COMPUTER SCIENCE

Database Management System

Transaction & Concurrency Control

Lecture_4

Vijay Agarwal sir

TOPICS TO BE COVERED

01 View Serializable

02 Problem due to concurrent execution

Conflict Serializable

1. Basic Concept
2. Testing (Precedence Graph Method)
3. Conflict Equal to Serial Schedule.

Conflict Equivalance.

S & S' Conflict Equal if all Conflict operation

[R-W, W-R, W-W] are Must be executed in Same Order in S & S'

if Conflict Equal to Any Serial then Conflict serializable

- Conflict Equal
- Same Set of transaction
- Precedence Graph Acyclic.

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

Schedule 1

T_1	T_2
read (A)	
A: = A – 100	
write (A)	
read (B)	
B: = B + 100	
write (B)	
commit	
	read (A) 1900
	temp := A * 0.1
	A := A – temp
	write (A)
	read (B) B=3100
	B := B + temp
	write (B)
	Commit

$S_1 < T_1\ T_2 >$

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) A = 1800
	read (B)
	B := B + temp
	write (B)
	Commit
A=1800 read (A)	
A: = A – 100	
write (A)	
read (B)	
B: = B + 100	
write (B)	
commit	

$S_2 < T_2\ T_1 >$

serial schedule where T_2 is followed by T_1

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

C_1

C1: 1710
+ 3290
5000
Consistent

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

C_2

A: 1900
+ B: 3300
5200
InConsistent

Equivalent Schedule

1. Result Equivalent
2. Conflict Equivalent
3. View Equivalent

Result Equivvalent :

Two Schedule are said to be Result Equivalent if they Produce same final Result for Some initial value of Data

$A = 100$

S_1	S_2
R(A)	R(A)
$A = A + 10$	$A = A * 1.1$
W(A)	W(A)

$A = 100$ | $A = 100$

$A = 110$ | $A = 110$

S_1 & S_2 are Result Equivalent.

Result Equivivalent. :

Two Schedule are Said to be Result Equivalent if they Produce same final Result for some initial value of Data

A = 200

S1	S2
R(A)	R(A)
A=A+10	A=A*1.1
W(A)	W(A)

S1: A=200, A=210

S2: A = 200, A=220.

S1 & S2 are Not Result Equivalent.

Complete Schedule : A Schedule is said to Complete Schedule if last operation of the each transaction is either Commit or Abort Called Complete Schedule

Otherwise Partial Schedule.

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) $T1 \rightarrow T3 \rightarrow T2$

(B) $T2 \rightarrow T1 \rightarrow T3$

(C) $T2 \rightarrow T3 \rightarrow T1$

(D) $T3 \rightarrow T1 \rightarrow T2$

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.11

$T_1 \rightarrow T_2$

Consider a schedule of transactions T_1 and T_2:

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

✓ $R_2(B) - W_1(B)$
✓ $W_2(B) - W_1(B)$
✓ $R_1(C) - W_2(C)$ ✓
✓ $R_2(D) - W_1(D)$

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]

$W_2(C) - R_1(C)$ ✗

(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

(b)
$R_2(B) - W_1(B)$ ✓
$W_2(B) - W_1(B)$ ✓
$R_1(C) - W_2(C)$ ✓
$R_2(D) - W_1(D)$ ✓

$W_1(B) - R_2(B)$
$W_1(B) - W_2(B)$ ✗
$W_1(D) - R_2(D)$

(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

Serializable

1. Conflict Serializable

2. View Serializable

Must be same for Each Data Item..

1. Initial Read
2. Final Write
3. Updated Read [Write-Read sequence]

View Serializability

S & S'

Let S and S' be two schedules with the same set of transactions. S and S' are **view equivalent** if the following three conditions are met, for each data item Q,

Initial Read

1. If in schedule S, transaction T_i reads the initial value of Q, then in schedule S' also transaction T_i must read the initial value of Q.
2. If in schedule S transaction T_i executes **read**(Q), and that value was produced by transaction T_j (if any), then in schedule S' also transaction T_i must read the value of Q that was produced by the same **write**(Q) operation of transaction T_j.

Final Write

3. The transaction (if any) that performs the final **write**(Q) operation in schedule S must also perform the final **write**(Q) operation in schedule S'.

for each Data Item.

