

CSCE-608 Database Systems

Spring 2023

Instructor: Dr. Jianer Chen

Office: PETR 428

Phone: 845-4259

Email: chen@cse.tamu.edu

Office Hours: TR 2:15 pm–3:45 pm

Teaching Assistant: Ya-Ru Yang

Office: PETR 445

Phone: (979) 969-0095

Email: yaruyang@tamu.edu

Office Hours: MW 2:00 pm–3:30 pm

COURSE PROJECT I

(Due March 21, 2023)

Overview. This project provides you with an opportunity to have experience in building a database application for a real-world domain of your own interests. You will create a database using SQL, populate the database, write programs that manipulate the database, and develop an interface for user's queries and modifications on the database.

The project is an individual project. Each student should work on her/his own project.

The project will consist of five parts, described in details as follows.

Part A. E/R Diagram

Your first step is to identify a domain you would like to manage with your database, and to construct an Entity-Relationship diagram for the database. It is suggested that you pick an application that you will enjoy working with. It is especially nice if you pick an application where you can populate your database using real, as opposed to fabricated, data.

Try to pick an application with a schema that is relatively substantial, but not too enormous. For example, your E/R design should have in the range of five or so entity sets, and a similar number of relationship sets. You should certainly include different kinds of relationships (e.g., many-one, many-many) and different kinds of data (strings, integers, etc.), but your application need not necessarily require advanced features such as weak entity sets and "is-a" relationships.

1. Write a one-page description of the database application you propose to work with. Your description should be brief, relatively complete and clear. If there are any unique or particularly difficult aspects of your proposed application, please point them out.
2. Specify an E/R diagram for your proposed database. Underline key attributes for entity sets and include arrowheads indicating the multiplicity of relationship sets. If there are weak entity sets or "is-a" relationships, make sure to notate them appropriately.

The appendix provides some suggested topics of the project for your consideration. You may revise yours to fit the applications of your own interests. You may also consult and discuss with the instructor or the teaching assistant on your proposed application.

Part B. Relational Schema

In this second part of the project, you will produce a relational schema from the entity-relationship diagram you came up with in Part A.

1. Using the method for translating an E/R diagram to relations, produce a set of relations for your database design.
2. For each relation in your schema, specify the nontrivial functional dependencies for the relation. Any functional dependencies that actually hold in the real-world scenario that you are modeling should be specified. It is fine if some of your relations have no nontrivial functional dependencies.
3. Check if each relation in your schema is in Boyce-Codd Normal Form (BCNF) with respect to the functional dependencies you specified. If not, decompose the relation into smaller relations so that each relation is in BCNF. No matter whether your relations from the E/R diagram are in BCNF or not, you should work through the relation normalization process, report and discuss the process and its results. Report how and why your relations are normalized if the normalization process produces new relations. If no new relations are produced, give descriptions on how your relations pass the BCNF test and provide explanations why your relations have no redundancies.
4. (Optional) Discuss whether the Third Normal Form or the Fourth Normal Form would further improve the structures of your relations. You may further improve your relation structures if you feel needed.

Part C. Creating Your Database using SQL

In this part of the project, you will create a database schema for your database application using SQL, and you will populate the relations in your database with data sets. To manage a database, you may use client software such as Unix software: `mysql` or `psql`. Graphical user-interface software might be helpful, including:

- pgAdmin (PostgreSQL) (<https://www.pgadmin.org/download/>)
- PHPMYAdmin (<https://www.phpmyadmin.net/downloads/>)
- XAMPP (<https://www.apachefriends.org/download.html>)
- MySQL Workbench (<https://dev.mysql.com/downloads/workbench/>)
- Microsoft SQL Server Developer edition
(<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>)

To create your database and populate it, you proceed as follows:

1. familiarize yourself with an SQL relational DBMS (referenced above),
2. create relations for your database based on your relational schema from Part B,
3. generate database data: write a program in any programming language you like that creates large files of records for each of your database relations.

If you have available real data for your database, then your program will need to transform the data into files of records conforming to your database schema and to MySQL's load format. Otherwise, you will need to write a program to fabricate data: your program will generate records conforming to your schema. Note that it is both fine and expected for your data values, strings especially, to be meaningless gibberish. The point of generating large amounts of data is so that you can experiment with a database of realistic size. The data you generate and load should be on the order of:

- At least two relations with thousands of tuples
- At least one relation with hundreds of tuples

If your application naturally includes relations that are expected to be relatively small (e.g., schools within a university), then it is fine to use some small relations, but please ensure that you have relations of the sizes prescribed above as well.

