COMPUTER SCIENCE

Database Management System

Transaction & Concurrency Control

Lecture_4

Vijay Agarwal sir

01 Serializable Schedule

02 Conflict & View Serializable

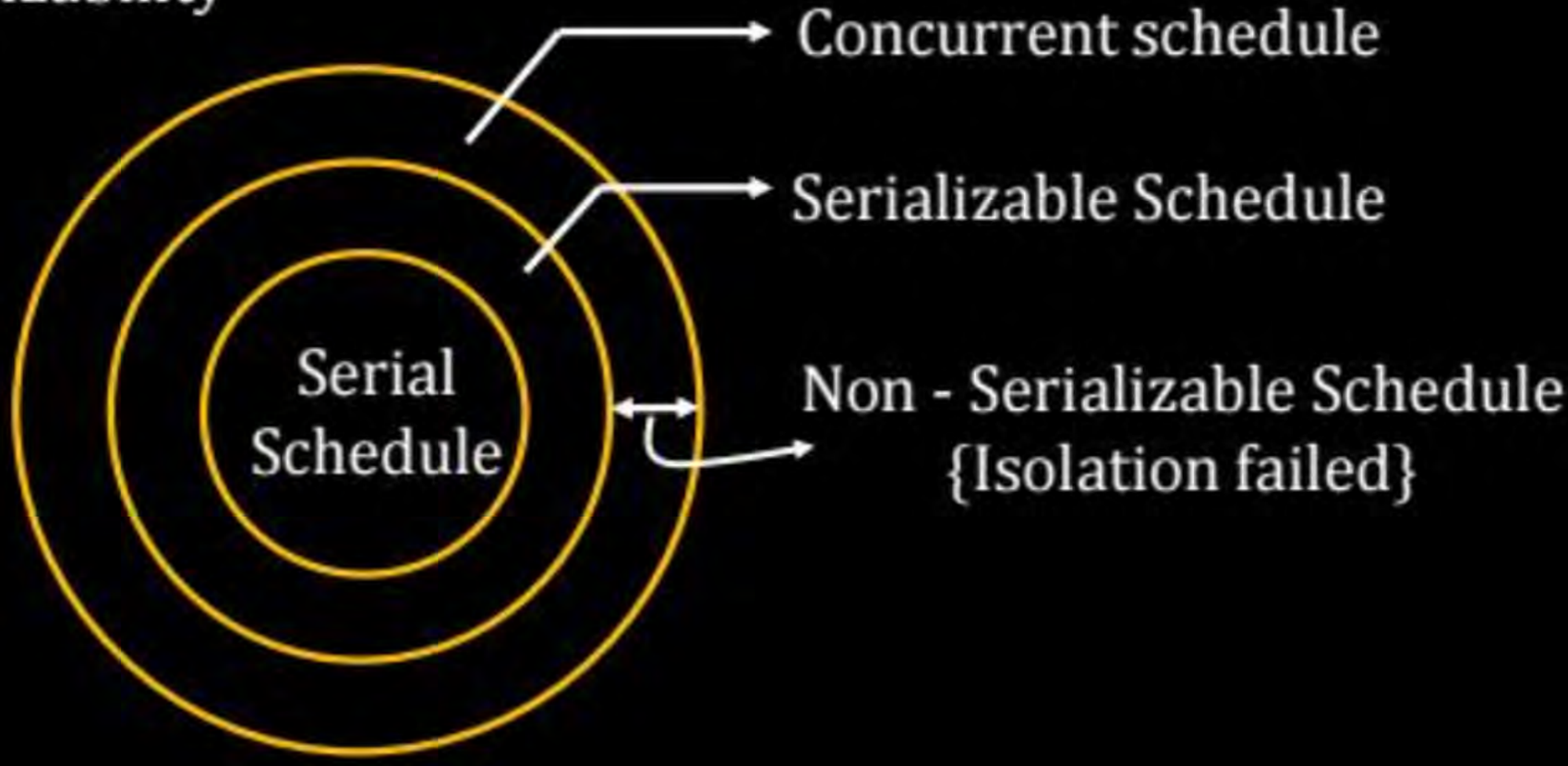
- Serial Schedule are Always Consistent.
- Non Serial Schedule May (or) May Not be Consistent.

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
2. Testing Method [Precedence Graph Method] (VIMP)
3. Conflict Pair
4. Conflict equivalent to Any Serial Schedule

① Conflict Instruction/operation.

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

$(i \neq j)$ & Same DATA ITEM

Conflict Inst^n/operation.
[Swapping Not possible].

(II) Non Conflict Instruction.

T_i T_j $(i \neq j)$ &

R(A)	R(A)
&	
T_i & T_j operating on Different Data Item	
W(A) — R(B)	
R(A) — W(B)	
W(B) — W(A)	

Non Conflict Instn [swapping Possible].

