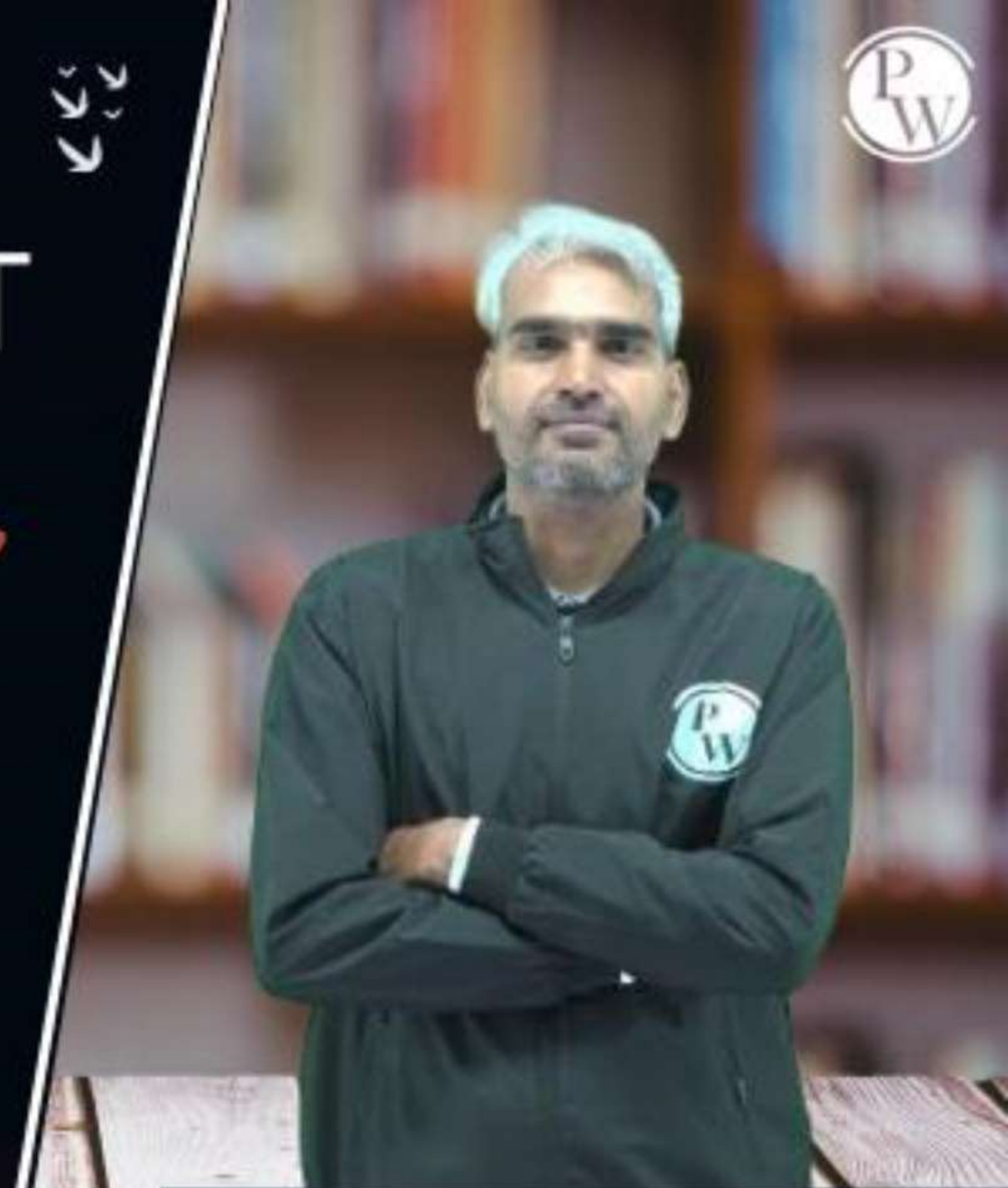
Computer Science & IT

Database Management System

Transaction
&
Concurrency control

Lecture No. 08



By- Vishal Sir

Recap of Previous Lecture

- ✓ **Topic** Precedence graph
- ✓ **Topic** Topological order
- ✓ **Topic** Practice questions on Conflict serializable schedule
- ✓ **Topic** View serializable schedule
- ✓ **Topic** View equivalence condition

