)

SELECT age\_group, COUNT(\*) as total\_orders

FROM age\_group\_sales

GROUP BY age\_group

ORDER BY age\_group;

"""

df = pd.read\_sql\_query(query, conn)

# Create a bar plot for total orders by age group

plt.figure(figsize=(8, 6))

plt.bar(df['age\_group'], df['total\_orders'], color='teal', width=0.6)

plt.title('Total Orders by Age Group', fontsize=16)

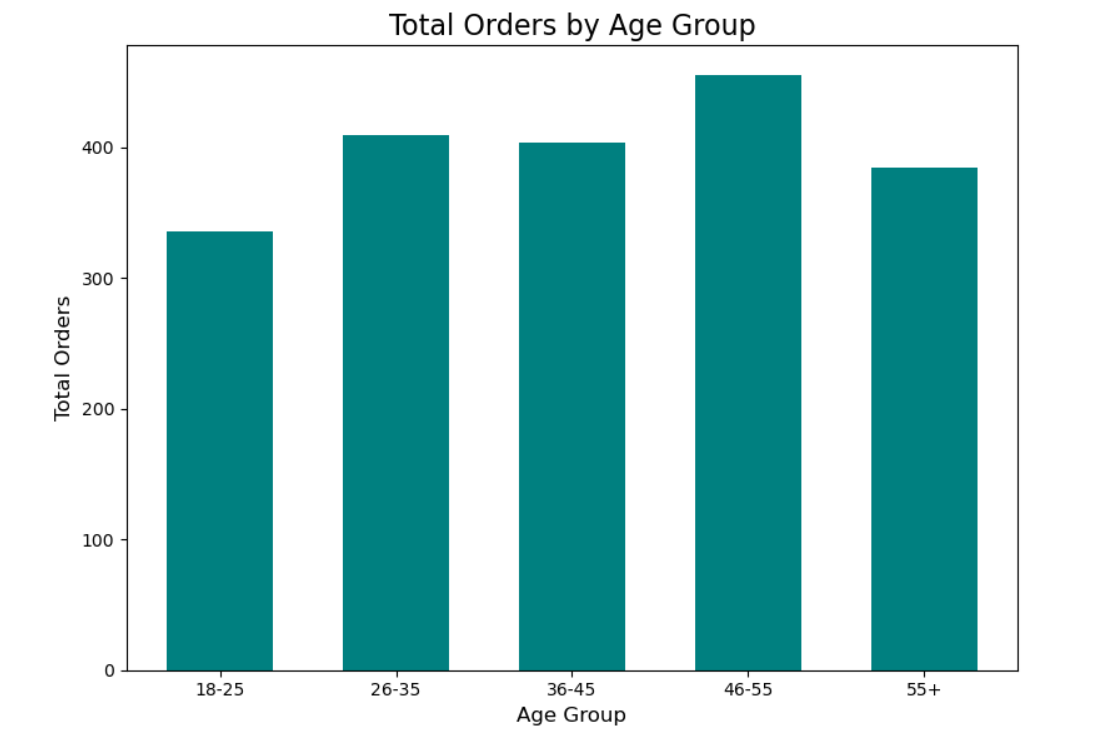
plt.xlabel('Age Group', fontsize=12)

plt.ylabel('Total Orders', fontsize=12)

plt.xticks(rotation=0) # No rotation needed for simple age group labels

plt.tight\_layout()

plt.show()



* **Create bar chart of Top 5 Customers by Total Sales**

pd.read\_sql\_query("SELECT customer\_id, sum(total\_sale) as Total\_sale from retail\_sales group by 1 order by 2 desc limit 5", conn)

# Create a bar plot

plt.figure(figsize=(10, 6))

plt.bar(df['customer\_id'], df['Total\_sale'], color='green', width=0.7)

