COMPUTER SCIENCE

Database Management System

Transaction & Concurrency Control

Lecture_2

Vijay Agarwal sir

TOPICS TO BE COVERED

01 Serializable Schedule

02 Conflict & View Serializable

Transaction Concept

Transaction State

ACID

A : Atomicity

C : Consistency

I : Isolation

D : Durablity

Schedule

- → Serial schedule (n!)
- → NonSerial schedule

Note ALL Serial schedule are always Consistent.

Note Non serial schedule May or May Not be Consistent.

But we execute Non Serial Schedule [Concurrent Execution].

Let T_1 transfer 100 Rs from A to B, and T_2 transfer 10% of the balance from A to B.

A:2000
+B:3000
5000

Schedule 1

T_1	T_2
read (A)	
A: = A – 100	
write (A)	
read (B)	
B: = B + 100	
write (B)	
commit	read (A)
	temp := A * 0.1
	A := A – temp
	write (A)
	read (B)
	B := B + temp
	write (B)
	Commit

A:1710
+B:3290
5000

Consistent

$S_1 < T_1\ T_2 >$

Serial schedule in which T_1 is followed by T_2:

Schedule 2

T_1	T_2
	read (A)
	temp := A * 0.1
	A := A – temp
	write (A)
	read (B)
	B := B + temp
	write (B)
read (A)	Commit
A: = A – 100	
write (A)	
read (B)	
B: = B + 100	
write (B)	
commit	

A=1700
+B=3300
5000

Consistent

$S_2 < T_2\ T_1 >$

serial schedule where T_2 is followed by T_1

ALL Serial Schedule

$S_1 < T_1\ T_2 >$ T_1 followed by T_2.

$S_2 < T_2, T_1 >$ T_2 followed by T_1

are always Consistent.

Schedule 3

T_1	T_2
read (A)	
A: = A – 100	
write (A)	
	read (A)
	temp := A * 0.1
	A := A – temp
	write (A)
read (B)	
B: = B + 100	
write (B)	
commit	
	read (B)
	B := B + temp
	write (B)
	Commit

A:1710
+ B:3290
5000
Consistent

$C_1 \equiv S_1 \langle T_1, T_2 \rangle$

Schedule 4

T_1	T_2
read (A)	
A: = A – 100	
	read (A)
	temp := A * 0.1
	A := A – temp
	write (A)
write (A)	
read (B)	
B: = B + 100	
write (B)	
commit	
	read (B)
	B := B + temp
	write (B)
	Commit

A:1900
+ B:3300
5200
Inconsistent

C_2

C_1 ✓ $\equiv S_1 \langle T_1, T_2 \rangle$

Non Serial Schedule C_1 is Consistent &

its equivalent to serial schedule $S_1 \langle T_1, T_2 \rangle$ T_1 followed by T_2.

Non serial Schedule C_2 is Not Consistent or In Consistent.

C_2 ✗

Serial Schedule

- □ After Commit of one transaction, begins (Start) another transaction.
- □ Number of possible serial Schedules with ‘n’ transactions is “n!”
- □ The execution sequence of Serial Schedule always generates consistent result.

Example

S : $R_1(A)$ $W_1(A)$ Commit (T_1) $R_2(A)W_2(A)$ commit (T_2).

T_1	T_2
R(A)	
W(A)	
Commit	
	R(A)
	W(A)
	Commit

T_1 followed by T_2

$<T_1, T_2>$

Advantage

- Serial Schedule always produce correct result (integrity guaranteed) as no resource sharing.

Disadvantage

- Less degree of concurrency.
- Through put of system is low.
- It allows transactions to execute one after another.

serializable schedule

If a Non Serial Schedule (Concurrent Execution) has been executed, that Could have Same effect on the Database, as a Schedule executed without Any Concurrent execution (Any Serial schedule) is Called Serializable schedule.

Note Serializable Schedule are always Consistent.

This Process is Called Serializablity.

