CS5600 Advanced Database

Written Assignment 4

Name: Venkata Lakshmi Sasank Tipparaju

Student ID: 700738838

CS5600 – 13892

Question 1. Transactions

Let schedule S1 has T1, T2 and T3 are transactions

T1	T2	T3
Read(Z)		
Write(Y)		
	Read(X)	
	Write(X)	
	Read(Z)	
		Read(X)
		Write(X)
	Write(Z)	
	Commit	
Read(X)		
Commit		
		Read(Y)
		Commit

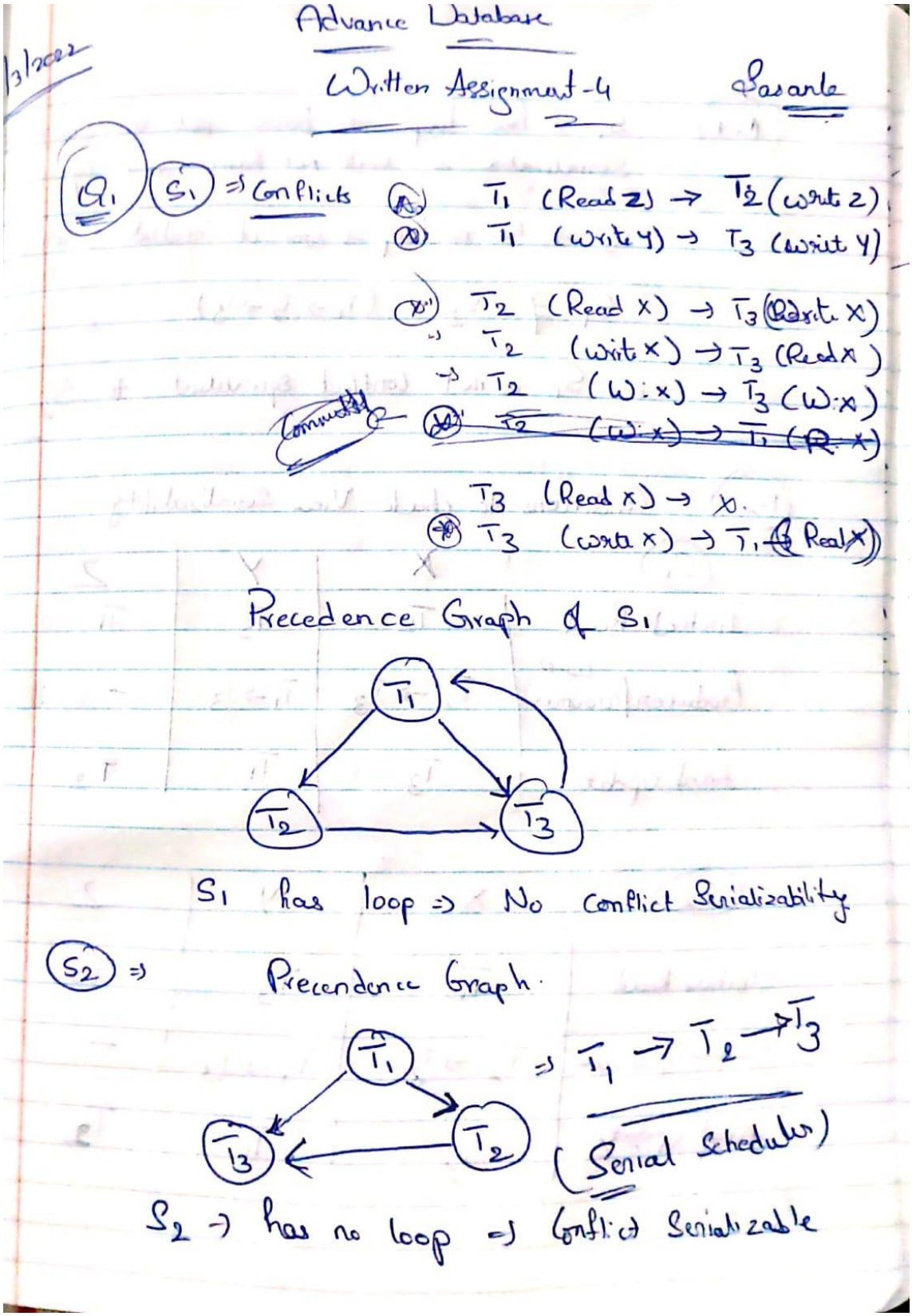
Let schedule S2 has T1, T2 and T3 are transactions

