CS 564 / Spring 2015: Course Project

Project Description

Students in teams of 3 to 4 are required to select an application that needs a database and build a database application from start to finish.

Goals

Finding an application for which database systems would be required  
Modeling the domain of the application, and defining the application functionalities  
Designing and implementing the schema  
Populating the database (this should not be the main focus of the project)  
Writing the code needed to embed the database system in the application.

Schedule

There are a number of intermediate deadlines that you must meet in order to ensure a successful project.

- Fri Feb 13: Group information due (Stage 1)  
- Wed Mar 4: Initial ER design due (Stage 2)  
- Fri Mar 20: ER translation due (Stage 3)  
- Fri Apr 10: SQL queries (Stage 4)  
- Fri Apr 24: Screenshots of the application (Stage 5)  
- Thur-Fri May 7-8: Application demo (Stage 6)

Suggested Application Domains (subject to change)

You are free to propose domains that you want to work on, as long as you think the domain that you want to work with will have to manage a lot of data and that this data will best be managed in an RDBMS. You can also pick one of the suggested domains below:

Movies domain:

In this domain you would be modeling entities movies, their actors, directors, genres, playing times, reviews, and so on. There exist several sources on the Web from which you can get data to populate such a database. You can support various queries such as finding specific playing times, finding movies playing in Madison, directed by a given director. You can also support updates to the reviews section of the database (e.g., viewers giving their own opinions). Another functionality is to provide personal profiles of people (i.e., the movies they like) and then try to recommend movies to them based on profiles of viewers with similar tastes.

Books domain:

In this domain you would be modeling entities such as books, their authors, topics (which may be a complex hierarchy). You may also model various attributes of the authors, the institutions they belong to, etc. You can support a buy/sell service of used books, books used in specific university courses. A personal profile, similar to the one for movies is also a possibility. Pointing an interested buyer to a web source to buy a book is also an interesting option.

Apartments:

This domain would require modeling apartments and their properties, areas of town and their various properties (e.g., bus lines, crime rate distance from various landmarks). You would provide an interface for offering apartments for rent, finding apartments.
Figure 1: Show all assignments in course CS 739.
Figure 2: Course table before inserting CS 564.

<table>
<thead>
<tr>
<th>courseid</th>
<th>course</th>
<th>year</th>
<th>semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>CS 729: Distributed Systems</td>
<td>2014</td>
<td>Fall</td>
</tr>
<tr>
<td>1</td>
<td>ECE 392 599 599: Rao Research Group Meeting</td>
<td>2014</td>
<td>Summer</td>
</tr>
<tr>
<td>2</td>
<td>CS 728: Context-Level Programming</td>
<td>2014</td>
<td>Fall</td>
</tr>
<tr>
<td>3</td>
<td>CS 736: Advanced Operating Systems</td>
<td>2014</td>
<td>Fall</td>
</tr>
<tr>
<td>4</td>
<td>CS 752: Advanced Computer Architecture</td>
<td>2014</td>
<td>Fall</td>
</tr>
<tr>
<td>5</td>
<td>CS/ECE 752: Advanced Computer Architecture I</td>
<td>2014</td>
<td>Fall</td>
</tr>
<tr>
<td>6</td>
<td>CS 646: Introduction to Computer Networks</td>
<td>2015</td>
<td>Spring</td>
</tr>
</tbody>
</table>

Figure 3: Course table after inserting CS 564.

<table>
<thead>
<tr>
<th>courseid</th>
<th>course</th>
<th>year</th>
<th>semester</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>CS 578: Context-Level Programming</td>
<td>2014</td>
<td>Fall</td>
</tr>
<tr>
<td>3</td>
<td>CS 736: Advanced Operating Systems</td>
<td>2014</td>
<td>Fall</td>
</tr>
<tr>
<td>4</td>
<td>CS 752: Advanced Computer Architecture</td>
<td>2014</td>
<td>Fall</td>
</tr>
<tr>
<td>5</td>
<td>CS/ECE 752: Advanced Computer Architecture I</td>
<td>2014</td>
<td>Fall</td>
</tr>
<tr>
<td>6</td>
<td>CS 646: Introduction to Computer Networks</td>
<td>2015</td>
<td>Spring</td>
</tr>
<tr>
<td>7</td>
<td>CS 564: Database Management Systems</td>
<td>2015</td>
<td>Spring</td>
</tr>
</tbody>
</table>
Figure 4: Course table after deleting CS 564.

Figure 5: WorksOn table before and after inserting grades for a student.
Figure 6: Querying a student’s average grade of a course.

```sql
fooboard=> select avg(grade) from worksOn w, assignment a where w.email = 'KaiZhao@wisc.edu' and a.assignmentId = a.assignmentId and a.courseId = 7;
               avg
            ---
           | 37.72
(1 row)
```

Figure 7: Querying all the due dates of assignments of CS 564.

```sql
fooboard=> select dueDate from assignment a, course c where c.courseName = 'CS 564' and a.courseId = c.courseId;
               dueDate
            ---------
            | 2015-01-20 18:25:29
          1 row
```

4