CS 5330 Group Project

The goal of the group project is to implement a database solution to the following application, using mySQL as the backend.

Application: Program Evaluation

Each program (degree) in a University are constantly under review. In order to facilitate reviews, data need to be corrected, stored and analyzed.

Programs, department, faculty and courses

The basis of evaluation is a (degree) program. For the sake of this project, you can think of a program as a degree. Each program has a (unique) name, and a person in charge (you need to store the person's name, University ID and e-mail address).

Programs are being run by departments. Each program will be assigned to one department. For each department, we would store the (unique) name of the department, and a (maximum) four-character code of the department (again, unique).

We would also store the list of faculty member of that department. For each faculty member, we store the following information of each: ID, name, e-mail, rank (either full, associate, assistant or adjunct).

Department offer courses. Each course has an ID which is a combination of the department code and a 4-digit number. Each course has a title (not necessarily unique), and a course description.

Every semester, courses are being offered. A course may have multiple sections being offered (each section is represented by a 3-digit number (from 001 to 999). Notice that not all courses are offered at each semester. (Every year there are 3 semesters: Fall, Spring and Summer). Each section of the course is assigned a faculty as the person-in-charge. We also need to record the number of students that enrolled in the course.

Learning objectives and outcome measurement

For program evaluation, each program is assigned a set of learning objectives (LO). There can be an arbitrary number of objectives. Each objective has a code (user can enter it, or else, the system should assign an automatic one). Each objective can have a set of subobjectives (which will be labelled with code.1, code.2, code.3 …, where code is the code of the LO). It also has a description of what the objective/subobjectives is about.

For each program, there are a set of courses that is associated with it that will be used for evaluation. For example, for the CS program, it may be the course CS 1100, CS 1200, CS 1300. And for Computer Engineering it will be CS 1100, CE 1100, CE 2200. Notice that a course may be associated with multiple programs.

For each (program, course) pair, there is a set of (sub)objectives that is associate to it. For every section of the course, we need to record whether each of those (sub)objectives has been meet. For each of those (sub)objective and each section, you need to record the following:

- How is the (sub)objective being evaluated. This can be things such as: exam, homework, interview etc.
- Number of students that met the (sub)objective.

Things to do

You need to design a relational database to store all the information. You need to store it in a relational database, using mySQL or MariaDB.

You will also need to develop an application that allow one to enter information to the database and retrieve information from it. Your application need to support the following operations:

- Data Entry
    - Enter basic information about
        - Departments
        - Faculty
        - Programs
        - Courses
        - Sections
        - Learning objectives / sub-objectives
    - Assigning courses to programs
    - Assigning learning (sub)objectives to (course, program) pairs (remember, a course can be associated with multiple programs, and for each program, the objectives can be different).
    - Enter evaluation results for a section.
- Querying. You should support the following queries:
    - Given a department:
        - List all its program
        - List all its faculty (including what program each faculty is in charge of, if there is one)
    - Given a program:
        - List all the courses, together with the objectives/sub-objectives association with year
        - List all the objectives
    - Given a semester and a program:
        - List all the evaluation results for each section. (If data for some sections has not been entered, just indicate that the information is not found)
    - Given an academic year (e.g. 23-24, which constitute summer 23, fall 23 and spring 24)
        - List all the evaluation results for each objective/sub-objective
            - For each objective/sub-objective, list the course/section that are involved in evaluating them, and list the result for each course/section.
            - For each objective/sub-objective, aggregate the result to show the number (and the percentage) of students

Implementation details

You are required to use either mySQL or MariaDB as the backend. (SqlLite is not allowed). You only need to host the database locally on your own machine. You program should read the user name/password/database name from a file that is to be included in the submission (You can name them whatever you want).

You are free to choose how to program all the functionality, but you should provide a standalone program to do the task. Under no circumstances should the user needs/be allowed to directly enter the database by the standard database GUI.

Your program should provide a (very basic) GUI for user's to enter data and queries. You can leverage a web browser to do it if you want to.

You are free to choose your language to implement your program.

Evaluation

Evaluation are done in 3 parts:

- Individual group demo. Each group will need to schedule a 30-minute time slot between 12/5-12/7 (Tue-Thu) for a demo session with me. I will be on campus 12/5 from 10:00am – 4:00pm. Otherwise we will need to schedule it via zoom. At the demo I will try to run your program (if via zoom, I will tell you what data to be entered). You should pre-populate your database with some data before the demo. The first part is for you to show your capability, and the second part is for me to test out the system.
- Group presentation. Each group will have a 10-minute presentation on 12/11 (Mon) from 11:30am – 2:30pm. For distance students, you will record a 10-minute video of your presentation and either upload the video or provide me a link (e.g. you upload it to YouTube and send me the link) and I will play it during the presentation.
  We will divide the time into 3 sessions (11:30 – 12:30, 12:30 – 1:30, 1:30 – 2:30), and you group will only need to be in the session that you are assigned. You should plan your presentation as a 'sell job' to highlight the capability of the system. You should also spend a couple of minutes to show that you have corrected the mistakes in the demo earlier. A schedule will be posted later.
- Final report. You should upload your final report as a zip file by noon 12/12 (Tue). You final report should contain:
  - The source code for your program.
  - A written report that contains the following:
    - The database schema, with comments on what each table represents (you can use the CREATE TABLE statements to describe the schema)
    - A brief installation/user manual to instruct someone (assume he/she is a junior CS student, but hasn't taken database yet) how to install and use the program
    - An implementation manual to describe the program. Describe the various functions/method/classes and what each of them do.

Notice that I expect all the functionalities to be implemented by the demo. If they are not correct, it can be corrected by presentation time.