Project PART 3

Your team is almost done with the logical design of the DB for Dr. Hope Smilow's business. Now you need verify your design, make few last improvements, and process to its implementation using SQL. After completing tasks described here, you should be able to present a working DB populated with sample data to your client. In addition to that, you will also present your client with a list of SQL queries that will allow her to retrieve specific data and create reports.

1. Review feedback provided for PART 2 and make necessary changes. Your (E)ERD, relational schema, and relational algebra should be fully correct and consistent. Do not proceed until these tasks are complete. Your entire team needs to work on improving and verifying the design. If your ERD was updated, describe all updates, and include new ERD and the original ERD from PART 2.
2. Apply process of normalization as learned in class to each table in your relational schema. At the end of the process all relations in your schema must be in BCNF. Normalization rules are applied to one relation at a time. **Make sure that your documentation shows 1NF-BCNF rules applied step-by-step to each relation and listed by relation and not by NF. For each relation on your final schema:**
   - Check that a relation is in 1NF and if it is not, bring it to 1NF. Explain the process and changes made.
   - List functional dependencies. Make sure to consider all the possible dependencies in each relation and not just the ones from your primary keys.
   - Determine the highest current normal form of that relation. Apply rules of 2NF, 3NF, and BCNF in the proper order. Explain the process and changes made if any.
   - You do not need to update ERD at this point, but you need to update your relational schema to ensure that after this step all relations are in BCNF.

Show your final relational schema after normalization.

3. Given your normalized relational schema, create a text file containing the SQL code to create your database and all the tables in your schema. Populate all tables in your DB with an appropriate number of records to test your queries and produce meaningful results. Recommended number of records per table is between 10-20 depending on table. However, that number can fluctuate depending on table's role in your DB. Save all your SQL code including INSERT statements used to populate tables with data. If your DB is deleted, you should be able to execute your SQL code as a script in proper order to fully recreate your DB including all tables, constrains, views, and data. Ensure that your code runs and produces correct results in either SSMS or SQLiteOnline (sqliteonline.com) as we will be using one of those platforms to test your code. Clearly indicate in your solution which one was used. Save all CREATE / ALTER TABLE STATEMENTS in a file called "CreateQueries.txt" and all applicable INSERT statements in a file called "InsertQueries.txt".

IMPORANT NOTE: For the following questions, if your relational schema cannot provide answers to these queries, revise your (E)ERD, relational schema, and SQL code in question 3 above to contain the appropriate data for constructing and running all the queries outlined below. On the other hand, if your database contains needed source data but in non-aggregated form, you should NOT revise your model but instead figure out how to aggregate it for the queries!

4. Given your relational schema, provide the SQL to perform the following queries that were previously documented in RA. If your schema cannot provide answers to these queries, revise your

ER Model, your relational schema, and your SQL code in question 3 to contain the appropriate information for these queries. These queries should be provided in a plain text file named "SimpleQueries.txt". Clearly label each query using SQL comments.

a. Create a list of patients and the medications they currently take. Sort your list by patient's last name and medication name in alphabetical order. Include other applicable details such as date prescribed and dosage.

b. Display patient information for patients who currently have Delta Dental insurance policy.

c. Generate a list of procedures and dates of service performed by doctor Smilow.

d. Print out a list of past due invoices with patient contact information. Past due is defined as over 30 days old with a balance over $10.

e. Find the patients who brought the most revenue in the past year. You can define how many records you want to display in the result of this query.

f. Create a list of doctors who performed less than 5 procedures this year.

g. Find the highest paying procedures, procedure price, and the total number of those procedures performed. Sort your list with highest paying procedures showing at the top of your list.

h. Create a list of all payment types accepted, number of times each of them was used, and total amount charged to that type of payment.

i. Find the name of the most popular insurance plan currently used by the patients.

5. For Project PART 2 question 4, you were asked to come up with three additional interesting queries that your database can provide. Provide the SQL to perform those queries. These queries should be

provided in a plain text file named “ExtraQueries.txt”. Clearly label each query using SQL comments. Each of your queries should include at least one of these. Make sure queries are sufficiently complex and utilize multiple tables and operations in addition to one of the required here:

a. outer joins

b. aggregate function (min, max, average, etc.)

c. “extra” entities from PART 1

6. CROSS_CHECK: Make sure that all your SQL code is properly formatted, easy to read, and label each query with SQL supported comments. DO not use any non-SQL supported contents in your scripts. They should execute as one unit by using copy / paste commands. All SELECT queries should produce meaningful results.

7. Document work being done for this portion of the project and team member contributions. Report any relevant team issues / praises / concerns.

8. Once you have completed all your work, create a ZIP archive containing:

   - A document showing your most current version of (E)ERD, relational schema, and relational algebra with PART 2 feedback addressed. **Submit a professionally written and well formatted report showing ALL your work. Your ERD, schema, RA, and all the written work must be submitted in one document. Do not submit separate files or links.**

- **A binary version of your database**, suitable for opening with either SSMS or the SQLiteOnline application (*sqlite, *.db). Clearly specify in your report and code comments if it was created in SSMS or SQLiteonline.
- Text formatted SQL files for questions 3-5:
  - **CreateQueries.txt**
  - **InsertQueries.txt**
  - **SimpleQueries.txt**
  - **ExtraQueries.txt**

  Before submitting your work: Make sure that the information presented in your (E)ERD, relational schema, and all your queries is fully consistent, and all your queries execute correctly and produce expected results! Remember that each of the SQL files should execute as a script, use SQL comments do identify each query, do not use any non-SQL compatible text or syntax in your code. Entire team is responsible to check for presence and correctness of all submitted work. **Clearly indicate what RDBMS you used to create your code so we can use the same one to test it!**

9. Save all your work as you will need to use it for next phase of the project.