Lab 9 & 10 -- Questions & Instructions (Oct 21, 2021 & Oct 28, 2021)

You are required to attempt all questions…
Answers need to be succinct and in your own words…
Verbosity is undesirable…

Total points = 120 points

Your submission will be a singular pdf file <Naming convention Roll_No._FirstName_Lab6.pdf> - containing detailed assumptions, observations, theoretical answers/expectations/predictions, and snapshots of code with outputs and observations… A demo of your code would be required for evaluations

Deadline for submission: Nov 3, 2021 .

1. Use the multi-table database for a library from Assignment 7&8
2. Simulate an environment that allows multithreaded database access for multi-transaction handling (References provided below) **(10 points)**
3. Simulate a deadlock situation derived from multiple transactions attempting to access & update the same tuple - **(10 + 15 * 3 = 55 points)**
    a. Justify that the situation is indeed one of a deadlock
    b. If you had to choose a victim to resolve the deadlock, what would your algorithm be? [Note: You are allowed to think of a mix of priority mechanisms + lock + timestamp mechanisms for deadlock resolution]
    c. Attempt execution of your algorithm for deadlock avoidance
    d. Comment on the conflict serializability of your algorithm
4. Simulate a deadlock situation derived from multiple transactions attempting to access & update the same table - **(10 + 15 * 3 = 55 points)**
    a. Justify that the situation is indeed one of a deadlock
    b. If you had to choose a victim to resolve the deadlock, what would your algorithm be? [Note: You are allowed to think of a mix of priority mechanisms + lock + timestamp mechanisms for deadlock resolution]
    c. Attempt execution of your algorithm for deadlock avoidance
    d. Comment on the conflict serializability of your algorithm
5. Create a real-life scenario in which an in-house application returns an error to the users, and users notify the development team about this error. The development team realizes that it is a deadlock issue, but they could not find the main reason for the problem. Under these circumstances, the team decides to receive consultancy service from an experienced database administrator. Do simulation for the same.

6. Take the following scenario & do the simulation for the same. Suppose we have two database sessions called A and B. Let's say that session A requests and has a lock on some data – and let's call the data Y. And then session B has a lock on some data that we will call Z. But now, lets say that session A needs a lock on data Z in order to run another SQL statement, but that lock is currently held by session B. And, let's say that session B needs a lock on data Y, but that lock is currently held by session A. This means that session B is waiting on session A's lock and session B is waiting for session A's lock.
7. Simulate a deadlock situation to prove that The deadlock state can be changed back to stable state by using Rollback statement.

8. Simulate the Wait-die situation in deadlock for When transaction Ti requests a data item currently held by Tj, Ti is allowed to wait only if it has a timestamp smaller than that of Tj (that is, Ti is older than Tj). Otherwise, Ti is rolled back (dies).

References:

1. https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15
2. https://www.ibm.com/docs/en/db2/11.1?topic=designing-concurrent-transactions-multi-threaded-database-access
3. https://www.columbia.edu/sec/acis/db2/db2a0/db2a082.htm
4. https://www.sqlite.org/threadsafe.html