Slide

Topics to be Covered

- ✓ Topic — View equivalence condition
- ✓ Topic — Practice questions on View serializable schedule
- Topic — Concurrency control protocols
- Topic — Simple use of shared and exclusive locks
- Topic — Basic Two phase lock (2PL)

Slide

Q:- Check whether the given schedule is a view serializable schedule or not? If view serializable schedule then identify all view equivalent serial schedules.

S

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 database :-

Data item	Transaction Reading the dataitem from initial database	Other writers (transactions) which are updating the data item	Constraint
A	No	T_1	No Constraint
B	T_2	(T_1, T_3)	$T_2 \rightarrow (T_1, T_3)$ i.e. $T_2 \rightarrow T_1$ $T_2 \rightarrow T_3$

Q:- Check whether the given schedule is a view serializable schedule or not? If view serializable schedule then identify all view equivalent serial schedules.

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

Let Ti updates the data item 'A', and transaction Tj reads the value update by transaction Ti, and let Tk is some other transaction which also updates dataitem A.

then in an Equivalent serial schedule transaction Tj must execute before transaction Tj } i.e. ⇒ Ti ⟶ Tj

(and)

and no other write of dataitem A should be allowed to execute b/w Ti & Tj ⇒ } i.e. "Tk" must not execute b/w Ti & Tj

Q:- Check whether the given schedule is a view serializable schedule or not? If view serializable schedule then identify all view equivalent serial schedules.

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

Dataitem	Transactions w.r.t Write-Read seq.	Constraint w.r.t. write-Read seq.
A	T_1: Updates A & T_2: Read the value updated by T_1. $T_k = \emptyset$ {as No other write of A}	$T_1 \longrightarrow T_2$ ↑ {No constraint w.r.t. intermediate writer as $T_k = \emptyset$}
B	No Write-Read seq w.r.t. B	No Constraint

Q:- Check whether the given schedule is a view serializable schedule or not? If view serializable schedule then identify all view equivalent serial schedules.

T_1	T_2	T_3
$W_1(A)$		
	$R_2(B)$	
	$R_2(A)$	
	$W_2(B)$	
$W_1(B)$		
		$\underline{W_3(B)}$

③ Identify the Constraints w.r.t. final update

Dataitem	Final writer of data item	Other writers of dataitem	Constraint
A	T_1	No other transaction	No Constraint
B	T_3	(T_1, T_2)	$(T_1, T_2) \rightarrow T_3$ i.e. $\left\{ \begin{array}{l} T_1 \rightarrow T_3 \\ T_2 \rightarrow T_3 \end{array} \right\}$

Note:- Once we have obtained Constraints w.r.t. all three Conditions, then Construct the dependency graph w.r.t overall Constraints.

If dependeny graph is cyclic, then schedule is not a View serializable schedule, and hence the corresponding schedule is not a serializable schedule

If dependeny graph is acyclic, then schedule is a View serializable schedule, and hence schedule is a serializable schedule, and view Equivalent serial schedules can be given by topological order of dependeny graph

W.r.t. above example,

① Constraints w.r.t. read from initial database, $T_2 \rightarrow T_1$
$T_2 \rightarrow T_3$

② Constraints w.r.t. write-read sequence, $T_1 \rightarrow T_2$

③ Constraints w.r.t final update, $T_1 \rightarrow T_3$
$T_2 \rightarrow T_3$

Set of overall Constraints

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

in the overall Constraint if we have $T_i \rightarrow T_j$ & $T_j \rightarrow T_i$, then we can directly say that schedule is not a View Serializable schedule

Otherwise construct the complete dependency graph and check if it is acyclic or cyclic