How to achieve serializable schedule

① Conflict Serializable

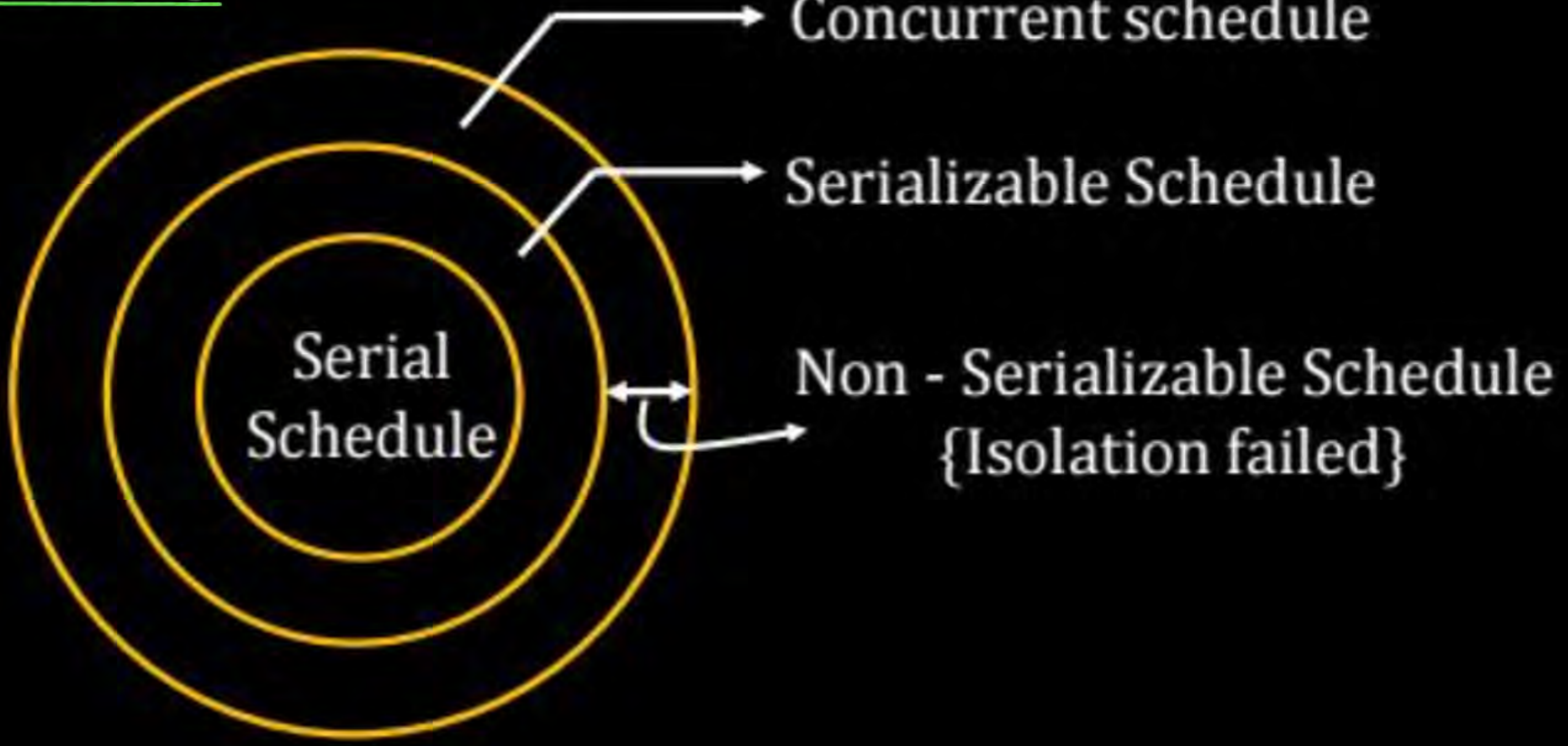
② View Serializable

Serializable Schedule

A Schedule is serializable Schedule if it is equivalent to a Serial Schedule.

(i)Conflict Serializability

(ii)View Serializability



Serializability

- □ Basic Assumption: Each transaction preserves database consistency.
- □ Thus, serial execution of a set of transactions preserves database consistency.
- □ A (possibly concurrent) schedule is serializable if it is equivalent to a serial schedule. Different forms of schedule equivalence give rise to the notions of:

1. Conflict serializability
2. view serializability

conflict serializable

1. Basic Concept

V.Imp. 2. Testing Method (Precedence Graph)

3. Conflict Equal to Any Serial Schedule.

conflict serializable

Conflict operation

Let us Consider Schedule S, in which there are Two Consecutive Instruction I_i & I_j of transaction T_i & T_j Respectively $[i \neq j]$

Same Data Item

T_i		T_j
R(A)	→	W(A)
W(A)	→	R(A)
W(A)	→	W(A)

Conflict Instruction/operation

[swapping Not Possible]

Non Conflict Operation/Instruction

T_i		T_j
R(A)	→	R(A)
R(A)	→	W(B)

Different Data Item

Non Conflict Instn/operation

[swapping Possible]

conflict serializable

S: Non Serial Schedule
[Given Question]

$$S \xrightarrow[]{\text{by Series of Swap of Non Conflicting Instruction/operation}} S'$$

S': Any Serial Schedule of S' (Of Given Question)

then Schedule S is Conflict Serializable.

Conflict Serializability

- If a schedule S can be transformed into a schedule S' by a series of swaps of non-conflicting instructions, we say that S and S' are conflict equivalent.
- We say that a schedule S is conflict serializable if it is conflict equivalent to a serial schedule.

Q

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

1710
3290
5000
<T1 T2>

S

C1

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

✓

<T1 | T2>

Q.1

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

S

C1

Not Possible to Convert S'<T2, T1>

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

S' <T2, T1>. T2 followed by T1.

Q.1

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

S

C_1

yse possible to swap & Convert into $S'<T_1 T_2>$

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

$S'<T_1, T_2>$. T_1 followed by T_2.

G in eg.

Here S is Conflict Serializable

to $S'\langle T_1, T_2 \rangle$ [T_1 followed by T_2].

Q.2

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

5200

Inconsistent

Not Possible

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

<T2 T1>

Q.2

T1	T2
R(A)	
	R(A)
	W(A)
W(A)	
R(B)	
W(B)	
	R(B)
	W(B)
	C2

Not possible

X s'(T1, T2)

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

C_2 is Not Possible to convert either

$S' \langle T_1, T_2 \rangle$

& $S' \langle T_2, T_1 \rangle$

So C_2 is Not Conflict Serializable.

C_2

Conflict Serializability (Cont.)

- Schedule 3 can be transformed into Schedule 6, a serial schedule where T_2 follows T_1, $<T_1, T_2>$ by series of swaps of non-conflicting instructions. Therefore Schedule 3 is conflict serializable.

