

# Quiz5 (Solution)

## Database Systems CE 373/ CS 355 (L1)

Fall 2024

Student Name: \_\_\_\_\_

Student ID: \_\_\_\_\_

### Question 1 (6 points)

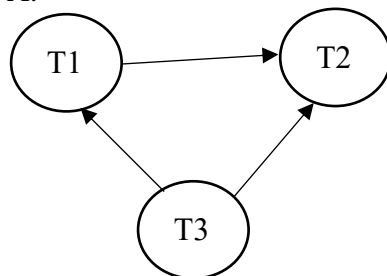
Consider the following three transactions:

T1	T2	T3
Read(A)	Read(A)	Read(T)
Read(B)	Read(S)	Read(A)
A = B + 1	Read(T)	T=A+T
Write(A)	A=S+T	Write(T)
Commit	Write(A)	Commit
	Commit	

- A. Provide a concurrent schedule of T1, T2, and T3 that is conflict-serializable and cascadeless.

T1	T2	T3
Read(A)		
Read(B)		
A=B+1		
		Read(T)
		Read(A)
		T=A+T
		Write(T)
		Commit
Write(A)		
Commit		
	Read(A)	
	Read(S)	
	Read(T)	
	A=S+T	
	Write(A)	
	Commit	

- B. Provide the precedence graph and serializability order of the concurrent schedule you developed in part A.



Serializability Order: [T3, T1, T2]

**Question 2 (2 points)**

Consider the following three transactions: T1, T2, and T3. Add the *lock-X*, *lock-S*, *unlock-X*, and *unlock-S* instructions to T1, T2, and T3 so that they observe the two-phase locking protocol.

T1(Without locks)	T2 (Without Locks)	T3 (Without Locks)	T1 (With Locks)	T2 (With Locks)	T3 (With Locks)
Read(A) Read(B) A = B + 1 Write(A)	Read(A) Read(S) Read(T) A=S+T Write(A)	Read(T) Read(A) T=A+T Write(T)	Lock-X(A) Read(A) Lock-S(B) Read(B) Unlock-S(B) A = B + 1 Write(A) Unlock-X(A)	Lock-X(A) Read(A) Lock-S(S) Read(S) Lock-S(T) Read(T) Unlock-S(S) Unlock-S(T) A=S+T Write(A) Unlock-X(A)	Lock-X(T) Read(T) Lock-S(A) Read(A) Unlock-S(A) T=A+T Write(T) Unlock-X(T)

**Question 3 (2 points)**

Consider two transactions: T1, T2

- a.) Rewriting these transactions by using SQL Views will always improve performance.  
(True / **False**)
- b.) A database system should be able to roll back from failure at any point during the execution of transaction to ensure atomicity. Consider the following transaction schedule. Assume that the system can suffer a failure at any line. Identify the line where a failure will result in an inability to rollback and thus a violation of atomicity property of a transaction. Justify your answer.

Line No.	T1	T2
1	Read(A)	
2	A=A+1	
3	Write(A)	
4		Read(A)
5		Read(B)
6		B=A+1
7		Write(B)
8		Commit
9	Commit	

Answer:

Line 9.

At line 4, T2 read the value written by T1 at line 3. Now, if T1 fails at Line 9, rollback process should also rollback T2. However, T2 has already committed at line 8. This schedule is an example of non-recoverable transaction schedule.