W.r.t. above example,

① Constraints w.r.t. read from initial database, $T_2 \rightarrow T_1$
$T_2 \rightarrow T_3$

② Constraints w.r.t. write-read sequence, $T_1 \rightarrow T_2$

③ Constraints w.r.t final update , $T_1 \rightarrow T_3$
$T_2 \rightarrow T_3$

Set of overall Constraints

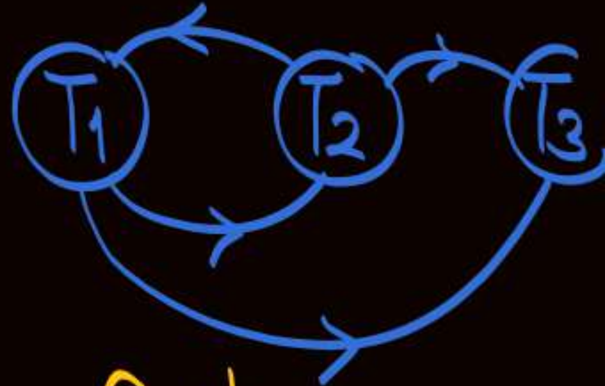
$T_2 \rightarrow T_1$

$T_2 \rightarrow T_3$

$T_1 \rightarrow T_2$

$T_1 \rightarrow T_3$

$\Rightarrow$ Dependency graph



Cyclic

∴ Given Schedule is not a V.S.S.

Q:- Check whether the given schedule is a view serializable schedule or not? If view serializable schedule then identify all view equivalent serial schedules.

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 database :-

Dataitem	Transaction Reading the dataitem from initial DB	Other writers of dataitem	Constraint
A	No transaction		No Constraint
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	(T_1, T_3)	$T_2 \rightarrow (T_1, T_3) \equiv T_2 \rightarrow T_1$ $T_2 \rightarrow T_3$

Cycle ⇒

⇒ is Not a View Serializable Schedule

② Constraints w.r.t. Write-Read seq.
Same as above question

③ Constraints w.r.t. final update
Same as above question

Q:- Check whether the given schedule is a view serializable schedule or not? If view serializable schedule then identify all view equivalent serial schedules.

T_1	T_2
$W_1(A)$	
	$R_2(A)$
	$W_2(B)$
$R_1(B)$	

① w.r.t read from initial database

A: No Constraint

B: No Constraint

② W.r.t. write-read seq.

A: $T_1 \rightarrow T_2$

No other write of A so no constraint w.r.t that

B: $T_2 \rightarrow T_1$

No other write of B so no constraint w.r.t that

Cyclic
∴ Not a View serializable schedule

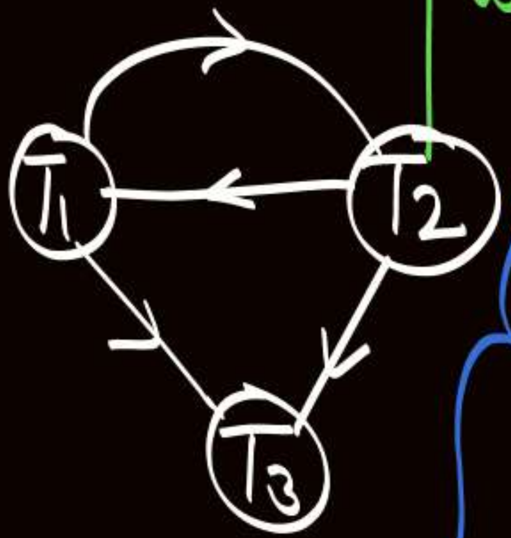
③ W.r.t. Final update

A: No Constraint

B: No Constraint

Q:- Check whether the given schedule is a view serializable schedule or not? If view serializable schedule then identify all view equivalent serial schedules.

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



Precedence graph is cyclic

∴ Not a C.S.S.

