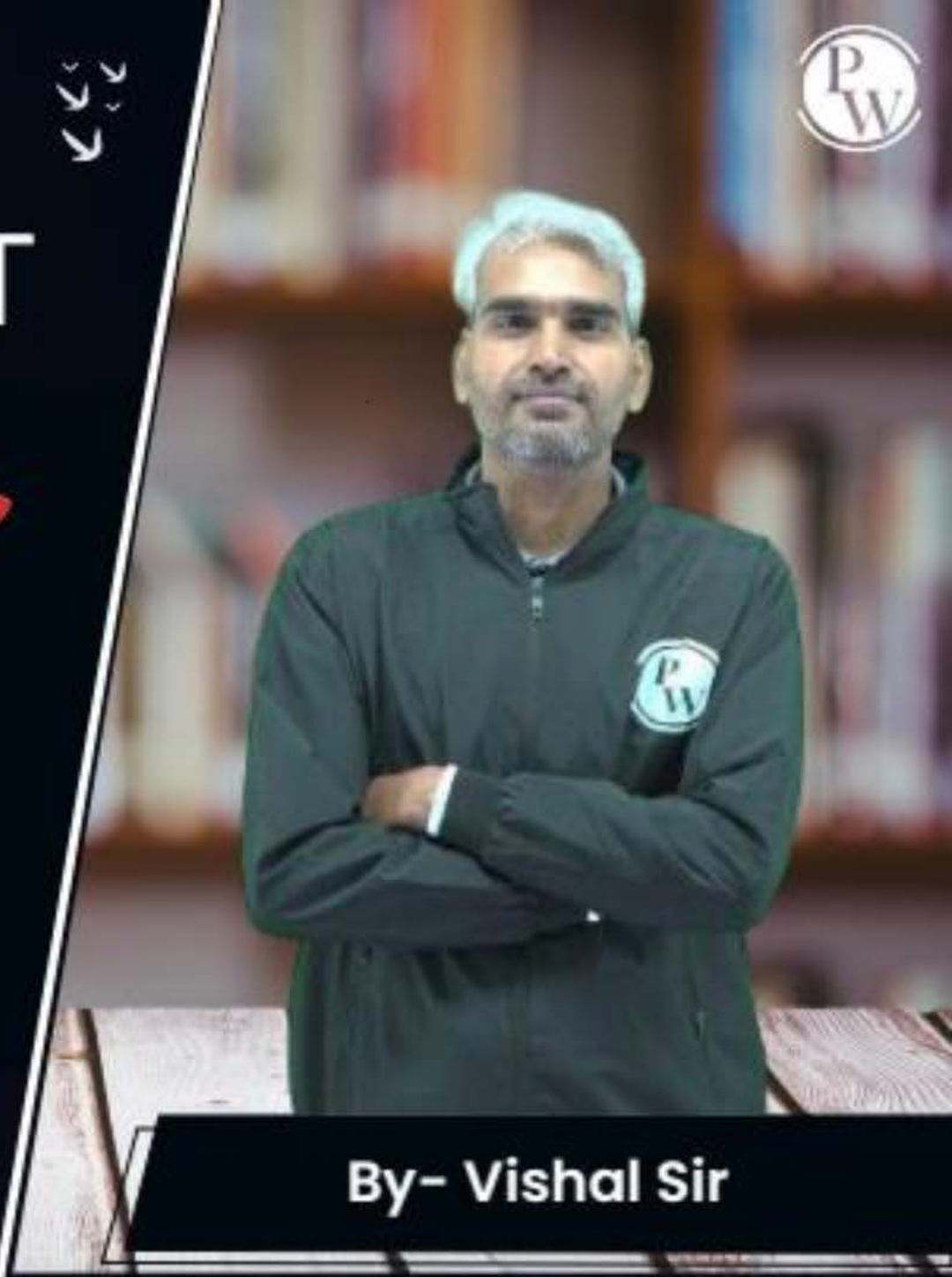
Computer Science & IT

Database Management System

Transaction & concurrency control

Lecture No. 07



By- Vishal Sir

Recap of Previous Lecture

Topic — Conflict serializable schedule

Topic — View serializable schedule

Slide

Topics to be Covered

- Topic: View serializable schedule
- Topic: Concurrency control component
- Topic: Simple use of shared and exclusive locks
- Topic: Two phase lock (2PL)

Slide

View Serializable Schedule

Topic : View serializable schedule

If given schedule is view equivalent to at least one of the serial schedule , then it is called a view serializable schedule.

Topic : View equivalent condition

Consider two schedules S1 and S2

Schedules S1 and S2 are called view equivalent if the following three conditions hold true for them –

- Condition-01 : For each data item X, if transaction Ti reads X from the initial database in schedule S1, then in schedule S2 also, Ti must perform the [redacted] read of X from the initial database.

Thumb Rule

- "Initial readers must be same for all the data items".

Topic : View equivalent condition

Condition-02 :

If transaction Ti reads the value of data item X updated by the transaction Tj in schedule S1, then in schedule S2 also, transaction Ti must read the value of data item X updated by the transaction Tj.