When writing a program to fabricate data, there are two important points to keep in mind:

- Make sure not to generate duplicate values for key attributes.
- Your database almost certainly includes relations that are expected to join with each other. For example, you may have a STUDENT relation with an attribute COURSENUM that is expected to join with attribute NUMBER in relation COURSE. When generating data, be sure to generate values that actually do join – otherwise all of your interesting queries will have empty results! In order to ensure this, you may generate records for multiple relations (e.g., COURSE and STUDENT) at the same time, or generate the records for one relation first, and then use the joining values for the other relation. For example, you could generate records for relation COURSE first, then use the COURSE.NUMBER values when creating values for STUDENT.COURSENUM.

Part D. Developing a user interface

The application should have a clear purpose, and the interface must be user-friendly. Users do not need to know the details of the database or the SQL language to manipulate the data. Graphical interface is more desired than a text-based interface. Your program should consist of a continuous loop in which:

- A list of at least five alternative options is offered to the user. (An additional alternative should be QUIT.)
- The user selects an alternative.
- The system prompts the user for appropriate input values.
- The system accesses the database to perform the appropriate queries and/or modifications.
- Data or an appropriate acknowledgment is returned to the user.

Both input and output in the interface should be in a format more convenient and pleasing than raw interactive SQL. Please include some "interesting" queries or modifications, i.e., operations that require some of the more complex SQL constructs such as subqueries, aggregates, set operators, etc. As a general example, if your database is a campus applicant database, then your interface might include in its menu a number of useful queries on the database, with some queries performing statistical analysis requiring multiple levels of grouping, and other queries.

To build the user interface,

1. you can choose any combination of a programming language and a database system for your application.
2. set up or install a web server if you choose to write a web application.
3. your software or web pages connect to the database using a connector. You will need to install a connector specific to the programming language of your application. The connectors include:
 - Java JDBC connector for MySQL server
<http://dev.mysql.com/downloads/connector/j/>
 - .NET Framework 4 and Microsoft SQL server (You can use languages VB, C#)
<http://msdn.microsoft.com/en-us/library/kb9s9ks0.aspx>

4. There are also online examples of connecting to a database through a program:

- PHP - MySQL example and function documentation
<http://dev.mysql.com/doc/apis-php/en/index.html>
- React - Firebase documentation
<https://firebaseopensource.com/projects/rakannimer/react-firebase/>
- C# Documentation of connecting to Microsoft SQL server
<http://msdn.microsoft.com/en-us/library/bb655891%28v=VS.90%29.aspx>

If you decide to build a website for this project, you may develop it in your local computer, and demonstrate to the teaching assistant, who will be the grader for the project. Alternatively, you may upload your web pages to the cloud service HEROKU and provide a link to the website in your project report. Check the documentation in devcenter.heroku.com for more details.

Since Heroku does not provide a free trial for users, you have to sign up for GitHub Student Developer Pack to get free credits. It will take a few hours for both GitHub and Heroku to verify your student identity. Check <https://www.heroku.com/github-students> for details.

Part E. Submitting your project

The project submission should include a README of how to install and run your programs, and a project report that provides all details of the project. If you are submitting a web application, please provide the URL of your website.

The project report should be printed single-spaced, and include the following components:

- (a) Project description of at least one page: it describes the application background of your system, and the functions and services your system will provide;
- (b) The Entity-Relationship diagram of your database. The E-R diagram should contain at least 5 entity sets and a similar number of relationships. Give discussions on how the diagram is produced and how it reflects the applications you described in item (a);
- (c) Table normalization: construct the relations based on the E-R diagram given in (b), and apply the table normalization process on the tables. Your tables should be at least in Boyce-Codd Normal Form. Discuss how the table normalization process changes/improves table structures. If the table normalization process does not change the table structures at all, explain why it does not;
- (d) Data collection: describe how the data are collected/generated. Explain how you ensure the uniqueness of key attributes and interesting joins among multiple relations;
- (e) User interface: describe how you build the system user interface and how users use your system. Give a list of functions that are offered by your system to the users. Explain how the functions are implemented in SQL;
- (f) Project source code: include the source code that implements your system;
- (g) Discussion: share your experience in developing a database application: what were the difficulties you encountered in the process and how you got over them; what you had learned from doing the project.

