Project PART 4

We are about to wrap up our DB design and development. The goal of this project is to finalize our design and documentation and develop few remaining components. At this point, all teams are expected to have fully correct logical design of the DB, implemented DB with tables, constraints, data, and initial set of SELECT statements all of which execute and produce results.

1. Review feedback provided for PART 3 and make necessary changes. Your (E)ERD, relational schema, and relational algebra should be fully correct and consistent. If any portions of your project (logical or physical design) had to be updated, list all updates, and show all your work before and after the updates.
2. Show your most current ERD and complete relational schema.
3. Verify that all your scripts from Part 3 now are logically correct, fully execute, and produce correct results. You will need to resubmit updated scripts for this part of the project. Points will be deducted again if any errors still exist.
4. Create new script that will contain additional INSERT and DELETE SQL statements that will demonstrate how data can be entered in the tables and then deleted. This script should not conflict with your INSERT SCRIPT from PART 3. 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. Statements must execute in proper order to first insert and then delete the same data, demonstrate referential integrity constrains compliance, and to keep your DB in a valid state. Save this script as “InsertDeleteQueries.txt”.
5. Suggest and provide description of two indexes that you want to implement in your DB to 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. Assume that PKs and FKs are already indexed by RDBMS.

6. For your database, propose at least two interesting views. 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. Do NOT reuse queries from previous steps.
7. List two sample transactions that you want to establish 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 at least 2 write operations on at least two tables, with appropriate error and constraint checks and responses. Provide valid SQL code for each transaction.
8. Document work being done for this portion of the project and team member contributions.
    a. Provide a list of team members and their contributions. Use this space to praise particular team members or share common concerns.
    b. 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.
9. 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 3 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.
- Most current versions of 5 text formatted SQL files:
    - **CreateQueries.txt**
    - **InsertQueries.txt**
    - **SimpleQueries.txt**
    - **ExtraQueries.txt**
    - **InsertDelete.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!**

10. Save all your work and have a team party! You have completed the project!