Schedule 3

T_1	T_2
read (A)	
Write (A)	
	read (A)
	write (A)
read (B)	
write (B)	
	read (B)
	write (B)

Schedule 6

T_1	T_2
read (A)	
write (A)	
read (B)	
write (B)	
	read (A)
	write (A)
	read (B)
	write (B)

Conflict Serializability (Cont.)

- Example of a schedule that is not conflict serializable:

T_3	T_4
read (Q) ✓	
	write (Q) ?
write (Q)	

T_1	T_2
R(Q)	
W(Q)	
	W(Q)

✗

T_1	T_2
	W(Q)
R(Q)	
W(Q)	

✗

- We are unable to swap instructions in the above schedule to obtain either the serial schedule < T3, T4 >, or the serial schedule < T4, T3 >

Conflict Serializable

A schedule is said to be conflict serializable if it is conflict equivalent to a serial schedule.

Same conflicting operation order in C_1 & S_1

∴ Its $\{C_1\}$ conflict is conflict serializable.

T_1	T_2
read(A)	
write(A)	
	read(A)
	write(A)
read(B)	
write(B)	
	read(B)
	write(B)

C_L

T_1	T_2
read(A)	
write(A)	
read(B)	
write(B)	
	read(A)
	write(A)
	read(B)
	write(B)

S_L

Conflicting Instructions

- Instructions l_i, and l_j of transactions T_i and T_j respectively, conflict if and only if there exists some item Q accessed by both l_i, and l_j, and at least one of these instructions write Q. $(i \neq j)$

1. l_i, = read(Q), l_j = read(Q). l_i and l_j don't conflict. (Non Conflict Instⁿ)
2. l_i, = read(Q) l_j = write(Q). They conflict.
3. l_i, = write(Q) l_j = read(Q). They conflict
4. l_i= write(Q) l_j = write(Q). They conflict

Conflict Instruction

- Intuitively, a conflict between l_i and l_j forces a (logical) temporal order between them.
  - If l_i, and l_j are consecutive in a schedule and they do not conflict, their results would remain the same even if they had been interchanged in the schedule.

Testing for conflict serializable

Precedence Graph Method

$G(V, E)$

(vertex) V: Set of Transactions.

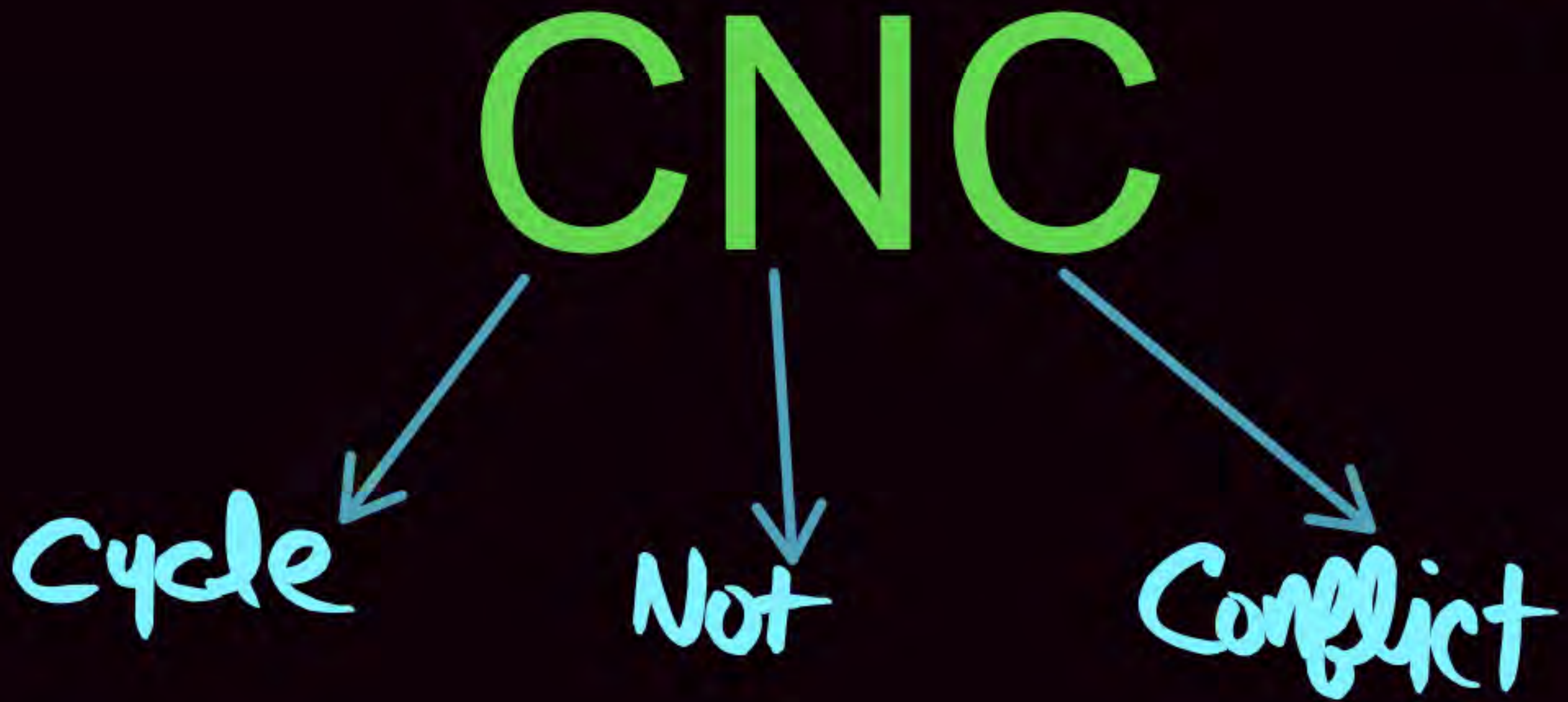
(Edge) E: Edge $T_i \longrightarrow T_j$ edge occur iff any one condition hold

$$\begin{array}{lcl} & T_i & & T_j \\ \text{Conflict operation} & R(A) & \longrightarrow & W(A) \\ & W(A) & \longrightarrow & R(A) \\ & W(A) & \longrightarrow & W(A) \end{array}$$

then edge Draw.