Grading policy (based on your submission in Part E)

- | | |
|---|--------------------------------|
| (a) Application Description: 10%, | (b) E/R Diagram: 10%, |
| (c) Database Schema and Normalization: 15%, | (d) Creating Tables/Data: 20%, |
| (e) User Interface and Functions: 30%, | (f) Project Report: 15%, |

APPENDIX. Sample Database Projects

(<https://www.lovelycoding.org/2013/11/top-18-database-projects-ideas-for-engineering-bca-mca-btech-bsc.html>)

1. INVENTORY CONTROL MANAGEMENT DATABASE PROJECT

Design goals: maintain a proper variety of required items, increase inventory turnover, reduce and maintain optimize inventory and safety stock levels, obtain low raw material prices, reduce storage cost, reduce insurance cost, reduce taxes

2. STUDENT RECORD KEEPING SYSTEM DATABASE PROJECT

Design goals: a student file that contains the information about student, a stream file, a marks file, a fee file, concession/scholarship etc.

3. ONLINE RETAIL APPLICATION DATABASE PROJECT

A customer can register to purchase an item. The customer will provide bank account number and bank name (can have multiple account number). After registration, each customer will have a unique customerid, userid and password. A customer can purchase one or more item in different quantities. The items can of different classes based on their prices. Based on the quantity, the price of the item and discount (if any) on the purchased items, the bill will be generated. A bank account is required to settle the bill. The items can be ordered to one or more suppliers

4. COLLEGE DATABASE DATABASE PROJECT

A college contains many departments. Each department can offer any number of courses. Many instructors can work in a department, but an instructor can work only in one department. For each department, there is a head, and an instructor can be head of only one department. Each instructor can take any number of courses, and a course can be taken by only one instructor. A student can enroll for any number of courses and each course can have any number of students.

5. RAILWAY SYSTEM DATABASE PROJECT

A railway system, which needs to model the following:

1. Stations
2. Tracks, connecting stations. You can assume for simplicity that only one track exists between any two stations. All the tracks put together to form a graph.
3. Trains, with an ID and a name
4. Train schedules recording what time a train passes through each station on its route.

You can assume for simplicity that each train reaches its destination on the same day and that every train runs every day. Also for simplicity, assume that for each train, for each station on its route, you store

- Time in,
- Timeout (same as time in if it does not stop)
- A sequence number so the stations in the route of a train can be ordered by sequence number.

1. Passenger booking consisting of train, date, from-station, to station, coach, seat and passenger name.

6. HOSPITAL MANAGEMENT SYSTEM DATABASE PROJECT

A patient will have unique Patient ID. Full description about the patient about personal detail and phone number, and then Disease and what treatment is going on. The doctor will handle patients, One doctor can Treat more than 1 patient. Also, each doctor will have unique ID. Doctor and Patients will be related. Patients can be admitted to hospital. So different room numbers will be there, also rooms for Operation Theaters and ICU. There are some nurses, and ward boys for the maintenance of hospital and for patient take care. Based upon the number of days and treatment bill will be generated.

Check Hospital Management System project in PHP

7. LIBRARY MANAGEMENT SYSTEM DATABASE PROJECT

A student and faculty can issue books. Different limits for the number of books a student and teacher can issue. Also, the number of days will be distinct in the case of students and teachers for issue any book. Each book will have different ID. Also, each book of the same name and same author (but the number of copies) will have different ID. Entry of all the book will be done, who issue that book and when and also duration. Detail of Fine(when the book is not returned at a time) is also stored.

8. PAYROLL MANAGEMENT SYSTEM DATABASE PROJECT

There will entry (Unique ID) of all the employee of any Organization. According to the date of joining and date up to which salary is created, Number of days will be entered. Basic pay will be defined according to the post of employee and department. Then component like DA, HRA, medical allowance, Arrears will be added, and Charges of Hostel/ Bus, Security, welfare fund and other will be deducted. The number of leaves taken by the employee.

9. HEALTHCARE ORGANIZATION DATABASE PROJECT

This organization provides the following functionalities

- Emergency Care 24x7
- Support Groups
- Support and Help Through calls

Any new Patient is first registered in their database before meeting the doctor. The Doctor can update the data related to the patient upon diagnosis (Including the disease diagnosed and prescription). This organization also provides rooms facility for admitting the patient who is critical. Apart from doctors, this organization has nurses and ward boy. Each nurse and ward boy is assigned to a doctor. Also, they can be assigned to patients (to take care of them). The bill is paid by the patient with cash and E-banking. Record of each payment made is also maintained by the organization. The record of each call received to provide help and support to its existing person is also maintained.