Conflicting Instructions

- Instructions I_i, and I_j of transactions T_i and T_j respectively, conflict if and only if there exists some item Q accessed by both I_i, and I_j, and at least one of these instructions wrote Q.
  1. I_i, = read(Q), I_j = read(Q). I_i and I_j don't conflict.
  2. I_i, = read(Q) I_j = write(Q). They conflict.
  3. I_i, = write(Q) I_j = read(Q). They conflict
  4. I_i= write(Q) I_j = write(Q). They conflict
- Intuitively, a conflict between I_i and I_j forces a (logical) temporal order between them.
  - If I_i, and I_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.

1 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.

S: Non Serial Schedule in Given Question.

S': Any Serial Schedule of S.

∴

(8) S

C1 : A: 1710
B: 3290
5000

S1<T1 T2>

C1 ≠ S2'<T2, T1>

C1 = S1'<T1 T2>

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

S1'<T1, T2>

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

S2'<T2, T1>

C_L is Not Possible to Convert Serial Schedule $S_2' \langle T_2, T_1 \rangle$.

C_1 is Possible to Convert into Serial Schedule $S_1' \langle T_1\ T_2 \rangle$.

So C_1 is Conflict Serializable & equivalent to Serial schedule $S_1 \langle T_1\ T_2 \rangle$

T_1 followed by T_2.

G: 1710
3290
5000 Consistent
$S_1 \langle T_1\ T_2 \rangle$

Conflict Serializability (Cont.)

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

S1 ⊂ T1 T2 ?.

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)

(8)

C1: 1900
3300
5200

C2

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

$C_2 \neq S_2' \langle T_2 T_1 \rangle$

$C_2 \neq S_1' \langle T_1 T_2 \rangle$

$S_1' \langle T_1, T_2 \rangle$

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

$S_2' \langle T_2, T_1 \rangle$

No serial Schedule C_2 Not Possible to convert

$S_1' < T_1\ T_2 >$

$S_2' < T_2\ T_1 >$

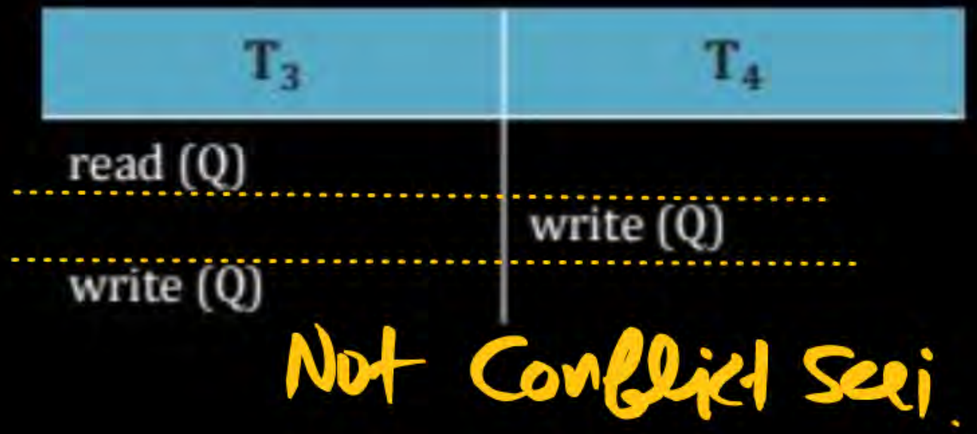
So Not Conflict Serializable.

C_2 :

A: 1900
B: 3300
5200

Conflict Serializability (Cont.)

- Example of a schedule that is not conflict serializable:



- 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

$\therefore$ 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 wrote Q.
  1. l_i, = read(Q), l_j = read(Q). l_i and l_j don't conflict.
  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
- 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.

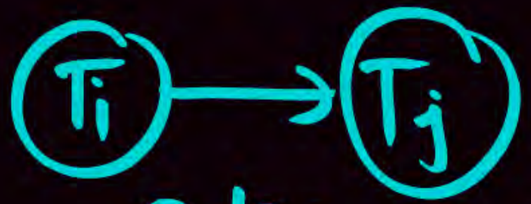
Testing for Conflict Serializability:

Precedence Graph method. (Directed Graph)

$G(V, E)$

V: Set of Transaction.

