Phase 2: Building a Relational Database Management System

Due: 3:30pm March 31, 2025

What to Turn In:

- Turn in a printed copy of your selected PHP code of the queries you have implemented.
- Submit an electronic copy of Phase 2 of your project by "***submit malamin 3100-S25-phase2 your_files***" from cs.uml.edu
  Your submission should include:
  1. a README file about how to install and run your software, and your names
  2. all of your HTML, PHP and SQL code for Phase2
- Pre-record an 10-15 minutes video demo and upload it to discussion board (either the mp4 file or a link to YouTube video)
- Demonstrate how your project works during class on March 31

(one submission/demo per team)

NOTE: Do not use client-side script such as JavaScript, CSS, etc. Only use plain HTML and PHP.

Phase 2 of the project contains two parts.

Part 1: Create and populate the database:

- Create a database called "DB2" for user "root". Do not specify password for root.
- In "DB2", create tables based on TA's solution run "DB2-tables.sql" to Phase 1 and populate the tables with fabricated data.
  - You can create tables in PhpMyAdmin or download and run "DB2-tables.sql" in MySQL command line to create tables.
  - You should not delete any attribute or table from "DB2-tables.sql".
  - You may create additional tables and/or add additional attributes to existing tables in "DB2-tables.sql" (Do not modify course, instructor, and time_slot table.)
  - Each table should contain at least 5 records.
  - Do not insert more than 100 records into a table.

Put all "CREATE TABLE" and "INSERT INTO" statements in one ***.sql*** file.

Part 2: Perform the following tasks as queries (some queries may include several steps):

1. A student can create an account and modify their information later. (The accounts for admin and instructors are created in advance.) *(10 points*)
2. The admin will be able to create a new course section and appoint instructor to teach the section. Every course section is scheduled to meet at a specific time slot, with a limit of two sections per time slot. Each instructor teaches one or two sections per semester. Should an instructor be assigned two sections, the two sections must be scheduled in consecutive time slots. *(10 points)*
3. A student can browse all the courses offered in the current semester and can register for a specific section of a course if they satisfy the prerequisite conditions and there is available space in the section. (Assume each section is limited to 15 students). *(10 points)*

4. A student can view a list of all courses they have taken and are currently taking, along with the total number of credits earned and their cumulative GPA. *(10 points)*
5. Instructors have access to records of all course sections they have taught, including names of current semester's enrolled students and the names and grades of students from past semesters. *(10 points)*
6. Teaching Assistants (TAs), who are PhD students, will be assigned by the admin to sections with more than 10 students. A PhD student is eligible to be a TA for only one section. *(10 points)*
7. Grader positions for sections with 5 to 10 students will be assigned by the admin with either MS students or undergraduate students who have got A- or A in the course. If there are more than one qualified candidates, the admin will choose one as the grader. A student may serve as a grader for only one section. *(10 points)*
8. The admin or instructor can appoint one or two instructors as advisor(s) for PhD students, including a start date, and optional end date. The advisor will be able to view the course history of their advisees, and update their advisees' information. *(10 points)*
9. Student-proposed functionality #1 *(10 points)*
10. Student-proposed functionality #2 *(10 points)*

You should have one *.html* file as user-interface that takes user input. You may have one short *.php* file for every query or one long *.php* file for all the queries combined.