Project Document Description

The final report must be a professionally presented, well-organized, typed document addressed to your client. It needs to be accompanied by a complete SQL database and a series of script files, submitted electronically to the Carmen Dropbox, all in a single ZIP file. This ZIP file needs to be neatly and professionally organized, with all filenames appropriately chosen, and all files suitably organized into subdirectories. Include a Table of Contents file named README.txt that explains the layout of your files, including where to find each of the following files in your file structure. See Syllabus and Project Overview document for detailed policies and due dates.

Outline

Part I – The Final Report

Section 1 - Database Description (Logical DB design)

a. Introduction and project summary
b. (E)ERD – Model properly documented and explained.
c. Relational schema properly documented and explained.
d. Relational algebra statements for all SELECT queries from CP02
e. Database fully normalized, with correct justifications

Section 2 - User Manual (DB Implementation)

a. Table description including table functions, keys, constrains, and data types.
b. A catalog of supplied SQL Queries with explanations and sample output for each.
c. INSERT and DELETE SQL code samples.
d. Two indexes properly explained, including SQL code.
e. Two views explained, including SQL code and data resulting from the execution.
f. Two transactions explained, including SQL code.

Section 3 - Team Reports and Graded Checkpoint Documents

a. Detailed description of all team member contributions.
b. Reflection on project completion process.
c. Description of feedback received, and revisions completed throughout the process.
d. Marked Project Checkpoints and Worksheets.

Part II – The SQL Database

*.sqlite, or *.db file: correctly formatted, and ready to open with SQLiteOnline
*.txt files: All CREATE, INSERT, DELETE, and SELECT SQL scripts required to reproduce and test your DB, submitted in separate text files.

Details

Part I – The Final Report

Section 1 - Database Description (Logical DB design)

a. **Introduction and project summary**
Provide several paragraphs to introduce your team and your project to the client, explain the purpose and main components of your DB, outline how it will address all the needs, and explain the benefits of extra features that it will offer. Your entire project document must have a title page, list all team members, display a table of contents, and page numbers. All sections must be consistently formatted, professionally organized, and easy to read.

b. **(E)ERD – Model properly documented and explained.**
Make sure that your ERD completely meets all listed and additional requirements. Ensure that your diagram is self-explanatory, easy to read, correctly identified all entities, relationships, and attributes, uses proper notation, and includes a legend. Each entity must have a proper key. All relationships must have cardinality and participation shown. Be mindful about using weak entities. Remember that (E)ER model cannot have any standalone entities, but only appropriate entities need to be associated in relationships. All relationships should be binary. Remember that FKs are not shown on ERD. Your ERD must fully and correctly represent your DB needs and supply all necessary information for the development of your relational schema.

c. **Relational schema properly documented and explained.**
A professionally presented, well-formatted relational schema for the database shown in your (E)ERD. Use mapping algorithm explained in this course in proper order to evaluate and map each element shown in your (E)ERD. Your relational schema must be fully consistent with your (E)ERD. Show and explain all the steps you take in the process. This schema must be annotated with the primary key for each table, and all required foreign keys. Make sure that connections between all PKs and FKs are clear. Introduce surrogate keys if proper natural PKs do not exist.

d. **Relational algebra statements for all SELECT queries from CP02**
List and explain each of RA statements created in CP02 questions 3 and 4. Make certain that they use correct syntax and agree with your most current relational schema.

e. **Database fully normalized, with correct justifications**
For each table of your relational schema, give a brief description of the level of normalization achieved for that table. Show all functional dependencies for each table. Ensure that each table in your schema is in 3NF and complete the required steps to bring each table to 3NF. If a table is not in BCNF, explain why. Once again, your relational schema must be fully consistent with all the tables discussed and shown in this section.

Section 2 - User Manual (DB Implementation)

a. **Table description including table functions, keys, constrains, and data types.**
List all your proposed DB tables. For each table describe its purpose, list all the fields, PKs, FKs, constraints for each field, and SQL approved data types. For FKs, identify matching tables and source PKs.

b. **A catalog of supplied SQL Queries with explanations and sample output for each.**
List all SQL queries created in CP03. For each query provide a short narrative explaining its purpose and results. Provide complete SQL code, list any assumptions or parameters if applicable, and show sample output. Remember that your DB should hold appropriate test data for all your queries to execute and produce meaningful results.

c. **INSERT and DELETE SQL code samples.**
Provide sample INSERT and DELETE SQL statements that will demonstrate how data can be entered in the tables and then deleted. Queries should execute. Include sample output showing the results of executing these queries. You should have 3-5 insert and 3-5 delete statements that will affect at least 2 tables each.

d. **Two indexes properly explained, including SQL code.**
Description of two indexes implemented in your DB that improve the performance. Explain their purpose and what you achieve by implementing them. Explain what type of indexing is used in each one of them (Clustering, Hash, or B-tree) and why. Provide valid SQL code for each index.

e. **Two views explained, including SQL code and data resulting from the execution.**
Describe two views that you created for your DB. Each view must involve joining at least two tables together and must include calculations/aggregation and/or nesting. Provide SQL code used for constructing your views and the output they produce.

f. **Two transactions explained, including SQL code.**
List two sample transactions that you established for your DB. Clearly document their purpose and function. Explain why it is crucial to execute those as one unit of processing. Each transaction should include read and write operations on at least two tables, with appropriate error and constraint checks and responses. Provide valid SQL code for each transaction.

Section 3 - Team Reports and Graded Checkpoint Documents

a. **Detailed description of all team member contributions.**
Provide a list of team members and their contributions. Use this space to praise any particular team members or share common concerns.

b. **Reflection on project completion process.**
Share your feedback on your teamwork dynamics, project timeframe, work schedules, project development process, comments, and suggestions. What would you do differently to make this process more efficient? Provide suggestions for future teams.

c. **Description of feedback received, and revisions completed throughout the process.**
List feedback you have received for each CP document and explain how you have addressed each feedback item and what other changes you have made to improve your design at each CP stage.

d. **Marked Project Checkpoints and Worksheets.**
Include all four previously submitted CP documents showing your work.

Part II – The SQL Database

1. *.sqlite, or *.db file: A binary version of your complete database, suitable for opening with the SQLiteOnline application.

2. Text formatted SQL files. These six (6) files described below should be able to recreate your database from scratch if your binary file is corrupted or lost. Make sure you provide instructions on how to use these scripts in a separate text file. These text formatted SQL files should only contain SQL script and your code should execute in SQLiteOnline without any changes. The entire SQL code should execute and produce correct and meaningful results and match all other components of your DB design.

- CreateQueries.txt
A text file containing all the scripts needed to create your database and all its tables and views with proper constraints, data types, keys, etc. This file should be properly commented and should execute properly if pasted into an SQL interpreter.

- InsertQueries.txt
A text file containing all the scripts needed to populate your entire DB with all its data. Ensure to have enough data in each table to allow to test your queries and produce meaningful results. The suggested number is at least 20 records for major tables. Keep in mind that it may be applicable to have less records in certain tables and more in others.

- InsertDeleteQueries.txt
A text file containing your additional INSERT and DELETE SQL statements created as part of Section 2 - User Manual question C to demonstrate how records can be inserted and deleted from your DB.

- SimpleQueries.txt
A text file containing a complete and working version of SQL SELECT queries required in question 3 of CP03.

- ExtraQueries.txt
A text file containing a complete and working version of SQL SELECT queries required in question 4 of CP03

- AdvancedQueries.txt
A text file containing a complete and working version of SQL SELECT queries required in question 5 of CP03