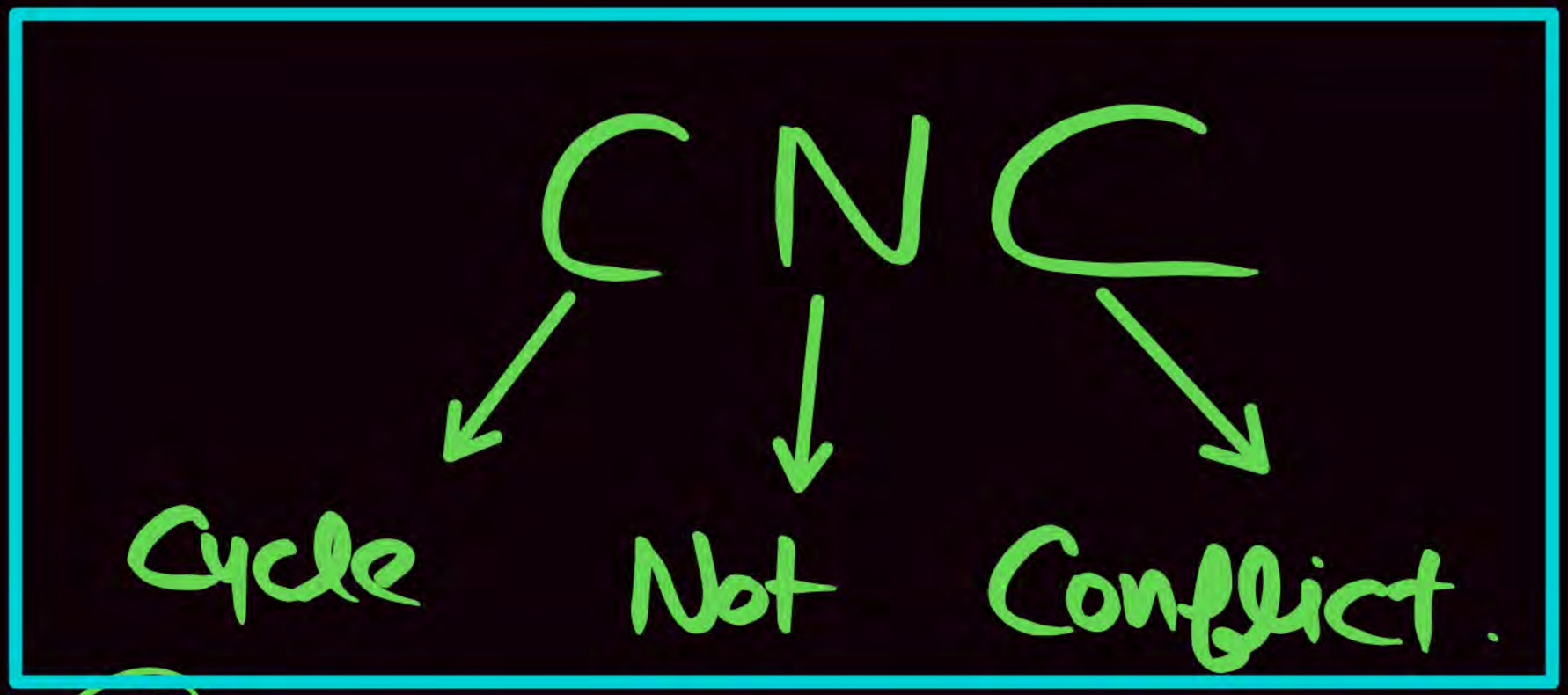
E: Edge Represent Conflict



edge occur. → Operation from Ti to Tj

Conflict Instr

R(A) – W(A)
W(A) – R(A)
W(A) – W(A)



Note

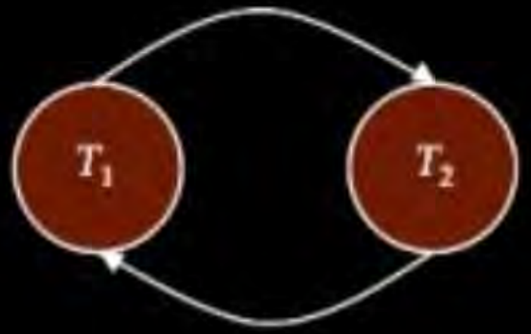
If Precdence graph Contain Cycle (Any one cycle) then Schedule is Not Conflict Serializable.

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.

Edge

Example:



Cycle Not Conflict.

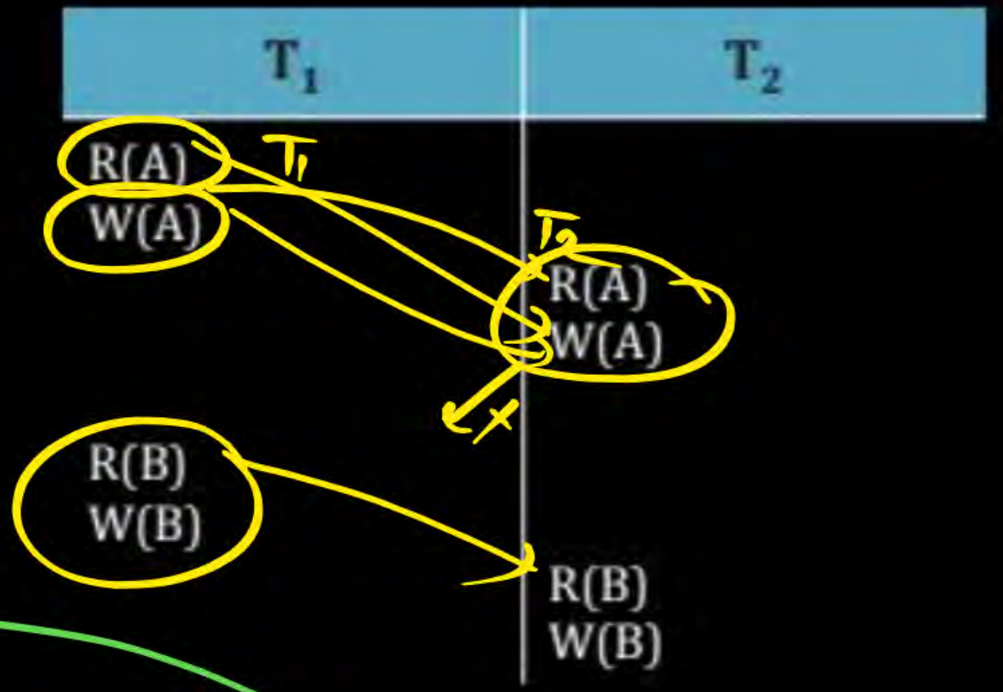
A schedule is conflict serializable if and only if its precedence graph is acyclic.

