CSC 370: Database Systems
Summer 2019

Assignment 3

Due: Friday, July 19, 2019, 11:55 pm, via git push

Important

All work for this assignment is to be done using PostgreSQL 10. Each student has been assigned their own account/database on a server instance of PostgreSQL10; the server is located at `studsql.csc.uvic.ca`.

In order to distribute to you the access information for your database (i.e., account, password, how to access, etc.) and also in order to provide a place in which you can prepare and submit your work, a `git` repository has been created for each student. (You will use this repo for both assignments #3 and #4.) To clone your repo, use the following URI:

```
https://gitlab.csc.uvic.ca/courses/2019051/CSC370/assignments/<netlink>/sql-coding.git
```

For example, if your netlink ID is `arlene`, then you could clone the repo as:

```
https://gitlab.csc.uvic.ca/courses/2019051/CSC370/assignments/arlene/sql-coding.git
```

In the root directory of your repo you will find a file named `ACCESS.md` which gives your username, password, and other needed connection details, *including those involving the Engineering firewall*. Note ***you will not be using your Netlink credentials*** to connect to the database server.

Problem Statement

The environmental-activist organization named "Green-not-Greed" (GnG) has grown to a point where some computerized record keeping would help with day-to-day operations. They would use online services for such record keeping, but are worried about security and surveillance given constant revelations on how some state-sponsored security services gather data on ordinary citizens via data transmitted over the Internet. They would rather build a home-grown system to help them with their work.}

The main purpose for which the group exists is to raise public awareness on emerging environmental issues that have relatively local impact (i.e., "local" here would mean a region up to a size of, say, "Vancouver Island"). Although GnG keeps in touch with other,

more globally-oriented environmental groups, and some GnG members belong to such groups, GnG's focus is on "as-local-as-possible" issues.

Attempts to raise public awareness and effect change are via "campaigns". These are person-on-the-street activities where volunteers are scheduled to be at street corners and public squares in towns or cities. There the volunteers use posters, placards, and other material to capture the attention of citizens in the public space and share with them the issues central to the campaign. These campaigns can last anywhere from two weeks to two months, and may have events taking place simultaneously in the same, or different, cities and towns and villages. There is a little bit of fundraising that also occurs during campaigns, but GnG is mostly funded by several large donors. Campaigns do have some costs associated with them. At present GnG operates out of a small downtown office, the rent of which is paid by some GnG supporters.

GnG also has a website. Although the computer system solving the problem described here will not be linked to the website, the group believes it makes sense that something be done to keep track of when and how campaigns (and the phases of each campaign) will be pushed out – and have been pushed out – to the website.

There are a few salaried employees (on very low salaries!) and most of the organization is based on the work of its volunteers. There are two tiers of volunteers: those who have participated in more than three publicity campaigns and then the others who have participated in two or fewer. There are also members who are not volunteers but who are interested in supporting the activities of GnG.

In essence the computer system is needed to help GnG keep track of campaigns, who is working on them, when activities need to happen, and the way funds actually flow in and out.

For this problem you must also use your own understanding of the way such groups work in order to "fill in the gaps" of the description above. The assumptions you make must be rooted in the real world and therefore when stating your assumptions please provide some justification. As you think through the problems in this assignment, feel free to look at how other, similar groups are organized in the community.

Deliverable A: ER Diagram

Prepare an ER diagram modelling the entities, relationships and constraints in this problem. Feel free to use whatever tool you wish for preparing the diagram; however, hand-drawn diagrams are acceptable. The finished diagrams must be available in PDF format.

Please do this step first! You may be tempted to develop the SQL schemas first and then prepare the ER diagrams, but this order of work may yield poor results and possibly a much lower assignment grade.

Deliverable B: A set of relations

After having prepared and reflected on your ER diagram (and you may have drawn several versions as you come to grips with the problem described), convert these into a set of relations. If your diagram includes ISA hierarchies, then choose what you believe to be the appropriate conversion approach (i.e., ER, O-O, or Nulls).

Deliverable C: SQL table creation commands

Prepare and implement the SQL statements (in PostgreSQL) needed to not only construct the table schemas corresponding to your relations, but also populating them with dummy data. (Use `insert` SQL commands to add tuples to tables.) Ensure any key and foreign-key constraints are listed and handled appropriately. You *do not need* to use attribute or table constraints for this assignment, but you are not forbidden from using them.

Deliverable D: At least ten SQL queries using your tables

You are to pose at least ten questions you would want to ask of the data in the tables. Provide at least one SQL query per question that finds the answer to that question.

Amongst all of the SQL must appear the important query constructs we have discussed in class: subqueries; subqueries using scalar values; set operations; use of *exists*, *any*, or *all*; join operations; grouping and aggregation; etc.

Each of your queries must be implemented as an SQL view where the view name corresponds to the number of the question (above) you are answering.

What to submit

All of your work is to be stored in your git repo (described at the start of this document) submitted via `git push` – that is, diagrams, notes, SQL files, etc. must be in the git project provided to you. E-mailed submissions will not be accepted.

- In a directory named `diagrams/` have all of your ER work (which may be scanned versions of your handdrawn diagrams).
- In a directory named `schemas/` have all of the SQL required construct tables, constraints, populate tables.
- In a directory named `queries/` have all of the ten SQL queries constructed for this assignment.

You may create other directories as needed in your git project to support your work. Please ensure these directories are properly `added` and `committed` before your final `push` for the assignment.

Evaluation

As there are many possible correct solutions to the problem, evaluation will be done via a demonstration of work in front of a member of the CSC 370 teaching team. Information on demos (i.e., where, when, how to sign up) will be distributed shortly before the assignment due date.

The marking scheme below will be used:

- A grade: An exceptional submission demonstrating creativity and initiative. The data modelling is thorough and shows insight, the database schema is well prepared, and required SQL features are intelligently (and clearly) used in assignment's queries.

- B grade: A submission completing the requirements of the assignment. The data modelling is thorough, database schema has been prepared, and required SQL features are used in the assignment's queries.

- C grade: A submission completing most of the requirements of the assignment. There may be problems with one of: data modelling; database schema; SQL queries.

- D grade: A serious attempt at completing the requirements of the assignment. There are many problems with the submitted work.

- F grade: Either no submission is given, or submission represents very little work.