Check clinic management system project in PHP

10. RESTAURANT MANAGEMENT DATABASE PROJECT

The restaurant maintains the catalog for the list of food and beverage items that it provides. Apart from providing food facility at their own premises, the restaurant takes orders online through their site. Orders on the phone are also entertained.

To deliver the orders, we have delivery boys. Each delivery boy is assigned to the specific area code. The delivery boy cannot deliver outside the area which is not assigned to the delivery boy (for every delivery boy there can be a single area assigned to that delivery boy).

The customer record is maintained so that premium customer can be awarded discounts.

11. DESIGN A SCENARIO AND AN ER DIAGRAM FOR AN IT TRAINING GROUP DATABASE PROJECT

It will meet the information needs of its training program. Clearly indicate the entities, relationships, and the key constraints. The description of the environment is as follows:

The company has 10 instructors and can handle up to 100 trainees for each training session. The company offers 4 Advanced technology courses, each of which is taught by a team of 4 or more instructors. Each instructor is assigned to a maximum of two teaching teams or may be assigned to do research. Each trainee undertakes one Advanced technology course per training session.

12. BLOOD DONATION SYSTEM DATABASE PROJECT

A system in which data of Patient, data of donor, data of blood bank would be saved and will be interrelation with each other

DATA OF PATIENT – Patient Name, Patient Id, Patient Blood Group, Patient Disease

DATA OF DONAR – Donar Name, Donar Id, Donar Blood Group, Donar Medical report, Donar Address, Donar Contact number

DATA OF BLOOD BANK – Blood Bank Name, Blood Bank Address, Blood bank Donors name, Blood Bank Contact Number, Blood Bank Address

Try to implement such scenario in a database, create a schema for it, an ER diagram for it and try to normalize it.

13. ART GALLERY MANAGEMENT DATABASE PROJECT

Design an E-R Diagram for an Art Gallery. Gallery keeps information about "Artist" their Name, Birthplace, Age & Style of Art about "Art Work," Artist, the year it was made, Unique title, Type of art & Prices must be stored. The piece of artwork is classified into various kind like Poetess, Work of the 19th century still life, etc. Gallery keeps information about Customers as their Unique name, Address, Total amount of Dollar, they spent on Gallery and liking of Customers.

14. HOTEL MANAGEMENT SYSTEM DATABASE PROJECT

A hotel is a hive of numerous operations such as front office, booking, and reservation, banquet, finance, HR, inventory, material management, quality management, security, energy management, housekeeping, CRM and more. The hotel has some rooms, and these rooms are of

different categories. By room category, each room has the different price. A hotel has some employees to manage the services provided to customers. The customer can book the room either online or by cash payment at the hotel. The customer record is stored in hotel database which contains customer identity, his address, check in time, check out time, etc. hotel provides food and beverages to their customers and generates the bill for this at the time of their check out.

15. SCHOOL MANAGEMENT SYSTEM DATABASE PROJECT

School Management System project in PHP

Design a database to maintain information about school staff(staff management system in ms access) and students satisfying the following properties:

1. Staff will have their id, name, and classes they are teaching
2. The student will be having the name, roll no, section, class
3. Another table containing the section, subject and teacher information
4. Next will contain fee information for students
5. One contains salary information for teachers
6. Rooms are assigned to classes keeping in mind that there is no time clash of same room or lab, students cannot be entered in more than one section, no student should be there who have not paid fees up to a particular date.

16. WHOLESALE MANAGEMENT SYSTEM DATABASE PROJECT

1. Maintain the details of stock like their id, name, quantity
2. Maintain the details of buyers from which manager has to buy the stock like buyer id, name, address, stock id to be bought
3. Details of customers i.e. name, address, id
4. Defaulters list of customers who have not paid their pending amount
5. List of payment paid or pending
6. The stock that is to buy if quantity goes less than a particular amount.
7. Profit calculation for a month.
8. Quantity cannot be sold to a customer if the required amount is not present in stock and date of delivery should be maintained up to which stock can be provided.

17. SALARY MANAGEMENT SYSTEM DATABASE PROJECT

1. Employee list to be maintained having id, name, designation, experience
2. Salary details having employee id, current salary
3. Salary in hand details having employee id, CTC salary, pf deduction or any other deduction and net salary to be given and also maintain details of total savings of employee
4. Salary increment to be given by next year if any depending upon constraints
5. Deduction in monthly salary if any depending upon any discrepancy in work and amount to be deducted.