CENG 352
Database Management Systems
Spring 2024
Written Assignment 3

Q1) Consider the following schedule involving three transactions T1, T2 and T3:

Time	T1	T2	T3
1	R(C)		
2			R(B)
3	R(A)		
4		R(C)	
5			W(B)
6	W(A)		
7		R(B)	
8			R(A)
9		W(C)	
10		W(B)	
11	Commit		
12		Commit	
13			Commit

Describe how the **strict two-phase locking protocol with deadlock detection** would handle the schedule by filling in the following table*. Indicate shared locks by $\mathbf{S_i}$ and exclusive locks by $\mathbf{X_i}$ where i is the transaction number. Indicate the operations with a subscripted transaction number (see the first operation in the table).

Operation	LOCKS			Waits-for Graph
	A	B	C	
$R_1(C)$			S_1	

* Add as many rows as you may need.

Q2) Consider the same schedule given in Q1. Assuming T1 is the oldest and T3 is the youngest transaction, describe at what points **strict two-phase locking protocol with wound-wait deadlock prevention** scheme would handle the schedule differently. State in which order the transactions commit eventually.

Q3) Consider the schedule S below. The symbol $R_i(X)$ stands for a read by transaction T_i to item X and $W_i(X)$ stands for a write by T_i to item X. Suppose **Basic Timestamp Ordering with commit bit** is used as the concurrency control protocol. Suppose furthermore that transaction T_i has timestamp i. By filling in the following table*, describe what happens during each operation below, justifying whether the operation is accepted, rejected, ignored, or delayed. Also show how the Read Timestamps (RT), Write Timestamps (WT) and Commit bits (C) of the data items are updated in each step. You can assume that the initial values for read and write timestamps of the data items A, and B are zero and commit bits are true.

S: $R_2(B)$ $R_3(B)$ $R_1(A)$ $R_1(B)$ $W_3(A)$ $W_2(A)$ $W_1(B)$ $W_4(B)$ commit T3 $R_4(A)$ Abort T4

Operation Accepted(A) Rejected (R) Ignored(I) A B
RTS WTS RTS WTS

Operation	Accepted(A) Rejected(R) Ignored(I) Delayed(D)	A			B		
		RT	WT	C	RT	WT	C

* Add as many rows as you may need.

Q3. Consider four transactions T1, T2, T3, T4 accessing data items A, B, C in a multiuser database. Assume that the database system uses Strict Two Phase Locking in conjunction with the crash recovery system which uses NO FORCE and STEAL policies. Assume only two pages are available in the buffer pool.

a) Show the contents of disk, buffer pool and log file as well as pages moving back and forth between memory and disk for the following schedule of the transactions, by filling in the table on the next page*.

T1	**T2**	**T3**
	R(A)	
R(B)		
W(B)		
	W(A)	
R(C)		
W(C)		
Commit		
		R(C)
	R(B)	
		W(B)
	W(B)	
		Commit
	Commit	

In filling in the table, use the following notation:

A^i : Page A with PageLSN i
S_i: Shared locked acquired by transaction i
X_i: eXclusive lock acquired by transaction i
R_i: Read operation by transaction i
W_i: Write operation by transaction i
<k T_i P^j P^k>: Log entry with LSN k for transaction i. Update P^j to P^k where j and k are pageLSNs.
<k T_i abort>: LSN k to record abort of transaction T_i.
<k T_i commit>: LSN k to record commit of transaction T_i.

* Add as many rows as you may need.

Operation	Locks			Buffer	Log	Disk Pages		
	A	B	C			A	B	C
$R_2(A)$	S_2			A^0		A^0	B^0	C^0

b) Assume the system crashes just before Transaction T3 writes B (i.e. just after transaction T2 reads B). Explain the Analysis, Redo and Undo phases over the log by writing down the contents of the transaction table and dirty page table. List the operations that are undone. Also write down the final state of the disk pages.

Analysis Phase:

Transaction Table*

Transaction id	LSN	status

Dirty Page Table*

Page	recLSN

Redo phase: Where does Redo start? List the log records that are redone.

Undo Phase: List of operations that are undone.

What is the final state of the database?

* Add as many rows as you may need.