NYU Polytechnic School of Engineering
Computer Science and Engineering Department
CS 3083
Prof. Phyllis Frankl

Feb 24, 2015

COURSE PROJECT PARTS I and II

Project Overview

The course project this semester is a `ClubHub`[1], a system for student clubs to post information about their events, and for members to join club, post comments, etc. All users can see public information about clubs and events. Registered users can log in to see and post further information or comments about clubs and events events sponsored by those clubs. Club members may have roles within the clubs that give them additional privileges, i.e. adding new club members, posting events, adding certain kinds of information, etc.

In part 1 of the project, you will design an Entity Relationship model for the database. In part 2, you will convert **my** E-R diagram (which I will post later) to a relation schema, write table definitions in SQL, and write some queries. Part 3 will be the most work: using **my** schema from part 2, you will revise your queries, if necessary, and write application code for the system. A detailed specification will be provided shortly after part 2 is due.

PART I

Draw an ER diagram modeling the data needed for the system. It should include entity sets representing `Person`, `Club`, and `Event`, possibly other entity sets, and several relationship sets. Remember to note primary keys of entity sets and cardinality constraints on relationship sets.

Each `Person` has a unique id, a password, a name, consisting of a first name and a last name, and an e-mail address (and, optionally, other attributes that you think would be useful).

There are two kinds of People – students and (faculty/staff) advisors. Advisors have phone numbers and students have gender and class (e.g. Freshman, sophomore, etc)

Each `Club` has a unique ID, a name (cname), and a description (and, optionally, other attributes that you think would be useful). Each club must have at least one advisor.

Clubs may have some keywords to help people find clubs they'd like to join. People can specify their interests (keywords), to help clubs find people who might want to join them.

Students can belong to clubs. A student may have some special role(s) within a club, such as president, treasurer, etc.

Clubs can sponsor events. Each event has a unique ID, a name (ename), a description, a date, a time, and a location, and an indication of whether it is public or just for club members (and, optionally, other attributes that you think would be useful).

People can sign up for events, can post comments about events, and can post comments about clubs. In some cases the system may restrict sign-ups and/or comments to club members. People making comments can designate those comments as being public or just for club members.

What You Should Do

Design an ER diagram for `ClubHub`. When you do this, think about: which information should be represented as attributes, which as entity sets or relationship sets? Are any of the entity sets weak

[1] Seemed like a cute name, but of course, it's already taken. So if you go out and found a start-up based on this project, you'll need to come up with a different name.

entity sets? If so, what is the identifying strong entity set? Is each attribute simple, composite, or multi-valued? Are there opportunities for specialization/generalization? What is the primary key (or discriminant) of each entity set? What are the cardinality constraints on the relationship sets? Think about whether a relation schema derived from your design would have redundant attributes, and, if so, go back and revise your design. (You do not have to hand in the relation schema for Part 1.)

- Draw the ER diagram **neatly**. You may draw it by hand or use a drawing tool.
- Write a short explanation of your design. Do not repeat things that are obvious from looking at the ER diagram, (such as "The Person entity set has attributes ID, name, ..."). Instead, explain any design decisions that you've made, e.g. "blah blah is represented by a relationship set because yada yada yada.").

Hand in a **HARD COPY** of your E-R diagram. MAKE SURE YOUR NAME AND YOUR PARTNER'S NAME ARE ON IT. Also hand in a scan or photo of the ER diagram electronically, as well as the explanation of your design.

