CSE 4/560 Project1: TinyHub - Part 2

Soft Deadline: 23:59 Nov. 15, 2020 EST
Hard Deadline: 23:59 Nov. 29, 2020 EST

November 10, 2020

This is a group project for designing DB schema. Each group is consisted of 1 to 3 member(s). Please note that the requirements of functions will be augmented during next few weeks. This description contains THE FIRST AND SECOND PART of requirements, there will be THREE parts at the end of this project.

1 Description

You are to design and implement database schema for TinyHub, which is a course enrollment system. It provides simple functions, main functions for TinyHub are the following:

- User management
- Department management
- Course management
- Student-Course management

Your DB schema must be able to support all the functions listed above. You are required to use E/R modeling (an online E/R diagram tool is `https://www.draw.io/`) to design and present your database, and map your E/R model to a relational database schema and implement the mapped schema using a RDBMS by a set of CREATE TABLE statements. A recommended RDBMS is:

- MySQL(`https://dev.mysql.com/downloads/`): you need to first install MySQL Community Server to use MySQL, then you'll need one of MySQL Shell and MySQL WorkBench for SQL development. If you need a GUI, you may prefer to use MySQL WorkBench.

You also need to briefly explain how your model supports all the required functions. The final product of this is a report (.pdf file), of which the structure will be given in section 3 in this description, and a SQL file (.sql file type) which contains all the CREATE TABLE statements you used for implementing your RDB schema.

Note: you ONLY need to design and implement the database schema for the data needed to support the functions, you are NOT required to implement the functions.

2 Requirements of functions

The specific requirements for each function of TinyHub are given in this section. **Note that you need to analyze and design the entity types, attributes and relationship types**, while this project description only gives some of them for clarifying the system requirements.

2.1 User management

1. Users sign up their accounts with their email addresses, and we can use email address to identify a unique account.
2. Account can be of three types: Students, Professors, or Staffs.
3. Accounts need to set their passwords.
4. One account has an optional unique display name. Users are allowed to change their display name as long as it is unique.
5. One email address can be used to register only one account.
6. Email, password, and display names are strings.

2.2 Department management

1. Each department has an identifier department number.
2. Every professor is hired by one department.
3. Every staff is hired by one department.
4. Departments offer different programs, which a program can only belong to one department.
5. Students are allowed to major in different departments.
6. Students can pursue different programs, but they must major in the department which is offering that program. e.g., Given a data science program offered by Computer science department, a student needs to major in computer science before pursue a data science program.

2.3 Course management

1. Each department offers different courses.
2. A course can be opened in different semesters.
3. A semester is identified by a year and a season, e.g. 2020 Spring.
4. A course does NOT have to be opened every semester.
5. Every opened course has a capacity, which is the max student number in the class.
6. Every opened course has TAs, and one instructor. However, it is possible to change TAs or the instructor even during the semester.
7. There might be multiple opened courses from the same course in a semester. e.g. CSE 101 has two sessions on 2020 Fall, which one of them has 100 as the capacity while the other session has 120.
8. An instructor must be a professor.
9. A TA must be a student.
10. A course may contain prerequisite courses.
11. A student may register in different semesters.

2.4 Student-Course management

1. A student can enroll different opened courses. But that student can enroll a course only if

   - That student has registered in the semester and the course is offered in that semester. i.e. A student need to register 2020 Spring before enrolling any course in 2020 Spring.
   - That student passes all the prerequisite courses
   - It is being offered by a department they are majoring in
   - The capacity of the course is not full

2. Students will have a grade (F/D/C/B/A) with the course, which will be given after the student finishes the course.
3. Students can post feedbacks for the instructor of the course in which they enrolled.
4. Each course has one or more exams. Students who take that course have letter grades on those exams.
5. Each exam has a number of problems. Students have scores on those problems.

3 Report template

Here's a template of the report, your report **must contain** each of the following sections, you can extend this template to include any necessary sections for your design. *To have feedbacks on each soft deadline, the report only needs to contain the E/R schema.*

- E/R schema: in this section you need to introduce your E/R schema and include the picture of your E/R schema.
- Relational database schema: in this section, you must
  - discuss briefly how you map the E/R schema to your relational database schema. If any design choice is made in the mapping process, illustrate and explain it briefly.
  - discuss briefly how your relational database schema satisfies all the requirements listed in section 2.
- Further discussion: in this section, you need to discuss the advantages and disadvantages of your design.

4 Submission

Failure to comply with the submission specifications will incur penalties for EACH violation.

- What to submit: A zip file has to be submitted through the 'submit_cse460' (if you are CSE460 student) or 'submit_cse560' (if you are CSE560 student) submit script by 11/29/2020 11:59PM EST. Only zip extension will be accepted, please **don't** use any other compression methods such as tar or 7zip. You can submit multiple times, note that **only** the last submission will be kept on the server. We only need one submission per each group.
- Zip file naming: Use *ubit_proj1* (**NO SPACE!**) for the filename, for example: *jsmith_proj1.zip*, where *jsmith* is the ubit of submitter. The project is an **GROUP** project, so each group only need to submit ONE zip file.
- Sub-structure of zip file: On unzipping the zip file, there should be a folder named with your ubit *ubit_proj1*, under the folder *ubit_proj1*, there should be two files: (1) a .pdf report, (2) a .sql SQL file, and (3) a .txt file contains ubit from all group members with one ubit per line.