

Important Dates:

A. ~~Phase I due Oct 09~~

B. ~~Phase II due Nov 08~~

C. Phase III due **Dec 08**

- *[group submission, one per group]* Recorded Video Presentations Dec 08 (11:59PM)
- *[group submission, one per group]* Complete Project Documentation
- *[individual]* Self-Reflection Document
- *[individual]* Individual Contribution Form for Group Project*
- *[individual]* Evaluating Group Presentations** Dec 11 (11:59PM)

* Instructor/Grader may request additional information or supporting documentation related to the individual member contribution for each group during the grading process of this project.

** On **Dec 09th**, a list of all class recorded video presentations will be made available to all students to review or evaluate. The review period will open on **Dec 11th** (until end of day). The evaluation will be in the form of an online survey that will all students to evaluate other projects for other groups.

The project is a group-based activity that must be divided **evenly** and **fairly** among the group members. Each group member will have an opportunity to evaluate their partners at the end of the term. **Such evaluations will reflect member's participation and contribution in the completion of the project and can affect student's final mark. This can also apply to any grades assigned for Phases I and II.**

Thank you for your understanding and cooperation.

Please do not hesitate to send me an email if you have any questions, doubts or concerns.

Overview (200 points)

In Phase II, you have created the schema and state for your database that is part of your application. In Phase III, you will work on creating the conceptual model of your database and develop a functional front end of your database application to demonstrate the major functionalities. The main purpose of this project is to enable you to gain experience and apply concepts and technologies presented in this course to real-world applications. The design and implementation of the application utilizes several of the concepts presented in class. You may use any database environment you like (i.e. MySQL, Microsoft SQL, Oracle, IBM DB2, etc...) or development tool to build your application. The following elements are the required deliverables for Phase III.

Part A: E-R diagram: (25 points)

Design and create an Entity Relational (ER) model for your project database.

For each entity you create, ask yourself:

- What are its attributes?
- What attributes can make up its key?
- Is the entity weak?
- Is the entity a subclass of another entity?
- What relations does it have with other entities?
- Are any attributes multivalued?

For each relation you create, ask yourself:

- What is its cardinality ratio?
- What are the participation constraints?
- Is it an identifying relationship?

Come up with at least one derived attribute and include it in your ER model.

Part B: Normalization (20 points)

Normalization allows you to optimize your database design (e.g. how many tables you should have). Your task for this section is to prove that your database satisfies the Boyce-Codd Normal Form (BCNF). That is, your database must be normalized to BCNF. You will need to demonstrate or prove that the database is in BCNF. A database schema is in BCNF if each relation schema is in BCNF. Hence, as part of your proof, you should provide possible functional dependencies for each relation schema. Then, use this information to prove that your database schema is in BCNF.

Part C: Revised Relational Schema (20 points)

Now that you have optimized your database to BCNF based on Parts A and B, this may require changes to your initial database schema that was created in Phase II. Therefore, revise your relational schema from Phase II to accommodate and/or reflect the changes that may have been introduced as a result of Parts A and B. **Your ER design model from Part A must map correctly to your revised relational schema.**

Create a SQL Script file (with **.sql** extension) that contains all SQL statements including:

- an illustrative diagram of the relational database schema (see Figure 3.7 from the textbook).
- the required SQL DDL statements for creating the database schema (see **Figure 4.1 from the textbook**)
Your schema should clearly specify all necessary integrity constraints (e.g. PRIMARY KEY, FOREIGN KEYS, NOT NULL, etc.) for constructing your database relations. A partial requirement for this database schema is to also include at least:
 - o **four** CHECK constraints,
 - o **four** attributes with initial default values, and
 - o **all** proper ON DELETE and ON UPDATE clauses associated with all of the defined foreign keys.
- SQL statements for creating sample data that reflects a real-world database application. Each relation should contain at least 10 tuples. Make sure that the populated data is suitable for the types of queries in Part C. Provide all the necessary INSERT INTO statements for all sample data in this section (group them by each table and provide necessary documentation)
- Ten non-trivial SQL Queries. Ensure that proper documentation is provided for each query including a query (a) number, (b) purpose and (c) summary of the expected result.

You may use the queries developed in Phase II Parts A, B and C. However, since you have revised your relational schema, this may require changes to the SQL queries developed in Phase II. Provide the necessary documentation throughout the entire SQL script. The SQL script must specify the working database platform that is used to run the SQL statement and must run without any errors on the specified platform.

Part D: Front End Interface (25 points)

You are asked to use your imagination in designing and developing an application that serves as the front-end of your database application. That is, the overall functionality of the system is delivered using multiple web pages. The project should be relevant to the course content. In particular, it should have some connection with the **databases, world wide web, scripting technologies, and markup Languages**. You may use **any** Web programming language or scripting technology to develop this front end (i.e. PHP, ASP.NET, JSP, CFM). That is, the front-end or the interface of your system should demonstrate the major functionalities of your system. In particular, your system should at least demonstrate a working functionality of the following features/components:

- **One web page** that serves as the main page of your project (e.g. homepage of the project, overview, project description, members' contact information, logo, title, etc.)
- **Four or more web pages** (or server-side script that generate appropriate HTML) that you implement that will provide useful features of your database application
 - o Web pages will execute **four of the SQL Queries from Phase II (Part C)** and display results in a tabular HTML format. These web pages should demonstrate or exhibit the **usefulness of your database application and overall features**. Each web page should provide or support one or

more (database) features of your database application. These can include, for example, retrieving data or manipulating data from/within a database.

