CSE5330.001 Fall 2019

# Database I: Term Project

This is a term project for the entire semester for students to understand what they are learning and to experience planning, designing, and implementing a database system and its application. This project is divided into 4 phases, planning the project, designing a logical database, designing and implementing a physical database, and finally implementing an application. Table 1 shows the summary of the four phases.

Table 1. Summary of database I term project by phases

| Phase | Description | Output | Due |
|---|---|---|---|
| Phase I (15%) | **Project Initiation**<br>o Problem Statement<br>o Plan your project | Documentation | 9/13 (Fri) |
| Phase II (30%) | **Conceptual/Logical Database Design**<br>o ER, EER modeling<br>o Relational modeling<br>o Relational Algebra | Documentation | 10/11 (Fri) |
| Phase III (25%) | **Physical Database Design/Implementation**<br>o Create Table and other objects<br>o Data Insert<br>o Formulate the queries | SQL statements & Documentation | 11/8 (Fri) |
| Phase IV (30%) | **Application Develop and Demonstration**<br>o Application implementation<br>o Final Documentation<br>o Demonstration | System & Documentation | 12/6 (Fri) |

## Stages of the Projects

## Phase I [15%]: Project Initiation

### Step 1: Form Project Team

You need to form a team with up to 2 members. Once you formed a team, you need to send an email to GTA (and cc to me) with the following information:

- Names of two members
- Preferred email addresses of the members
- Student IDs

GTA then assigns you project team ID, which will be used throughout the rest of the phases. Try to choose an interesting problem in your domain or area of interest/expertise which requires data modeling and supports for a database management system (DBMS).

### Step 2: Problem Definition (Chapter 3)

In this step, you need to choose a topic, write a detailed description of the application, such as the requirements of what data needs to be represented and captured. Some examples of the applications include:

- Retail shop management: Video, Nail care, doughnut, etc.
- Human Resources Management System

- Auto part shops
- Travel/Air Line/Hotel/Rental car reservation system
- Inventory management in grocery store
- Online bidding system

, but exclude:

- University/Company examples in the textbook.

More examples will be discussed in the classroom. Please, make sure the description is as detailed as possible. Otherwise, you have to revise this phase until you have all the details, and cannot proceed to the next phase.

**Step 3: Formulate 10 realistic English queries (Chapter 3)**

The queries would be useful to somebody using the data. The queries must be realistic and you should have a few complex queries. Note that the queries that the database system must be able to answer ultimately determine what information needs to be maintained in the database. For example, the following queries could be good if you select Hotel Reservation system:

- On this coming weekend, are there available rooms for two adults and one kid for two nights in Orlando, FL?
- Show me all the rooms (or hotels) that cost less than $100 a night in Dallas, TX.

**Step 4: Specify the assumption about the database in English (Chapter 3)**

Here you talk about attributes, keys, the nature of relationships between entities, etc. Do not discuss something that is very obvious (e.g., that a student can take several courses). In addition, don't make too many simplifying assumptions.

**[Submission guideline for Phase I]**

You should submit the full and legible report with a cover page that includes your name, the title of the project, and abstraction (brief description of the project). The body of Phase I documentation should include sufficient detail to describe all steps of Phase I:

- (Step1) Team Information: team ID, name, email address, student ID, etc
- (Step2) Problem Statement: description of your database
- (Step3) List at least 10 queries in English
- (Step4) Any assumptions

In conclusion, describe what you have learned from the project, and what you need for the better project output. The GTA and the instructor may recommend modification or additions to the document submitted in Phase I. After the feedback from GTA, you can move to the next phase. You are strongly recommended to contact GTA before you submit Phase I documentation.

The name of your submitted documentation file should be your “Team#_phase1”. You can use your team ID for #. For example, “Team1_phase1.doc”. Submit this to the link in the Canvas system.

## Phase II [30%]: Conceptual / Logical Database Design

### Step 5: EER modeling (Chapter 3 and 4)

Based on the Step 1 ~ 4 in the previous phase, in the phase II, draw an Enhanced Entity Relationship (EER) diagram for the problem you choose. This EER modeling should adhere to the principles that you learned in the course. It should also use the notation that you learned in the lecture. Entity should have keys clearly identified and relationships should have the cardinality information. The resulting EER diagram should contain:

- At least 5 entity types (user or customer related entity types are counted 1)
- At least 5 relationship types (should include at least one 1:1, 1:N, and M:N)
- Some attributes on relationships
- At least 1 Generalization/Specialization

### Step 6: Creating Relations (Chapter 9)

Based on the EER diagram, you should create relations using the relational model in the lecture. Make sure that you need to identify the primary key and foreign keys in all the relations.

