

Final Project

CSSE 333—Introduction to Database Systems

This document describes the requirements for your final project and lists the project deliverables and deadlines.

Team Composition

Teams will consist of about 3 members each. We are willing to allow smaller teams, even “teams” of one*, if you have a good reason for wanting to do so. We may consider allowing larger teams, but would require a very strong argument and a very interesting project idea before allowing this.

Because a significant amount of class time will be devoted to project work during the last third of the term, we require that all members of a team attend the same section.

Project Selection

Software engineering is a creative endeavor and can be a lot of fun. We want you to choose a project that is interesting to you. We impose some constraints below so that the project will also let you demonstrate your skills relative to the learning objectives of the course.

Each team will initially have a project idea, based on the Lightning Talks. Your team’s first task is to work with the instructors and assistants to refine that idea so that it satisfies the constraints below.

Database Complexity

The database for your project should include at least 6 tables. The data in these tables must be sufficiently interrelated so that there are valid foreign key relationships between the tables. You also must include appropriate primary and secondary indexes on your tables, and justify the design of your indexing scheme. You must include appropriate views and stored procedures in your database, or else argue (in your final report) why these are not necessary.

Data Import

Your project should have some initial data to work with. This data may come from some “real-world” source or may be generated randomly.

In addition to this initial data, your project must also deal with importing data after the work is underway. This simulates the common situation where a customer will request a new database, but will continue gathering data in the old format while the database is being developed.

Your data import does not have to be automatic. You can manually manipulate the data in a text editor or Excel, for example, before adding it to your database.

You have two options for this requirement:

* Solo efforts are strongly discouraged. In the past all students who worked alone indicated that they wished they had done otherwise.

1. You may identify an on-line source of data that changes regularly. In this case, you must load your database with data from the source initially, and then add additional data from this source later in the term.
2. Alternatively, you may just specify the schema for your data. In this case, you must invent initial data for your database. Later, you will create some additional data based on your schema and import it into your existing database.

It is very likely on real-world database projects that the customer-supplied data will not exactly follow the specified requirements. Whichever option you choose for data import, you should expect that you'll have to manipulate the additional data to make it fit in your database.

We will ask you to provide reports from your database before and after data import. These reports should demonstrate that you successfully imported the data. We will also ask you to document how your system guards against the entry of invalid data.

Interface

Your project must include an end-user interface that works with the database. This should include screens that let users add, change, and view data from the database. Your interface might also include a program that uses the database but doesn't necessarily expose the user to the fact that you have a database. For example, in a multiplayer role-playing game the interface might let users explore some world even though the user doesn't know that internally that world is represented by a database.

Your interface may be written in whatever language you choose. However, we recommend Java and JDBC so that we can be of the most help if you have questions.

Please note that although we are requiring an end-user interface, this is still a databases project. Be careful not to spend so much effort on your interface that you shortchange the database aspects of your project. Doing so might negatively affect your grade.

Deliverables

Below is a list of the deliverables that your team must produce, along with the deadline for submitting the deliverable. Following each is a brief description of what is required.

All written deliverables are to be uploaded, one copy per team, to the appropriate drop box in Angel. Drop boxes will close at midnight on Fridays. Late submissions will not be accepted.

Because deliverables are cumulative, we want you to do well on each. Thus, any deliverable that scores below 70% may be resubmitted. Resubmissions are worth at most 90%.

Initial problem statement

End of 5th Week

Your team must develop a problem statement for your project. Your statement must follow the Function-Form-Economy-Time model. A template for this model and a sample document are available on Angel. Your document should be about 2 to 3 pages long. We will grade your initial problem description and perhaps suggest some changes. You will turn in a revised problem statement at the end of the project.

Entity-Relationship diagram

End of 5th Week

Your team must submit an entity-relationship diagram for your database.

Relation schemas diagram for the database

End of 6th Week

Your team must submit a diagram showing the relation schemas for your database. The document should give the relations, their attributes and primary keys, and show the foreign key relationships between the relations.

Security and data integrity analysis

End of 6th Week

To help you consider the privacy, security, and data integrity issues involved in database design, your team must submit a short document (3 to 4 pages) discussing these issues.