Thumb Rule

- "Write-read sequence must be same.".

Topic : View equivalent condition

Condition-03 :

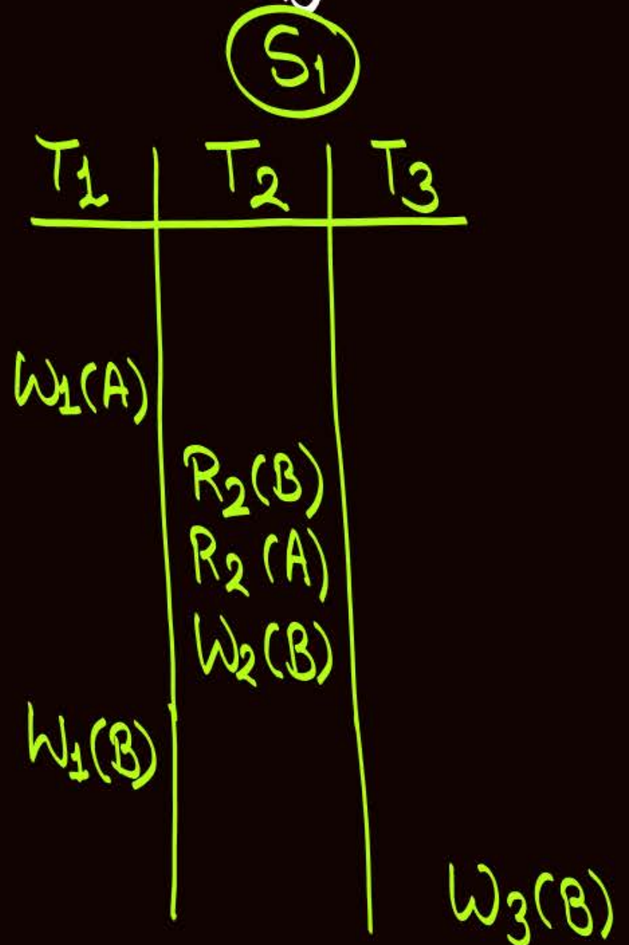
For each data item X, if X is finally updated by transaction Ti in schedule S1, then in schedule S2 also, X must be finally updated by transaction Ti.

Thumb Rule

- "Final writers must be same for all the data items".

- For two schedules to be view equivalent all three Conditions must hold true

① Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"



Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"

S_1

T_1	T_2	T_3
$W_1(A)$		
	$R_2(B)$	
	$R_2(A)$	
	$W_2(B)$	
$W_1(B)$		
		$W_3(B)$

① Identify the Constraints w.r.t. Read from initial DB

Data item	Transaction that reads the dataitem from initial DB	other transactions that writes that dataitem	Constraints w.r.t. view Equivalent Serial schedule
A	No	T_1	No Constraint
B	T_2	T_1, T_3	$T_2 \rightarrow (T_1, T_3) \equiv T_2 \rightarrow T_1$ & $T_2 \rightarrow T_3$

① Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"

S_1

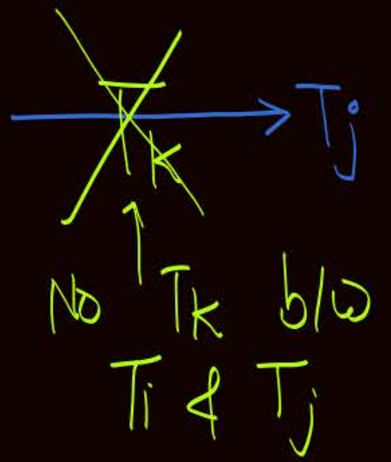
T_1	T_2	T_3
$W_1(A)$		
	$R_2(B)$	
	$R_2(A)$	
	$W_2(B)$	
$W_1(B)$		
		$W_3(B)$

② Identify the Constraints w.r.t. Write-read sequence

Concept: Let transaction T_i updates the dataitem X in the given schedule and transaction T_j reads the value of X updated by transaction T_i, also assume that T_k is some other transaction in the schedule which also update the dataitem 'X', then in a view equivalent serial schedule,

① T_i must Execute before T_j i.e $T_i \rightarrow T_j$

and ② No T_k should be allowed to execute b/w T_i & T_j



① Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"

(S1)

T_1	T_2	T_3
$W_1(A)$		
	$R_2(B)$	
	$R_2(A)$	
	$W_2(B)$	
$W_1(B)$		
		$W_3(B)$

② Identify the Constraints w.r.t. Write-read sequence

Data item	Constrain w.r.t Write-read sequence
A	T_1: writes A & T_2 reads the value update by T_1 ∴ $T_1 \rightarrow T_2$ No other writer of A ∴ No Constraint w.r.t T_K ∴ Final Constraint $T_1 \rightarrow T_2$
B	No Write-Read Sequence ∴ No Constraint

① Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"

