

CSC 4710/6710

Final Project Description

In the final project you are assigned to design and implement a database using a relational database management system (MySQL). The phases of the project will be described in the following section. The project will be an individual one and a grading rubric is given in the end. Before you start your project, you are expected to pick a mini-world for which you will design a database.

Phase I

The first part of the database design is getting the requirements from your client. However, you do not have any clients. Therefore, you need to research the data you will be storing. Knowing your data is one of the most important parts of your design process. Note that we do not want oversimplified databases, but we also do not want vastly complex databases¹. In this phase, decide on which real world entities/objects/concepts you will have information about, the attributes of these entities and the relationships between/among these entities. Additionally, you will need to list your functional requirements (you will turn these into queries/views/triggers in the later phases).

After Phase 1, the first part of your “Project Report” will be ready. It will include an overview about the database you are designing. Then, you will identify and list all the requirements. It is expected that you write a short introduction about your mini-world and list your data and functional requirements.

Phase 2

The second part of this project is about creating your ER Model (conceptual). You will use EER (Extended Entity Relationship) diagrams in this phase. Using the requirements you have identified in Phase 1, you will first determine the entities, then the attributes of these entities and the relationships among them. After that, you are expected to identify key attributes of entities, the minimum and maximum participation constraints of the relationships as well as your specialization/generalization constraints, if you have any. When drawing your ER model, you must use the min-max notation, which we have covered in the class.

¹ Just to give a hint of what is expected – the number of entities/relationships in your database should be about 10. We can have exceptions on these constraints, if you bring an innovative topic.

Phase 3

In the third phase of the project, your assignment is to perform the mapping between the conceptual schema (ER Model) to relational schema. My suggestion is following the steps shown in the class one by one. Some of the mappings may be tricky and can be done in multiple ways. For such, do not refrain asking help from the TA and/or me. Note that it is always better to map your regular entities, then weak entities, and relationships.

Phase 4

After creating the relational model, your next task is to create a data dictionary. A data dictionary defines the basic organization of the tables which are to be stored in the database. In a data dictionary, the tables, the attributes (or columns) of the tables are demonstrated. The relationship between attributes are also shown (foreign key relationships). Different from the relational model, in a data dictionary, you specify the domain of each attribute, as well as other constraints that might be important during the implementation phase. Note that the domain of an attribute includes syntactic and semantic interpretations of an attribute. For example, the domain for the birth date of an employee is a 'date (or 'datetime' in some DBMSs). However, you may only store the day, month, and year values. Similarly, the model year for a car can also be stored using 'datetime' type attributes, but you may only store the year and not months, hours or time zones. There may also be semantic constraints (e.g., for the birth date example we mentioned, the recorded time cannot be more than the current time.) You will be given an example data dictionary template to follow.

Phase 5

This phase will be the implementation of your project. Using your implementation-ready relational model, and well-prepared data dictionary, you will define your tables and constraints and create your database. You must use 'create table' statements for creating your tables. In this phase, all your primary and foreign key constraints must also be mapped from your relational model. The second part of this phase is to insert data into the database. You may use simulated (artificially created) data or real data. For inserting data you can use, a series of 'insert' statements or you can use database loading tools, or SQL statements such as 'copy' or 'load data'. After the insertion of your data, you will implement the functional requirements as the queries or a series of queries. You are expected to write five SQL queries and these need to be advanced ones. Your queries are expected to correctly answer the functional requirements listed.

Phase 6 [Optional]

The Phase 6 is optional and will get you extra credit. In the Phase 6, you will create two views and two triggers. The views and their function should be mentioned in the functional requirements section (see Phase 1). Please make sure you use 'create view' statements when creating views. You can use these views in your queries if needed. Next, you will create

triggers. One of the triggers is expected to be run 'before' the modification (e.g., checking a semantic constraint on insertions) and the other trigger is expected to run 'after' modification. The modifications can be 'insert', 'delete', or 'update' statements. You also need to mention the triggered actions in your functional requirements.