- Be creative in terms of what features you offer through these web pages. For example, searching for a customer information is **not as creative** as identifying the most commonly sold items for a particular category over the last ten-day period for an online store database application.
- Web pages should be presentable and utilize creative CSS theme(s) for demonstrating the functionalities of your database application. We are not requiring groups to incorporate advanced appearance or presentation (i.e. advanced CSS design), but an appearance that is comparable in complexity to that of the assignments 3 and 4.
- Web pages ultimately use HTML in browsers to display information. Hence, you may use any scripting technology for generating the HTML of these web pages. In the assignments, you learned PHP. As part of Module 7, a tutorial on how to develop web pages in ASP.NET using Microsoft Visual Studio is provided. Hence, there is no restriction in terms of what technology you can utilize for supporting the delivery of the requirements of this project.

All groups are **strongly** encouraged to incorporate the Google Cloud Platform (GCP) or any cloud platform of your choice for deploying your database application on an existing cloud platform (i.e. similar to Assignments 3 and 4).

Part E: Project Documentation and Final Report (50 points)

Your project-related work must be your own original work. You must document all instances whether you are citing, paraphrasing or utilizing someone else's words, ideas or intellectual material. It is the student's responsibility to ensure that all submissions for the course material abide by the policies regarding UWT [academic integrity](#). For more information on how to avoid plagiarism, please click [here](#). Thank you.

The final report should reflect the purpose, design and technical implementation of your database application. This report should be 3 - 5 pages (~1000 words and excluding appendices for diagrams), Times 11 font and single column. The report must include an introduction, objectives and scope of the project, relation to other existing work that closely relate to your database application, an overview of the architecture or design diagrams (should describe all provided diagrams), conclusions with contribution, thoughts about any future work and references. The report should also reflect the overall goals of the project, its relationship to the course, design and implementation. Your report should also include proofs of a working or functional SQL queries used throughout this phase. That is, include screenshots of all queries used (e.g. query followed by a showing an output running the query). In addition, the final report should include screenshots of the interface and/or front-end of your system.

Part F: Implementation and Coding (20 points)

The grading for this section depends on the project itself. That is, if you implement a system then how well the system is implemented, what is the level of complexity of the functionality, the level of creativity in implementing the system, the level of documentation of the coding, performance analysis of the system when compared to similar existing systems that closely resemble your project idea, and the overall usability.

Part G: Prototype (40 points)

Develop an 6-10 minute video that presents your project idea and a real-time demo of your project/prototype (file named demo.mp4). The short video should exhibit your overall TCSS445 project, interface design, running SQL queries, ER design, among other features. You are encouraged to include PowerPoint slides while presenting your project and demo/prototype as part of this video. Not all students need to be part of the video demonstration and you do not need to appear in person in the video. However, all students are encouraged to participate in the demo if possible (e.g. recording Zoom session). Samples from previous quarters are provided in the drop-box for the Phase III on Canvas. A suggested format of the video with approximate timing for each component is provided below.

- Slide 1: title, logo and names of students participating in the group (0.5 min)
- Slide 2: overview of your database application including (1 min)
 - o objective (why you built this application)
 - o target audience (who will use this application)
 - o main features of your application
- Slide 3: ER design (1 min)
 - o explain the main entities and defend your conceptual design
- Slide 4: Relational Schema (1 min)
 - o Discuss schema and any major differences between Phase II and Phase III schemas
- Slide 5: Normalization Proof (1 min)
 - o demonstrate that database is in BCNF on slide, can discuss one or two relations
- Slide 6: Prototype/Demo (3-5 mins)
 - o demonstrate a working demo exhibiting working functionalities of the front-end
- Slide 7: Challenges (1 min)
 - o Discuss any challenges and potential future work enhancements

Bonus: You can receive up to **2 bonus points** if prototype employs web services or APIs and is deployed on a cloud platform (e.g. Google Cloud Platform) of your choice.

Deliverable (submit electronically via CANVAS: drop box Phase III):

Each group is required to submit a compressed file (".zip" extension) with the filename being the group's number (i.e. **group_number.zip** where **number** represents your **group number**) containing the answers/requirements outlined in Parts A-G. Below is an outline of the recommended format for your final submission:

- i. **PDF document (.pdf file named FinalReport.pdf) containing**
 - Cover Page (with group name, members' names and logo)
 - **Part E:** Final Report
 - **Part A:** Appendix A: ER Diagram (one single page)
 - **Part B:** Appendix B: Normalization Proof to BCNF (maximum of two pages)
 - **Part C:** Appendix C: Relational Schema Diagram (one single page)
 - **Part E (cont'd):** Appendix D: Screenshots of functional SQL Queries (with proper documentation)
 - **Part D (cont'd):** Appendix E: Screenshots of functional Web Interface (with proper documentation)
- ii. **SQL Script (.sql file named queries.sql)**
 - **Part C:** An SQL script file with ".sql" extension (with proper documentation)
- iii. **Implementation/Coding (.zip file named code.zip)**
 - **Part F:** Complete database application files that including source code files, web pages, images, etc. You are required to also include a **readme.txt** file that describes how to run your project, what development platform was used, location and structure of your files/folders, etc.
- iv. **Presentation Video (mp4 file named demo.mp4)**
 - **Part G:** The video describing your project and demonstrating a working prototype of your TCSS445 database application including front-end interface.
- v. **[for bonus: deployment on cloud platform] (text file named deploy.txt)**
 - BONUS: Include the following elements:
 - URL indicating the address where the database application is deployed
 - username/password of the DBMS server and how to access it (e.g. MySQL and address of phpMyAdmin)
 - front-end interface address URL(s)