S_1

T_1	T_2	T_3
$W_1(A)$		
	$R_2(B)$	
	$R_2(A)$	
	$W_2(B)$	
$W_1(B)$		
		$W_3(B)$

③ Identify the Constraints w.r.t. Final Write

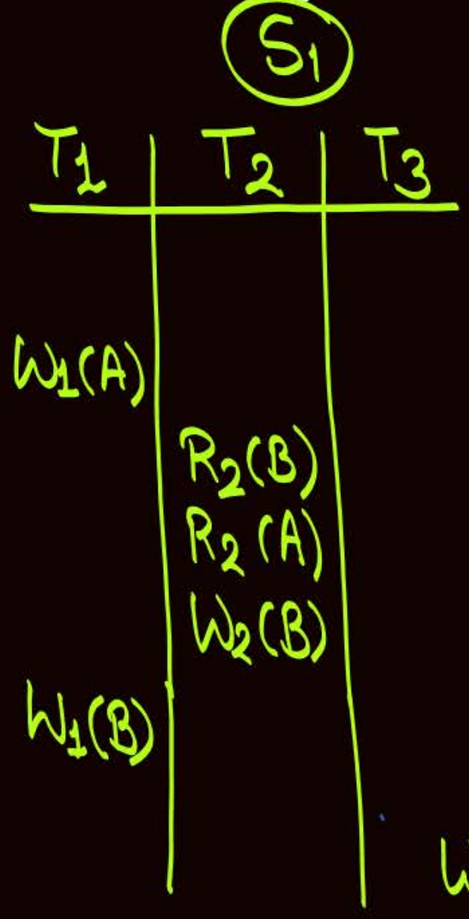
Data item	Transaction with the final update of data item	Other transactions that update the data item	Constraints w.r.t Equivalent serial schedule
A :	T_1	NO	No Constraint
B	T_3	T_1, T_2	$(T_1, T_2) \rightarrow T_3 \equiv$ $T_1 \rightarrow T_3$ & $T_2 \rightarrow T_3$

Note :- Once we have identified the constraints w.r.t. all three conditions construct the dependency graph w.r.t. overall constraints

- If dependency graph is cyclic then schedule is not a view serializable schedule, and hence the schedule is a non-serializable schedule

- If dependency graph is acyclic then schedule is a view serializable schedule, and hence the schedule is a serializable schedule

(1) Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"



→ Set of overall Constrains identified

- w.r.t Read from initial database : $T_2 \rightarrow T_1$, $T_2 \rightarrow T_3$
- w.r.t Write-Read sequence : $T_1 \rightarrow T_2$
- w.r.t Final write : $T_1 \rightarrow T_3$, $T_2 \rightarrow T_3$

In the overall Constraint if we observe the Constraints of the type

$T_i \rightarrow T_j$ & $T_j \rightarrow T_i$,

then dependency graph is definitely cyclic

∴ Schedule is not a View Serializable Schedule

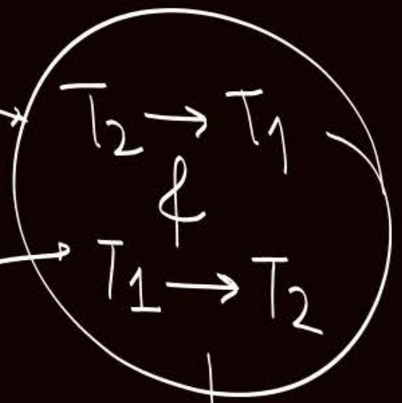
(1) Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"

S_1

T_1	T_2	T_3
$W_1(A)$		
	$R_2(B)$	
	$R_2(A)$	
	$W_2(B)$	
$W_1(B)$		
		$W_3(B)$

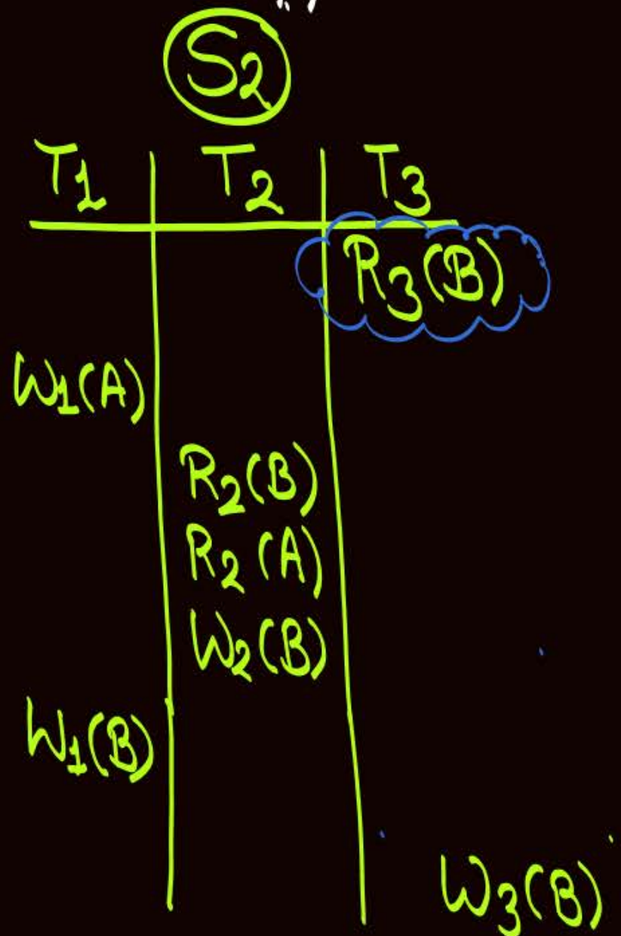
→ Set of overall Constrains identified

- w.r.t. Read from initial database : $T_2 \rightarrow T_1$, $T_2 \rightarrow T_3$
- w.r.t. Write-Read sequence : $T_1 \rightarrow T_2$
- w.r.t. Final write : $T_1 \rightarrow T_3$, $T_2 \rightarrow T_3$