if Acyclic [Not cycle] then Conflict Serializable.

NOTE: CNC [Cycle not conflict serializable]

Q.1

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



Not cycle

Conflict Serializable

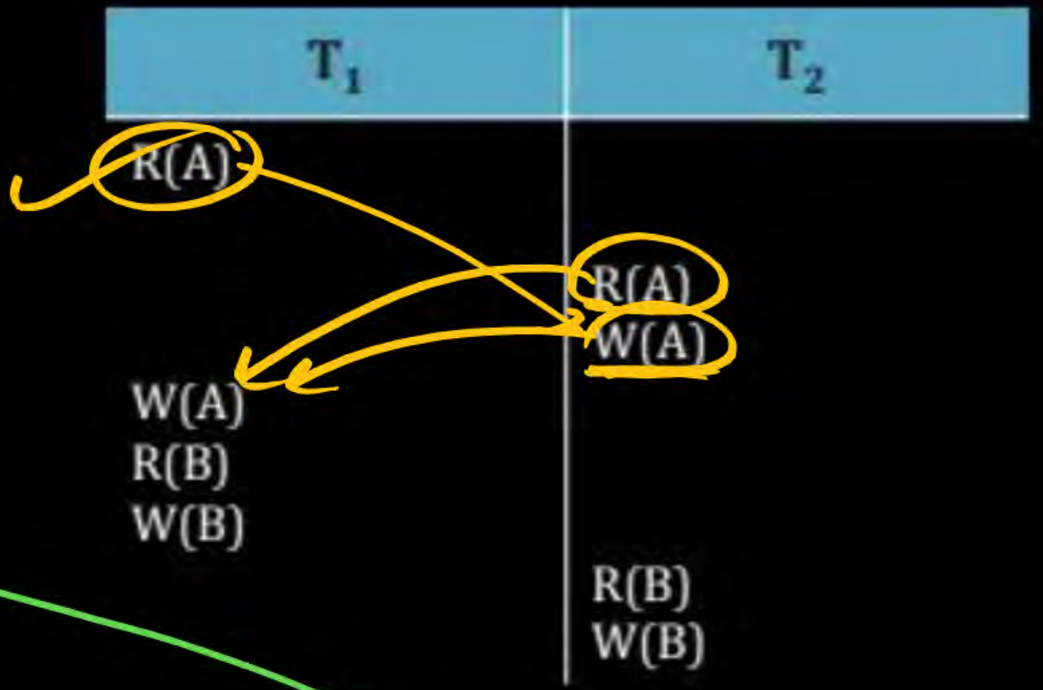
$\langle T_1\ T_2 \rangle$

C1 1710

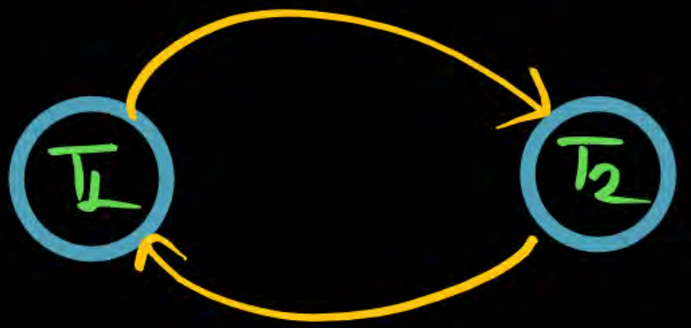
3290

5000 Consistent $S_1 \langle T_1, T_2 \rangle$

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



CNC



Cycle
Not Conflict

C2
A: 1900
B: 3300
5200
Inconsistent

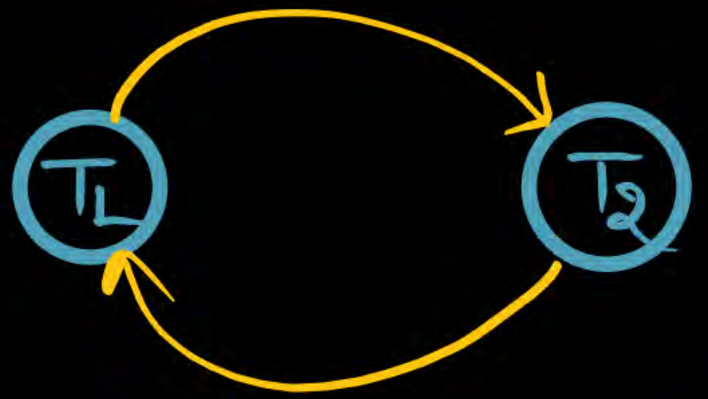
Q.3

