Name: Zhang / Hexiao Student#: 20932780 Date: 11/15

This is an individual exercise; each student must submit their own worksheet.

MSBD 6000L: Database Systems

Lecture 19 Exercises
Concurrency Control: Lock-based Protocols

Exercise 4: Show that the following schedule is conflict serializable according to 2PL by adding lock-s(), lock-x() and unlock() instructions, as necessary, to the schedule. If possible, add the lock-s(), lock-x() and unlock() instructions so that *no transaction is required to wait.*

T_1	T_2	T_3
lock-s(X) read(X)		
	lock-s(X) read(X)	
		lock-s(Y) read(Y)
lock-x(Z) unlock(X) read(Z)		
	lock-s(Y) read(Y)	
	lock-x(X) write(X)	
	unlock(X) unlock(Y)	lock-x(X) read(X)
		write(X) unlock(X)
write(Z) unlock(Z)		

Give the equivalent serial schedule

T1, T2, T3

Exercise 5: *In which positions, A to E, can an unlock(X) instruction be inserted if the schedule is according to:*

a) *strict 2PL (circle the correct answer)*
   i. {A} {B} {C} {D}
   (ii) {A} {B} {C} {D} {E}
   iii. {A} {C} {D}
   iv. {B} {E}
   v. {A} {C} {D} {E}

b) *rigorous 2PL (circle the correct answer)*
   i. {A} {B} {C} {D}
   ii. {A} {B} {C} {D} {E}
   iii. {A} {C} {D}
   (iv) {B} {E}
   v. {A} {C} {D} {E}

T_1	T_2
lock-s(X) read(X)	
	lock-s(X)
lock-x(Y) **{A}**	
read(Y) write(Y)	
	read(X) **{C}**
commit unlock(Y) **{B}**	
	{D}
	commit **{E}**

Upload this completed exercise worksheet to Canvas by 11 p.m. on Thursday of this week.

Name: _____ / _____ Student#: _____ Date: _____
Family/Last (PRINT) Given/First (PRINT)

This is an individual exercise; each student must submit their own worksheet.

MSBD 6000L: Database Systems

Lecture 19 Exercises
Concurrency Control: Lock-based Protocols

Exercise 6: Consider the schedule shown below.

T_1	T_2	T_3	T_4
lock-s(X)			
read(X)			
lock-x(X)			
write(X)			
	lock-s(X)		
	read(X)		
		lock-s(Y)	
		read(Y)	
		lock-x(Y)	
		write(Y)	
	lock-x(X)	commit	
	write(X)	unlock(Y)	
	commit		lock-s(Y)
	unlock(X)		read(Y)
lock-x(Y)			commit
write(Y) < commit			unlock(Y)
unlock(Y)			
unlock(X)			

a) Is the schedule conflict serializable? ☑ Yes ☐ No

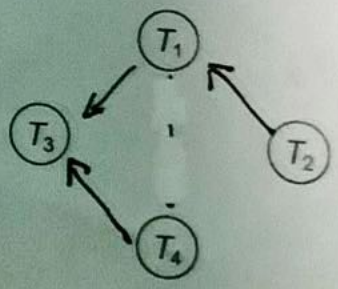
If yes, give the equivalent serial schedule T_3, T_4, T_1, T_2

b) If T_3 aborts at the end of this schedule, which other transactions will be rolled back? T_4

c) If T_1 aborts at the end of this schedule, which other transactions will be rolled back? T_2

d) Construct the wait-for graph that results from this schedule if all locks are only exclusive-locks (lock-x), no locks are released, and the execution process runs to the point of lock-x(Y) just before write(Y) in T_1.

Is the system in a deadlock state? ☐ Yes ☑ No



Wait-for graph

e) Add lock-s(), lock-x() and unlock() instructions to the schedule according to strict 2PL. Indicate which transactions, if any, are required to wait and which instructions cause the wait.

read(X) in T_2 waits for write(X) in T_1.
read(Y) in T_4 waits for write(Y) in T_3