import streamlit as st

import pandas as pd

import psycopg2

import seaborn as sns

import matplotlib.pyplot as plt

# Database configuration

DB\_CONFIG = {

'host': 'localhost',

'database': 'SQLTEST',

'user': 'postgres',

'password': 'Admin'

}

# Function to load data from PostgreSQL

def load\_data():

conn = psycopg2.connect(\*\*DB\_CONFIG)

query = """

SELECT

c.name AS category\_name,

v.phone AS vendor\_phone,

SUM(o.total\_amount) AS total\_order\_contribution

FROM

public.product\_names pn

LEFT JOIN

public.categories c ON pn.category\_id = c.id

LEFT JOIN

public.products p ON pn.id = p.name\_id

LEFT JOIN

public.vendors v ON p.vendor\_id = v.id

JOIN

public.product\_ratings pr ON p.id = pr.product\_id

JOIN

public.orders o ON pr.order\_id = o.id

WHERE

o.deleted\_at IS NULL -- Ensure we only consider non-deleted orders

GROUP BY

c.name, v.phone

ORDER BY

c.name, v.phone;

"""

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

conn.close()

return df

# Set up the Streamlit app

st.title("Dynamic Heatmap Dashboard")

# Load data

data = load\_data()

# Check if data is loaded

if not data.empty:

# Create a sidebar for user interaction

st.sidebar.header("Filter Options")

# Display the raw data

if st.sidebar.checkbox("Show Raw Data"):

st.subheader("Raw Data")

st.write(data)

# Pivot the data to create a heatmap-friendly format

pivot\_table = data.pivot(index='category\_name', columns='vendor\_phone', values='total\_order\_contribution').fillna(0)

# Create a heatmap

plt.figure(figsize=(12, 8))

sns.heatmap(pivot\_table, annot=True, fmt=".1f", cmap="YlGnBu", linewidths=.5)

plt.title("Heatmap of Order Contributions by Product Categories and Vendors")

plt.xlabel("Vendor Phone")

plt.ylabel("Category Name")

# Display the heatmap in Streamlit

st.pyplot(plt)

else:

st.write("No data available to display.")