Phase 7 [Optional]

Phase 7 is again an optional part with extra credit. In this phase, your task is to design and implement a client application. This application will use a high level programming language of your choice, yet you need to ask for my permission if it is not from my choice of languages (and my choice of programming languages are Java and Python). The client must be able to access or modify any table in the database and its necessity and utility must be identified in the functional requirements. It must also implement and/or utilize all the functional requirements.

Final Project Report

You will prepare the project design report which will be constituting 60% of your final project grade (**15% of overall course grade**). The report will be graded over **60 points**. You will be provided with a template for formatting purposes. In the final report you will have the following sections:

- **Introduction:** Briefly describe what the designed database is about (your project topic), what is your motivation for selecting this project topic, how this database can be used, and some of the important aspects of your design.
- **Requirements Analysis:** The second part of your report will be related to the database requirements analysis. Describe your requirements (the ones you have researched in Phase 1) one by one. Additionally, include specific constraints related to your database (min-max, semantics, and specialization/category types). Another important part is the functional requirements. State all your functional requirements (including modifications and queries) of your database. Note that these functional requirements should be sophisticated (inserting a tuple in your database does not need to be reported as a functional requirement). You will be expected to implement all the functional requirements in the end.
- **EER Model:** Create your EER model using the requirements you have provided in the earlier phase. Include min-max notations [as we have seen in the class] and specialization or category relationships. Make sure you specify your key attributes as well.
- **Relational Model:** Prepare and demonstrate your relational model. For each of the relations, show how you transformed the entities and relationships into relations. Discuss how the cardinalities, generalizations, union types affected your decision if you have taken a non-trivial step. Otherwise simply demonstrate the correspondence between entities/relationships and relations/attributes.
- **Data Dictionary:** In this section, show your data dictionary for each table (or relation you mapped). Demonstrate the constraints (null, unique, or other check constraints such as [check age > 0]) on the data, show the data types and the descriptions of each attribute. Add necessary information about candidate, primary, and foreign keys.

- **Implementation:** In this section, show how you implemented your functional requirements. Include your SQL statements as code blocks. The names of the tables, types of the attributes must be the same as the ones specified in your data dictionary. Do not include insert statements. However, include your SQL queries. If you choose to implement triggers and views (Phase 6), show your SQL code, and for the application (in Phase 7) describe what your application can do.
- **Summary:** Summarize all your work, and the contribution of your project.

Implementation

The implementation of the project will constitute **40%** of final project grade (**10% of overall course grade**). The implementation will be graded over **40 points** and should include the following:

- **SQL/** folder: Create a folder called "SQL", and put the following files in this directory.
 - The first file should be named as "create.sql" and it must include DDL commands, such as create and alter table statements.
 - The second one should be named as "queries.sql". In this file you must have the SQL queries, that refers to the functional requirements you have listed in your project reports. Comment and format both of the files, properly. If your SQL file does not show which functional requirement it addresses, it will not be graded.
 - The third file must be named as "insert.sql", it should have the insert table statements. When evaluating, we will only run "create.sql" and "insert.sql" once (and as a whole); and check whether your functional requirements work properly. Therefore, test all your code and make sure it runs correctly.
 - [Optional] The fourth file must be named "views.sql" and it should include SQL statements that will create your views in Phase 6. You need to have a subsection in your report.
 - [Optional] The fifth file must be named "triggers.sql" and it should include SQL statements that will create your triggers in Phase 6. You need to have a subsection in your report.
- **Application/** folder: Create a folder called "Application". In this directory, you must have all your code and executables (if necessary) for the optional application part from Phase 7. You should include a readme file along with the code. The readme file must include which systems (with the versions) are required to run the code, as well as the step-by-step instructions on how to run your code.

Compress all of these folders (zip or tar only), and submit it as one file. I will only consider the very last file you will upload when grading; and that should be the compressed file..

Grading Rubric

Category	Item/Feature	Percentage
Report (60%)	Requirements Analysis (Data + Functional Requirements)	12
	EER Model	12
	Relational Model	12
	Data Dictionary	8
	Implementation Description (SQL + Optional)	16
Implementation (40% + extra 20%)	Create table statements and constraints	15
	Data insertion statements	10
	Queries (select statements)	15
	Trigger statements	6 (3 each)
	View (create view statements)	4 (2 each)
	Application	10