BA in Computer Science
BA in Mathematics and Computer Science
Michaelmas Term 2021

DATABASES

Second Practical Assignment – PHP

Introduction

The Databases practical uses the same of the database as the prior one. As in the prior practical, you should use your private copy of the data. Recall the schema of the database:

- `student(`**sid**`, sname, sex, age, year, gpa)`
  Here `sid` is a unique student identifier (a number), and `gpa` is the student's grade point average.
- `dept(`**dname**`, numphds)`
  A department is identified by its name, and we also record the number of PhD students in the department.
- `major(`**dname**`, `**sid**`)`
  A student takes a degree in one or more departments, and this is recorded in the major relation.
- `prof(`**pname**`, dname)`
  Each professor is identified (perhaps unrealistically) by a name, and the professor is associated with a department.
- `course(`**cno**`, cname, `**dname**`)`
  A course is identified by its course number and the department where it is given. The course also has a name.
- `section(`**dname**`, `**cno**`, `**sectno**`, pname)`
  A course is split up into sections. Each section is taught by a specific professor.
- `enroll(`**sid**`, grade, `**dname**`, `**cno**`, `**sectno**`)`
  The enroll relation records which students are taking a section, and we also record their grade for this section. The grade is a number between 0 and 4.

Your database should be already populated with data from the previous practical; please connect to Postgres and check the tables.

Framework

You will be writing a Web interface for a university administrator to view and modify the database. Information on how to create PHP scripts and HTML can be found off of the Practicals webpage (see the "PHP Info" link). Examples are included there.

The entry point should allow the user to choose from several administrator services, one for each question below. For the questions below, each service will consist of a page where the administrator gives some input (e.g. the administrator asks about the grade of a student) followed by a page that displays the results (e.g. shows the grades of the students).

Query Questions

Create services for each of the following:

1. Administrator gives as input the number of a course; the response is a list of names of students who have taken the course number and the department, along with the grade the student achieved in that course. If there are several students with the same name, they should be shown in separate rows.

2. Administrator gives as input a number (call it K), the resulting pages displays the names of departments that have one or more students majoring in the department who are under (strictly less than) K years old.

3. Administrator gives as input a number (call it K); resulting page lists the department name, course number, and section number of all sections that have less than K students in them.

Note: as before, your answers should work on *any data*.

Optional

Write an interface that allows the administrator to enter a studentid; in response the script should display a page listing all the courses (department name and course number) for classes that student has taken, *in a drop-down list*.

The administrator can then choose from the drop-down list a particular course for that student, and the script that follows it will display the grade (assume for this assignment that a student takes at most one section of a given course, and hence gets at most one grade for a given course number).

Assignment Submission

You should turn in well-documented scripts and sample output.

To gain S+ you must complete the optional part.