COMP3013 Database Management System
Course Project Requirements
2024 Fall

Introduction

This is a group project for four to five students.
This project is to build up an dynamic website for a real-world application. It can be an information system for beauty parlor, kid's training institutes, fitness centers, restaurants, etc. The topic is unlimited. You are free to choose anything you are interested.

Requirements

Front End Requirements

1. Your website needs at least ten pages.
2. The static pages can be implemented by HTML + CSS, while the dynamic contents can be handled by PHP and Javascript.
3. Your website may include navigation bars, logo, footers, data validation, etc.
4. There is no requirement on the opensource platforms. Using them can give you a fancy look to your website, but no extra marks.
5. Your website needs to be deployed on an Apache server. Use XAMPP.
6. Your website should allow user registration and login. At least two user types should be offered. For example, suppose you are building a bookstore website, your user types could be:
   - owner
   - registered user
   - anonymous visitor

7. Your website needs to include at least four features. The more features you have, the higher score you will earn. Feature is a workflow that can allow the user to perform a complete task. Take the example of bookstore, the features could be:
    - The owner add new books to the store.
    - The owner view purchasing data analysis, etc.
    - The user orders books and leaves the books in the shopping cart.
    - The user check out and pay.
    - The owner, the user and the visitor search for books.

Back End Requirements

1. Your ER diagram should have at least 8 entities and 6 relationships.
2. On average, each table must have no less than 10,000 records. There are at least two tables consisting of more than 50,000 records. (Note: Not all the records have to be real but they should be realistic. You may generate records using a program.)
3. Your system takes less than 1 second to insert a record to any table.
4. Your system takes less than 2 seconds to delete or search a record.
5. Display the running time of your query on the webpage. You can use *microtime()* function in PHP.
6. The logical design of the database must follow the normal forms (**BCNF** or **3NF**).
7. Use foreign keys. For example, the table *student* has students' information and the table *registration* has courses registered by each student. If one tuple is removed from *student*, the corresponding tuple(s) will also be removed from *registration*.

Presentation Requirements

1. 10-12 minutes for each group.
2. Make some good slides.

Documentation Requirements

1. Briefly introduce the purpose of this project. You should define which real-life problem you are solving, address the difficulty of the problem, give the abstraction of the problem, and major goal of the project.
2. List all front end functions that you have implemented.

3. To design the database, you may need to make some assumptions for modeling because real-life problems sometimes do not provide enough information. But, all your assumptions have to be realistic.

4. Your report must include your final ER diagram and a brief description of each entity and relationship set.

5. You need to give all functional dependencies and schemas.

6. If your schemas are in the normal forms, explain why. If not, you need to decompose them and show the steps in detail.

7. You need to describe the primary keys.

8. Workload of each team member.

Bonus (1% for each)

1. Use BLOB to store pictures.

2. Use triggers to implement constraints other than *not null, primary key, unique,* or *referential.* For example, the table *student* has an attribute *credit* showing the total credits earned by each student. Then, if a student finishes a course of 3 credits, his/her *credit* will be increased by 3 automatically.

3. Secure the connection. The administrator account of the database system has the highest authority of everythin in databases. If you use the administrator account to make the connection, it is not safe at all. For the security issue, your database connection can only access partial information subject to users authority.

Tips

For presentation and report:

- The presentation should follow the top-down procedure. Please start your presentation from the problem definition and assume that audiences know nothing about your project.

- Describe your database design on a high level, ER diagram for example.

- Highlight the critical points, constraints and triggers for example.

- Prepare some good slides.

- Do **NOT** make your presentation as function demonstration.

- Time is limited. You cannot show all details in a short presentation. Attract people's attention by some shining gold.

- A simple instruction **for presentation** can be found at “Project Presentation Guideline.docx”.

For web design and implementation, you are referred to w3school. A set of startup codes is given in “DBPJ_example.zip”. You can try to get it run first.
For XAMPP, Apache, phpMyAdmin, you are referred to Apache Friends. A simple tutorial is also given in “SQLLab.pdf”.
For generating data, see “DataGenerator.zip”.

Timeline

To make sure that nobody starts working on the project in the last minute, we setup the following schedule.

1. **Oct. 20** - grouping. Students need to select teams on iSpace in 2 or 3 people. Then, we will **randomly** pair two teams as one group of 4-5 students. The number of teams of 2 or 3 people is limited on iSpace. First come first serve! The grouping link will be available at **8pm Oct. 17** on iSpace.
2. **Nov. 10** - a problem description (5%, another 5% will be with the final submission), about 1-2 pages long (11pt font size, single spacing, and default margin). You should define the problem that you are solving, why you are interested in this problem, the information that you want to model in your system, and the features that your system will include.
3. **Nov. 24** - the first draft of the ER diagram (5%, another 5% will be with the final submission).
4. **Dec. 1** - static webpage implementations (10%, another 10% will be with the final submission). You only need to submit HTML and CSS files, which is the structure of the information system. All dynamic contents are replaced by static pages.
5. **Dec. 11∼17** - project presentation (20%). And in the night before it, you need to submit your presentation slides The presentation will be scheduled for sections differently.
6. **Dec. 22** - the final submission, which includes
   - the report,
   - the finallized ER diagram,
   - webpages, and
   - your database (exported from the database server in the format .sql), only one file for all tables. Please double check your file because we cannot give you marks if your database cannot be imported into the system.

7. Each partial submission worth 50% of the corresponding field except the presentation. If you miss the DDL, you will lose marks.

8. Don't worry if your partial submission is not perfect. You can still improve it before the final submission and fight for the other 50%.

Grading

- Report 30%
  - Problem Description - Partial Submission 5%
  - Problem Description - Final Submission 5%
  - ER Model - Partial Submission 5%
  - ER Model - Final Submisssion 5%
  - Logical Design - Final Submission 5%
  - Report Writing - Final Submission 5%
- Presentation 20%
  - Organization 10%
  - Oral Language 10%
- Website Implementation 20%
  - Partial Submission 10%
  - Final Submission 10%
- Database Implementation - Final Submission 30%
- Bonus 1% for each

One group only submits **one copy** to iSpace **by the teamleader**. For partial submission, rename your files as "COMP3013_24F_Group###_filename", where "filename" is one of the report, ER diagram draft, or PPT, and ### is your gourp number. And also rename your final submission package as "COMP3013_24F_Group###".