T1	T2	T3
Read(Z)		
Write(Y)		
Read(X)		
Commit		
	Read(X)	
	Write(X)	
	Read(Z)	
	Write(Z)	
	Commit	
		Read(X)
		Write(X)
		Read(Y)
		Commit

1.1) S1 is a conflict equivalent to S2 or not? And why? (3 points)

1.2) S1 is a view equivalent to S2 or not? And why? (3 points)

Answers:


Advance Database
Written Assignment-4
Q1
S1 ⇒ Conflicts
T1 (Read Z) → T2 (write Z)
T1 (write Y) → T3 (write Y)
T2 (Read X) → T3 (write X)
T2 (write X) → T3 (Read X)
T2 (W:X) → T3 (W:X)
T3 (Read X) → X.
T3 (write X) → T1 (Read X)
Precedence Graph of S1
T1
T2
T3
S1 has loop ⇒ No conflict Serializability
S2 ⇒ Precendence Graph.
T1
T3
T2
⇒ T1 → T2 → T3
(Serial Schedule)
S2 → has no loop ⇒ Conflict Serializable

Solution

1-1 S_1 ⇒ has loop ⇒ hence not a conflict Serializable ∴ will not have any equivalent Serial schedule

S_2 ⇒ has no loop ⇒ Conflict Serializable ⇒ Serial schedule

$S_1 \not\equiv S_2$ $(T_1 \to T_2 \to T_3)$

∴ S_1 is not conflict Equivalent to S_2

1-2 Conditions to check View Serializability.

S_1

	X	Y	Z
Initial Read	T_2	T_3	T_1
Producer/(co-n) Consumer	$T_2 \to T_3$	$T_1 \to T_3$	–
final update	T_3	T_1	T_2

S_2

	X	Y	Z
Initial Read	T_1	T_3	T_1
Producer/consumer	$T_2 \to T_3$	$T_1 \to T_3$	–
final update	T_3	T_1	T_2

As S_1 & S_2 conditions are not same at (X) reason

S_1 is not View Equivalent to S_2 $(S_1 \not\equiv S_2)$

Question 2. Locking Protocol

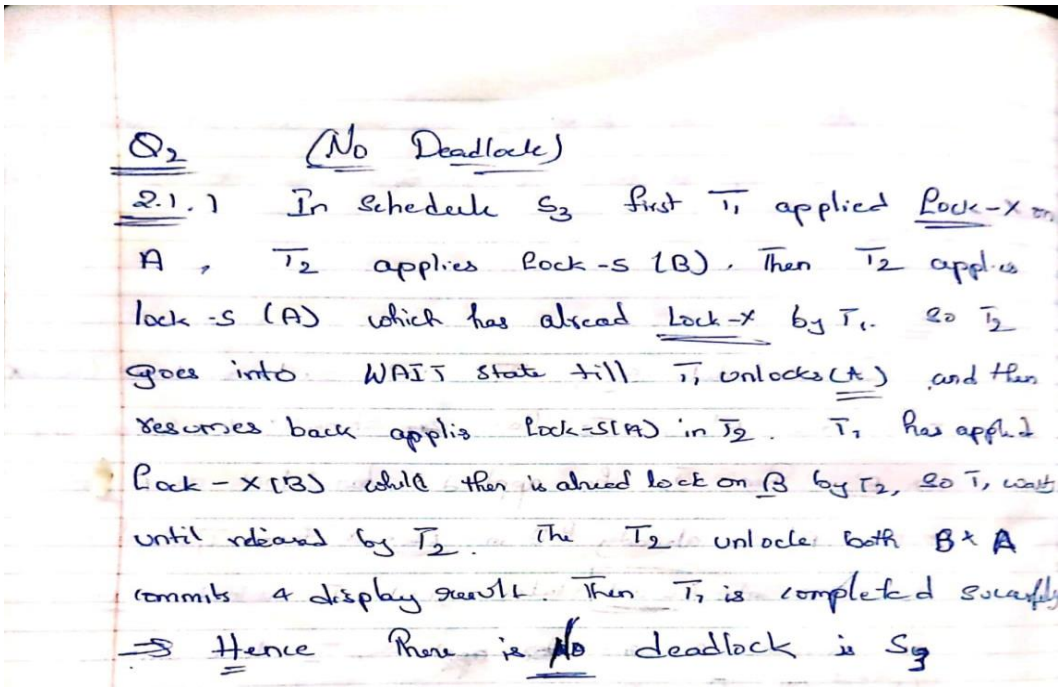
2.1 Let schedule S3 contains transaction T1 and T2.