1. Initial Read
2. Final Write
3. Write → Read Sequence

S

T_1	T_2	T_3
	W(A)	
		R(A)

S'

T_1	T_2	T_3
	W(A)	
		R(A)

Now Practically Check the

Importance of all 3 Condition one by One.

View Serializability

- **View Serializable Schedule:** View equivalent serial schedule.
- **View Equivalent:** S_1 and S_2 said to be view equivalent.

  Only if

  [1] initial reads of S_1 and S_2 should be same.

@ YES

ⓑ NO

Example

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

S_1

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

S_2

$S_1 \neq S_2$

View Serializability

- **View Serializable Schedule:** View equivalent serial schedule.
- **View Equivalent:** S_1 and S_2 said to be view equivalent.

Only if

[1] initial reads of S_1 and S_2 should be same.

(a) YES

(b) NO

Example

200

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

A = 50000

B = 60000

B = 200

Initial Read on A: T_1

B: T_1

S_1

200

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

B = 200

A = 50000

B = 200

Initial Read on A: T_1

B: Not T_1

S_2

$S_1 \neq S_2$

View Serializability

2. Final updations for every data item should be same in S_1 and S_2

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

S_1

final write on
A: T_3
B: T_3

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

S_2

Final write on
A: T_2
B: T_3.

$S_1 \neq S_2$

View Serializability

2. Final updations for every data item should be same in S_1 and S_2

25000 1lakh 2crore

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

S_1

Final write on A : T_3

25000 1lakh 2crore

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

S_2

Final write on A : T_2

View Serializability

3. Write-Read sequence should also be equal. (Updated Reads should be same

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

S_1

(Write - Read Sequence)
Updated Read
$T_2 \to T_3$

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

S_2

Write - Read Sequence
Updated - Read
$T_1 \to T_3$

$S_1 \neq S_2$

View Serializability

3. Write-Read sequence should also be equal. (Updated Reads should be same

1lakh 25lakh

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

S_1

1lakh 25lakh

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

S_2

$S_1 \neq S_2$

T_1	T_2
R(A)	
W(A)	
	R(A)
	W(A)
R(B)	
W(B)	
	R(B)
C_1	W(B)

1710
3290
5000 = $S_1 \langle T_1, T_2 \rangle$

$\langle T_1, T_2$

Conflict Serializable $\langle T_1, T_2 \rangle$

& it already View Serializable.

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

$\langle T_1, T_2 \rangle$

$\langle T_2, T_1 \rangle$

Dummy.

for each Data Item.

1. Initial Read
2. final Write
3. Write - Read (Updated Read) sequence

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

① Initial Read

A: T_1

B: T_1

② Final Write

A: T_2

B: T_2

③ Write-Read (Updated Read) Sequence

A: $W_1(A) - R_2(A)$: $T_1 \rightarrow T_2$

B: $W_1(B) - R_2(B)$: $T_1 \rightarrow T_2$

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

① Initial Read

A: T_1

B: T_1

② Final Write

A: T_2

B: T_2

③ Updated Read (Write-Read sequence)

A: $W_1(A) - R_2(A)$: $T_1 \rightarrow T_2$

B: $W_1(B) - R_2(B)$: $T_1 \rightarrow T_2$

Its View Serializable $\langle T_1, T_2 \rangle$

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

C_2

A : 1900
B : 3300
+ 5200

Inconsistent

CNC

Cycle Not Conflict

But Now check for View Serializable ?

Not View

Not Serializable

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

$\langle T_1, T_2 \rangle$

Dummy

$\times \langle T_2, T_1 \rangle$

① Initial Read $(B : T_1)$

② final write

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

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

(T_2, T_1)