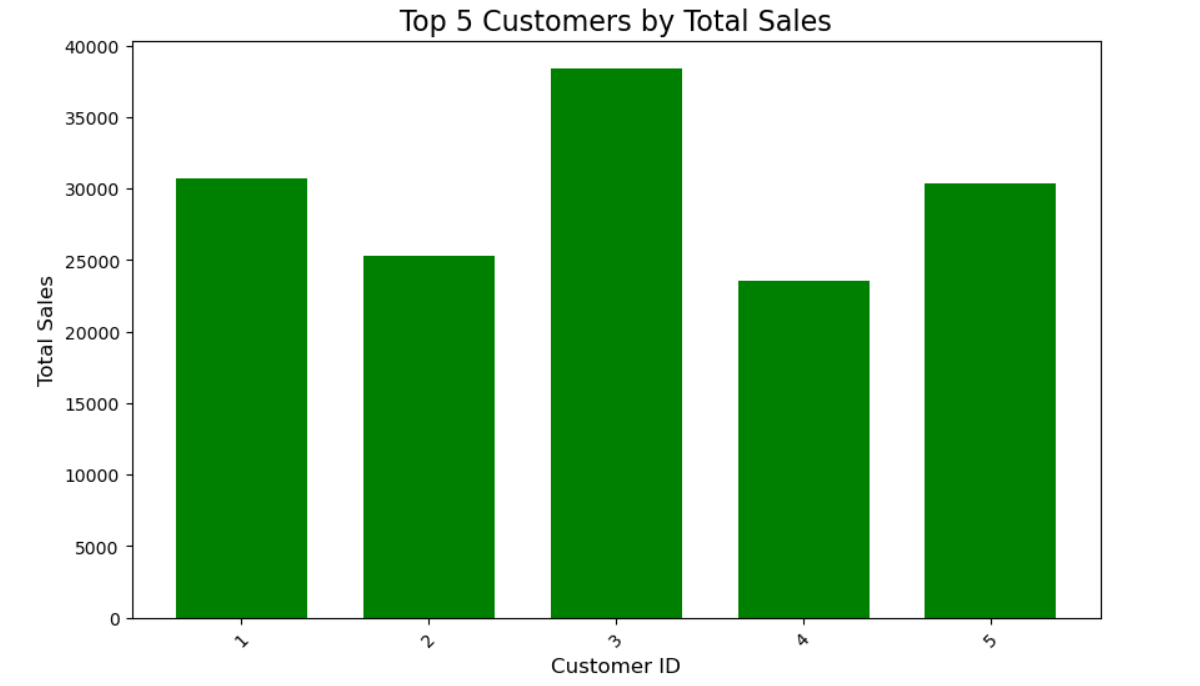
plt.title('Top 5 Customers by Total Sales', fontsize=16)

plt.xlabel('Customer ID', fontsize=12)

plt.ylabel('Total Sales', fontsize=12)

plt.xticks(rotation=45)

plt.show()



* **Create a bar chart of total sales by time shift**

query = """

WITH hourly\_sale AS (

SELECT \*,

CASE

WHEN EXTRACT(HOUR FROM sale\_time) < 12 THEN 'Morning'

WHEN EXTRACT(HOUR FROM sale\_time) BETWEEN 12 AND 17 THEN 'Afternoon'

ELSE 'Evening'

END AS Shift

FROM retail\_sales

)

SELECT Shift, COUNT(\*) as total\_orders

FROM hourly\_sale

GROUP BY Shift;

"""

# Execute the SQL query

df = pd.read\_sql\_query(query, conn)

# Create a bar plot for total orders by shift

plt.figure(figsize=(8, 6))

plt.bar(df['Shift'], df['total\_orders'], color='yellow', width=0.6)

plt.title('Total Orders by Time Shift', fontsize=16)