Testing for conflict serializable

Precedence Graph Method



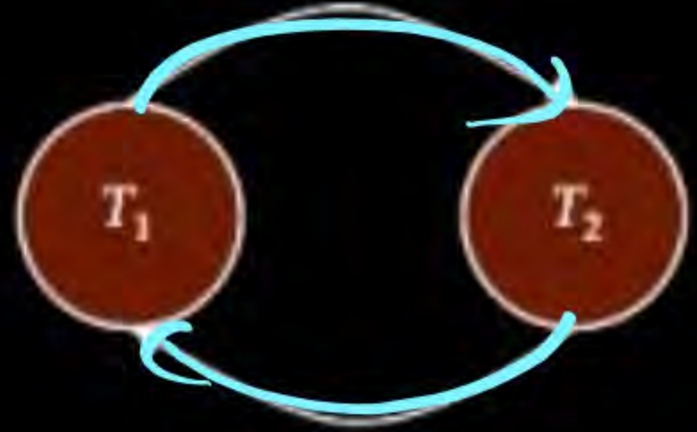
CNC

Note If Precedence Graph Contain cycle [Any one cycle] then Schedule is Not Conflict serializable

Testing for Serializability

- Testing for conflict serializability.
  - Consider some schedule of a set of transactions T_1, T_2, ...T_n
  - Precedence graph — a direct graph where the vertices are the transactions (names).
  - We draw an arc from T_i to T_j if the two transaction conflict, and T_i accessed the data item on which the conflict arose earlier.
  - We may label the arc by the item that was accessed.

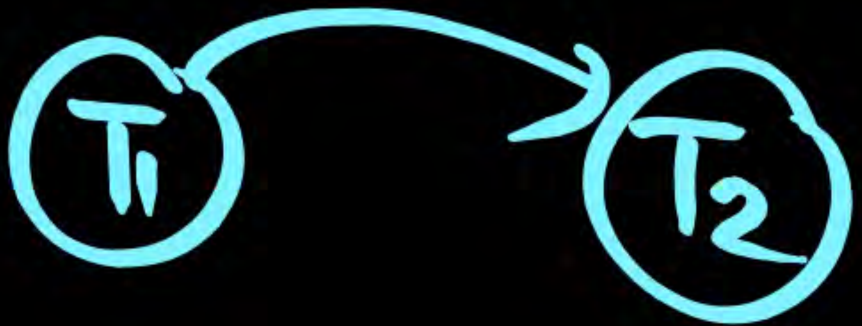
Example:



CNC

Cycle Not Conflict Serializable.

A schedule is conflict serializable if and only if its precedence graph is acyclic.



Conflict serializable

NOTE: CNC [Cycle not conflict serializable]

Q.1

$S: R_1(A)\ W_1(A)\ R_2(A)\ W_2(A)\ R_1(B)\ W_1(B)\ R_2(B)\ W_2(B)$

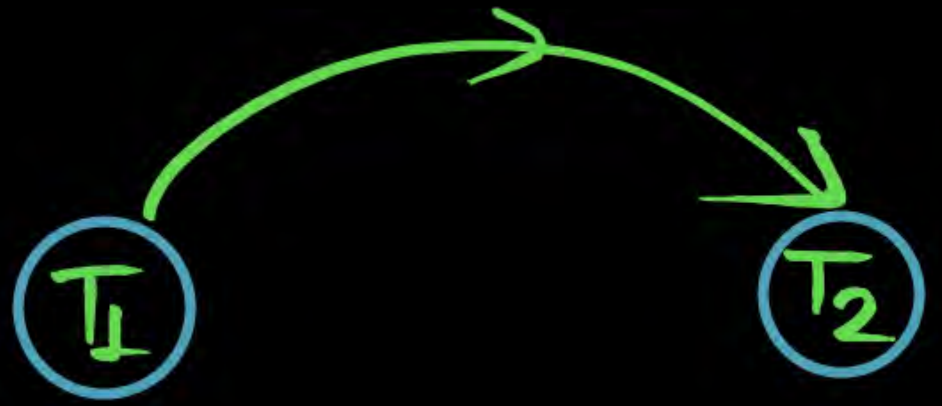
T_1	T_2
R(A)	
W(A)	
	R(A) ✓
	W(A) ✓
R(B) ✓	
W(B)	
	R(B)
	W(B)

