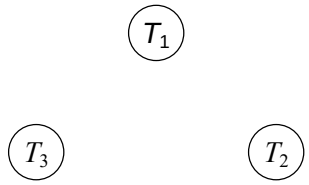
COMP 3311: Database Management Systems

Tutorial 10
Transactions and Concurrency Control

Exercise 1: For the following schedule, state whether it is serializable, recoverable and cascadeless. Justify your answers.

T_1	T_2	T_3
		read(Y)
		read(Z)
read(X)		
	read(Y)	
read(Y)		
	read(Z)	
write(Y)		
		write(Z)
write(X)		



Precedence graph

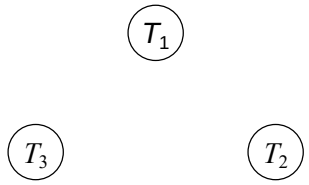
Serializable: ☐ Yes ☐ No Justification?

Recoverable: ☐ Yes ☐ No Justification?

Cascadeless: ☐ Yes ☐ No Justification?

Exercise 2: Show that the following schedule is conflict serializable and give the timestamp-ordering, serializable schedule (i.e., assign timestamps to T_1, T_2 and T_3 so that the schedule is serializable).

T_1 [TS=]	T_2 [TS=]	T_3 [TS=]
		read(Y)
		read(Z)
read(X)		
write(X)		
		write(Y)
		write(Z)
	read(Z)	
read(Y)		
write(Y)		
	read(Y)	
	write(Y)	
	read(X)	
	write(X)	



Precedence graph

Do not upload this exercise sheet to Canvas.

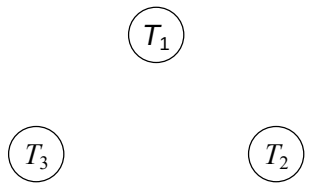
Name: _____ **Student#:** _____ **Date:** _____

COMP 3311: Database Management Systems

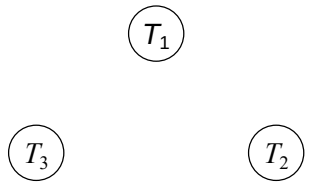
Tutorial 10
Transactions and Concurrency Control

Exercise 3: Is the following schedule conflict serializable? If yes, give the equivalent serial schedule. If no, show, using 2PL, how and where the schedule fails.

T_1	T_2	T_3
	read(Z)	
	read(Y)	
	write(Y)	
		read(Y)
		read(Z)
read(X)		
write(X)		
		write(Y)
		write(Z)
	read(X)	
read(Y)		
write(Y)		
	write(X)	



Precedence graph



Wait-for graph

You must upload this completed exercise sheet to Canvas by **11 p.m. today.**

Name: _____ **Student#:** _____ **Date:** _____

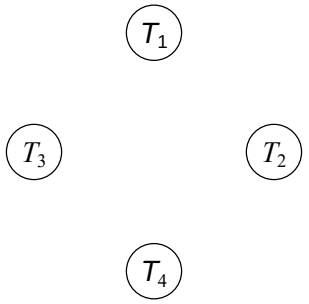
COMP 3311: Database Management Systems

Tutorial 10
Transactions and Concurrency Control

Exercise 4: Consider the following schedule consisting of transactions T_1, T_2, T_3 and T_4 (note: r_1 means T_1 read, w_1 means T_1 write and so on):

Schedule: $r_1(X)$, $w_1(X)$, $r_2(X)$, $r_3(Y)$, $w_3(Y)$, $w_2(X)$, $r_4(Y)$, $w_1(Y)$

a) Show that the schedule is conflict serializable by constructing the precedence graph.



b) What is the equivalent serial schedule?

c) Can the schedule be rewritten so it becomes recoverable, but not cascadeless by adding commit operations in the appropriate locations in the schedule? Explain.

d) Can the schedule be rewritten so it becomes both recoverable, and cascadeless by adding commit operations in the appropriate locations in the schedule? Explain.

You must upload this completed exercise sheet to Canvas by **11 p.m. today.**