∴ Cyclic, Hence not a V.S.S. & Hence not a Serializable Schedule

(2) Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"



(2) Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"

S_2

T_1	T_2	T_3
		$R_3(B)$
$W_1(A)$		
	$R_2(B)$	
	$R_2(A)$	
	$W_2(B)$	
$W_1(B)$		
		$W_3(B)$

(1) Identify the Constraints w.r.t. Read from initial DB

Data item	Transaction that reads the dataitem from initial DB	other transactions that Writes the dataitem	Constraints w.r.t. view Equivalent Serial schedule
A	No	—	No Constraint
B	T_2	T_1, T_3	$T_2 \rightarrow (T_1, T_3) \equiv T_2 \rightarrow T_1$, $T_2 \rightarrow T_3$
B	T_3	T_1, T_2	$T_3 \rightarrow (T_1, T_2) \equiv T_3 \rightarrow T_1$, $T_3 \rightarrow T_2$

$T_2 \rightarrow T_3$ & $T_3 \rightarrow T_2$ ⇒ Cycle ⇒ Not a V.S.S.

② Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"

(S2)

T_1	T_2	T_3
		$R_3(B)$
$W_1(A)$		
	$R_2(B)$	
	$R_2(A)$	
	$W_2(B)$	
$W_1(B)$		
		$W_3(B)$

① Identify the Constraints w.r.t. Read from initial DB

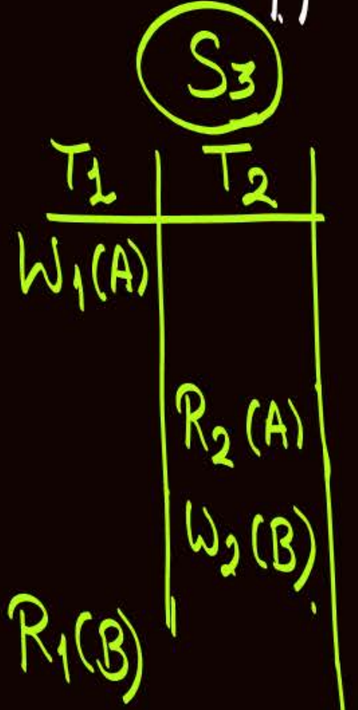
Data item	Transaction that reads the dataitem from initial DB	other transactions that writes the dataitem	Constraints w.r.t. view Equivalent Serial schedule

on Previous Slide

② Identify the Constraints w.r.t. Write-Read Seq (Same as Previous question)

③ Identify the Constraints w.r.t. Final-Write (Same as Previous question)

③ Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"



① Constraints w.r.t. read from initial DB:
A: No Constraint ✓
B: No Constraint

② Constraints w.r.t. write read seq ✓
A: $T_1 \xrightarrow[(No\ T_K)]{} T_2$
B: $T_2 \xrightarrow[(No\ T_K)]{} T_1$

③ Constraints w.r.t. final write
A: No Constraint
B: No Contraint

Overall Constraints

$T_1 \rightarrow T_2$ & $T_2 \rightarrow T_1$ ∴ Cyclic

Hence, Not a V.S.S.

Hence not a Serializable Schedule

④ Q:- Check whether the following schedule is a view serializable schedule or not, if view serializable schedule then identify all "view equivalent serial schedules"

(S4)

T_1	T_2	T_3
	R(B)	
	R(A)	
W(A)		
		W(A)
	W(A)	

① Identify the constraints w.r.t. Read from initial DB

A: $T_2 \rightarrow (T_1, T_3) \equiv T_2 \rightarrow T_1$, $T_2 \rightarrow T_3$

B: No constraint

② Identify the constraints w.r.t. Write-Read Seq

A: No constraint

B: No constraint

③ Identify the constraints w.r.t. Final-Write

A: $(T_1, T_2) \rightarrow T_3 \equiv T_1 \rightarrow T_3$, $T_2 \rightarrow T_3$

B: No Constraint

Overall Constraints

$T_2 \rightarrow T_1$

$T_2 \rightarrow T_3$

$T_1 \rightarrow T_3$

⇓

Dependency graph

Acyclic: Schedule S_4 is a V.S.S.

⇓

View Equivalent serial schedule is $T_2 \rightarrow T_1 \rightarrow T_3$

#Q. Check whether the schedule is view serializable schedule or not?

If view serializable schedule then identify all view equivalent serial schedules.

