CS 448 - Homework 4 - Extra Credit Assignment

Transaction, Concurrency, and Recovery
(100 points)

Spring 2020

Due on: Friday May 1, 2020, 11:59PM

Note: The deadline for this extra credit assignment is firm. There are no late days. It will not be accepted past its deadline. It can give you credit equivalent to an entire homework assignment. If you have done all your homeworks successfully, it can cover difficiencies in your midterm or in your project grades.

1 Part 1: Transaction Management

1. Consider a database with object X and Y and assume that there are two transactions T1 and T2. Transaction T1 reads objects X and Y and then writes object X. Transaction T2 reads objects X and Y and writes objects X and Y.

   (a) Give an example schedule with actions of transaction T1 and T2 on objects X and Y that results in a write-read conflict.

   (b) Give an example schedule with actions of transaction T1 and T2 on objects X and Y that results in a read-write conflict.

   (c) Give an example schedule with actions of transaction T1 and T2 on objects X and Y that results in a write-write conflict.

   (d) For each of the schedules, show that strict 2PL disallows the schedule.

2. Consider the following schema:
   Suppliers(sid: integer, sname: string, address: string)
   Parts(pid: integer, pname: string, color: string)
   Catalog(sid: integer, pid: integer, cost: float)

   The Catalog relation lists the prices charged for parts by Suppliers.
   For each of the following transactions, state the SQL isolation level that you would use and explain why you chose it.

   (a) A transaction that adds a new part to a supplier's catalog.

   (b) A transaction that increases the price that a supplier charges for a part.

   (c) A transaction that determines the total number of items for a given supplier.

   (d) A transaction that shows, for each part, the supplier that supplies the part at the lowest price.

2 Part 2: Concurrency Control

1. For each of the following schedules, state whether it is conflict-serializable and/or view-serializable. If you cannot decide whether a schedule belongs to either class, explain briefly.

   (a) T1:R(X) T2:R(X) T1:W(X) T2:W(X)

   (b) T1:W(X) T2:R(Y) T1:R(Y) T2:R(X)

   (c) T1:R(X) T2:R(Y) T3:W(X) T2:R(X) T1:R(Y)

   (d) T1:R(X) T1:R(Y) T1:W(X) T2:R(Y) T3:W(Y) T1:W(X) T2:R(Y)

   (e) T1:R(X) T2:W(X) T1:W(X) T3:W(X)

2. Recoverability and serializability are both important properties of concurrent transaction schedules. They are also orthogonal. Serializability requires that the schedule be equivalent to some serial ordering of the transactions. Recoverability requires that each transaction commits only after all of the transactions from which is has read data have also committed.

   Using the following two transactions:

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

   give examples of schedules that are:

   (a) recoverable and serializable

   (b) recoverable and not serializable

   (c) not recoverable and serializable

3 Part 3: Recovery

Consider the execution below:

LSN	LOG
00	update: T1 writes P2
10	update: T1 writes P1
20	update: T2 writes P5
30	update: T3 writes P3
40	T3 commit
50	update: T2 writes P5
60	update: T2 writes P3
70	T2 abort

1. Extend the figure to show prevLSN and undonextLSN values.
2. Describe the actions taken to rollback Transaction T2.
3. Show the log after T2 is rolled back, including all prevLSN and undonextLSN values in log records.