Quiz 4 Answers

1. [10pts] For the following questions, clearly circle True or False.

1. Using 2PL guarantees that there will be no deadlocks during a concurrent execution of transactions.
   FALSE

2. All four SQL isolation levels guarantee that there will be no dirty reads during a concurrent execution of transactions.
   FALSE

3. The database is allowed to become temporarily inconsistent in the middle of a transaction execution.
   TRUE

4. Serializability of a transaction schedule implies recoverability.
   FALSE

5. WW conflicts cannot arise if all transactions perform blind writes.
   FALSE

2. [10pts] Consider the following two transactions and the given interleaved schedule:

   T1 : R(A), W(A), R(B), W(B), Commit
   T2 : R(B), R(C), W(C), W(B), Commit

   Is the schedule serializable? If you claim yes, write an equivalent serial (non-interleaved) execution of the two transactions. If you claim no, explain why it is not serializable.

   ANSWER: No, it is not serializable. There is a write-write conflict on B; the update by T1 is lost, since T2 overwrites it after reading an older value of B. Thus, it is not equivalent to either T1 → T2 or T2 → T1.

3. [10pts] Give a clear example of an interleaved schedule with three transactions on a database with three distinct data objects A, B, and C such that the schedule has some
concurrency but still satisfies Strict 2PL. Both of your transactions should have at least 2 write actions on at least two distinct data objects.

ANSWER: Many correct answers exist, as mentioned in class. Below is one possibility:

T1 : R(A), W(A), R(B), W(B), Commit
T2 : R(C), W(C), R(B), W(B), Commit
T3: R(A), W(A), R(B), W(B), Commit

The following interleaved schedule satisfies strict 2PL but still has interleaving of transactions and thus, some concurrency.

\[ R_{T1}(A), W_{T1}(A), R_{T2}(C), W_{T2}(C), R_{T1}(B), W_{T1}(B), Commit_{T1}, \]
\[ R_{T2}(B), R_{T3}(A), W_{T2}(B), W_{T3}(A), Commit_{T2}, R_{T3}(B), W_{T3}(B), Commit_{T3} \]

To explain the above, T1 first acquires X locks on A and B and holds them till it commits as per 2PL. T2 acquires X lock on C concurrently with T1 but waits till T1 commits for X lock on B (as per 2PL). Once T1 commits, T2 acquires X lock on B, while T3 acquires X lock on A concurrently. But T3 has to wait till T2 commits for X lock on B (as per 2PL), following which it acquires that lock. Note that this schedule is equivalent to the serial schedule \( T1 \rightarrow T2 \rightarrow T3 \) but has more concurrency.

4. [5pts]: In terms of the ACID properties, explain briefly why recoverability of a schedule is important.

ANSWER: Check class notes.

5. [5pts]: Out of WW, WR, and RW(R) conflicts, which ones are avoided by the READ COMMITTED isolation level of SQL?

ANSWER: WW conflict is avoided by all four levels. In addition, READ COMMITTED also avoids WR conflicts.