

Phase 2: Building a Relational Database Management System

Due: 11am March 28

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 yli2 3100-S23-phase2 your_files**" from cs.uml.edu
- Your submission should include:
- a README of how to install and run your software, and your name(s)
 - all of your HTML, PHP and SQL code for Phase 2
- Pre-record an 8-10 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 28
- (one submission/demo per team)

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

During Phase 1, you have designed the E-R diagram and relational schema of the database. In Phase 2, you will enforce the constraints and implement the desired functionalities as specified in Phase 1's project description.

Phase 2 of the project contains two parts.

Part 1: Create and populate the database:

1. Create a database called "DB2" for user "root". Do not specify password for root.
 2. In "DB2", create tables based on the solution to Phase 1 and populate them with fabricated data.
 - a. You may run "DB2-tables.sql" to create tables. You should not delete any attribute or table from "DB2-tables.sql". You may create additional tables and/or add attributes to the tables in "DB2-tables.sql".
 - b. Each table should contain at least 5 records except admin which may contain only one tuple. Do not insert more than 100 records into any table.
- Put all "CREATE TABLE" and "INSERT INTO" statements in one .sql file.

Part 2: Perform the following tasks as queries:

1. A user will be able to create an account. Students will be able to update their own accounts; parents will be able to update both their own and their children's accounts. Admin will be able to modify meetings and materials tables.
2. All users can find information about reading groups and past and future meetings.
3. Groups are determined based on grade levels. For example, there are an English group for 5-6 graders and a Math group for 4-5 graders, etc. A student may join more than one groups.
4. The admin can create meetings. All meetings are held on Saturday or Sunday. Each meeting is one hour long.
5. A student can join all future meetings or join only one meeting by the end of Thursday of the week for their group; The student can quit one meeting or all meetings at any time. Parents can sign up meetings for their children. Students can also be assigned to meetings by the admin. Each meeting can have at most 6 students.

6. Students will be able to see a list of other students of the meetings they are joining, including names and email addresses.
7. A meeting with fewer than 3 students will be canceled on Friday by the admin. All participants of the meeting will be notified of the cancelation. To simplify, you can generate notification files that contain the names and email addresses of the users who should be notified.
8. The admin will post study materials to each meeting every Friday for the following week. Students can see the current and past materials for each meeting they have enrolled in.

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.