EECS 484 Database Management Systems

Homework 5 (100 points)

Due Monday, April 10th. 11:55PM

Please read the following instructions before starting the homework. Failing to follow the submission rules will lead to -4 points (e.g., wrong file format, wrong group submission).

You are *allowed* to work on this homework in pairs.

Deliverables: You need to submit all your solutions in a single PDF **on CANVAS**. Your solutions can be either handwritten or created electronically, as long as they are clear. If you are working in pair, make sure to **JOIN THE SAME GROUP** on Canvas (you need to join the group every time for different homework and projects) and make only one group submission.

*As usual, **No late days for homeworks!** If you miss the due date, you get 0 points. No exceptions on this.*

By submitting this homework, you are agreeing to abide by the Honor Code:
I have neither given nor received unauthorized aid on this assignment, nor have I concealed any violations of the Honor Code.

Part 1: Serializability (66 points)

Question 1 (16 points)

Consider the following (incomplete) schedule involving transactions T1, T2, T3, and T4. *Your answers should consider only what is contained in the schedule snapshot below (i.e., do not worry about what may happen afterward).*

T1	**T2**	**T3**	**T4**
		R(Z)	
		W(Z)	
	R(X)		
	W(X)		
	R(Y)		
R(X)			
			R(Y)
			COMMIT
COMMIT			
	COMMIT		

(A) **[4 points]** Is this schedule recoverable? Explain.
(B) **[4 points]** Does this schedule avoid cascading aborts? Explain.
(C) **[4 points]** Could it be generated by non-strict 2PL? Explain.
(D) **[4 points]** Could it be generated by strict 2PL? Explain.

Question 2 (32 points)

Determine if the following schedules are: conflict serializable, and/or serializable. If it is, provide a possible equivalent serial schedule.

Schedule A [8 points]

T1	T2	T3
R(A)		
	W(A) COMMIT	
		R(A)
		W(A) COMMIT
W(A) COMMIT		

Schedule B [8 points]

T1	T2	T3
R(A)		
		R(B) R(A)
	W(A)	
R(B)		
	W(B) COMMIT	
W(B) COMMIT		
		COMMIT

Schedule C [8 points]

T1	T2	T3
		R(A)
R(B) W(B) COMMIT		
		W(B)
	R(B) W(A)	
		W(A) COMMIT
	ABORT	

Schedule D [8 points]

T1	T2	T3	T4
			R(A)
R(B) R(D)			
			W(B)
		R(B) W(C)	
R(C) W(B) COMMIT			
	R(C) W(B) COMMIT		
		COMMIT	
			R(C) W(C) W(B) COMMIT

Question 3 (18 points)

Determine if the **schedules B, C, D in Question 2 will deadlock** if **2PL** is used, and **state which transactions deadlock** if any.

Assume that lock requests are filled in the order they arrive in the schedule. If a transaction has a request in queue, it will not ask for other locks.

A shared lock is forced into the queue if there is an exclusive lock request on the same object and is also in queue. This is to prevent starvation.

Furthermore, a lock upgrade request (read lock to write lock) is granted immediately if the requesting transaction is the only holder of the shared lock on the object. Otherwise, the requester must wait until all other shared locks are released.

Part 2: ACID and ARIES Protocol (34 points)

Question 4 (15 points)

Consider the following example of operations on a DBMS:

1. Transaction 1 modifies Page 1
2. Transaction 1 modifies Page 3
3. Transaction 2 modifies Page 2
4. Crash!!!!

Assume that:

- The memory can only accommodate 2 pages
- Avoid writing pages back to disk if not necessary (do not force)
- No logging or any other kind of book keeping technique is used here

(A) **[3 points]** At timestamp 3, What does the DBMS have to do to make sure there is space to modify Page 2?

(B) **[4 points]** After timestamp 4, the DBMS came back without running any recovery protocol. Which property in ACID does it violates now? Explain it in 1~2 sentences.

Now consider another scenario:

1. Transaction 1 modifies Page 1
2. Transaction 1 modifies Page 3
3. Transaction 1 commits
4. Transaction 2 modifies Page 2
5. Crash!!!!

Again, assume that:

- The memory can accommodate 4 pages
- Avoid writing pages back to disk if not necessary (do not force)
- No logging or any other kind of book keeping technique is used here

(C) **[4 points]** After timestamp 5., the DBMS came back without running any recovery protocol. Which property in ACID does it violates now? Explain it in 1~2 sentences.

(D) **[4 points]** How can we restore this property without using logging? Is there any tradeoff?

Question 5 (19 points)

Consider the following log file, which was found on disk, and the ARIES recovery protocol:

LSN	LOG
1	T1 writes to P1
2	T2 writes to P2
3	T3 writes to P3
4	T1 writes to P1
5	T2 writes to P2
6	T1 commit
7	T1 end
8	begin checkpoint
9	end checkpoint (along with the dirty page table and transaction table)
10	T3 writes to P1
11	T3 commit
12	T3 end
13	T2 writes to P3
14	CRASH

(A) **[4 points]** Describe the dirty page table at the end of the ANALYSIS phase of recovery. For example:

Page ID	**recLSN**
P500	7
P605	9

(B) **[4 points]** Describe the transaction table at the end of the ANALYSIS phase of recovery. For example:

Txid	**lastLSN**	**Satus**
T1000	4	U
T2000	3	U

(C) **[3 points]** Which transactions are “loser” transactions, whose result does not persist?

(D) **[4 points]** Are there any new compensation log records (CLRs) written during the REDO phase of recovery? Explain.

(E) **[4 points]** How many CLR records are added in the UNDO phase? Briefly describe them.