PART II

1. Using the posted E-R diagram (*my solution to Part I*), use the procedures we studied to derive a relational schema, then write SQL CREATE TABLE statements and execute them in your database system. Remember to include primary key and foreign key constraints. (You don't have to hand in the schema diagram, only the create table statements, but you may find it useful to draw a schema diagram first.) Use the following data types:
2. Based on this schema, write SQL `CREATE TABLE` statements. Remember to include primary key and foreign key constraints. Use the following attribute domains:
   - `Person.pid` is varchar(30)
   - The ID attributes in all the other tables are integers. Optionally, you can designate these as `autoincrement`.
   - `Person.passwd` is char(32). The stored value will be an md5 hash of the user's plain-text password.
   - `Event.edatetime` is a DATETIME (or something equivalent if the DBMS you're using doesn't have this type.)
   - Advisor.phone is char(12) (for 10 digit phone number + two hyphens).
   - `is_public_c` and `is_public_e` are Boolean or int.
   - All other attributes are varchars of some reasonable length.
3. Write SQL INSERT statements to capture the following situation:
   (a) There are two clubs, the frisbee club (clubid = 1, advisor's pid = 'Ann') and the theater club (clubid = 2, advisor's pid ='Bob').
   (b) There are at least four people, two faculty, Ann and Bob, and at least two students, including one with `pid` "Pedro". For simplicity when we test, the plain-text of each Person's password is the same as their `pid`, so the stored `passwd` is the md5 hash of that string. Use the md5() funciton to compute it, e.g. `INSERT INTO Person VALUES ('Ann', md5('Ann'), ...)`. One student belongs to both clubs and one just belongs to the theater club.
   (c) Each student is interested in some topics and each club is about some topics.
   (d) Frisbee club is sponsoring at least one public event. Theater club is sponsoring at least one public event and at least one private event.

(e) Each of the two students is signed up for at least two events that he or she is eligible for.

(f) One student has posted at least two comments about events and another student has posted at least one comment about an event.

(g) Use reasonable values for the attributes that aren't specified, such as names, phone numbers, comment texts, etc. (Do not use any inappropriate language in these.)

(h) Include a comment with a few sentences explaining the data, along the lines of "Sam is a member of both clubs; he has signed up for event 3 (event name) and event 5 (event name). He posted comments X and Y about Z.".

4. Write queries to find the following:

   (a) People with username 'Pedro' and password md5('Pedro')

   (b) Events (eid, title, date and time) in the next seven days that the user with `pid` 'Pedro' has signed up for. (Hint: Look at the MySQL documentation for DATETIME type and the functions it supports.)

   (c) Clubs that Pedro belongs to.

   (d) For each upcoming event of the theater club, list the eid, the name, and the number of people who have signed up.

   (e) Check whether the event with eid 3 is public or whether Pedro is a member of the club that's sponsoring that event.

   (f) **For team projects:** note how the schema would change if events can be co-sponsored by multiple clubs and write a query to check whether the event with eid 3 is public or whether Pedro is a member of any club that's sponsoring that event.

   You are strongly advised to add some more data to the tables in order to test your queries thoroughly, but you do not have to hand in the results of this additional testing.

What to hand in for Part II:

Use electronic handin to hand in a single text file called `ClubHubPart2.txt`, with the CREATE TABLE, INSERT, and SELECT statements in that order. We should be able to execute this file, so comment out anything that isn't SQL. The comment delimiter is double hyphen. If you're working with a partner, be sure to include both of your names as comments.

The total project grade will be 25% of your course grade. Part 1 counts for about 15% of the project grade. Part 2 counts for about 10 to 15% of the project grade. There may also be a quiz or exam question(s) based on the project.

Instructions for working with a Team

You may work alone or with a partner. You may work on part 1 alone then choose a partner for parts 2 and 3, but you may not switch partners. If you work with a partner, you will be required to add some extra features (which I will specify) to the application, but the total lines of code per person will be substantially less than if you work alone. Teams are encouraged to try pair programming, rather than simply dividing up the work. I may consider allowing three person teams, but any such teams will need to extend the project in a substantial and instersting way. **If you are doing part 2 of the project with a team, and did not already notify us, you must notify me by sending e-mail to the grader by March 2.** His address is posted in the Syllabus section on Classes.

Teams will be required to submit a work plan, indicating who will do what, and will be required to submit an evaluation at the end. Note that each partner is expected to contribute roughly equally to each aspect of the project and each partner is responsible for understanding the entire system. Normally all team members will receive the same grade, but I may deduct points from individuals who are not pulling their weight on a team.