import mysql.connector  
import pandas as pd  
import matplotlib.pyplot as plt  
import statistics as stats  
import numpy as np  
  
from sqlalchemy import create\_engine  
  
conn = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 password="riyayadav",  
 database="springdb")  
  
cursor = conn.cursor()  
  
query = "select \* from customers\_1"  
  
cursor.execute(query)  
  
result = cursor.fetchall()  
  
# for x in result:  
# print(x)  
  
df\_c = pd.read\_sql\_query(query, conn)  
print(df\_c)  
# print(df\_c.info())  
# print(df\_c.describe())  
# print(df\_c.drop\_duplicates())  
# print(df\_c.info())  
  
# engine = create\_engine('mysql+mysqlconnector://root:riyayadav@localhost/springdb')  
  
# Write the DataFrame to the MySQL table  
# df.to\_sql(name="customers\_1", con=engine, if\_exists='replace', index=False, schema=None)  
conn = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 password="riyayadav",  
 database="springdb")  
  
cursor = conn.cursor()  
  
query = "select \* from sales\_1"  
  
cursor.execute(query)  
  
result = cursor.fetchall()  
  
# for x in result:  
# print(x)  
  
df\_s = pd.read\_sql\_query(query, conn)  
print(df\_s)  
  
conn = mysql.connector.connect(  
 host="localhost",  
 user="root",  
 password="riyayadav",  
 database="springdb")  
  
cursor = conn.cursor()  
  
query = "select \* from products\_1"  
  
cursor.execute(query)  
  
result = cursor.fetchall()  
  
# for x in result:  
# print(x)  
  
df = pd.read\_sql\_query(query, conn)  
print(df)  
  
# x = ["quantity"]  
# y = ["total\_sales"]  
  
df\_3 = pd.merge(df\_c, df\_s, on='customer\_id', how='inner')  
print("\nMerged DataFrame:")  
print(df\_3)  
print(df\_3.describe())  
  
merged\_df = pd.merge(df\_s, df, on="product\_id", how="inner")  
print("\nMerged dataframe:")  
print(merged\_df)  
print(merged\_df.describe())  
  
#category analysis  
category\_price = merged\_df.groupby('category')['price'].sum()  
print("category wise total price: ", category\_price)  
  
category\_q = merged\_df.groupby('category')['quantity'].sum()  
print("quantity wise category: ", category\_q)  
  
total\_spending = merged\_df.groupby('category')['total\_sales'].sum()  
print("category wise total sale: ", total\_spending)  
  
#product analysis  
product\_sale = merged\_df.groupby('product\_id')['total\_sales'].sum()  
print("sum of product sales: ", product\_sale)  
  
pquantity\_sale = merged\_df.groupby('product\_id')['quantity'].sum()  
print("quantity wise productt sale: ", pquantity\_sale)  
  
high\_performing= merged\_df.groupby("category")["product\_id"].sum()  
print("high performing category based on product id sales:", high\_performing)  
  
top\_products = product\_sale.nlargest(5)  
print("top 5 products: ", top\_products)  
least\_products = product\_sale.nsmallest(5)  
print("least 5 products: ", least\_products)  
top\_pquantity = pquantity\_sale.nlargest(5)  
print("top 5 quantity wise products: ", top\_pquantity)  
least\_pquantity = pquantity\_sale.nsmallest(5)  
print("least 5 quantity wise products: ", least\_pquantity)  
  
#time analysis sale  
yearly\_revenue = merged\_df.groupby('sale\_date')['total\_sales'].mean()  
print("yearly revenue: ", yearly\_revenue)  
  
quantity\_sales = merged\_df.groupby('quantity')['total\_sales'].sum()  
print("quantity wise sales: ", quantity\_sales)  
  
month\_sale = merged\_df.groupby('month')["total\_sales"].sum()  
print("monthly sales: ", month\_sale)  
  
