DTSC660: Data and Database Management with SQL
Module 7
Assignment 5

Purpose

For this assignment you have been asked to look into some consumer sales trends and to perform some analysis. For this assignment you will rely on the skills obtained from Module 7 to successfully create the sales database and query it to retrieve the appropriate responses. Note that many, but not necessarily all, of the tools you learned in Module 7 will be applied in this assignment.To complete this assignment, download and import the dataset and then create queries that respond to each prompt. Please make sure that you only use postgres language conventions. **Each question is all or nothing. Graders will not attempt to correct or interpret malformed SQL queries.** You will be responsible for testing your code on the provided data set before submission. The question will be graded based on whether or not it generates the correct output and addresses all requirements specified in the question. Extraneous columns will not count against you as long as correct results are obtained.

Submission

You will submit a total of **1** sql files to CodeGrade. Files should be named appropriately and be in .sql format. Each file must use the postgres standards taught in the course. Use of other SQL languages such as T-SQL will result in an automatic 0 for the assignment. **Each question is all or nothing. Ensure your file runs in its entirety in pgAdmin. This means ensuring each query ends in a semicolon (;). Graders will not attempt to correct or interpret malformed SQL queries.** You will have **one submission attempt** for this assignment.

- ***File 1***: You must submit a SQL document called <LastName>_Assignment5. This document must include ALL queries requested in the instructions below.

- You will submit the file to the Assignment 5 Submission link to CodeGrade.

Instructions

As in the practice assignments you will be querying a large dataset to gather insights about that data. You have been provided a SQL file with all the necessary data. Make sure you take a moment to familiarize yourself with the columns, their data types, and any other pertinent information.It is expected that if you have questions or difficulties with any portion of this assignment that you utilize the assignment discussion board or email the GAs to gain clarity (dtsc_ga_660@eastern.edu).

Make sure to complete the steps below to prepare the data file for importing:

- In the Assignment 5 folder, download the ***customer_spending.csv*** file
- Place this file in a public folder on your computer
- Take note of the path to this file (copy the path)

PART 1 Creating the Table and Importing the Data

1. Create the table with appropriate data types
    a. Name the table *customer_spending*
    b. Reuse the column titles from the csv, *do not change these*
    c. When selecting data types for your tables, ensure they accommodate the full range of values and avoid truncating any values. Ensure that decimals, lengths, and other specifications are sufficient for all data to be imported completely.
2. Write the copy statement to bring the data into the database
    a. Remember that if you choose an incompatible data type, you can enter the command *DROP TABLE customer_spending* to remove the table and restart.
3. Run a basic select statement that verifies the data is present and matches what is in the csv file.

Make sure to include the table creation and copy statement in your code as this is a component of your grade as demonstrated in the rubric below.

Part 2 (Queries)

To complete this part, download the assignment_5_template.sql file from the Assignment 5 folder. Rename this file using the naming convention: <LastName>_Assignment5. Complete each query in the identified space in this document. Once you are done, submit the document to the Assignment 5 Submission link.

Please note that column names in **bold** to help you identify the columns from the table to be used in each question. However, these aren't necessarily the columns that your query output should return. Be sure to read each question carefully. Note that a request for a list of values expects no duplicate values unless specified otherwise.

1. Write a query that returns each **category** and the corresponding total **revenue** for that category for the **sale_year** 2016. The output should be arranged alphabetically.

2. Write a query that returns a list of **sub_categories** and their corresponding average **unit_price**, average **unit_cost**, as well as the difference between these two values (name this column *margin)* for the **sale_year** 2015. Organize the results alphabetically.

3. Write a query that returns the total number of female buyers (**gender**) who made purchases in the Clothing **category**.

4. Write a query that returns the **age**, **sub_cateogry**, average **quantity**, and average **cost** of products purchased by each **age** and **sub_category**. Output should show the columns in the same order they are listed. Organize the data by **age**, oldest to youngest, and then by **sub_category** alphabetically.

5. Write a query that returns a list of **countries** where more than 30 transactions were made by customers between the **ages** of 18-25 (inclusive).

6. Write a query that returns a list of **sub_categories** along with their average **quantity** and average **cost,** both rounded to 2 decimal places, and named as *avg_quantity* and *avg_cost* respectively. Only include **sub_categories** that have at least 10 records in the data set. Organize the data by **sub_category** alphabetically.

7. Write a query that returns each **sub_cateogry** and their respective total **quantity** and total **revenue** for male buyers (**gender)**, in the **sale_year** of 2016.

8. Write a query to determine each **country's** total **revenue** generated from sales, sorted alphabetically.

9. Write a query to determine the highest **unit_cost**, lowest **unit_cost**, and average **unit_cost** for each **gender** in each **category**. The output columns should be **gender**, **category,** *high_cost*, *low_cost*, *avg_cost.* Organize the results by **gender** and then **category**.

10. Write a query to return the **country** that has the highest average **revenue**. Your output columns should be **country** and *high_sales.*

Hint: To obtain the highest value, you will need to return only one row from your results. Think of how you'll need to sort the output and then see this link if you need guidance on how to select a single row.

******************************GRADING RUBRIC ON NEXT PAGE********************************

This assignment will be graded on the following rubric. Incorrect syntax, extraneous results, or incorrectly addressing all question requirements will result in loss of points. Please see Module 0 for our course's guidelines on point allocation and deduction. Graders will NOT attempt to correct malformed sql code. :

Question Number	**Points**
Creating Table	10
Importing Data	10
1	10
2	10
3	10
4	10
5	10
6	10
7	5
8	5
9	5
10	5
Total	100