Initial Read on B: T_2

T_1 Not Read initial Value of B

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

① Initial Read

A: T_1, T_2

B: T_1

② Final write

A: T_1

B: T_2

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

$\langle T_1\ T_2 \rangle$ X

① Initial Read

B: T_1

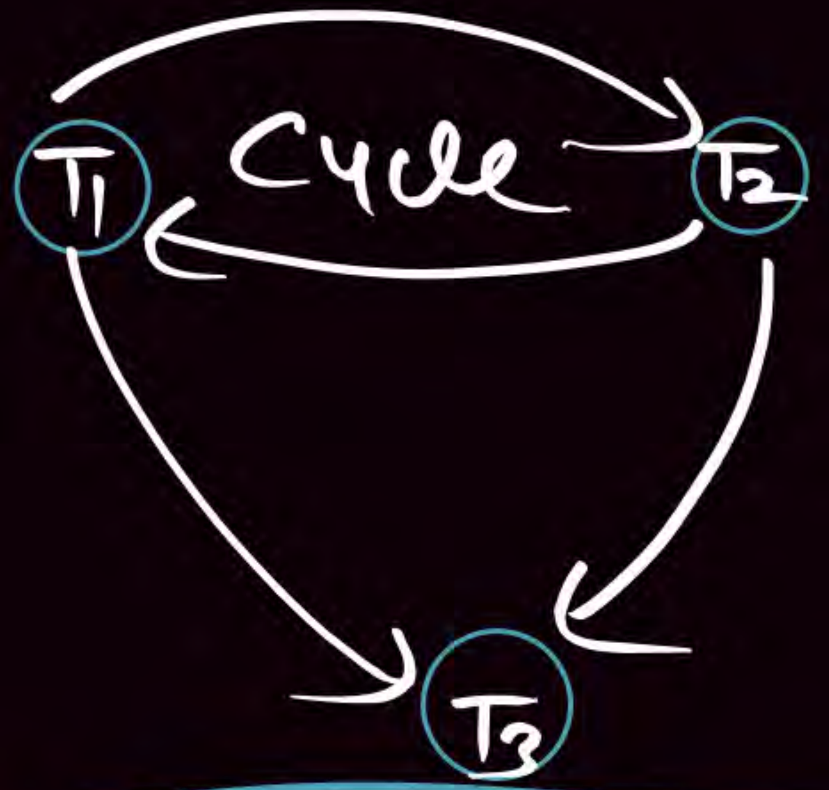
A: Only T_1 Not T_2

final write on

A: T_2

Fail.

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



Cycle Not Conflict

How to Select Dummy

① Initial Read : A : T_1

② Final Write : A : T_3

3 Transaction.

$3! = 6$ Serial Schedule

Dummy

$\langle T_1, T_2, T_3 \rangle$

$\langle T_1\ T_3\ T_2 \rangle$

$\times \langle T_2, T_1, T_3 \rangle$

$\times \langle T_2, T_3, T_1 \rangle$

$\times \langle T_3, T_1, T_2 \rangle$

$\times \langle T_3, T_2, T_1 \rangle$

Blind Write : Write without Reading.

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

(1) Initial Read A : T_1

(2) final Write A : T_3

No Write-Read
(No Updated Read)

Not Conflict
But View serializable

∴ serializable

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

(1) Initial Read A : T_1

(2) Final Write A : T_3

(No Updated Read)

View Serializability (Cont.)

Note

- A schedule S is **view serializable** if it is view equivalent to a serial schedule.

Note

- Every conflict serializable schedule is also view serializable.
- Below is a schedule which is view-serializable but not conflict serializable.

T_{27}	T_{28}	T_{29}
read(Q)		
	write(Q)	
write(Q)		
		write(Q)

Note:

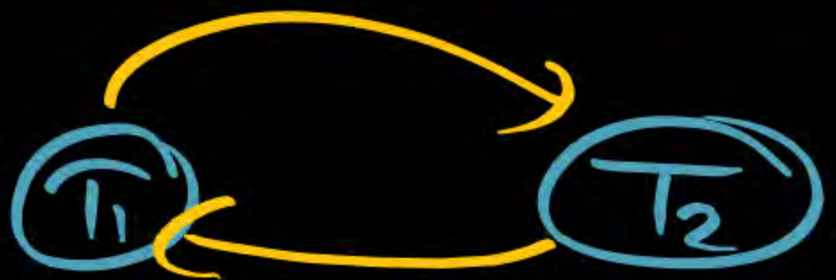
Every view serializable schedule that is not conflict serializable has **blind writes.**

Q.

Dummy

$\langle T_2, T_1 \;\; T_3 \rangle$

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



Cycle Not Conflict

Now Check for View

Q.