With regard to privacy, discuss what personal data might be stored in the database. Try to think broadly about what various stakeholders might consider to be personal data. Discuss what limits should be placed on different users of the system with regard to viewing various kinds of data in your system. For example, if your system stores Social Security numbers, you would discuss which users should have access to that data and why.

With regard to security, discuss ways in which the security of your database might be breached. Discuss the implications of a security breach on the integrity of the data in the database and on the privacy of personal information. Discuss the effect of a security breach on the various stakeholders of your system.

Finally, discuss the data integrity constraints that your system must enforce. This should include a discussion of entity integrity and referential integrity, including the specific constraints that you are applying to your database in SQL Server. If there are business logic constraints between tables (like adjusting inventory when products are shipped to customers), then you should discuss this. Finally, you should discuss how your system guards against the entry of invalid data.

Front-end prototype demonstration

During 7th Week

In the past some teams have had very rough 10th week experiences because they postponed all work on their user interface. To help you avoid this, we are requiring demonstrations of front-end prototypes early in the project. Your demonstration should show at least one screen, web page, dialog box, or command-line interaction (whatever is appropriate for your project). Your demonstration should also show that your user interface can successfully connect to your back-end database.

Prototype demonstrations can be done during lab time in class. In order to complete the demonstrations quickly, you should rehearse yours beforehand.

Sample reports

End of 8th Week

Based on your relation schemas and your data integrity analysis, we will ask for some specific reports from your database. We will be playing the role of a customer in this respect.

You should create appropriate views or stored procedures in your database for generating these reports, as we will require updated versions of the reports later in the project. You should also turn in the SQL code to create your views or stored procedures.

Sample reports after adding new data

End of 9th Week

After generating the sample reports required above, you will be importing additional data into your database. Depending on which option you chose for Data Import, you will either import additional data from an external source, or else we will provide additional data based on your relation schemas.

After importing the data you must submit updated versions of the sample reports.

Final problem statement

End of 9th Week

Your team should turn in a revised version of your problem statement that incorporates the changes we suggested in response to your initial problem statement. We will compare your initial and final problem statements and will also use your final problem statement in evaluating your project.

Final project presentation

During 10th Week

Each team will give a short presentation on its project in class. Presentations will tentatively be about 10 minutes long and should include some time for questions. The exact length of the presentations may change slightly depending on the total number of teams.

Your presentation should focus on clearly defining the problem and explaining the key challenges you faced. You should provide a brief description of how you overcame those challenges. You should also give a *brief* demonstration of your system or show some screen shots.

Program demonstration

During 10th Week

Your team must schedule a time during 10th week to meet with your instructor and demonstrate your program. You'll have about 15 minutes for your demo. You should plan on showing your end-user interface and on walking through your back-end database.

We'll be particularly interested in:

- admiring your handiwork and
- seeing how your project satisfies all the constraints listed under the Project Selection section above.

Final project report

End of 10th Week

Your team must prepare a written final report on your project, not to exceed 5 pages.

Your report should include: a description of the problem (derived from your problem statement) and your solution; a discussion of the key challenges you faced; and a description of your database design, including the integrity constraints that you applied, your indexing scheme, any views or stored procedures used, and a discussion of the pros and cons of your design.

Your final project report should be self-contained. It should not assume that the reader is already familiar with the problem that you are solving.

XML schema ~~_____~~ **End of 10th Week**

~~Your team should prepare an XML schema for a *portion* of your database that might be shared with external customers. The schema should be in the XML Schema language (see section 27.2.2 of the text).~~

Peer evaluation forms **End of 10th Week**

Each team member must independently complete a form evaluating his or her contribution to the team, and the contributions of the other team members. We will provide this form.

Grading

Your project will be graded based on the quality of all deliverables. We will not necessarily assign the same grade to each member of a team. We reserve the right to adjust individual grades up or down based on peer evaluations and our observations about your teamwork. A team member can hurt his or her team either by being a slacker or by running roughshod over other team members. Thus, taking over a project and doing all the work might actually hurt your grade.