CSCI 340 – Database Design; Final Project Assignment

**Description:**

This semester you have acted as the *Database Designer* of a complex Mini world, or universe of discourse (UOD) of your choosing. Up until now you have Gathered requirements (both logical data and functional) from various types of end users (through informal interviews) to gain a clearer understanding of the Mini world you are modeling, and to validate various assumptions you thought to be true. You then took those logical data requirements and translated them into a conceptual data model (in the entity relationship (ER) and/or extended entity relationship (E-ER) diagram, a form of the conceptual model) ensuring that what you had modeled fully reflected the data requirements requested by your end users.

Your last task will be to perform the data model mapping step (using the step-by-step algorithm from Chapter 9 in the textbook) to produce a relational schema from your ER / E-ER Diagrams. Then you will implement that relational schema (with its relations, attributes, domains, and constraints) then seed it with sample data using SQL (applying various DDL and DML statements) to define, construct, manipulate, and share your databases. All of this with the help of the MSSQL RDBMS in Microsoft Azure.

Finally, you will implement a few controllers for a simple CRUD API database application using Java Spring Boot (which will start to define how external applications and users can interact with your database system).

**Deliverables:**

* **Data Requirements.**
  + Please include the full list of logical data and functional requirements for your Mini world. These should be reflective of the format found in the textbook and should be comprehensive.
* **ER / E-ER Diagram.**
  + Please include the entity relationship (ER) / extended entity relationship (E-ER) diagram created from the logical data requirements. Be sure your diagram includes both participation and cardinality constraints (following the standards and best practices covered in chapters 3 – 4 of the textbook), and optionally represent Disjointness and completeness constraints (for situations in which subclass / superclass relationships are required). The diagram should be neatly organized and fulfill all the requirements requested by your end users.
* **Relational Schema.**
  + Please include a relational schema diagram created by mapping the conceptual diagram using the algorithm outlined in chapter 9 for “data model mapping”. This diagram should note any assumptions, and include relation names, attribute names, primary keys, and foreign keys. The relational schema should follow the standards and best practices covered in the textbooks for relational schemas. The relations must also be normalized through third normal for (please include and/or explain any changes since your initial mapping from the ER / E-ER diagram).
* **SQL (DDL).**
  + Please include all the SQL used to construct your database tables, constraints, types, etc... (You will need to execute these scripts on your database prior to your final presentation).
* **SQL (DML).**
  + Please include all the SQL used to seed the database with sample data (You will need to execute these scripts on your database, and each relation should have between 5 – 10 tuples of data for your presentation and/or demonstration to be meaningful).
* **Java Spring Boot API.**
  + Please include in your final project document a link to your Java Spring Boot API project on GitHub. Your API project should include the following components...
    - At least four controllers in which you have implemented one of each of the following; GET, POST, PUT, and DELETE operations.
    - Of the GET operations that you implement, at least one should include the ordering of the results, one should include a complex SQL query (i.e., one with a JOIN condition, aggregation of Data, etc...). For those queries, please refer to the section on advanced SQL in chapter 7 of the textbook and utilize one of the techniques from that chapter in your final project. Finally, one should include a query utilizing an index (which you will create to make that query more performant and will be specified in the DDL).
    - One or more instance(s) of a semantic constraint(s) i.e., constraints that cannot be enforced by the DDL in SQL (you will implement this on one of the POST, PUT, or DELETE operations in your API).
    - An instance of using a referential triggered action (for a DELETE or UPDATE operation in the database) other than the default RESTRICT option i.e., SET NULL, SET DEFAULT, or CASCADE qualified with the ON DELETE or ON UPDATE keyword (also specified in the DDL).
  + Lastly, choose one or more functional requirements for your database i.e., what action(s) will the user(s) be doing in your database system? Show the class how you might utilize the API to fulfill the user request(s) i.e., a user of an online shopping database might need to place an order. Placing the order should also decrement the inventory of the item in tandem with inserting a tuple into the order table, etc... this may involve use of many of the endpoints in your Java Spring Boot API. Give one or many such examples in your API and be prepared to show the class at least one of the examples (by demonstrating the use of your Java Spring Boot API in Postman).

**Grading:**

***Due Friday May 10th @ 8:00 a.m.***

Final Presentations (Same day from 8:00 a.m. – 10:00 a.m. in Social Science 362) – Students will be evaluated on their understanding of the database design process as it relates to their modeled Mini world i.e., be ready to give a defense to the class for the design choices you made throughout the semester in any of the following... requirements gathering, conceptual design, logical (implementation) design, and physical design. The deliverables and/or or links to them on GitHub should be included in a formal final project document (which you will submit to Moodle prior to the deadline). Each student will have 5 minutes to present their project in front of the class (so please focus the attention of your final presentation on the following topics)...

* Brief overview of your modeled Mini world (it’s purpose, types of end users, etc...).
* Brief overview of your conceptual model (i.e., ER / E-ER diagram).
* Brief overview of your logical or implementation model (Relational Schema).
* Demonstration of one or more functional requirements using your Java Spring Boot API.

**Additional Notes:**

**Late projects will not be accepted** (i.e., are exempt from the late policy of the class outlined in the Syllabus). If you cannot be present on final presentation day**, you must get former written permission from me**, and submit a 5-minute video presentation of your final project to me no later than the deadline (in combination with the submission of your final project document, through Moodle)\*