Final Exam Solutions

Question 1: Deferred Update Technique

In table 01, there are two transactions T1 and T2 running some read and write operations in an interleaved manner. After each operation, what will be the values of X, Y, and Z in the system log and in the database? Note that the recovery technique used is "deferred update". Moreover, before T1 and T2, the values of X, Y, and Z in the database are: X0=20, Y0=18, Z0=3.

For T1:

- Read item(X) - No change yet, as it's a read operation.
- Write item(X,20,25) - The system log will record the new value of X as 25, but the database will still have X as 20 until the commit.

For T2:

- Read item(Y) - No change yet, as it's a read operation.
- Write item(Y,18,19) - The system log will record the new value of Y as 19, but the database will still have Y as 18 until the commit.
- Read Item(Z) - No change yet, as it's a read operation.
- Write Item(Z,3,5) - The system log will record the new value of Z as 5, but the database will still have Z as 3 until the commit.
- Commit - At this point, all deferred writes are made to the database, so the database will now have X=25, Y=19, and Z=5.

Question 2: Concurrency Control Problem

Consider the following two concurrent transactions in the scenario to update shared variable

winner_name and the corresponding private variable prize money. Suppose these two transactions are executed concurrently, and T1 starts first. Each step of transaction is viewed as below:

a) Isolation levels:

- Level-0 Isolation (Read uncommitted): This level allows dirty reads. If T1 reads the winner name after T2 writes but before T2 commits, it may read uncommitted data, which could lead to inconsistent data.
- Level-1 Isolation (Read committed): This level does not allow dirty reads. T1 can only read the winner name after T2 has committed, ensuring that only committed data is read.

b) Concurrency problem:

- The concurrency problem that may arise here is a dirty read. If T1 reads the value of winner_name after T2 has written to it but before it has committed, T1 may take actions based on uncommitted data that might never be committed if T2 fails or is rolled back.