R·I·T	Rochester Institute of Technology Golisano College of Computing and Information Sciences Department of Information Technology

4002-484/784 Fundamentals of Database Client Services

Course Project

Faculty Activity Database

Background

Faculty are evaluated annually based on a variety of factors. These include their teaching, their publications and grants (scholarship), their service activities, and other miscellaneous things. Every professor is responsible for tracking his/her activities and reporting them at the end of the year. The problem is that items are often forgotten and not reported. A database application might help address this problem. If we were to store all this information as it happened, it would be easy to generate a report at evaluation time.

Overview

Your job is to develop a system that will allow these users to perform functions that are appropriate for them. Users for you to consider are:

- Faculty
- Chair
- Administrative Assistant

Ideally, this would be an online system. However, if your group does not have expertise developing in PHP (or similar), you may develop a desktop system. There is no "extra credit" for doing an online system so do not take it on unless your group is more comfortable with that platform than with Java or .NET. It is more important that you adhere to good design principles.

Overall Requirements

- One database will be shared by all users.
- Faculty members will have the ability to review their entries for teaching, service, and scholarship.
- An Administrative Assistant will have the ability to review, edit, or insert entries for teaching, service, and scholarship.
- The chair will have the ability to review, edit, or insert all entries.
- The interface must be intuitive and very easy to use

Starting Point

You will be given a populated database upon which you can base your work. The database will be a MySQL database; you may convert it to Oracle if you wish. You may not use Access. If you have another DBMS in mind, talk to me about it.

You may change the structure of the database if you wish, but be sure you document your changes.

Deliverables

The following schedule for your deliverables may seem a bit odd to you, but it really does make sense. You are being required to submit documentation before you submit code. That means you have to think about what you want to create before you create it. Failure to meet a due date will result in 10 points being deducted from your project grade! This includes peer reviews.

Deliverable	Contents	Due Date
1	Design Documentation	Oct. 13, 2012, 11:59 PM
2	Database Abstraction (and Business) Layer	Oct 20, 2012, 11:59 PM
3	User Documentation	Oct. 27, 2012, 11:59 PM
4	Presentation Layer	Nov. 3, 2012, 11:59 PM
5	Final Code and Docs	Nov. 10, 2012, 11:59 PM
6	Peer Reviews	Nov. 14, 2012, 5:00 PM
7	Presentation and Demo	Nov. 14, 2012, 8:00 AM or sooner

Deliverable details:

- Design Documentation includes
    - a sketch of the overall architecture
    - choice of languages for each layer
    - rough sketch of the user interface(s)
    - description of key functionality
    - identification of areas of particular note or concern
- Database Abstraction Layer includes
    - all code for working with the database
    - code should be appropriately commented
- User Documentation includes

        - description of the functions available and how to use them
        - sketches and menu trees should be included
        - this document should be user-readable
        - this document will be the design spec for the Presentation Layer
    - Presentation Layer includes
        - all code for interacting with the user
        - interface should be well-designed
    - Demo
        - presentation by your group
        - not everyone has to speak