C1

1710
3290
5000

Consistent

<T1, T2>



Acyclic

Conflict serializable

<T1, T2>

Q.2 $R_1(A)\ R_2(A)\ W_2(A)\ W_1(A)\ R_1(B)\ W_1(B)\ R_2(B)\ W_2(B)$

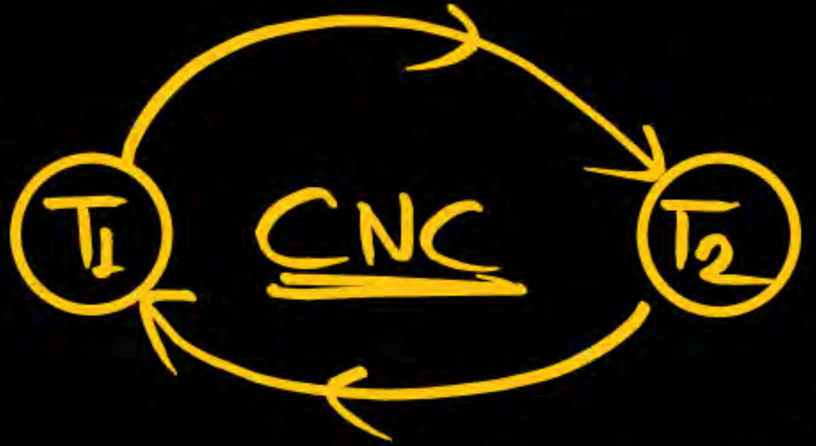
T_1	T_2
R(A)	
	R(A)
	W(A)
W(A)	
R(B)	
W(B)	
	R(B)
	W(B)

