Assignment 2: Query the Database

CSE 511: Data Processing at Scale - Spring 2023

Available: 02/01/2023 **Due Date**: 02/11/2023 11:59 PM **Points:** 100

Introduction & Background

In Assignment 1, you learned how to design and load a database. The assignment gave you an opportunity to create a database from scratch and perform optimized data insertion. Assignment 2 has the same background information as Assignment 1, and this will help you to understand how to make different queries to the tables we created in Assignment 1, and also give you the opportunity to explore how 2 queries with different clauses can have a very different impact on execution time as well. For this assignment, we will use the tables you created as a part of Assignment 1.

Problem Statement

Considering the **four** created tables **submissions, comments, authors and subreddits,** from Assignment 1. Your task is to implement the following SQL queries

1. `query1`: Write a SQL query to return the total number of comments authored by the user **`xymemez`.**
    a. Your column names MUST be: '`count of comments`'
2. `query2`: Write a SQL query to return the total number of subreddits for each subreddit type.
    a. Your column names MUST be: '`subreddit type`', '`subreddit count`'
3. `query3`: Write a SQL query to return the top 10 subreddits arranged by the number of comments. Calculate average score for each of these subreddits and round it to 2 decimal places.
    a. Your column names MUST be: '`name`', '`comments count`', '`average score`'
4. `query4`: Write a SQL query to print `name`, `link_karma`, `comment_karma` for users with >1,000,000 average karma in descending order. Additionally, also have a column '`label`' which shows 1 if the `link_karma` >= `comment_karma`, else 0
    a. Your column names MUST be: '`name`', '`link karma`', '`comment karma`', '`label`'

Copyright © Zhichao Cao, Ph.D.

    b. You can write this query with both having and where clauses (both will be considered correct and submit only one), however, try doing both just to see the speed difference. (if you do try it) let us know the results in the README along with your theory for why!
    c. To fairly compare times between 2 queries, you need to clear the postgres cache! A helpful link: See and clear Postgres caches/buffers? - Stack Overflow

5. `query5`: Write a SQL query to give count of comments in subreddit types where the user has commented. Write this query for the user `` `[deleted_user]` ``
    a. Your column names MUST be: '`sr type`', '`comments num`'

6. `query6`: Write a SQL query to print the datetime (UTC format) of the created time of comments, subreddit name and comments for the user `` `xymemez` `` in the subreddit `` `starcraft` ``
    a. Your column names MUST be: '`utc time`', '`subreddit`', '`comment`'

7. `query7`: Write a SQL query to get the 4 most upped submissions (if any) from the 4 oldest under 18 subreddits
    a. Your column names MUST be: '`submission`', '`ups`', '`subreddit`'
    b. Using posgres functions might be a good idea for this!

8. `query8`: Write a SQL query to get the author and ups for the most upvoted and least upvoted comment on reddit.
    a. Your column names MUST be: '`author`', '`upvotes`'
    b. Sub-queries or temp tables are a great option for such questions

9. `query9`: Write a SQL query to display the number of comments made by the author, `` `xymemez` `` according to the utc date arranged in ascending order.
    a. Your column names MUST be: '`date`', '`count`'

10. `query10`: Write a SQL query to get the month when reddit was most active along with the 10 top subreddits and the number of posts in the subreddits in that month.
    a. Your column names MUST be: '`month`', '`subreddit`', '`count`'

Above all, your script **MUST** generate **ten tables**, namely, "query1", "query2", …, "query10" respectively for each query.

Copyright © Zhichao Cao, Ph.D.

NOTE

- All table names and attribute names **must** be in lowercase and exactly the same with the specification.
- Do not put "create/select/drop database", or "set system settings or encoding" in your script. This may lead to point deductions. **Don't re-create tables of assignment 1 and don't load any data.**
- You are free to create any other temp/permanent views, temp/permanent tables or functions to help your queries.
- You should use the following command to save your query result to a table.

  CREATE TABLE query0 AS YOUR SQL STATEMENT

  For instance, select the user from the users table which has userID = v1 and store it in query0 and rename the "username" column to "userfullname".

  CREATE TABLE query0 AS SELECT username AS userfullname FROM users WHERE users.userid = :v1

How to work on the assignment and test your code

You can work choose to work on your local setup, or to use the **docker image** with all the tables created and ready to go. For those not familiar with docker, we recommend using the container to give you a better grasp of using Docker containers.

- PostgreSQL 14 is already setup (but not running) in the docker image. Find the configurations below
  - **username**: postgres
  - **password**: postgres
  - **Database Name**: postgres
  - **Database IP**: 127.0.0.1:5432
- The database and tables from Assignment-1 are also set up in this docker image.
- Docker is a lightweight virtualized runtime environment that is used for running applications and will be used more in the future assignments. Make use of the below references on familiarize yourself with Docker! You only need the basics right now, but it always good to explore aas much as you can!
  - Overview | Docker Documentation
  - Tutorial: Get started with Docker apps in Visual Studio Code | Microsoft Learn
  - A Docker Tutorial for Beginners (docker-curriculum.com)
  - The Docker Handbook – Learn Docker for Beginners (freecodecamp.org)
  - How To Install and Use Docker on Ubuntu 22.04 | DigitalOcean

Copyright © Zhichao Cao, Ph.D.

Grading

- The assignment will be graded using automated scripts. Make sure to follow the naming conventions as mentioned.
- 100 points of the assignment are equally divided into **ten** queries
- To help you along the way, please find the sample output ONLY for query 1 and query 2
    - Query 1:

	count of comments bigint
1	18

    - Query 2

	subreddit type text	subreddit count bigint
1	employees_only	1
2	gold_restricted	1
3	private	506
4	public	322262
5	restricted	27633
6	user	563664

Submission Requirements & Guidelines

- Assignment 2 is due **02/11/2023 at 23:59:00**. Submit the assignment following the below guidelines
- This is an **individual** assignment
- Maintain your code on the **provided private GitHub repository for assignment-2** in the [SPRING-2023] [CSE511] Data Processing at Scale Organization). Make sure you don't use any other repository.
- **What to submit on canvas?**
    - One SQL file with your queries
- **Naming format:**
    a. You **MUST** name your .sql file as **assignment2.sql**

Submission Policies

1. Late submissions will *absolutely not* be graded (unless you have verifiable proof of emergency). It is much better to submit partial work on time and get partial credit than to submit late for no credit.
2. Every student needs to *work independently* on this exercise. We encourage high-level discussions among students to help each other understand the concepts and principles. However, a code-level discussion is prohibited, and plagiarism will directly lead to the failure of this course. We will use anti-plagiarism tools to detect violations of this policy.

Copyright © Zhichao Cao, Ph.D.