① w.r.t. read from initial DB

A: $T_2 \rightarrow T_1$

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

② write-read seq:

A: No WR seq

B: No wr seq

③ w.r.t. Final update:

A: No Constraint

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

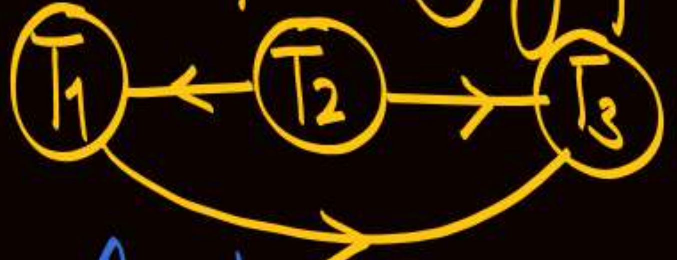
Overall constraints

$T_2 \rightarrow T_1$, $T_2 \rightarrow T_3$

$T_1 \rightarrow T_3$

⇓

Dependency graph

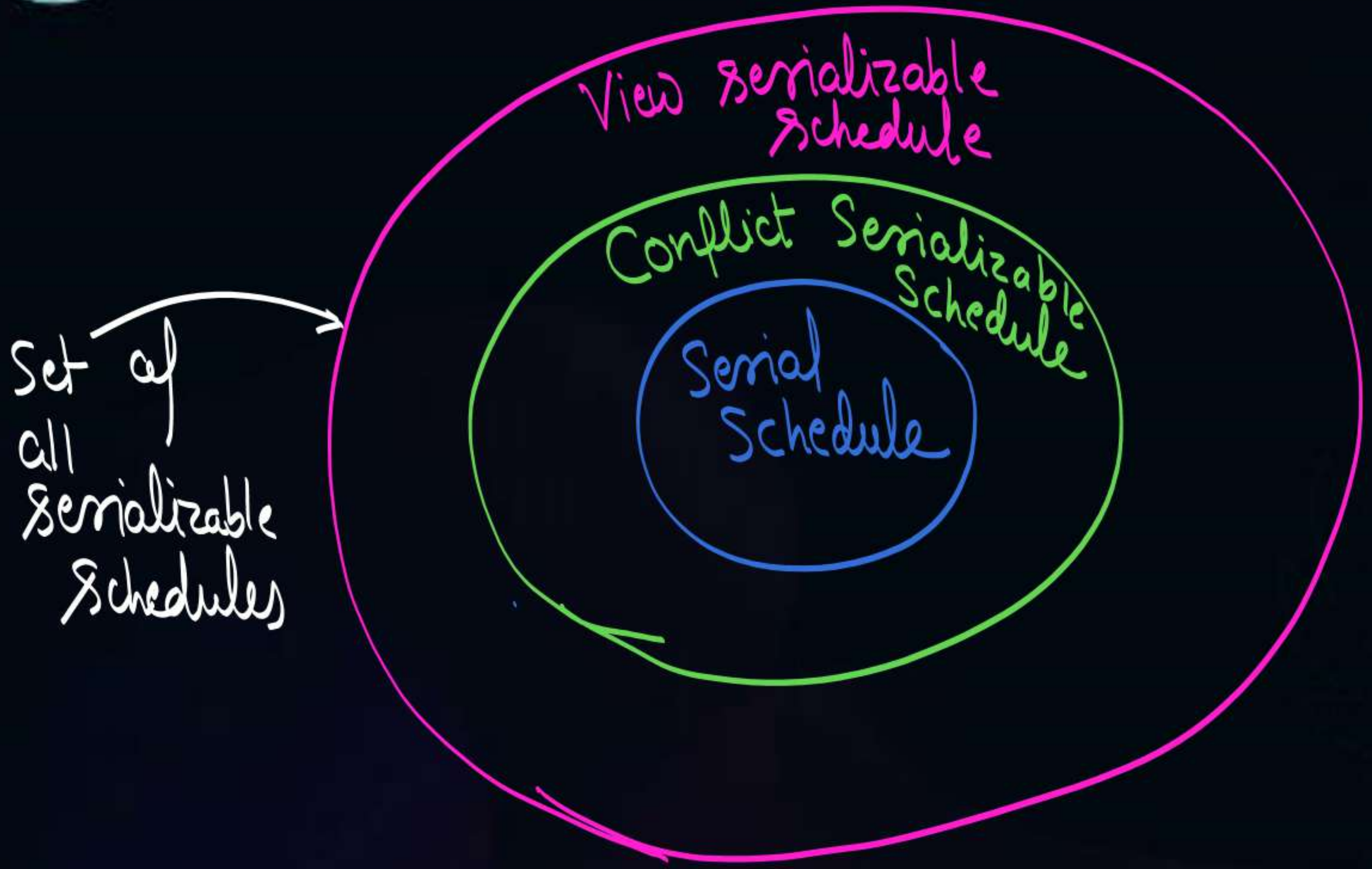


Acyclic,

∴ View serializable schedule

& View equivalent serial schedule is $T_2 \rightarrow T_1 \rightarrow T_3$

Topic : Conflict serializable & View serializable schedule


View serializable schedule
Conflict Serializable Schedule
Serial Schedule
Set of all serializable schedules

- * Every C.S.S. is also a V.S.S.
- * Every V.S.S. need not be a C.S.S.
- * A schedule which is not V.S.S. can never be a C.S.S.

H.W.

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

(1) If view serializable schedule then identify all view equivalent serial schedules.

(2) Find the number of serial schedule that view equivalent to given schedule, but 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)

H.W.

#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)	

Concurrency control Protocols :-

└→ Concurrency Control Component is responsible for avoiding the execution of non-serializable schedule

Topic : Concurrency Control Protocols

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

Topic : Concurrency Control Protocols

- There are two types of concurrency control protocols.

1. Lock based Protocols

   Two Phase Locking Protocol (2PL)

   (i) Basic 2PL

   (ii) Strick 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 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.

{At least one write opn must be there}

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.

   Transaction will request for a shared lock on a dataitem if transaction wants to perform only read op^n on that dataitem