T1	T2
Lock-X(A)	
Read(A)	
A=A+100	
	Lock-S(B)
	Read(B)
	Lock-S(A)
Write(A)	
Unlock(A)	
Lock-X(B)	
	Read(A)
	Unlock(B)
	Commit
	Unlock(A)
	Display(A+B)
Read(B)	
B=B-100	
Write(B)	
Commit	
Unlock(B)	

2.1.1) The deadlock occurs in schedule S3 or not? And why? (3 points)

2.1.2) How to modify schedule S3 for guarantee about serializability (T1 followed by T2) (4 points)

Answer:



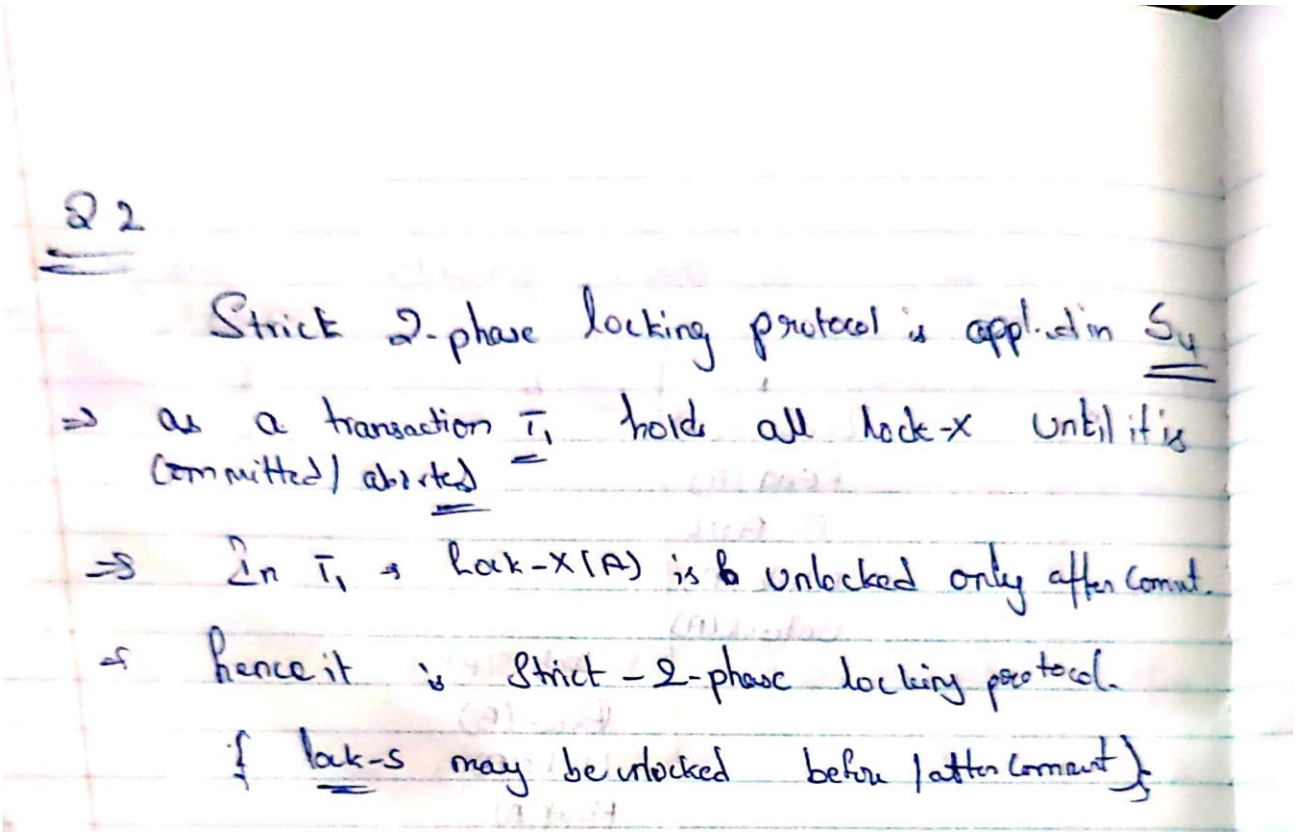
2.1.2 To ensure guarantee serializability for S_3