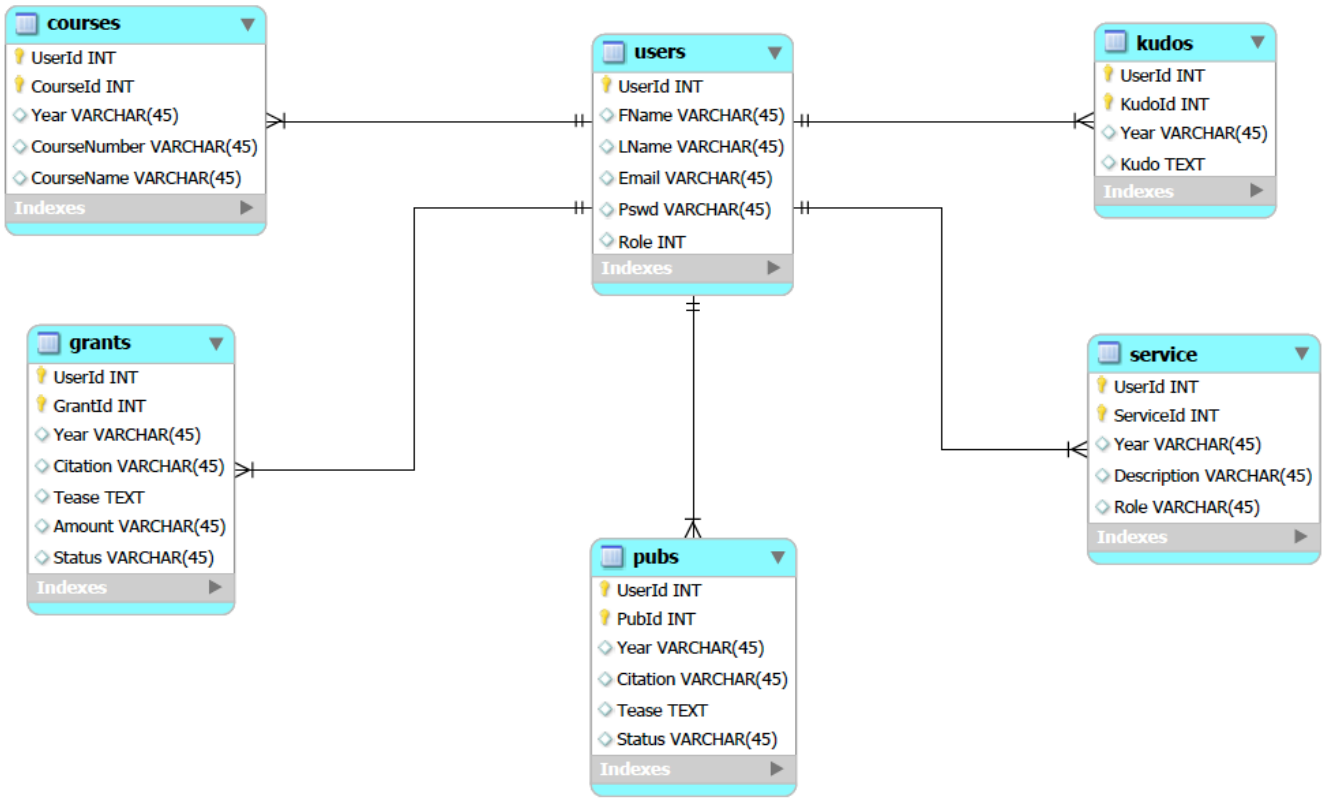
Additional Info

- You must use a layered architecture with at least a total isolation of the database and presentation layers.
- You may talk to me or anyone else in order to learn more about the user requirements.
- You will be required to demo your work at the end of the course.

Potential Functional Requirements

- Minimum functionality will get you a B at best
- You should consider adding additional functionality. Examples include:
    - Complete security system
    - Admin control of lookup information

Database Schema



Data Dictionary

The **users** table holds basic user information. It is keyed on UserId—a simple integer counter. A password and email field are included for authentication purposes. If you opt to use LDAP, the password field becomes unnecessary.

The child tables all contain a field named **year**. This is the result of denormalization. A fully normalized structure would include a table for academic year and then have all these other child tables hanging off of it. You can leave the structure as-is, or change it.

The **courses** table contains information regarding what courses this person taught. It is double keyed on UserId and CourseId—a simple integer counter. Course number should be the full course number, e.g. 4002-484-01.

The **grants** table contains information regarding what grants this person has applied for and received. It is double keyed on UserId and GrantId—a simple integer counter. Tease is a short paragraph that describes the purpose of the grant. Status will be either "Submitted", "Pending", or "Awarded".

The **pubs** table contains information regarding what publications this person has. It is double keyed on UserId and PubId—a simple integer counter. Tease is a short paragraph that describes the purpose of the publication. Status will be either "Submitted", "Pending", or "Published".

The **service** table contains information regarding what service activities the person has engaged in. It is double keyed on UserId and ServiceId—a simple integer counter. Description is a short description of the service, e.g., "Eisenhart Committee", or "Tenure Committee". Role indicates a person's role in the activity, e.g., "Chair" or "Member".

The **kudos** table is for everything else, both positive and negative. It is double keyed on UserId and KudoId—a simple integer counter. Description is a short description of the kudo, e.g., “Came in to work Open House”, or “Skipped faculty meeting on 10/1/12”.