### Step 7: Relational Algebra (Chapter 8)

Formulate the queries in Step 3 using relational algebra. If, at this point, you are unable to state the queries in relational algebra, this is probably because your database is not powerful enough. Then, you must go back to step 2 and see how the database can be improved.

### [Submission guideline for Phase II]

You should submit the full and legible report with a cover page that includes Team name, names of all members, the title of the project, and abstraction (brief description of the project). The body of Phase I documentation should include sufficient detail to describe all steps of Phase I:

- (Step5) ER diagram of the database
- (Step6) Relational database schema (including diagram)
- (Step7) Give the relational algebra expressions for the queries

In conclusion, describe what you have learned from the project, and what you need for the better project output.

The name of your submitted documentation file should be your “Team#_phase2”. You can use your team ID for #. For example, “Team2_phase2.doc”. Submit this file to GTA (hardcopy) and the link in the Canvas system (softcopy).

## Phase III [25%]: Physical Database Design/Implementation (Chapter 6 and 7)

In this phase, students should implement the database tables from the normalized set of relations created in the previous phase. Sample data should be supplied for each table. This phase is called as Physical database design.

**Step 8: Creating DDL Script** **`Filename: dbDDL.sql`**

Make a file containing the SQL statements that create your entire database schema, named dbDDL.sql. This includes the tables with their constraints, view, indexes, triggers, and all other database objects if you have them.
To keep the project consistent, make sure you have at least 8 tables. You should have at least 1 database object among trigger, view or stored procedure.

**Step 9: Creating DML Script** **`Filename: dbDML.sql`**

Make a file containing INSERT statements which populate the table created in Step 8, named dbDML.sql. This script will contain SQL commands to fill data in your data. Each table should have around 7 ~ 10 sample data. If needed, other DML statement, such as UPDATE, and DELETE can be included here

**Step 10: Creating Drop Script** **`Filename: dbDROP.sql`**

Create a script that will drop all the objects you have created for your project including table, trigger, index, and etc. This will be used to start from a clean state after some inserts and deletes have been added to your application to check the correctness of your queries. You should be able to clean everything through this script and re-create the database instance by redoing step 8 and 9.

**Step 11: Creating SQL Script** **`Filename: dbSQL.sql`**

Create a script with queries from the relation algebra in Step 7, named dbSQL.sql. This script should contain at least 5 queries on your database. Use the comment facility in SQL*Plus (starting a line with -- , or /* */ ) to write the English version of your query, followed by the SQL version of the query. Also show the expected output in the file. These queries need to satisfy the following:

- Should be at least join queries (some involving more than 2 relations)
- At least two of them should be aggregate queries including GROUP BY and HAVING clauses with ORDER BY clause as well

**[Submission guideline for Phase III]**

Before you submit, you should test all four SQL script files using SQL*Plus by giving the command

SQL> @<filename>
or
SQL> start <Filename>

Make a single zip file. Submit this zip file the link in the Canvas system (softcopy). The name of your submitted documentation file should be your "Team#_phase3". You can use your team ID for #. For example, "Team3_phase3.doc".

## Phase IV [30%]: Application Develop and Demonstration (Chapter 10)

### Step 12: Application Development

In this phase, you will develop a front-end application that will interface with your DBMS at the backend. The user will interact with the DBMS only through this interface. It is strongly recommended to use GUI to retrieve and update the database without having to write SQL queries. The interface will have a menu-driven input through which all interaction with the DBMS is accomplished. The results will be displayed to the user as well.

You can use DBMSs such as Oracle, MySQL, and MS SQL. You can also use any kind of programming language that can support ODBC or JDBC. The database application can be either standalone windows app or web based one. But try NOT to use supporting tool such as phpMyAdmin, instead try to create your own source code using general programming language. The emphasis in grading phase of the project will be on the functionality and not on the GUI design. However, the GUI should be easy to understand and easy to use.

### Step 13: Project Demonstration

Sign up for a demo (A sign-up sheet will be made available at the appropriate time by GTA). Be prepared for the demo about 5 to 10 minutes. Your demo may include

- Introduction:
- Executive summary of the project
- Demonstration. If your application needs setting up on our machine, make sure it can be done in as less time as possible by automating it using compilation scripts etc. If you are using some specific web servers or tools, you need to demo on your own laptop.
- Conclusion

### Step 14: Final Report and Project delivery

Prepare the final project report including:

- Cover page (Same as Phase I)
- Phase I documentation. You need to revise it if needed
- Phase II documentation. You need to revise it if needed
- Sources of SQL scripts created in Step 8 ~ 11
- System configuration of the project (usually system diagram)
- Sources of Applications generated in Step 12
- In conclusion, describe what you have learned from the project, and what you need for the better project output.

Put everything including documents, SQL files, and source files into a zip file. Then, submit it to the link in the Canvas system on the demo day.