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

Not Conflict But View serializable

∴ Serializable

① Initial Read

A : T_2, B : T_2

② Final write on

A : T_3

B : T_1

③ Updated Read (Write - Read) sequence

$T_1 \rightarrow T_3$

$\langle T_2, T_1, T_3 \rangle$

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

① Initial Read on

A : T_2, B : T_2

② Final Write on

A : T_3

B : T_1

③ Updated Read (write - Read) Sequence

$T_1 \rightarrow T_3$

MCQ **Q.13**

Consider the following transactions with data items P and Q initialized to zero:

T_1: read (P);
read (Q);
if P = 0 then Q := Q + 1 ;
write (Q).

T_2: read (Q) ;
read (P);
if Q = 0 then P := P + 1 ;
write (P).

T_1	T_2
r(P)	
	r(Q)
	r(P)
	w(P)
r(Q)	
w(Q)	

Not Conflict

Not View Serializable

Not Serializable

Any non-serial interleaving of T_1 and T_2 for concurrent execution leads to

[GATE-2012-CS: 1M]

A. a serializable schedule

B. a schedule that is not conflict serializable

C. a conflict serializable schedule

D. a schedule for which a precedence graph cannot be drawn

Q.14

Consider the following schedule S of transactions T_1 and T_2:

Which of the following is TRUE about the schedule S? [2004: 2 Marks]

(A) S is serializable only as T_1, T_2

(B) S is serializable only as T_2, T_1

(C) S is serializable both as T_1, T_2 and T_2 T_1

(D) S is not serializable either as T_1 or as T_2

T_1	T_2
Read(A)	
A = A–10	
	Read(A)
	Temp = 0.2 * A
	Write(A)
	Read(B)
Write(A)	
Read(B)	
B = B +10	
Write(B)	
	B = B + Temp
	Write(B)

MCQ **Q.15**

Consider a simple checkpointing protocol and the following set of operations in the log.

(start, T4); (write, T4, y, 2, 3); (start, T1);
(commit, T4); (write, T1, z, 5, 7);
(checkpoint);
(start, T2); (write, T2, x, 1, 9); (commit, T2);
(start, T3); (write, T3, z, 7, 2);

If a crash happens now and the system tries to recover using both undo and redo operations, what are the contents of the undo list and the redo list?

[GATE-2015-CS: 2M]

A) Undo: T3, T1; Redo: T2

B) Undo: T3, T1; Redo: T2, T4

C) Undo: none; Redo: T2, T4, T3, T1

D) Undo: T3, T1, T4; Redo: T2

Problem due to Concurrent Execution:

1. WR [Write - Read] / Uncommitted Read / Dirty Read Problem
2. RW [Read - Write] / Non/un Repeatable Read Problem.
3. WW [Write - Write] / Lost Update Problem.
4. Phantom Tuple Problem

   Incorrect Summary Problem.

Finding Total Number of schedule.

Concurrent Schedule

Serial Schedule.

Non serial Schedule

Finding Total Number of schedule.

$$\text{Concurrent Schedule} \left[\text{Serial Schedule} + \text{Non serial Schedule} \right]$$

m Transaction

Total Number of serial schedule = $m!$ serial schedule.

$$\boxed{\text{Non Serial Schedule} = \text{Total Concurrent Schedule} - \text{Serial Schedule}(m!)}$$

1

T_1 T_2

L_1 L_3

L_2 L_4

2

A I

B II

AB I II

I II AB

AI B II, AI II B

IA II B, IAB II

3

T_1 T_2

0 1

0 1

00 11

11 00

0101, 0110

1010, 1001

6

$L_1 L_2 L_3 L_4$

$L_3 L_4 L_1 L_2$

$L_1 L_3 L_2 L_4$ or $L_1 L_3 L_4 L_2$

$L_3 L_1 L_4 L_2$ or $L_3 L_1 L_2 L_4$

Finding Total Number of concurrent Schedule

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

T_1	T_2
L_1 L_2	L_3 L_4

T_1	T_2
0 0	1 1

$$\left.\begin{array}{l} L_1L_2L_3L_4 \\ L_3L_4L_1L_2 \end{array}\right\}$$

$L_1L_3L_2L_4$ (or) $L_1L_3L_4L_2$

$L_3L_1L_4L_2$(or)$L_3L_1L_2L_4$

(1)

