SQL Assignment

Instructions

This assignment would capture SQL queries implementation on PostgreSQL installed on the system locally.

The Database

This assignment will use a fictional database containing college-type data. The data is in seven flat files. You can find all the flat files in this zip file.

Your first goal is to import this data into relations in PostgreSQL. The schema of the database that you should create is as follows

```
student (sid, sname, sex, age, year, gpa)
dept (dname, numphds)
prof (pname, dname)
course (cno, dname, cname)
major (dname, sid)
section (dname, cno, sectno, pname)
enroll (sid, dname, cno, sectno, grade)
```

course (cno, cname, dname)

enroll(sid, grade, dname, cno, section)

Note: Keys are in styles, italic represents foreign key and underlined are primary key

The Schema Understanding

Based on the above given Tables and their fields, assignment would require creating the ER Diagram of the database.

To import a flat file into PostgreSQL, here are the steps you should follow:

1. Create a new directory for this assignment.
2. Copy all the data files into your new directory.
3. Create all the tables with appropriate fields. (Keep all the "CREATE TABLE" commands in a single SQL file named `*createtable.sql*`).
4. Import the flat files one-by-one to the tables. (If using **psql** command line tool, you can invoke `***\copy {name_of_table} from {respective_file}***`.
   For e.g., `*\copy enroll from enroll.data*`
5. Verify that the data made it to PostgreSQL correctly.

Note: As we are having foreign keys, data needs to be imported in a specific sequence which candidate need to find based on schemas. Not following the sequence would result in errors while importing.

The Queries

Design SQL queries that answer the questions given below (one query per question) and run them using PostgreSQL. Your queries should be correct with respect to the design of this dataset. In other words, we should be able to use your SQL queries on another dataset with the same schema and get correct answers even if the actual data within is different.

The query answers should be duplicate free, but you should use **distinct** only when necessary, in general. Determining whether you need **distinct** should not depend on the particular dataset you have before you, but on its design.

It is most important to write queries that are correct, but good style is also equally important.

- There is generally little agreement on a single standard for formatting and capitalizing your SQL code (indentation, etc). For this assignment, we'll need to understand following points
    - All SQL keywords should be in upper-case
    - If the query is large, it should be split into multiple lines with appropriate indentations.
- You should write your queries as briefly as you can. Redundant or gratuitous additional pieces make your queries harder to understand and maintain. In particular, be wary of using nested queries. All the nested queries should be addressed using CTE (Common Table Expression).
- Do not use natural joins. Most SQL developers agree that this convenient shortcut is an opportunity for unexpected bugs to occur.

Here are the queries you should write. Put a comment above each query with the number and description so we can easily tell which query is which.

1. Print the names of professors who work in departments that have fewer than 50 PhD students.
2. Print the names of the students with the lowest GPA.
3. For each Computer Sciences class, print the class number, section number, and the average gpa of the students enrolled in the class section.
4. Print the names and section numbers of all sections with more than six students enrolled in them.
5. Print the name(s) and sid(s) of the student(s) enrolled in the most sections.
6. Print the names of departments that have one or more majors who are under 18 years old.
7. Print the names and majors of students who are taking one of the College Geometry courses.
8. For those departments that have no major taking a College Geometry course print the department name and the number of PhD students in the department.
9. Print the names of students who are taking both a Computer Sciences course and a Mathematics course.
10. Print the age difference between the oldest and the youngest Computer Sciences major.
11. For each department that has one or more majors with a GPA under 1.0, print the name of the department and the average GPA of its majors.
12. Print the ids, names and GPAs of the students who are currently taking **all** the Civil Engineering courses.

All of your SQL queries should be contained in a single file called queries.sql.

The Submission

In this assignment there are three major outcomes that are highlighted in this manner. All three outcomes would be required to be added in the Presentation along with basic understanding of Databases.

At the time of review, candidate can be asked out of the box question to perform on the database as well. So, don't just focus on solving the above problems, rather focus on learning the SQL commands and relative logics.

Any case found of cheating will not be tolerated and will result in strict reviews for both the parties.