T1	T2	T3	T4
R(A)			
	R(A)		
		R(A)	
			R(A)
W(B)			
	W(B)		
		W(B)	
			W(B)

① w.r.t. Read from initial DB
A: No Constraint
B: No Constraint

② w.r.t. Write-Read Seq
A: No Constraint
B: No Constraint

③ w.r.t. Final Update
A: $(T_1, T_2, T_3) \rightarrow T_4$
B: No Constraint

Overall Constraints
$T_1 \rightarrow T_4$
$T_2 \rightarrow T_4$
$T_3 \rightarrow T_4$

Dependency graph

Acyclic, ∴ V.S.S.

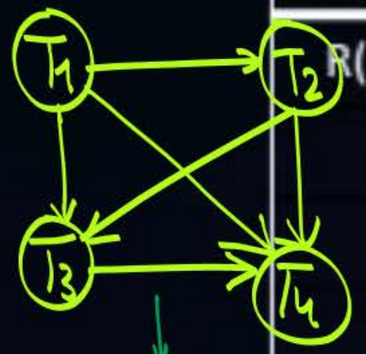
3! = 6 View equivalent Serial schedules

#Q. Check whether the schedule is view serializable schedule or not?

If view serializable schedule then identify all view equivalent serial schedules, that are not conflict equivalent to given schedule.

T1	T2	T3	T4
R(A)			
	R(A)		
		R(A)	
			R(A)
W(B)			
	W(B)		
		W(B)	
			W(B)

Precedence graph



Topological order

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4$

∴ given schedule is Conflict Equivalent to serial schedule $T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4$

→ Schedule is view equivalent to '6' serial schedules

$T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4$

$T_2 \rightarrow T_1 \rightarrow T_3 \rightarrow T_4$

$T_2 \rightarrow T_3 \rightarrow T_1 \rightarrow T_4$

$T_1 \rightarrow T_3 \rightarrow T_2 \rightarrow T_4$

$T_3 \rightarrow T_1 \rightarrow T_2 \rightarrow T_4$

$T_3 \rightarrow T_2 \rightarrow T_1 \rightarrow T_4$

These are the '5' serial schedule that are view equivalent to given schedule, but not conflict equivalent to given schedule.

#Q. Check whether the schedule is view serializable schedule or not?

If view serializable schedule then identify all view equivalent serial schedules.

T1	T2	T3
R(A)		
	R(B)	
	W(B)	
		R(A)
		W(A)
R(B)		
		R(B)
	R(A)	
	W(A)	