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

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

T_i T_j

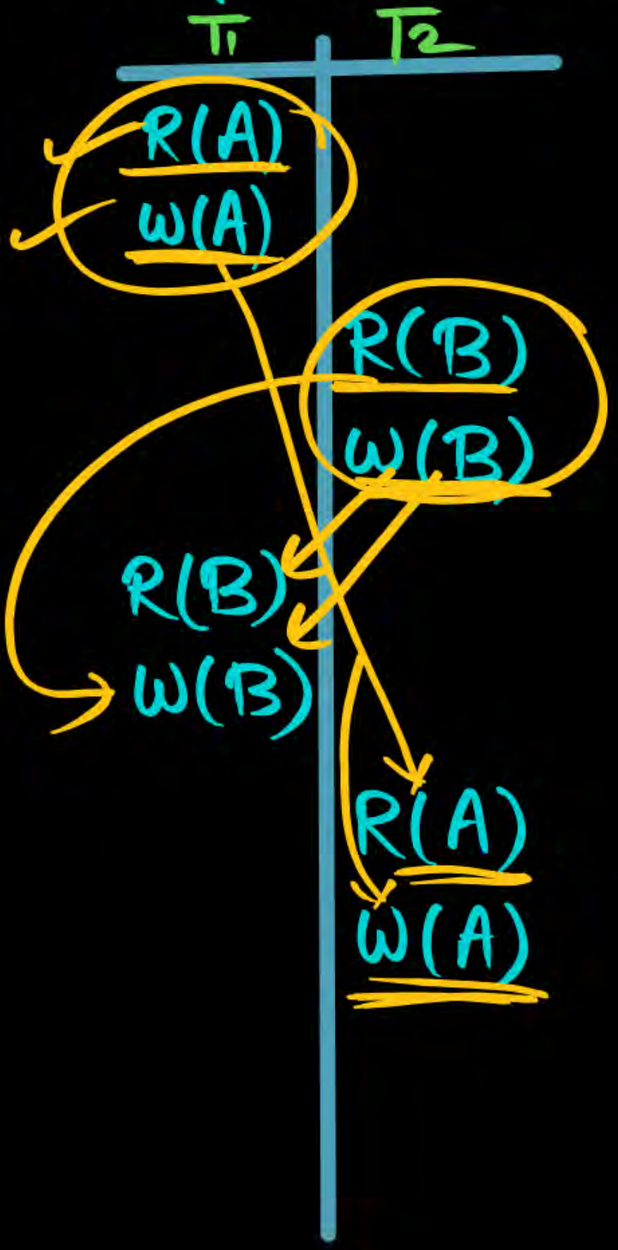
R(Q) - W(Q)
W(Q) - R(Q)
W(Q) - W(Q)

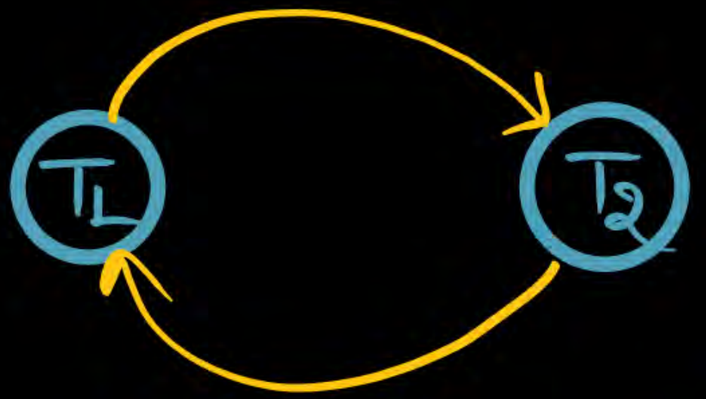


Cycle Not Conflict

Q.3

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





Cycle Not Conflict

Serializability Order

Important Point 1:

1. If S_1, S_2 Schedule are conflict equal then precedence graph of S_1 and S_2 must be same.
2. If S_1 and S_2 have same precedence graph then S_1 and S_2 may or may not conflict equal.