T_1	T_2
Lock-X(A)	
Read (A)	
A = A+10	
Write (A)	
Unlock(A)	
	Lock-S(B)
	Read (B)
	Lock-S(A)
	Read (A)
	Unlock (B)
	Commit
	Unlock (A)
	Display (A+B)
Lock-X(B)	
Read (B)	
B = B+100	
write(B)	
Commit	
Unlock(B)	

2.2 Let schedule S4 contains transaction T1 and T2.

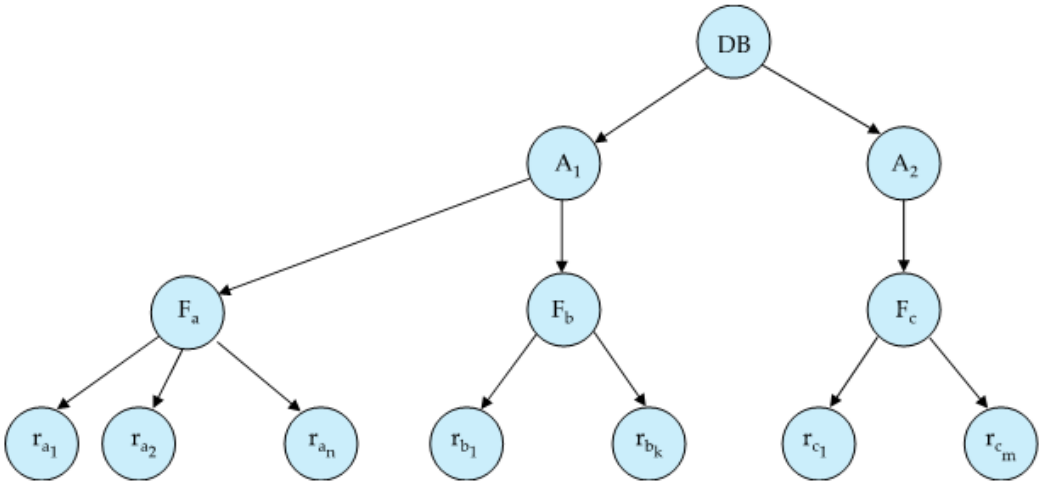
T1	T2
Lock-X(A) Read(A) A=A+100 Write(A)	
	Lock-S(B) Read(B) Lock-S(A)
Lock-S(B) Read(B) Unlock(B) Commit Unlock(A)	
	Read(A) Unlock(A) Unlock(B) Display(A+B) Commit

What kind of 2-Phase Locking Protocol in schedule S4? And why? – **Strict 2-phase Locking – holds all Lock-X until the transaction is committed /aborted – Lock-X(A) in T1 is unlocked after commit.**



2.3 Granularity of locking: The corresponding tree is starting from the top level are

- Database
- Area
- File
- Record.