Conflict operation

$R(Q) \rightarrow W(Q)$

$W(Q) \rightarrow R(Q)$

$W(Q) \rightarrow W(Q)$



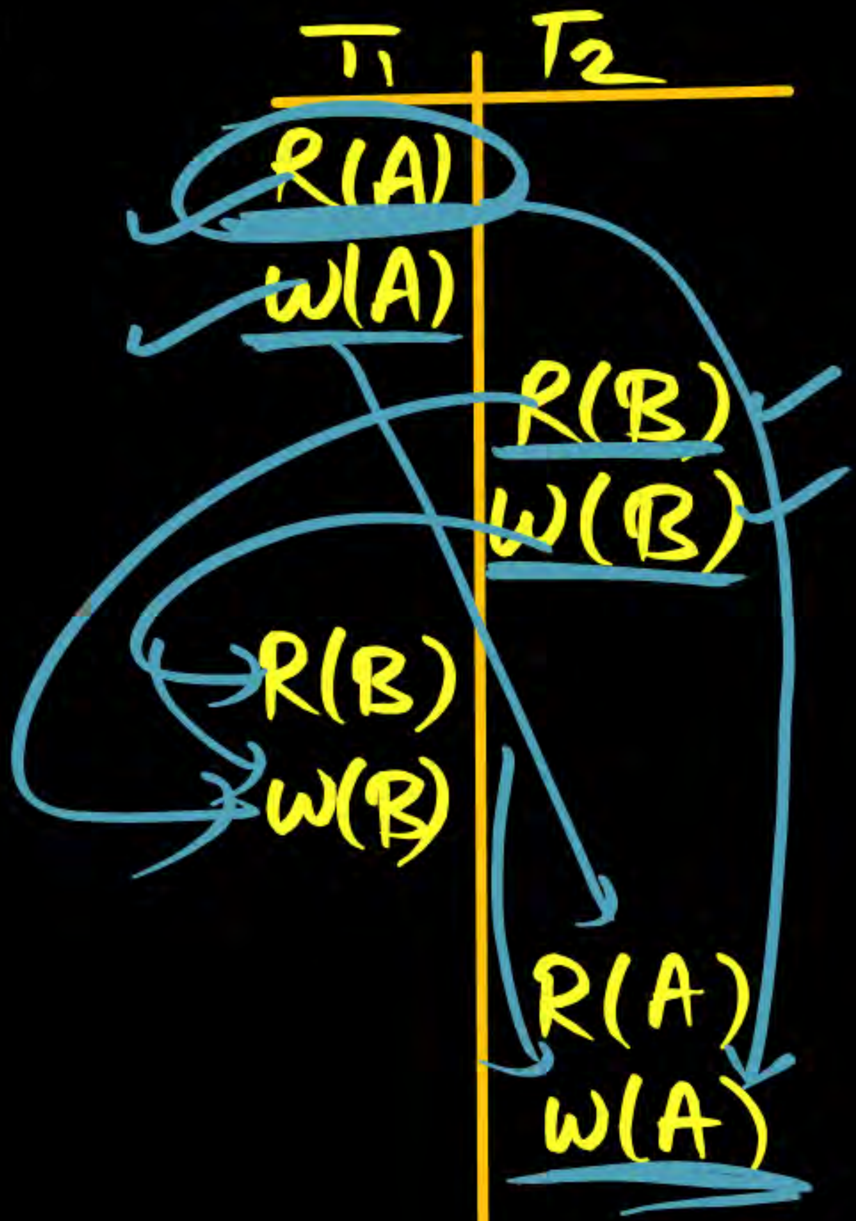
Cycle Not Conflict

C_2

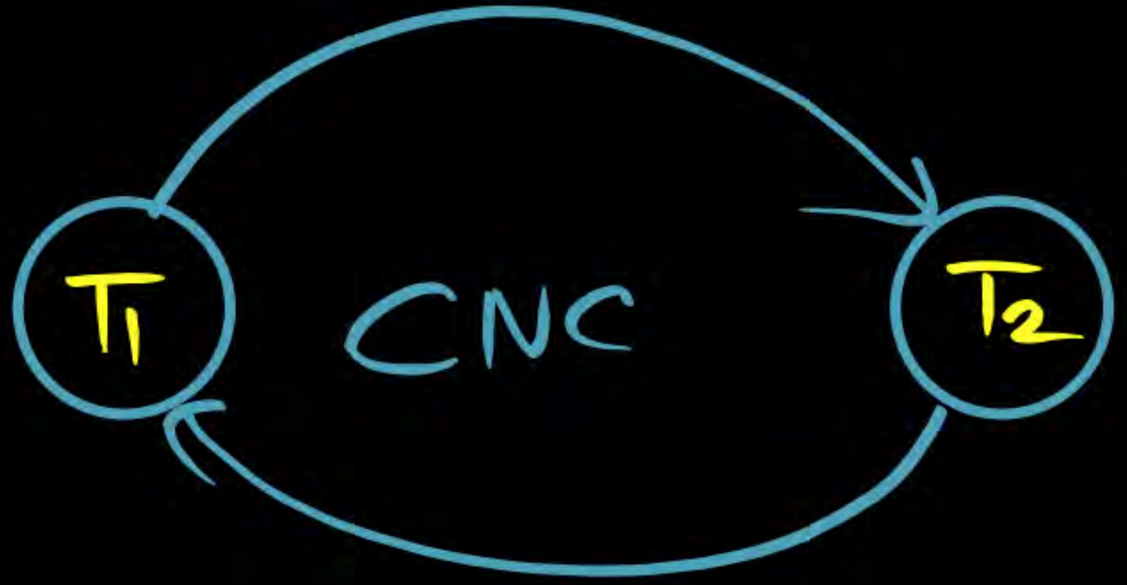
5200 Inconsistent

Q.3

$R_1(A)\ W_1(A)\ R_2(B)\ W_2(B)\ R_1(B)\ W_1(B)\ R_2(A)\ W_2(A)$



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



Cycle Not Conflict

Q.1

Consider the following schedules involving two transactions. Which one of the following statements is TRUE?

S_1: $r_1(X); r_1(Y); r_2(X); r_2(Y); w_2(Y); w_1(X)$

S_2: $r_1(X); r_2(X); r_2(Y); W_2(Y); r_1(Y); w_1(X)$

[2007: 2 Marks]

A Both S_1 and S_2 are conflict serializable

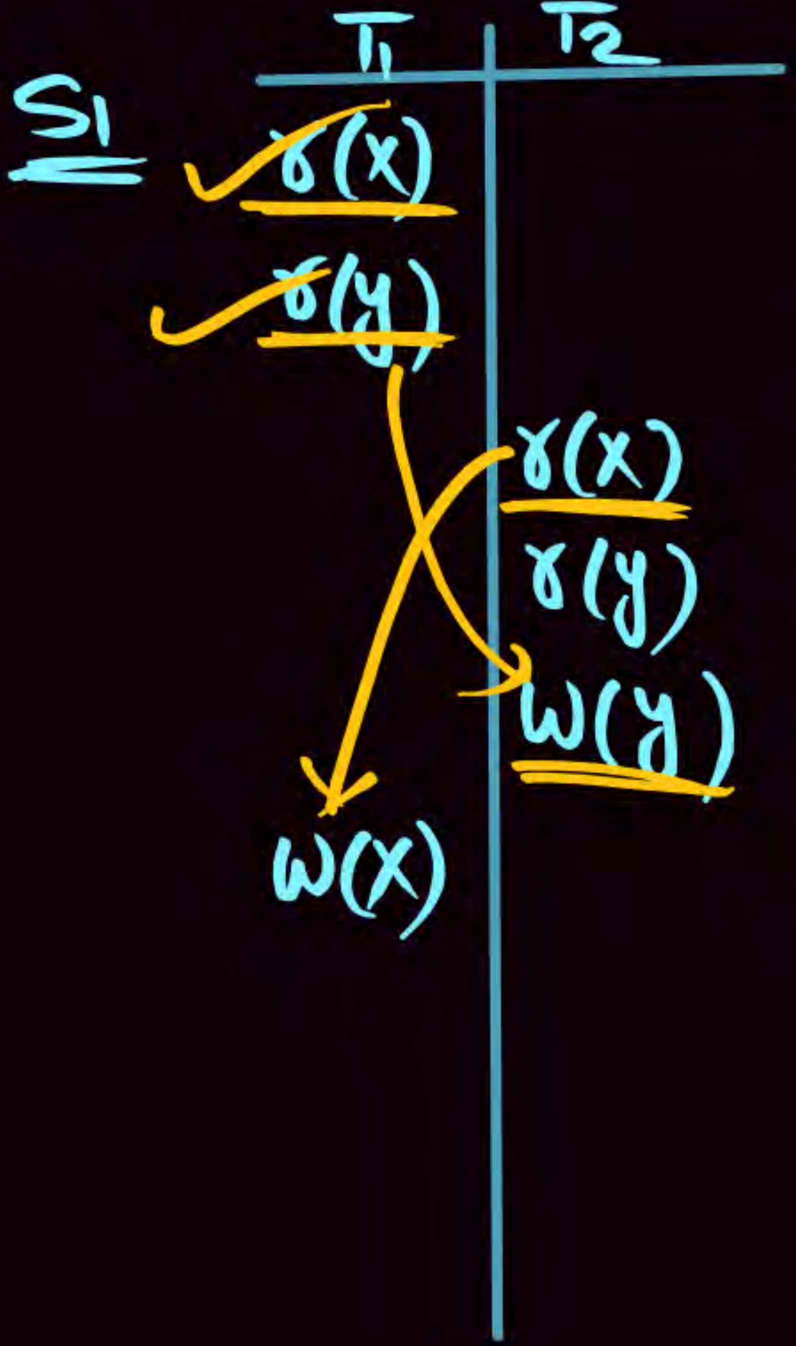
B S_1 is conflict serializable and S_2 is not conflict serializable