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

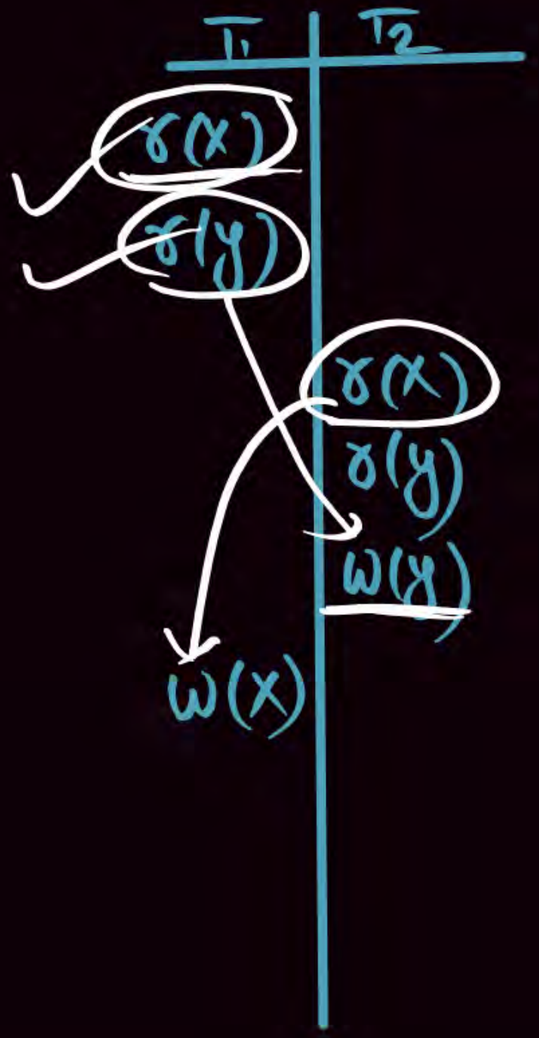
B. S_1 is conflict serializable and S_2 is not conflict serializable

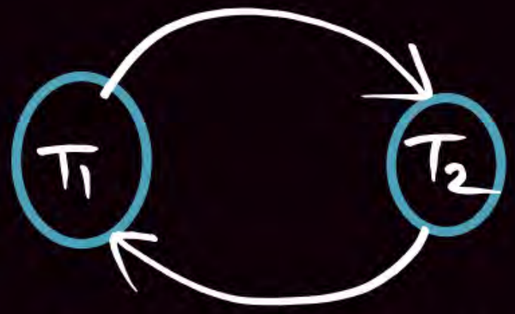
C. S_1 is not conflict serializable and S_2 is conflict serializable

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: $r_1(X)$ $r_2(X)$ $r_2(y)$ $w_2(y)$ $r_1(y)$ $w_1(X)$





Cycle Not Conflict.

Conflict Serializable $\langle T_2\ T_1 \rangle$.

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?

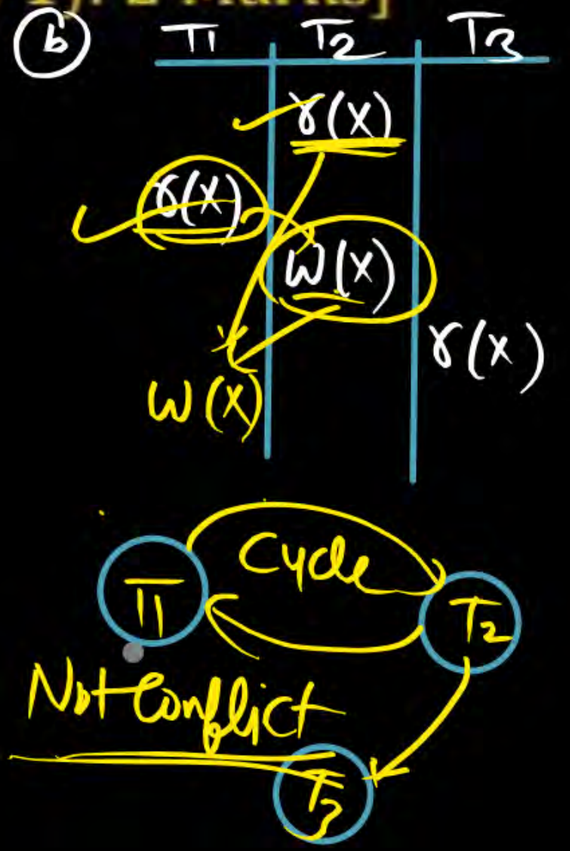
[2014(Set-1): 2 Marks]

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



T_1: $\sigma_1(x)\ w_1(x)$

T_2: $\sigma_2(x)\ w_2(x)$

T_3: $\sigma_3(x)$

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?

