CSC 370: Database Systems
Spring 2020

Assignment 3

Due: Tuesday, March 17, 2020, 11:55 pm, via *git push* (to *gitlab*)

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 `pgstudent.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 `gitlab.csc` repository has been created for each student. (You will use your repo for both assignments #3 and #4.) In order to clone your repo, use the following URI:

```
https://gitlab.csc.uvic.ca/courses/2020011/CSC370/assignments/<netlink>/sql-coding.git
```

For example, if your netlink ID is `jtrudeau`, then the address you use to clone the repo is:

```
https://gitlab.csc.uvic.ca/courses/2020011/CSC370/assignments/jtrudeau/sql-coding.git
```

In the root directory of your repo you will find a file named `ACCESS.md` which provides your username, password, and other needed `pgstudent.csc` connection details. Note that ***you will not be using your Netlink credentials*** to connect to the database server.

Problem Statement

You are tasked to model the data needed for a fitness-centre organization called "Bob's Artisanal Abs" (or "BAA" for short).

The organization consists of *members* for which we must track a *name*, *fitness-center ID*, *mailing address*, *contact details*, *membership status* (i.e., full, guest, drop-in, and others), and *membership type*. Each *membership type* has a full name and indicates the kinds of services of the centre for which members are eligible to access. There is a system of *passes*, ranging in duration to single drop-ins, to a set of 10 drop-ins, to a quarterly pass, to an annual pass; passes are also specific to that kind of service to be used by the member. Passes must, of course, be purchased, and these *transactions* must be recorded.

BAA also offers additional services that are specially scheduled, such as *camps* which offer specific kinds of training of several weeks for training in certain kinds of events in the community (e.g., training for 10K competitions). These camps have *instructors* (who may or may not also be members) along with lists of enrolled members. Enrolment in camps is in addition to membership passes.

Lastly there is some merchandise that is sold at the front desk, such as T-shirts, towels, locks for change-room lockers, etc. Such merchandise sales are not necessarily restricted to members.

The details given above have been left deliberately imprecise as there is some room for creativity in this assignment. In order to obtain more precision on the problem, please feel free to fall back on your own knowledge of the way fitness centres and gyms are organized and managed.

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 `gitlab` 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.