[✓ C] S_1 is not conflict serializable and S_2 is conflict serializable

D Both S_1 and S_2 are not conflict serializable

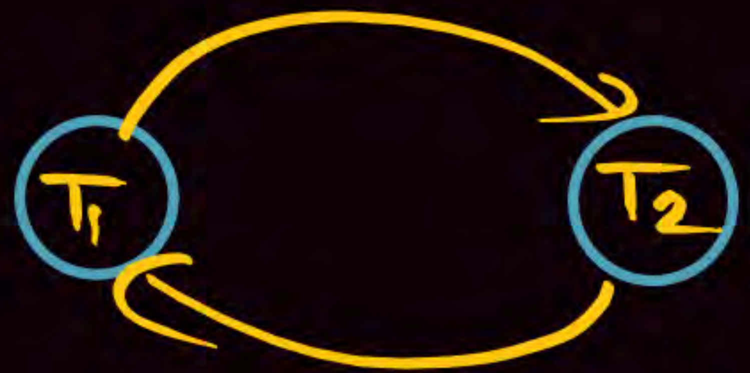
S_1 : $r_1(x)$ $r_1(y)$ $r_2(x)$ $r_2(y)$ $w_2(y)$ $w_1(x)$

S_2 $r_1(x)$ $r_2(x)$ $r_2(y)$ $w_2(y)$ $r_1(y)$ $w_1(x)$



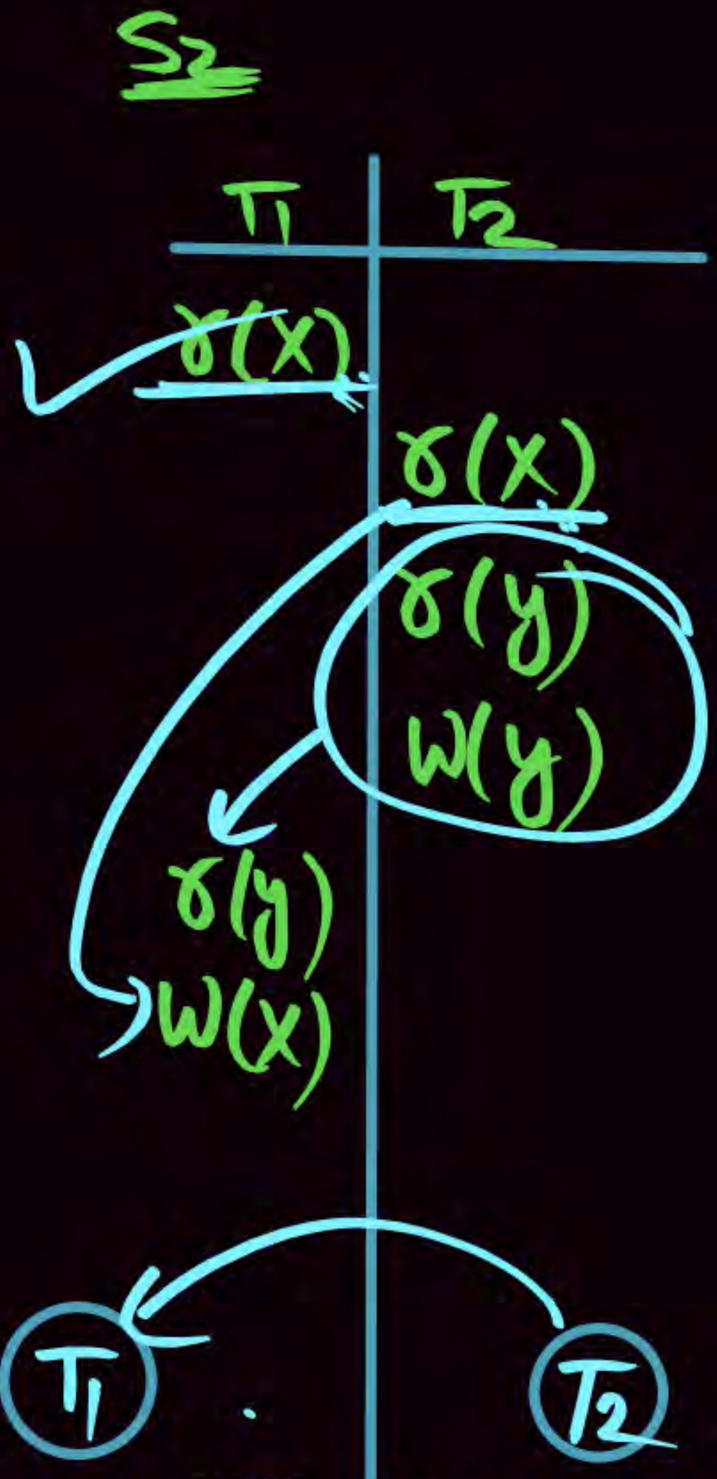
S_1

T_1	T_2
r(x)	
r(y)	
	r(x)
	r(y)
	w(y)
w(x)	



Cycle Not Conflict

S_1 is Not Conflict



S_2

T_1	T_2
r(x)	
	r(x)
	r(y)
	w(y)
r(y)	
w(x)	

$\langle T_2, T_1 \rangle$

S_2 Conflict serializable

Q.2

Consider the following four schedules due to three transactions (indicated by the subscript) using read and write on a data item x, denoted by r(x) and w(x) respectively. Which one of them is conflict serializable?

