CSE 3241 Project Checkpoint 04

Functional Dependencies, Normal Forms, Indexes, Transactions

In a NEATLY TYPED document, provide the following:

1. Provide a current version of your ER Diagram and Relational Model as per Project Checkpoint 03. If you were instructed to change the model for Project Checkpoint 03, make sure you use the revised versions of your models.

2. Check that each relation in your schema is in 1NF and if they are not, bring them to 1NF. For each relation schema (table) in your model, indicate the functional dependencies. Make sure to consider all the possible dependencies in each relation and not just the ones from your primary keys.

3. For each relation schema in your model, determine the highest normal form of the relation. If the relation is not in 3NF, rewrite your relation schema so that it is in at least 3NF.

4. For each relation schema in your model that is in 3NF but not in BCNF, either rewrite the relation schema to BCNF or provide a short justification for why this relation should be an exception to the rule of putting relations into BCNF.

5. For your database, propose at least two interesting views that can be built from your relations. These views must involve joining at least two tables together and must include calculations/aggregation/and /nesting. Provide SQL code for constructing your views along with the English language description of these views and what they do.

6. Description of two indexes that you want to implement in your DB. Explain their purpose and what you want to achieve by implementing them. Explain what type of indexing would be most appropriate for each one of them (Clustering, Hash, or B-tree) and why. To properly answer this question, look at your queries to identify the best candidates for indexing. Provide valid SQL code for each index.

7. Two sample transactions that you want to establish in your DB. Clearly document their purpose and function. Explain why it is crucial to execute each transaction you have created 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.

8. Think about your DB design and the buyer/seller financial portal needs and possible inquiries and reports they may have. If this was not a classical relational DB, what data would you want to pre-aggregate/calculate and store in order to speed up information retrieval and meet user's information needs? Provide at least 3 separate examples. Do not make any changes to your DB design or SQL code based on this question.