DTSC660: Data and Database Management with SQL
Module 3
Assignment 2

Purpose

For this assignment, you have been asked to perform some basic filtering of a data set that was scraped from Github. The goal is to better understand the Github data and to practice the various querying techniques learned in Module 3. Note that many, but not necessarily all, of the tools you learned in Module 3 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. You will be responsible for testing your code on the provided data set before submission. Each 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 **<u>one</u>** submission attempt for this assignment.

- ***File 1***: You must submit a SQL document called <LastName>_Assignment2. This document must include ALL queries requested in the instructions below.

- You will submit the file to the Assignment 2 link to CodeGrade.

Instructions

As in the practice assignments you will be querying a large dataset to gather insights about that data. The data set you will be working with is scraped data from Github about topics of discussion on the site, the users who have created those topics, and how popular the topics are based on their “star” count. 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).

Vocabulary
Repo: Short for repository, think of this as a single discussion thread in a discussion board.
Topic: An organization of repository discussions. Similar to a discussion board.
Rep link: A unique URL that identifies where the code repository is located.
Star Count: Stars are similar to likes, the more stars the more popular the topic.

PART 1 Creating the Table and Importing the Data

1. In the assignment 2 folder, download the ***github_data.csv*** file
2. Place this file in a public folder on your computer
3. Take note of the path to this file (copy the path)
4. In the assignment 2 folder, copy and paste the SQL commands into the query tool from the ***Github Table Creation and Importing*** page. (The database you choose to put this in is up to you).
5. Update the pasted code with the path to the github_data.csv file that you took note of in step 3.
6. Run the commands and this should create the table and import the github data. Table must be named "github". **Failure to name (and query) your table as github will result in a 0 as the grader will not be able to run the queries. (Examples of incorrect queries: SELECT <columns> FROM github_table, SELECT <columns> FROM public.github)**
7. Run a basic select statement that verifies the data is present and matches what is in the csv file.

Part 2 (Queries)

This is the part you will be graded on. To complete this part, download the assignment_2_template.sql file from the assignment 2 folder. Rename this file using the naming convention: <LastName>_Assignment2. Complete each query in the identified space in this document. Once you are done, submit the document to the Assignment 2 link to CodeGrade. Remember that you will not be penalized for extraneous columns, but you will be for extraneous rows (data). At times, returning an extra column may return incorrect rows, so be cognizant. There may be multiple correct ways to solve some queries.

1. Write a query that returns all data for the github table
2. Write a query that returns a list of the topics on github (no duplicates)
3. Write a query that returns the repo name and star count from largest to smallest
4. Write a query that returns a list of topics in alphabetical order (no duplicates)
5. Write a query that returns all repo names that have a star count greater than 2000
6. Write a query that returns all repo names that have a star count greater than 3000 and are in the 3d topic
7. Write a query that returns all repo names that are in the aws, azure, or chrome topics and that have a star count less than 1000

8. Write a query returning the user name, repo name, and repo link where the link contains ‘ext’ (note this is NOT case sensitive)
9. Write a query that returns all columns for chrome topics whose star count is larger than 5000
10. Write a query that returns all the username and respective repo name where the star count is greater than 1000 and less than 15000
11. Write a query that returns an alphabetical list of usernames who have repos with star counts of higher than 15000 (no duplicates)
12. Write a query that returns a list of usernames that start with ‘Add’ or end with ‘on’ (Note this IS case sensitive). (no Duplicates)
13. Write a query that returns an alphabetical list of topics that have at least one repo with a star count greater than 100,000 (no duplicates)
14. Write a query that returns a list of topics that contain null star counts (if any exist) (duplicates allowed)
15. Write a query that returns a list of topics, usernames, and star_count if the star count is at least 100,000 but no more than 200,000 and whose topic starts with ‘a’

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

This assignment will be graded on the following rubric. Remember that questions are ALL OR NOTHING. Incorrect syntax, extraneous results, or incorrectly addressing all question requirements will result in loss of points for that question. Graders will NOT attempt to correct malformed sql code. :

Question Number	**Points**
1	2
2	2
3	3
4	3
5	5
6	5
7	5
8	5
9	10
10	10
11	10
12	10
13	15
14	5
15	10
Total	100