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

$S_1<T_1T_2>$

(2)

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

$S_2<T_2T_1>$

(3)

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

(4)

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

(5)

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

(6)

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

T_1 - n_1 operation

T_2 — n_2 operation

$$\boxed{\text{Total \# Concurrent Schedule} = \frac{(n_1+n_2)!}{(n_1)!\,(n_2)!}}$$

$T_1 \rightarrow$ 2 operation

$T_2 \rightarrow$ 2 operation

$$\text{Total Concurrent} = \frac{(2+2)!}{(2!)\,(2!)} = \frac{4!}{2\times 2}$$

$$= \frac{\cancel{4}\times 3\times 2}{\cancel{2\times 2}} = 6 \text{ Concurrent}$$

Serial schedule $= 2! = 2$

Non Serial $= 6-2 = 4$

$$\boxed{\text{Total \# Concurrent Schedule} = \frac{(n_1+n_2)!}{(n_2)!(n_2)!}}$$

$T_1 \rightarrow n_1$ operation
2 operation

$T_2 \rightarrow n_2$ operation
2 operation

$$= \frac{(2+2)!}{(2)!(2)!} = \frac{4\times3\times2}{2\times2} = 6$$

$$\boxed{\text{Total Concurrent} = 6}$$

Total non serial Schedule = Total Concurrent - Serial schedule(m!)
m: # of transaction

2

$$= 6 - 2$$

$$\boxed{\text{Serial} = 2}$$

$$\boxed{\text{Total non Serial} = 4}$$

6-2=4

NOTE:

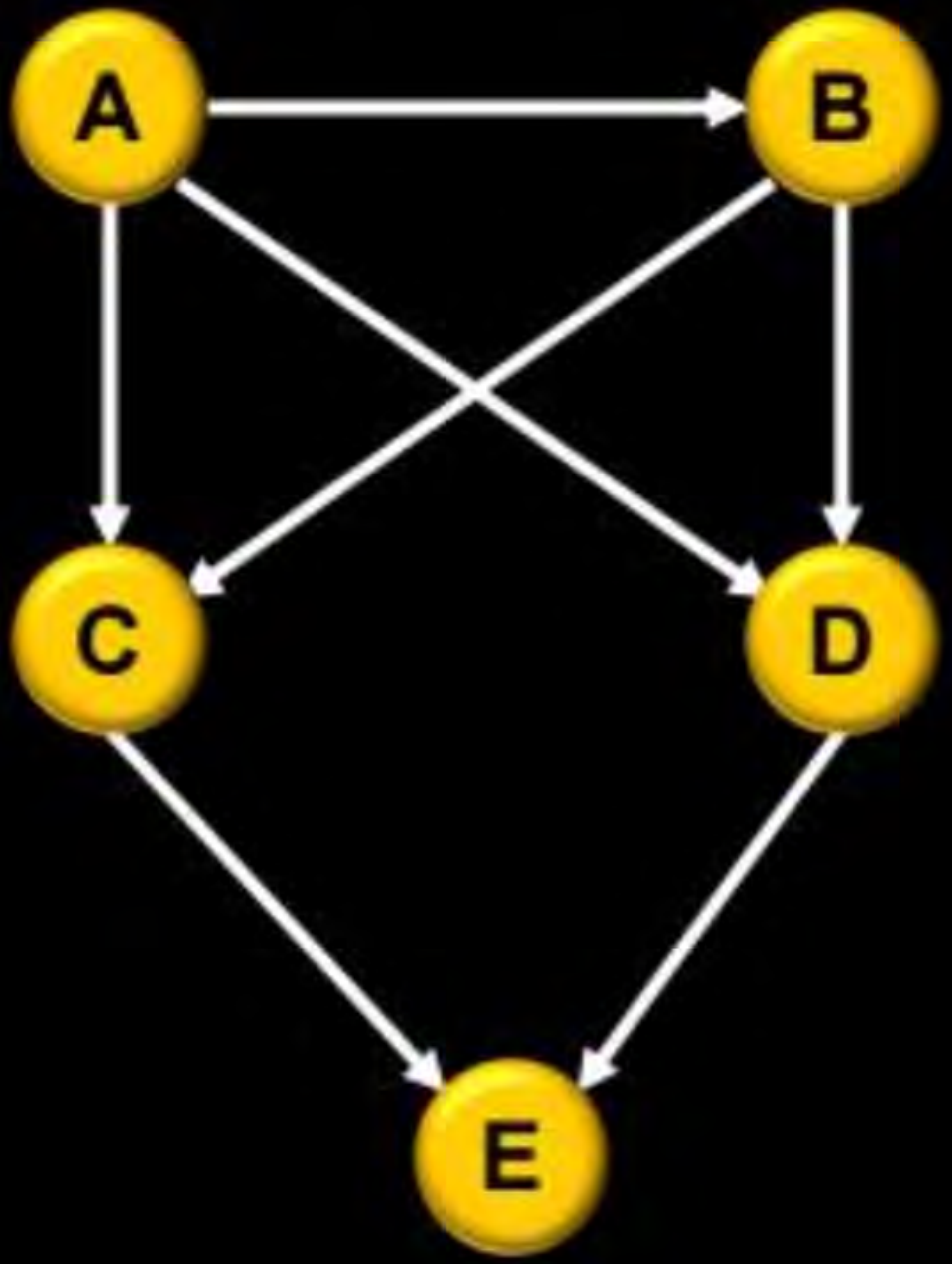
The Number of Concurrent schedule that can be formed Over m transaction having $n_1\ n_2\ n_3 \ldots n_m$ operation respectively