Overall Constraint

① A: $T_1 \rightarrow (T_2, T_3)$ & $T_3 \rightarrow T_2$

B: No Constraint

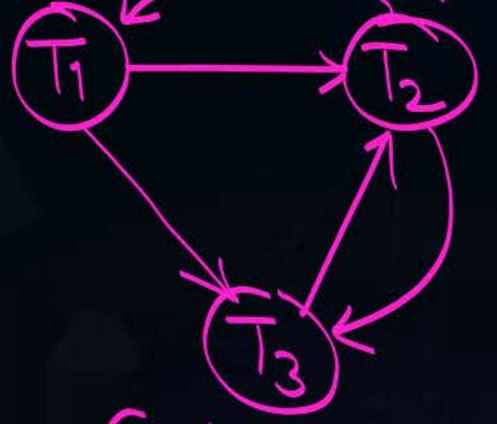
② A: $T_3 \rightarrow T_2$

B: $T_2 \rightarrow T_1$ & $T_2 \rightarrow T_3$

③ A: $T_3 \rightarrow T_2$

B: No Constraint

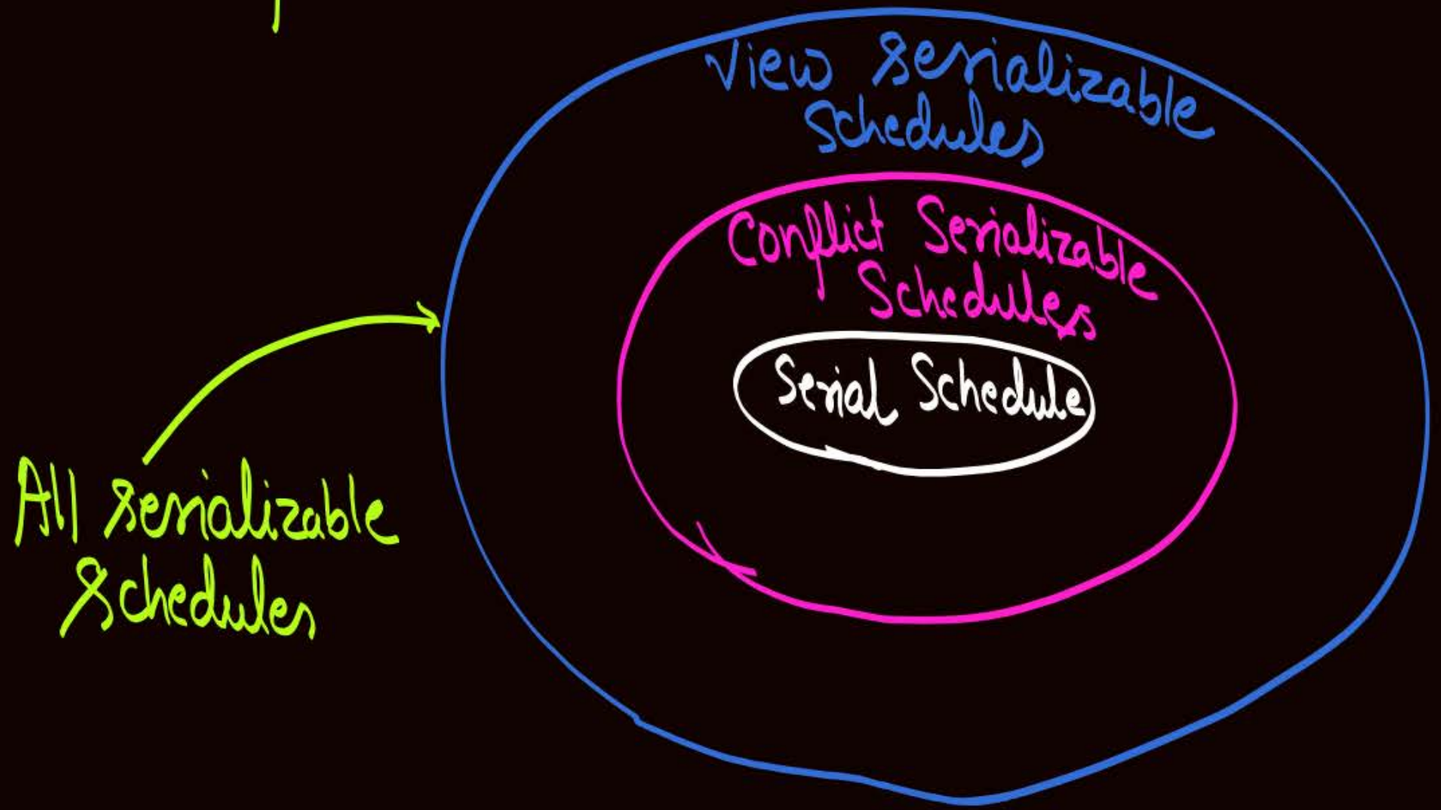
Dependency graph



Cyclic

∴ Not a V.S.S.

Note: Every Conflict serializable schedule is a View serializable schedule, but Every View serializable schedule need not be Conflict serializable schedule



Q. Check whether the schedule is a V.S.S. or not

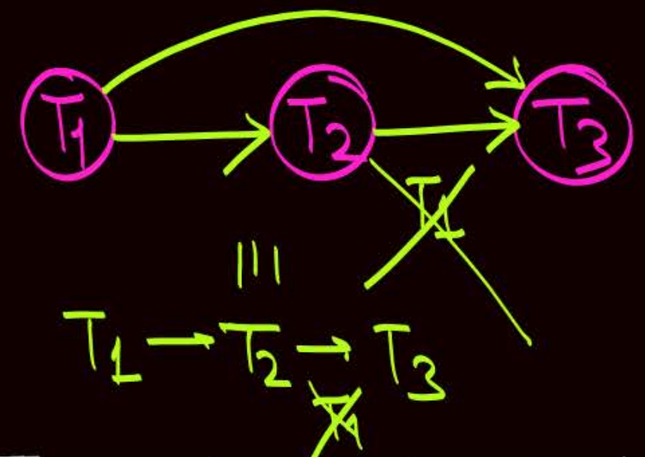
T_1	T_2	T_3
$W_1(A)$		
	$W_2(A)$	
		$R_3(A)$
$R_1(B)$		
		$W_3(B)$

① A:
B: $T_1 \rightarrow T_3$

② A: $T_2 \longrightarrow T_3$ ($T_k = T_1$)
B:

{ T_1 must not execute b/w T_2 & T_3 }

③ A: $T_1 \longrightarrow T_2$
B:



Concurrency Control Component

Topic : Concurrency Control Protocols

- Concurrency Control Protocols are responsible for avoiding the execution of non-serializable schedules.

- Concurrency Control Protocols are suggested to ensure serializability

* If schedule is allowed to execute using a valid Concurrency Control Protocol, then schedule is a Serializable Schedule
Converse of the statement need not be true

Topic : Concurrency Control Protocols

- There are two types of concurrency control protocols.

