

CS 166: Database Management Systems

FALL 2004

PROGRAMMING ASSIGNMENT 1

Assigned: October 15, 2004

DUE: Thursday October 28, 2004, 11:59am (Electronic Turnin)

Hardcopy, due Thursday October 28, 2004, 3:40pm in class

This is an individual assignment – no group submissions are allowed

Introduction

This assignment will use a database containing data about a university. The data for all relations are available at the class website for you to download.

The schema of the database is provided below (keys are underlined in bold, field types are omitted):

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

Before you start writing SQL, it is a good idea to take a look at the database and familiarize yourself with its contents. Note that all data are delimited by a symbol '@'. The database must satisfy the following conditions:- (hint: use foreign key constraints)

- Only the student in student relation can exist in the enroll relation, i.e. a person not in the student database is not allowed to enroll in any classes.
- Each student can enroll in as many classes as he/she likes being offered by various departments. Each course (from various departments) can offer more than one sections. It is also possible for a student to enroll in all sections offered by the same department and course number. Make sure that all departments, course numbers, and section numbers involved in all relations are valid.
- A professor can only belong to the department listed in the dept relation. However, a professor can be affiliated to more than one departments, and only professors affiliated to the school are allowed to teach a course. You can assume that each section must be taught by exactly one professor.
- Not all students are required to have a major, but each student with a major must belong to the student relation.
- Make sure that all the Integrity constraints are captured. For example, a course can only be offered by a valid department name listed in the dept relation. Inserting an invalid department name into course table should give you a similar following error message.

```
Mydb=# INSERT INTO course (cno, dname, cname)
Mydb=# VALUES (850, 'Food Engineering', 'Intro to Food Eng');
ERROR: $1 referential integrity violation - key referenced from
course not found in dept.
```

The hardcopy you are turning in should demonstrate ALL integrity constraints in your tables, by showing error messages similar to the above for each of the violations made.

Do not forget to take care of the integrity constraints on DELETE or DROP as well (CASCADE, NO ACTION, etc). Make sure you put all the codes to create tables in `create.sql`, and the codes to bulk-load the data into tables in `load.sql`

Queries

Write SQL queries that answer the questions below (one query per question) and run them on PostgreSQL. The query answers must not contain duplicates, but you should use the SQL keyword **distinct** only when necessary. You should copy the data files from the class directory into your personal directory before you start writing queries. For this assignment, creation of temporary tables is *not* allowed, i.e., for each question, you have to write exactly ONE SQL statement.

Questions

Write the following SQL queries, and report the results obtained from PostgreSQL.

1. Print the names of professors who work in departments that have fewer than 50 Ph.D. students.
2. Print the name(s) of student(s) with the lowest gpa.
3. For each Computer Sciences class, print the *cno*, *sectno*, and the *average gpa* of the students enrolled in the class.
4. Print the course names, course numbers, and section numbers of all classes with less than six students enrolled in them.
5. Print the name(s) and sid(s) of the student(s) enrolled in the most classes.
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. (Hint: use “like” predicate and the string matching character in your query)
8. Print the names of students who are taking both a Computer Sciences course and a Mathematics course.
9. Print the age difference between the oldest and youngest Computer Sciences major(s).
10. 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.

What to turn in

This is an individual assignment – no group submissions are allowed. You should turn in a directory named ***assign1_<login_name>*** that contains the following files *electronically* through the turnin site.

- ***create.sql***, ***load.sql***, ***drop.sql***, and ***query.sql*** containing the codes to create all the tables, bulk-load all the data files into tables, delete all the tables, and all the SQL queries, respectively. Note that ‘*query.sql*’ only needs to contain the ten SQL queries. The results returned by each query will be included in the writeup, ‘*answer.txt*’.
- ***answer.txt***, containing all ten SQL queries, each followed by the results returned by PostgreSQL.

HARDCOPY:- The printout should contain 1) the PostgreSQL session showing the ‘Integrity Constraints’ for each table (if applicable), i.e. showing the PostgreSQL error messages when trying to insert violating entries, and 2) the ten SQL queries for the ten questions above, each followed by the query results returned by PostgreSQL.

Note: Please direct all the questions/discussions regarding this assignment to the class mailing list, cs166@lists.cs.ucr.edu.