Total # of Concurrent Schedule	$= \dfrac{(n_1+n_2+n_3+\cdots n_m)!}{(n_1!)(n_2!)(n_2!)\ldots(n_m!)}$
Total # of Non Serial Schedule	$= \dfrac{(n_1+n_2+n_3+\cdots n_m)!}{(n_1!)(n_2!)(n_2!)\ldots(n_m!)} - m!$

Serial Schedule

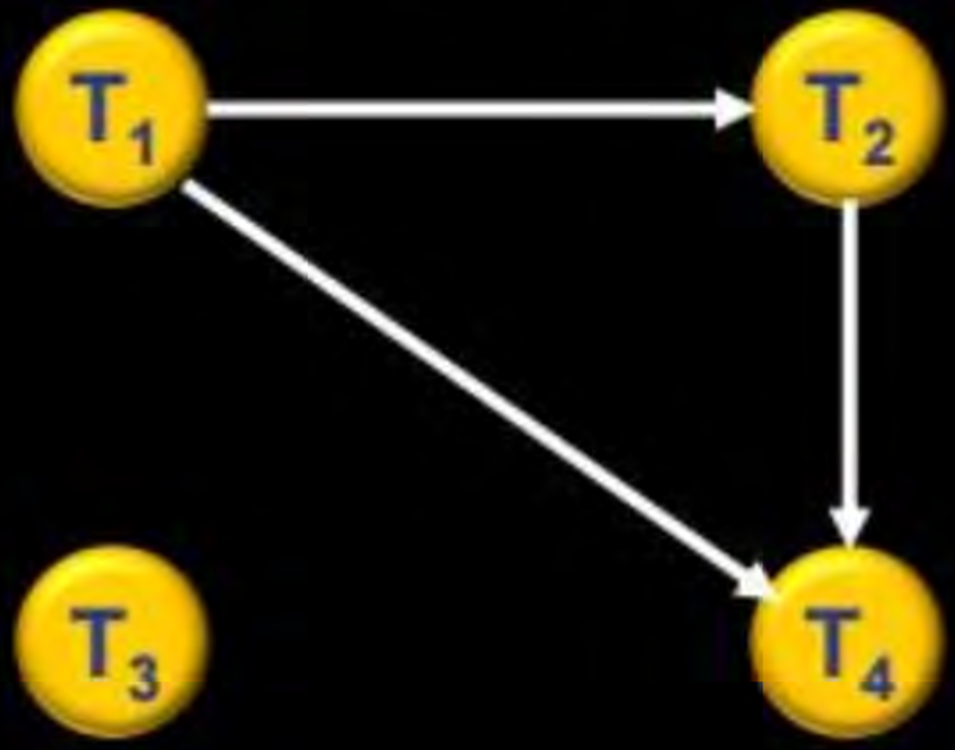
Topological Sorting

Q.



