

Database Application Development

Due date: Last day of classes (in the assignment box and electronically to an email to be announced)

Value: 15% of final marks

Price: best project gets ONE copy of a DB text book

The focus of this project is to do conceptual and logical database design, to create a JDBC-based code that accesses a PostgreSQL database, and to create parts of the application that reside at the middle and presentation tiers of the three tier architecture.

Description of the requirements — The University of Canada needs to store information about its researchers, areas they work in, volumes they edit, residences they live in, papers they write, and graduate students that are supervised by the researchers that are professors. Consider the following information about UofC:

- Every researcher has an SIN, a name, a salary, and a hiring date.
- Each researcher lives at an address with a phone number. Researchers may share the same address. The later may have more than one phone numbers, but no phone number can belong to two different addresses. You must assume here that telephones have numbers and reside at some location that has an address.
- Researchers specialize in various areas: a researcher must specialize in one or more areas, and an area may, on the contrary, be without specialist at UofC.
- An area has an area identifier, a name, a description, and a key journal.
- A researcher may have edited zero or more volumes. A volume may have been edited by researchers of UofC.
- A volume has an ISBN number, a title, a classification (e.g., proceedings, book, journal, etc), a publisher, and a publishing date. Notice that UofC keeps track of all volumes that have appeared in the literature.
- Each researcher may have written several papers, and a paper may have several authors from UofC. A paper has an identifier, a title, an abstract, and a bibliographic entry.
- Each paper must appear in at most one volume, and a volume may contain papers written by UofC researchers.
- There are two categories of researchers, namely professors and graduate students. Professors are in turn subdivided into two categories: untenured and tenured. Graduate students have a student identifier, a cgpa, a prospective graduation date, and thesis topic. Untenured professors are characterized by a rating used for their promotion to tenure. Finally, tenured professors have a tenure date, and responsibilities.

- Each graduate student must have at most one professor as supervisor, and each professor may supervise several graduate students.
- An untenured professor must have at most one tenured professor as a mentor, and a tenured professor may mentor several untenured professors.

Based on the information given above, do the following.

1. **(30 points)** – Draw an ER diagram for the UofC database. Doing so, indicate the various attributes of each entity and relationship set; also specify the key and participation constraints for each relationship set. Specify any necessary weak entity sets, looping relationship sets, aggregations, and hierarchies that might be present in the requirements. Translate your ER diagram into a relational database schema by writing a PostgreSQL SQL script. Use appropriate constraints in your SQL script. (Submit this deliverable by Oct 20, 2006 at 6PM)
2. **(50 points)** – Write JDBC code that performs database access and manipulation. The interface to your system should offer at least the following functionality:
 - (a) A user can search the records stored in the database by the names of researchers, the titles of volumes, and the titles of papers. For a nice example of a terse (but elegant) interface for a similar web-based information system, see the “search” rubric at <http://www.informatik.uni-trier.de/~ley/db/>.
 - (b) Note that a user can run appropriate queries against the different relations that constitute the database schema. Find such appropriate queries and provide an interface for them. Such an interface will comprise further meaningful search predicated that you may imagine for your the database schema.
 - (c) The Java interface to your system should be functional, and appealing.

(Submit this by the last day of classes)
3. **(20 points)** Furthermore, design a (simple) web-based presentation tier, and a middle tier using one of the technologies available: e.g. HTML pages, CGI scripts, Java servlets, JavaServer pages, or something else. Integrate code used for the Java-based part. Design your web pages by following the hints below:
 - Users should log on to the site. An user that has already logged on cannot log on again. A JSP cookie could be used to implement this functionality.
 - Read Sections 6.6 and 7.8 to get an idea of how to design the architecture of your system.

(Submit this by the last day of classes)

Some Resources: Links to material related to JSP, Servlets, XML, JDBC, HTML, etc:

http://www.cs.wisc.edu/dbbook/openAccess/www_resources.html
<http://www.java.sun.com/products/jdbc>
<http://www-db.stanford.edu/~ullman/fcdb/oracle.html>
<http://www.utoronto.ca/webdocs/HTMLdocs/Book/Book-3ed>
<http://www.postgresql.org/docs>

To submit by the last day of classes: A CD containing

- your ER diagram, SQL script, and Java code;
- any further CGI code written or the like.

Also, you will demonstrate your system to a TA at the end of term on a day to be determined.