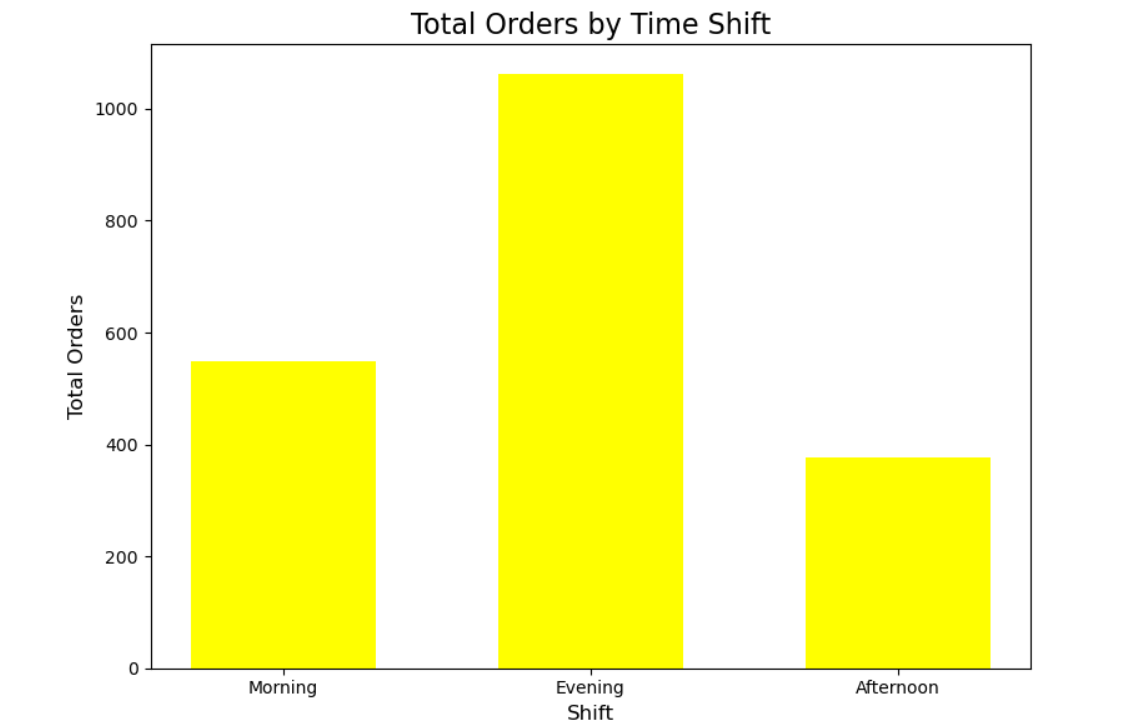
plt.xlabel('Shift', fontsize=12)

plt.ylabel('Total Orders', fontsize=12)

plt.xticks(rotation=0) # No rotation needed for simple shift labels

plt.tight\_layout()

plt.show()

****

**Findings -**

* **Total Sales -**
  + The total number of sales transactions was calculated using SQL queries.
* **Customer Insights –**
  + The dataset contains distinct customer IDs, and the average age of customers was found.
  + Customers who made purchases of more than $1,000 were identified.
* **Sales by Category –**
  + Sales were analyzed by product categories, showing distinct patterns in purchasing behavior.
  + A pie chart was created to visualize net sales by category.
* **Sales by Gender –**
  + A breakdown of sales by gender was conducted, with results displayed through a pie chart.
* **Category Sales by Gender –**
  + The sales data was further segmented by gender within each category, and a bar chart was created to illustrate this comparison.
* **Sales by Age Group –**
  + Customers were grouped into age brackets (e.g., 18-25, 26-35), and a bar chart was generated to show total orders by age group.
* **Top Customers-**
  + The top 5 customers, based on total sales, were identified and visualized with a bar chart.
* **Time-Based Sale –**
  + The analysis broke down sales into time shifts (morning, afternoon, evening), showing total sales by these periods through a bar chart.

These insights can guide strategic decisions such as customer targeting, inventory management, and promotional timing.

**Conclusion-**

The retail sales analysis identified key trends that can inform business strategy. Customers were segmented by demographics, revealing valuable insights such as the average age and high-value customers who contribute significantly to revenue. Sales performance varied by product category, with some categories outperforming others, offering guidance for inventory and promotional efforts. Sales trends also differed by gender and time of day, suggesting opportunities for more targeted marketing and time-sensitive promotions. Overall, the findings highlight the importance of focusing on top-performing customers and categories, optimizing operations around peak sales periods, and refining marketing strategies to enhance business performance.