2. Exclusive LOCK [X]: It is read as well as write lock.

T

~~R(A)~~

not allowed because transaction 'T' has not acquired any lock on data item 'A'

Requested for a shared lock on dataitem A

T

S(A) (granted)

R(A)

~~W(A)~~

~~R(B)~~

Not allowed because dataitem 'A' is locked using shared lock

Not allowed because 'B' is not locked by T

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.

Transaction can request for an Exclusive lock on a dataitem if transaction wants to perform at least one write opn On that dataitem

T

~~W(A)~~

not allowed because A is not locked by T.

Requested for an Exclusive lock on dataitem 'A'

T

X(A) (let granted)

R(A) : Allowed

W(A) : Allowed

S(B) : (let granted)

R(B) : Allowed

W(A) : Allowed

~~R(C)~~ Not allowed

~~W(B)~~: Not allowed

U(B) : unlocked

R(A): Allowed

~~R(B)~~ : Not allowed

Topic : Lock Compatibility Table

Lock requested by transaction T_j

Dataitem let 'A'	S	X
S	✓(Allowed)	✗ (Not allowed)
X	✗ (Not allowed)	✗ (Not allowed)

lock held by Transaction T_i

Note:- If we use same lock for both read as well as write opn, then Even readers are not allowed to execute Concurrently.

simple lock for both read as well as write operation

T_1	T_2
l(A) (let granted)	
$R_1(A)$	
	~~l(A)~~ ← it will be denied because T1 has already acquired the lock on dataitem A.
	$R_2(A)$
$R_1(A)$	
	$R_2(A)$

{Simple lock does not differential b/w Read & write opn}

{T_2 is not allowed to Perform the opn Concurrently with T_1 even when both are just performing the read opn}

∴ We need seperate locks for read only And (read+write) opn, Hence we use Shared & Exclusive locks

2 mins Summary

Topic View equivalence condition

Topic Practice questions on View serializable schedule

Topic Concurrency control protocols

Topic Simple use of shared and exclusive locks

Topic Basic Two phase lock (2PL)

Next class

Slide

THANK - YOU