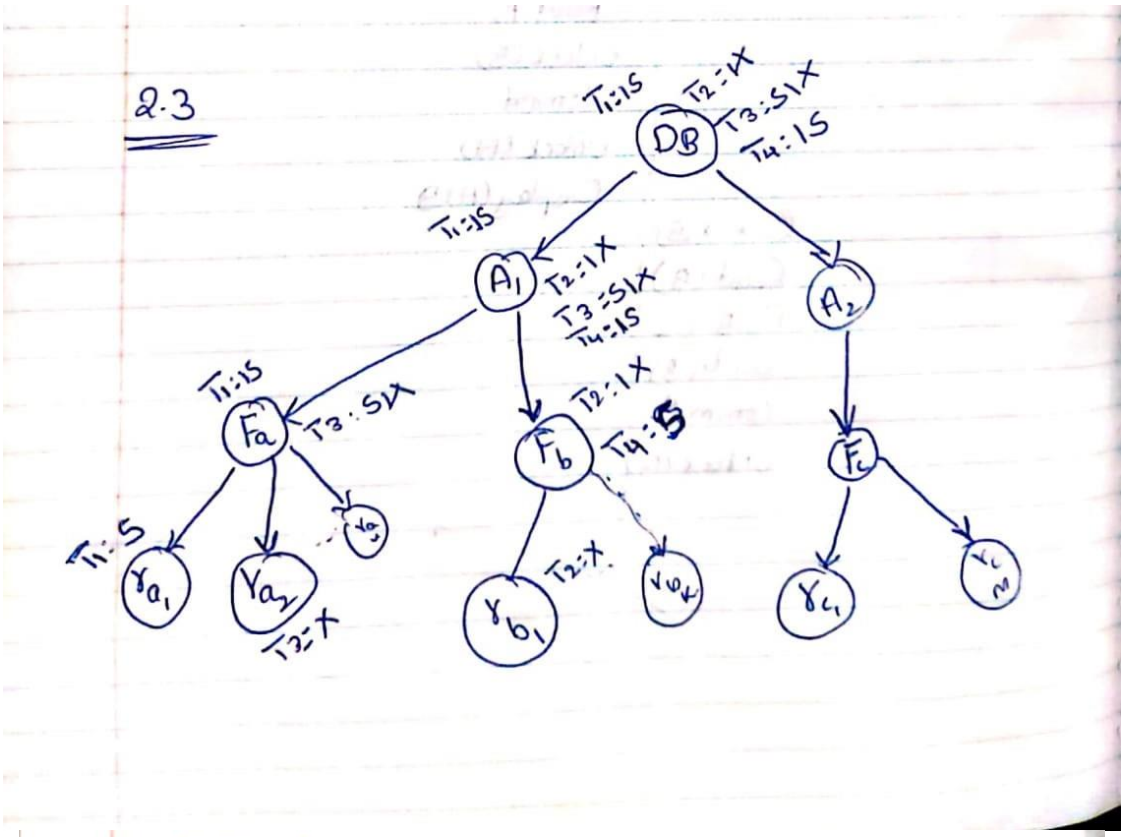


Let T1 read R_{a1}, T2 write R_{b1}, T3 read F_a and write R_{a2}, T4 read F_b.

Complete the following table by filling in the appropriate lock mode. (4 points)

	DB	**A_1**	**F_a**	**F_b**	**R_{a1}**	**R_{a2}**	**R_{b1}**
T1	**IS**	**IS**	**IS**		**S**		
T2	**IX**	**IX**		**IX**			**X**
T3	**SIX**	**SIX**	**SIX**			**X**	
T4	**IS**	**IS**		**S**			