[2014(Set-1): 2 Marks]

A $r_1(x); r_2(x); w_1(x); r_3(x); w_2(x)$

B $r_2(x); r_1(x); w_2(x); r_3(x); w_1(x)$

C $r_3(x); r_2(x); r_1(x); w_2(x); w_1(x)$

D $r_2(x); w_2(x); r_3(x); r_1(x); w_1(x)$

MCQ **Q.3**

Consider the transactions T1, T2 and T3 and the schedules S1 and S2 given below.

T1: r1(X); r1(Z); w1(X); w1(Z)
T2: r2(Y) ; r2(Z) ; w2(Z)
T3: r3(Y); r3(X); w3(Y)
S1: r1(X); r3(Y); r3(X); r2(Y); r2(Z); w3(Y); w2(Z); r1(Z); w1(X); w1(Z)
S2: r1(X); r3(Y); r2(Y); r3(X); r1(Z); r2(Z); w3(Y); w1(X); w2(Z); w1(Z)

Which one of the following statements about the schedules is TRUE?

[GATE-2014-CS: 2M]

A Only S1 is conflict-serializable.

B Only S2 is conflict-serializable.

C Both S1 and S2 are conflict-serializable.

D Neither S1 nor S2 is conflict-serializable.

Q.4

Let $r_i(z)$ and $w_i(z)$ denote read and write operations respectively on a data item by a transaction T_i. Consider the following two schedules.

S_1: $r_1(x)\ r_1(y)\ r_2(x)\ r_2(y)\ w_2(y)\ w_1(x)$

S_2: $r_1(x)\ r_2(x)\ r_2(y)\ w_2(y)\ r_1(y)\ w_1(x)$

Which one of the following options is correct? [MCQ: 2021: 2M]

A S_1 is conflict serializable, and S_2 is not conflict serializable.

B S_1 is not conflict serializable, and S_2 is conflict serializable.

C Both S_1 and S_2 are conflict serializable.

D Neither S_1 nor S_2 is conflict serializable.

Q.5

Let $R_i(z)$ and $W_i(z)$ denote read and write operations on a data element z by a transaction T_i, respectively. Consider the schedule S with four transactions.

S: $R_4(x), R_2(x), R_3(x), R_1(y), W_1(y), W_2(x), W_3(y), R_4(y)$

Which one of the following serial schedules is conflict equivalent to S? [2022: 2 Marks]

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

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

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

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

Q.6

Consider the following transaction involving two bank accounts x and y.

read(x); x: = x – 50; write (x); read (y); y: = y + 50; write (y)

The constraint that the sum of the accounts x and y should remain constant is that of [2015(Set-2): 1 Marks]

A Atomicity

B Consistency

C Isolation

D Durability

MCQ Q.7

Which one of the following is NOT a part of the ACID properties of database transactions?

[GATE-2016-CS: 1M]

A Atomicity

B Consistency

C Isolation

D Deadlock-freedom

MCQ Q.8

Suppose a database schedule S involves transaction T_1......, T_n. Construct the precedence graph of S with vertices representing the transactions and edges representing the conflicts. If S is serializable, which one of the following orderings of the vertices of the precedence graph is guaranteed to yield a serial schedule?

[GATE-2016-CS: 2M]

(A) Topological order

(B) Depth-first order

(C) Breadth-first order

(D) Ascending order of transaction indices

MCQ **Q.9**

Consider the following schedule for transactions T1, T2 and T3:

T1	T3	T3
Read(X)		
	Read (Y)	
		Read (Y)
	Write (Y)	
Write (X)		
		Write (X)
	Read (X)	
	Write (X)	

Which one of the schedules below is the correct serialization of the above?

[GATE-2010-CS: 2M]

A T 1 →T 3 →T 2

B T 2 →T 1 →T 3

C T2 → T3 → T1

D T3 →T 1 →T 2

MCQ **Q.10**

Consider two transactions T_1 and T_2, and four schedules S_1, S_2, S_3, S_4 of T_1 and T_2 as given below:

T_1: $R_1[x]\ W_1[x]\ W_1[y]$;

T_2: $R_2[x]\ R_2[y]\ W_2[y]$;

S_1: $R_1[x]\ R_2[x]\ R_2[y]\ W_1[x]\ W_1[y]\ W_2[y]$;

S_2: $R_1[x]\ R_2[x]\ R_2[y]\ W_1[x]\ W_2[y]\ W_1[y]$;

S_3: $R_1[x]\ W_1[x]\ R_2[x]\ W_1[y]\ R_2[y]\ W_2[y]$;

S_4: $R_2[x]\ R_2[y]\ R_1[x]\ W_1[x]\ W_1[y]\ W_2[y]$;

Which of the above schedules are conflict serializable?

[GATE-2009-CS: 2M]

A S_1 and S_2

B S_2 and S_3

C S_3 only

D S_4 only

THANK YOU!