Topological Sorting

Q.



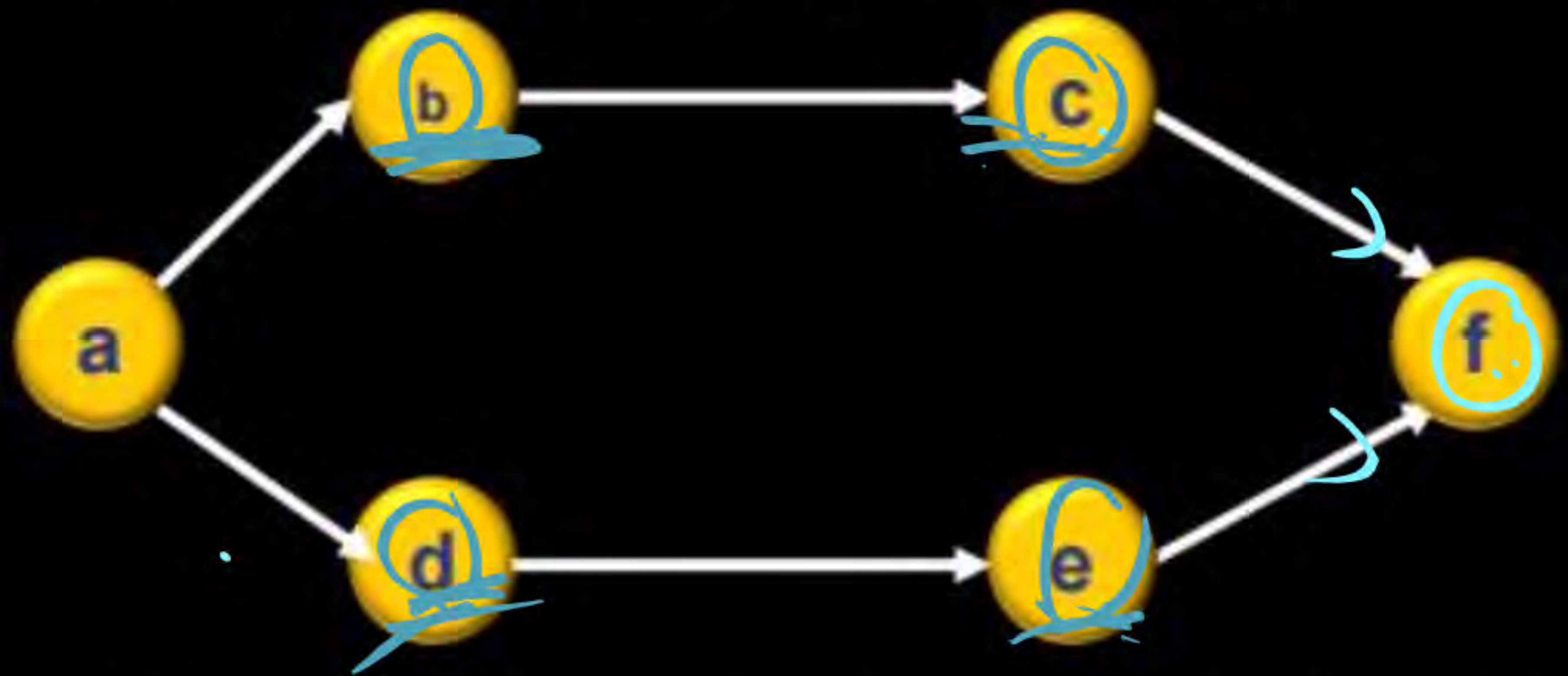
Topological Sorting

Q. $R_4(x)\ R_2(x)\ R_3(x)\ W_1(y)\ W_2(x)\ R_3(y)\ W_2(y)$

Topological Sorting

Q. Consider the following directed graph:

6



a b c d e f
a d e b c f
a b d c e f
a b d e c f
a d b e c f
a d b c e f

The number of different topological ordering of the vertices of the graph is ___6___.

[MCQ: 2016]

THANK YOU!