#customer analysis  
customer\_spending = merged\_df.groupby('customer\_id')['total\_sales'].sum()  
print("customer spending: ", customer\_spending)  
  
top\_customers = customer\_spending.nlargest(5)  
print("top 5 customers: ", top\_customers)  
  
tp\_price = df['price'].nsmallest(5)  
print(tp\_price)  
  
totrevenue = df\_s["total\_sales"].sum()  
avgorder = df\_s["quantity"].mean()  
avgprice = df["price"].mean()  
category\_avgprice = merged\_df.groupby('category')['price'].mean()  
print("category wise average price: ", category\_avgprice)  
print("total revenue: ", totrevenue)  
print("avergae order: ", avgorder)  
print("average price: ", avgprice)  
  
#product sale analysis  
  
plt.figure(figsize=(7,5))  
pquantity\_sale.plot(kind='bar', color='skyblue')  
plt.title( 'Quantity Wise sales of product')  
plt.xlabel('Product ID')  
plt.ylabel('Quantity Sold')  
plt.show()  
  
plt.figure(figsize=(7,5))  
product\_sale.plot(kind='bar', color="lightgreen")  
plt.title("sum of product sale")  
plt.xlabel("product Id")  
plt.ylabel("sum of sale")  
plt.show()  
  
plt.figure(figsize=(7,5))  
top\_products.plot(kind="bar", color="red")  
plt.title("top 5 product id as per sales ")  
plt.xlabel("product id")  
plt.ylabel("sale")  
plt.show()  
  
plt.figure(figsize=(7,5))  
top\_pquantity.plot(kind="bar", color="red")  
plt.title("top 5 product id as per quantity ")  
plt.xlabel("product id")  
plt.ylabel("quantity sold")  
plt.show()  
  
plt.figure(figsize=(7,5))  
least\_products.plot(kind="bar", color="red")  
plt.title("least 5 product id as per sales ")  
plt.xlabel("product id")  
plt.ylabel("sale")  
plt.show()  
  
plt.figure(figsize=(7,5))  
least\_pquantity.plot(kind="bar", color="red")  
plt.title("least 5 product id as per quantity ")  
plt.xlabel("product id")  
plt.ylabel("quantity sold")  
plt.show()  
  
  
  
#customer sale analysis  
  
plt.figure(figsize=(7,5))  
top\_customers.plot(kind="bar", color="yellow")  
plt.title("top 5 customers id as per sale")  
plt.xlabel("customer id")  
plt.ylabel("sale")  
plt.show()  
  
plt.figure(figsize=(7,5))  
customer\_spending.plot(kind="bar", color="pink")  
plt.title("sum of customers sale")  
plt.xlabel("customer id")  
plt.ylabel("sale")  
plt.show()  
  
#category sales  
plt.figure(figsize=(7,5))  
category\_price.plot(kind="bar", color="purple")  
plt.title("category wise sales")  
plt.xlabel("category")  
plt.ylabel("sale")  
plt.show()  
  
y = np.array([209, 206])  
plt.pie(y)  
plt.title("total quantity of category")  
mylabels = ("Apparel= 209", "Electronics=206")  
plt.pie(y, labels=mylabels)  
plt.show()  
  
y = np.array([153.12, 150.92])  
plt.pie(y)  
plt.title("average price of category")  
mylabels = ("Apparel= 153.12", "Electronics=150.92")  
plt.pie(y, labels=mylabels)  
plt.show()  
  
  
y = np.array([ 2296.91,2263.94 ])  
plt.pie(y)  
plt.title("sum\_of\_category\_price")  
mylabels = ("Apparel= 2296.91", "Electronics=2263.94")  
plt.pie(y, labels=mylabels)  
plt.show()  
  
#time analysis  
  
plt.figure(figsize=(7,5))  
month\_sale.plot(kind="bar", color="pink")  
plt.title("monthly sale")  
plt.xlabel("month")  
plt.ylabel("sale")  
plt.show()