

# CSCE-608 Database Systems

Fall 2017

**Instructor:** Dr. Jianer Chen

**Office:** HRBB 315C

**Phone:** 845-4259

**Email:** chen@cse.tamu.edu

**Office Hours:** T,Th 2:00pm-3:30pm

**Project TA:** Vaibhav Rawat

**Office:** N/A

**Phone:** 739-7293

**Email:** vrawat106@gmail.com

**Office Hours:** by appointment

## COURSE PROJECT #1

(Due October 10, 2017)

Although database programming is not the focus of this course, this project provides you with an opportunity to have some experience in building a real database system. In the current project, you will create a database with large volume of data using a DBMS of your choice, and write SQL programs to populate, manipulate, and query the database. You can pick an application of your own interests. It will be more interesting if it is something that you would use in your daily life. You will design a user-friendly interface such as web pages or local software for the application. The following are some examples.

- *Personal contact management*

It keeps track of your contacts and their relationship. You can put their contact information, pictures, sounds, and even video in the database so you can search and view them from anywhere that has internet connections.

- *Personal photo database*

This system will help you get your personal photos organized. You enter keywords and comments for each photo, and link it to other photos and people. You can search your photos by date, size, resolution, and other interesting aspects.

The following are some detailed instructions and requirements for the project.

### A. Collect or generate a large amount of data.

For your data, design a relational database schema that consists of a number of tables. Some tables should reference other tables. Minimize the redundancy of the data in your tables. Note that you should design your own database schema.

1. One of the tables should contain at least 1K tuples.
2. You can create SQL scripts with INSERT commands to fill the database, or simply create tab-separated files and load the data from the files.

### B. Setup a database using a DBMS.

The CSE department provides accounts of PostgreSQL and MySQL database servers for each student, which can be activated through CSNET (<http://csnet.cse.tamu.edu>). It is recommended that you read this page [https://wiki.cse.tamu.edu/index.php/How\\_to\\_Use\\_MySQL](https://wiki.cse.tamu.edu/index.php/How_to_Use_MySQL), which contains detailed information about how to activate your database account and how to access the database. You may also find other useful information on CSE helpdesk website (<http://helpdesk.cse.tamu.edu>). Please note that after activating your database account,

you will be asked to setup a password for it. When connecting to the database, use your \*CS\* user ID instead of your NetID. Also, you must use VPN when accessing CSNET and the database when you are off campus. VPN instructions are also available on CSE helpdesk website.

You can also download the Microsoft SQL Server Compact from the MSDN Academic Alliance. You should have received an email including your account and password because you are taking a CSE course.

To manage a database, you may use client software such as Unix software: mysql or psql. They are available on the department servers `compute.cs.tamu.edu` and `linux2.cs.tamu.edu`. Graphical user-interface software might be helpful, such as MySQL Workbench, PHPMyAdmin, pgAdmin, or SQL Server Management Studio Express.

### **C. Develop an interface that allows users to manipulate the data.**

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.

1. You can choose any combination of a programming language and a database system for your application. The application should apply SQL commands including SELECT, UPDATE, INSERT, and DELETE. When a tuple is inserted or deleted, deal carefully if the tuple references tuples in other tables.
2. Set up or install a web server if you choose to write a web application. The department provides a web server that hosts web pages written in Perl, PHP and C/C++. Note that you need VPN to access the website. Find the instructions on the CSE helpdesk website.
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. CSE web server already has the Perl & PHP connector installed for MySQL and PostgreSQL database. Some other 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. For your reference, there are 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>
  - 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 are recommended to use the department web server that supports PHP, Perl, and HTML. Read this page for more information: [https://wiki.cse.tamu.edu/index.php/CSE\\_Web\\_Pages#Web\\_Project\\_Pages](https://wiki.cse.tamu.edu/index.php/CSE_Web_Pages#Web_Project_Pages). Basically, you need to create a folder called `web_project` in your network drive as the root folder of your website. Then, access your website at [http://project.cse.tamu.edu/#CS\\_user\\_ID](http://project.cse.tamu.edu/#CS_user_ID).

### **D. Submit your project through CSNET.**

This should include a README of how to install and run your programs, and a project report that provides all details of the project (see Item F next). If you are submitting a web application, please provide the URL of your website.

## **F. Submit a project report.**

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) Data collection: describe how the data are collected/generated;
- (c) The Entity-Relationship diagram of your database. The E-R diagram should contain at least 3 entity sets and at least 3 relationships. Give discussions on how the diagram is produced and how it reflects the applications you described in item (a);
- (d) Table normalization: construct the relations based on the E-R diagram given in (c), 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 the table structures. The table normalization process may not change the table structures at all. If this is the case, explain why it does not.
- (e) User interface: describe how you build the system user interface and how users use your system;
- (f) Project source code: include the source code that implements your system;
- (g) Discussion: share your experience in developing the database system: what were the difficulties you encountered in the process and how you got over them; what you had learned from doing the project.

## **G. Grading policy**

Data collection (15%),	Database schema (15%),	Functionality (35%),
User interface (15%),	Report (20%)	

## **H. Questions?**

You can

- post your questions in the class discussion forum in Piazza;
- contact the project TA: Vaibhav Rawat at [vrawat106@gmail.com](mailto:vrawat106@gmail.com), or (979)739-7293, who is the main person in charge of this project; or
- if the above two did not work out, contact the course instructor and the course TA Ye Yuan at [ye.yuan@tamu.edu](mailto:ye.yuan@tamu.edu) or (832) 591-4096.