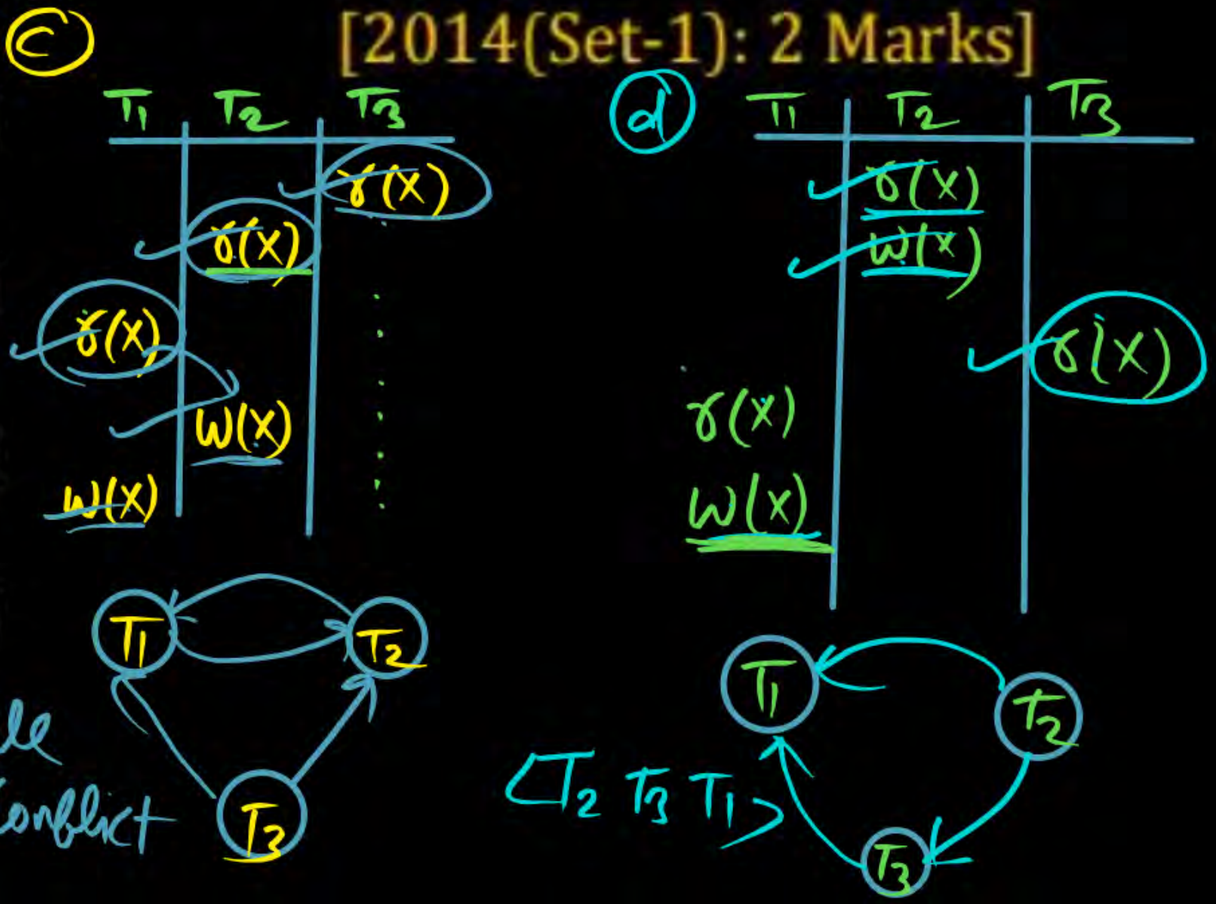
[2014(Set-1): 2 Marks]

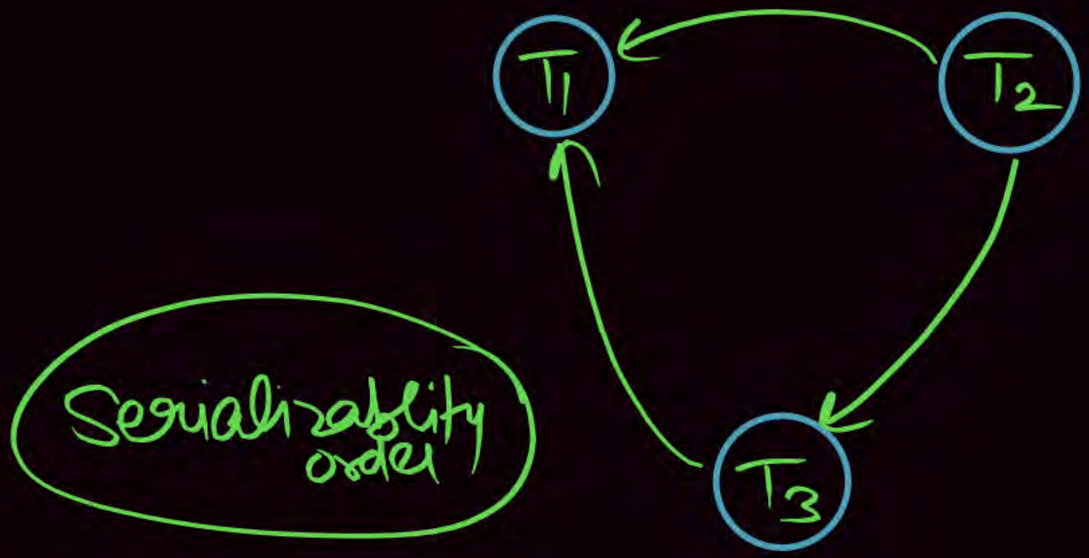
A $r_1(x); r_2(x); w_1(x); r_3(x); w_2(x)$