1. Lock based Protocols

   Two Phase Locking Protocol (2PL)

   Versions of 2PL
   - (i) Basic 2PL
   - (ii) Strict 2PL
   - (iii) Conservative 2PL
   - (iv) Rigorous 2PL

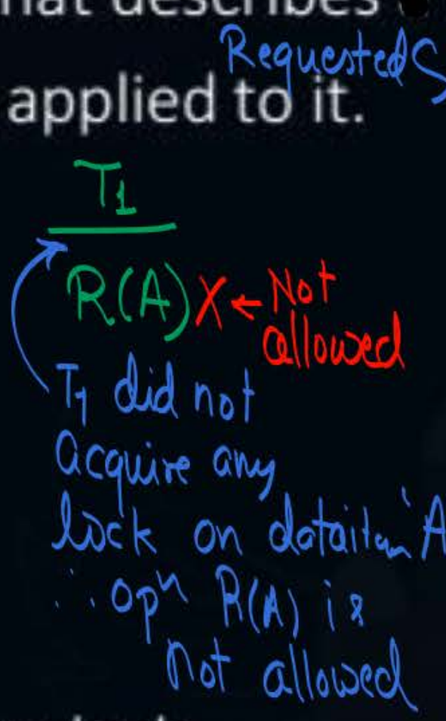
2. Time-stamp based Protocols
   - (i) Basic Time stamp ordering Protocol.
   - (ii) Thomas write Time-stamp ordering Protocol

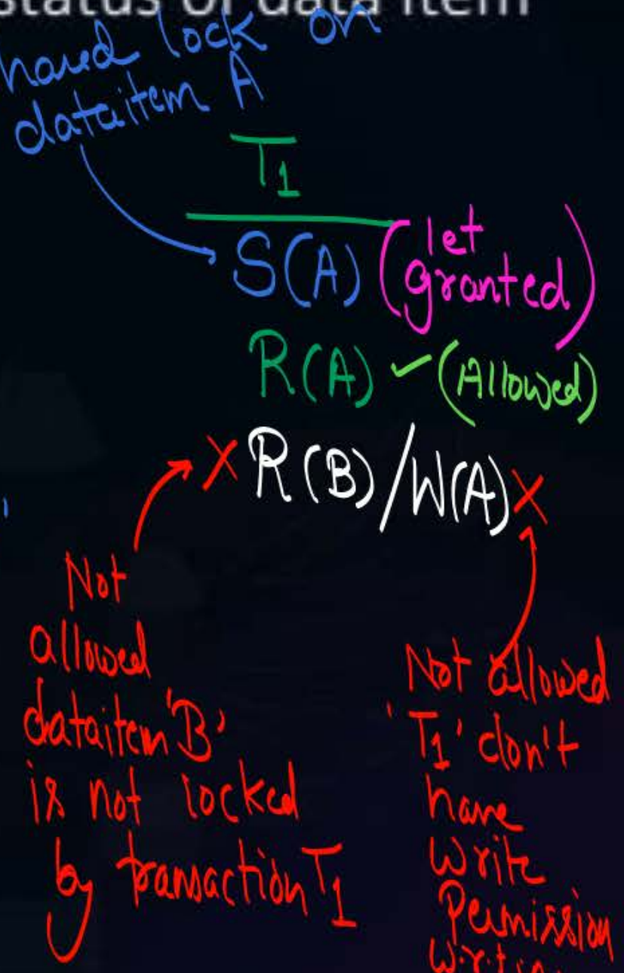
Topic : Lock Based Protocols

- Lock is a variable associated with a data item that describes status of data item with respect to possible operation that can be applied to it.

- There are two types of locks used,
  1. Shared LOCK [S]: It is read only lock.
  2. Exclusive LOCK [X]: It is read as well as write lock.

If transaction wants to perform only read opn on a dataitem, then transaction will request for a shared lock on that dataitem





Topic : Lock Based Protocols

Lock is a variable associated with a data item that describes a status of data item with respect to possible operation that can be applied to it.

- There are two types of locks used,
  1. Shared LOCK [S]: It is read only lock.
  2. Exclusive LOCK [X]: It is read as well as write lock.

T
X W(A)
Not allowed

Requested for Exclusive lock on dataitem 'A'

T
X(A) (let granted)
W(A) ✓(Allowed)
S(B) (granted)
R(B) ✓(Allowed)
R(A) ✓(Allowed)
X R(C)/W(B) X
Not allowed

{Transaction can request for an Exclusive lock on dataitem A only if transaction wants to perform at least one write oph on dataitem 'A'}

Topic : Lock Compatibility Table

Locks held by Transaction Ti

data item 'A'	Shared lock 'S'	Exclusive lock 'X'
Shared lock 'S'	(✓) Allowed	Not allowed
Exclusive lock 'X'	Not allowed	Not allowed

Locks requested by transaction Tj

Note :- To ensure serializability a non-serializable schedule must not be allowed to execute.

Topic : Simple use of shared and exclusive locks

↳ i.e, No restriction { i.e, dataitems can be locked/unlocked in any order at any time }

Consider the following Schedule.

'S'

T_1	T_2
X(A) ✓	
$R_1(A)$ ✓	
$W_1(A)$ ✓	
Unlock 'A' → U(A)	
	S(A) ✓
	$R_1(A)$ ✓
	S(B) ✓
	$R_2(B)$ ✓
	U(A)
	U(B)
X(B) ✓	
$R_1(B)$ ✓	
$W_1(B)$ ✓	
U(B)	

Given schedule is a non-serializable schedule

↓

But it is allowed to Execute using Simple use of Shared and Exclusive locks

A non-serializable schedule may be allowed to execute using Simple use of Shared & Exclusive locks

i.e. Simple use of Shared & Exclusive lock does not Ensure Serializability.