2.3



	DB	A_1	Fa	F_b	R_{a_1}	R_{a_2}	R_{b_1}
T_1	IS	IS	IS		S		
T_2	IX	IX		IX			X
T_3	SIX	SIX	SIX			X	
T_4	IS	IS		S			

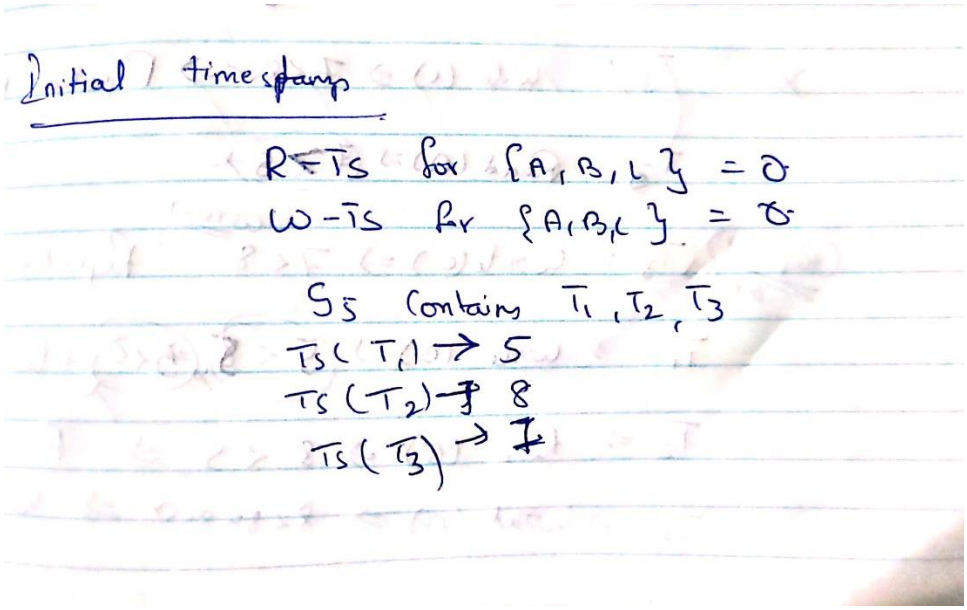
Question 3. Timestamp-Based Protocol (5 points)

Let schedule S5 contains transaction T1, T2 and T3 with initial timestamp 5, 8 and 7 respectively. The initial read timestamp(R-TS) for data items A, B and C = 0. The initial write timestamp(W-TS) for data items A, B and C = 0.

T1	T2	T3
Read(A) Write(A) Read(B) Write(B)	 Read(C) Write(C) Read(A) Read(B) Write(A) Display(A+B)	 Write(C) Read(A) Write(A)

3.1) What are the final R-TS and W-TS of data item A, B, and C? (3 points)

3.2) Which transaction(s) can commit, and which transaction(s) need to rollback or killed? (2 points)



$(T_1, T_2, T_3) = (5, 8, 7)$

	(A)	(B)	(C)
R-Ts	∅ 5 8	∅ 5 8	∅ 8
W-Ts	∅ 5 8	∅ 5	∅ 8

✓ T_1 : Read(A) ⇒ (5 > 0) ⇒ R-TS(A) ⇒ 5

✓ T_1 Write(A) ⇒ (5 < 5) (5 < 0) Else ⇒ W-Ts(A) ⇒ 5

✓ T_2 : Read(C) ⇒ 8 > 0 ⇒ R-Ts(C) = 8

✓ T_1 : Read(B) ⇒ 5 > 0 ⇒ R-TS(B) ⇒ 5

✓ T_2 : Write(C) ⇒ (8 < 8) (8 < 0) Else ⇒ W-Ts(C) ⇒ 8

✓ T_2 Read(A) ⇒ 8 > 5 ⇒ R-Ts(A) ⇒ 8

✗ T_3 : Write(C) ⇒ 7 < 8 ✓ ⇒ Rejected & Rollback T_3

T_3 : Read(A) ⇒ (7 > 5) ✗

(Rejected & killed) T_3 : Write(C) ⇒ 7 < 8 (WTs(C)) ⇒ Rejected & kill T_3

T_1 ⇒ Write(B) ⇒ (5 < 5) (5 < 0) ⇒ W-Ts(C) ⇒ 5

T_2 ⇒ Read(B) ⇒ 8 < 5 ⇒ R-Ts(B) ⇒ 8

T_2 ⇒ Write(A) ⇒ 8 < 5 > 8 < 0 ⇒ W-Ts(A) ⇒ 8

Final

3.1

	A	B	C
R-TS	8	8	8
W-TS.	8	5	8

3.2 T_1 & T_2 transactions can commit.

T_3 transaction is rejected & killed