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

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

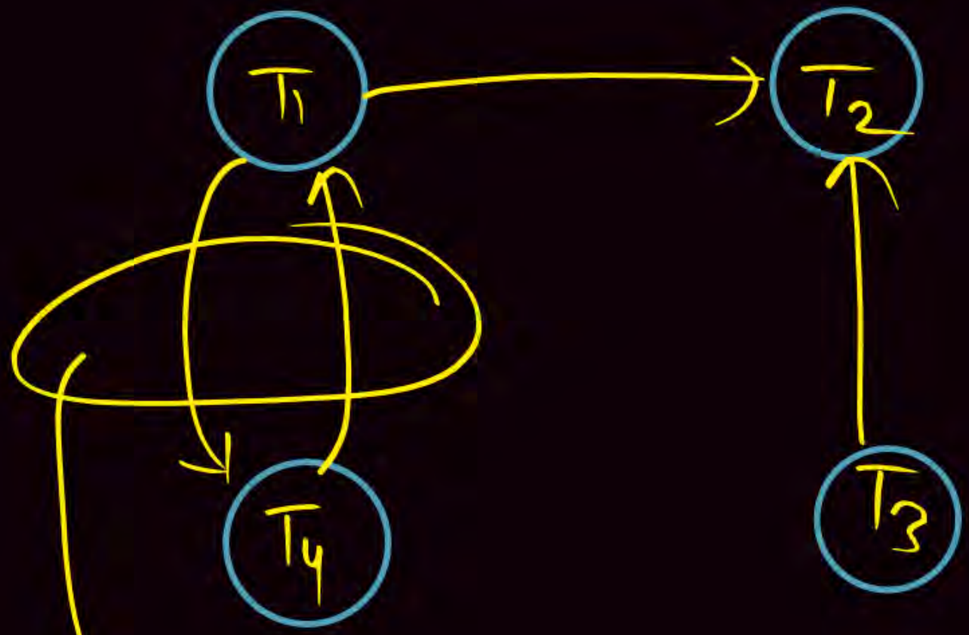
D $r_2(x); w_2(x); r_3(x); r_1(x); w_1(x)$





<T2 T3 T1>

Topological Sorting



Any One Cycle then Not Conflict Serializable

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:

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

[GATE-2010-CS: 2M]

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

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

Conflict Equivalent Schedule

Two schedule are said to be conflict equivalent, if all conflicting operations in both the schedules must be executed in the same order.

Q.

S_1: $R_1(x)$ $W_1(x)$ $R_2(y)$ $W_2(y)$ $R_1(y)$

S_2: $R_1(x)$ $W_1(x)$ $R_1(y)$ $R_2(y)$ $W_2(y)$

Q.

S_1: $R_1(A)$ $W_1(A)$ $R_2(A)$ $W_2(A)$ $R_1(B)$ $W_1(B)$

S_2: $R_1(A)$ $W_1(A)$ $R_2(A)$ $R_1(B)$ W_2 (A) $W_1(B)$

Q.11

Consider a schedule of transactions T_1 and T_2:

T_1	RA			RC		WD		WD	Commit	
T_2		RB	WB		RD		WC			Commit

Here, RX stands for "Read(X)" and WX stands for "Write(X)". Which one of the following schedules is conflict equivalent to the above schedule?

[2020: 2 Marks]

A

T_1					RA	RC	WD	WB	Commit	
T_2	RB	WB	RD	WC						Commit

B

T_1				RA	RC	WD	WB		Commit	
T_2	RB	WB	RD					WC		Commit

C

T_1	RA	RC	WD	WB					Commit	
T_2					RB	WB	RD	WC		Commit

D

T_1	RA	RC	WD	WB					Commit	
T_2					RB	WB	RD	WC		Commit

MCQ **Q.12**

Consider the following three schedules of transactions T1, T2 and T3. [Notation: In the following NYO represents the action Y (R for read, W for write) performed by transaction N on object O.]

S1: 2RA 2WA 3RC 2WB 3WA 3WC 1RA 1RB 1WA 1WB

S2: 3RC 2RA 2WA 2WB 3WA 1RA 1RB 1WA 1WB 3WC

S3: 2RA 3RC 3WA 2WA 2WB 3WC 1RA 1RB 1WA 1WB

Which of the following statements is TRUE? [GATE-2008-CS: 2M]

A. S1, S2 and S3 are all conflict equivalent to each other

B. No two of S1, S2 and S3 are conflict equivalent to each other

C. S2 is conflict equivalent to S 3, but not to S1

D. S1 is conflict equivalent to S2, but not to S3

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

$\therefore$ 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

Home Work.

1. Normal FORM PYQ's..
2. Today class PYQ's

THANK YOU!