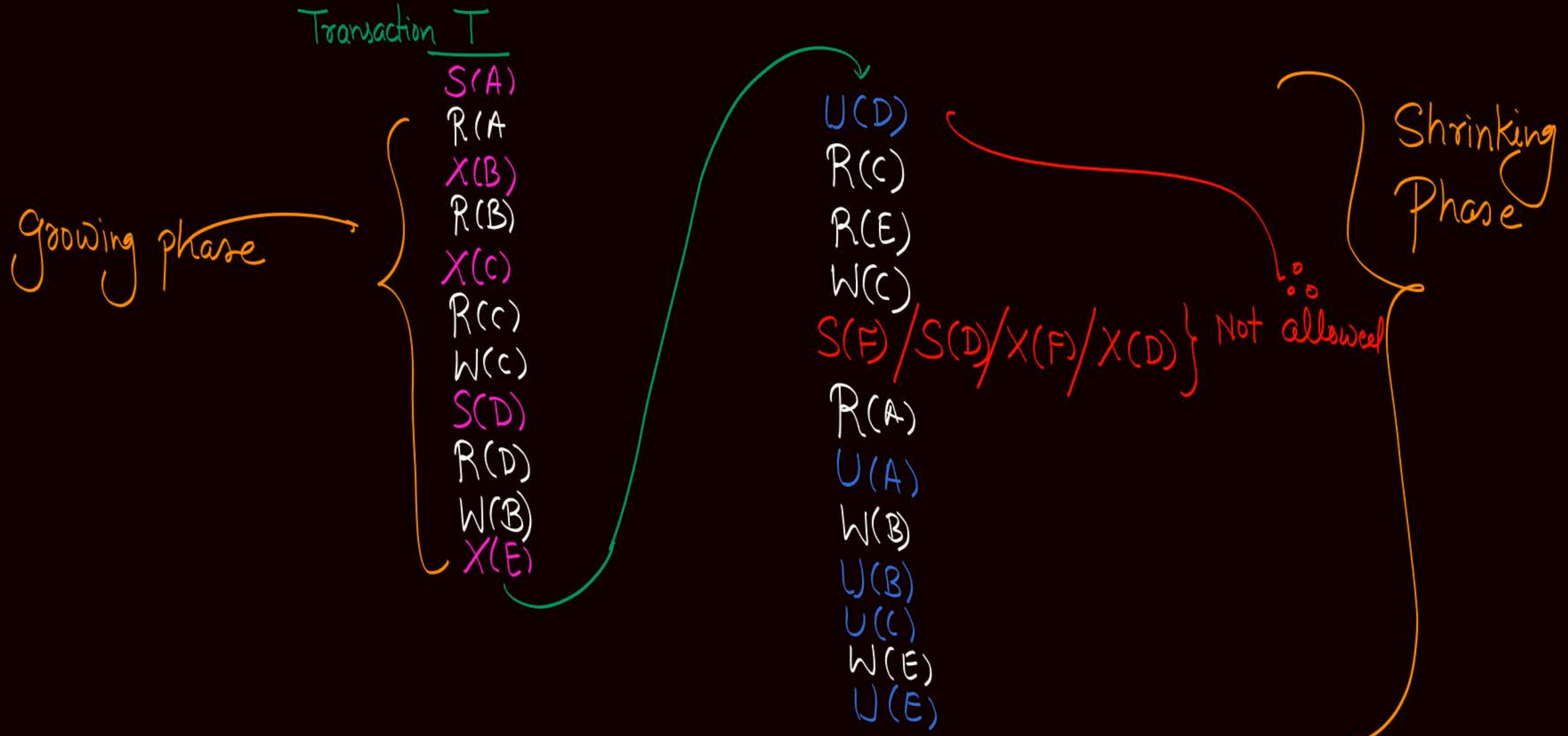
Topic : Basic Two Phase Locking Protocol

There are two phases in Two phase locking protocol

1. Growing phase / Lock acquiring phase
2. Shrinking phase / Lock releasing phase

In Two Phase Locking Protocol, transaction T is allowed to request for a lock only (on any dataitem) if transaction T has not performed any unlock operation.

→ The moment transaction perform any unlock operation it can request for a lock on any dataitem


Transaction T
S(A)
R(A
X(B)
R(B)
X(C)
R(C)
W(C)
S(D)
R(D)
W(B)
X(E)
Growing phase
U(D)
R(C)
R(E)
W(C)
S(F)/S(D)/X(F)/X(D)} Not allowed
R(A)
U(A)
W(B)
U(B)
U(C)
W(E)
U(E)
Shrinking Phase

H.W. Q: Check whether the schedule is allowed to execute using 2PL or not?

S1

T_1	T_2
$R_1(A)$	
$W_1(A)$	
	$R_2(A)$
	$R_2(B)$
$R_1(B)$	
$W_1(B)$	

2 mins Summary

- Topic: View serializable schedule
- Topic: Concurrency control component
- Topic: Simple use of shared and exclusive locks
- Topic: Two phase lock (2PL)

Slide

THANK - YOU

Slide