Transaction & Concurrency Control - Part V

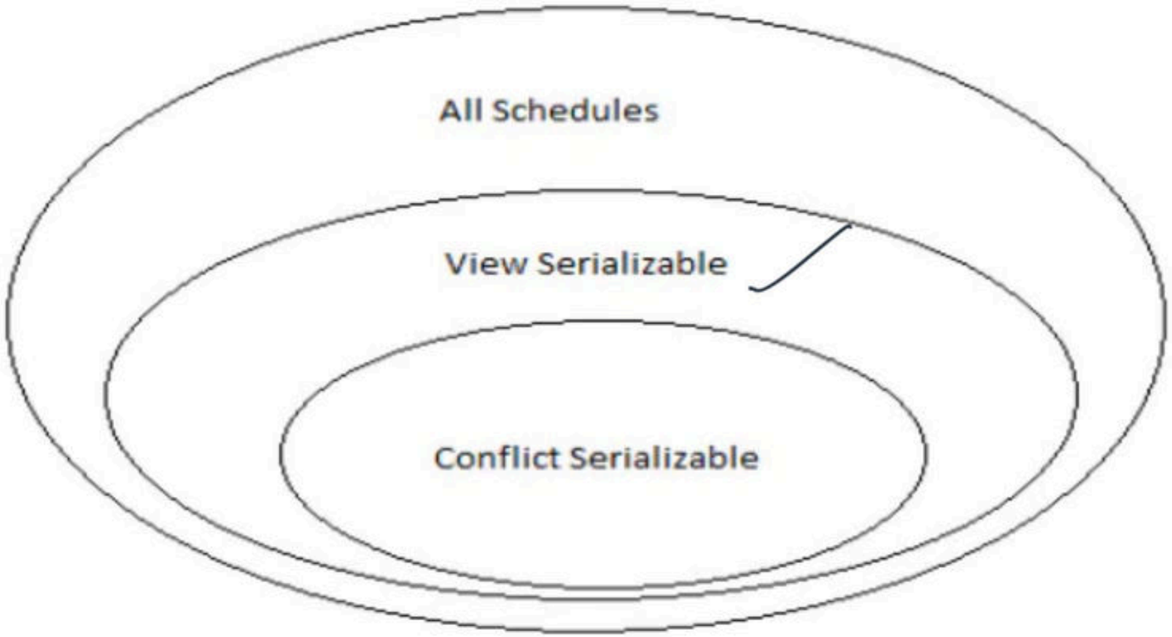
Complete Course on DBMS for GATE

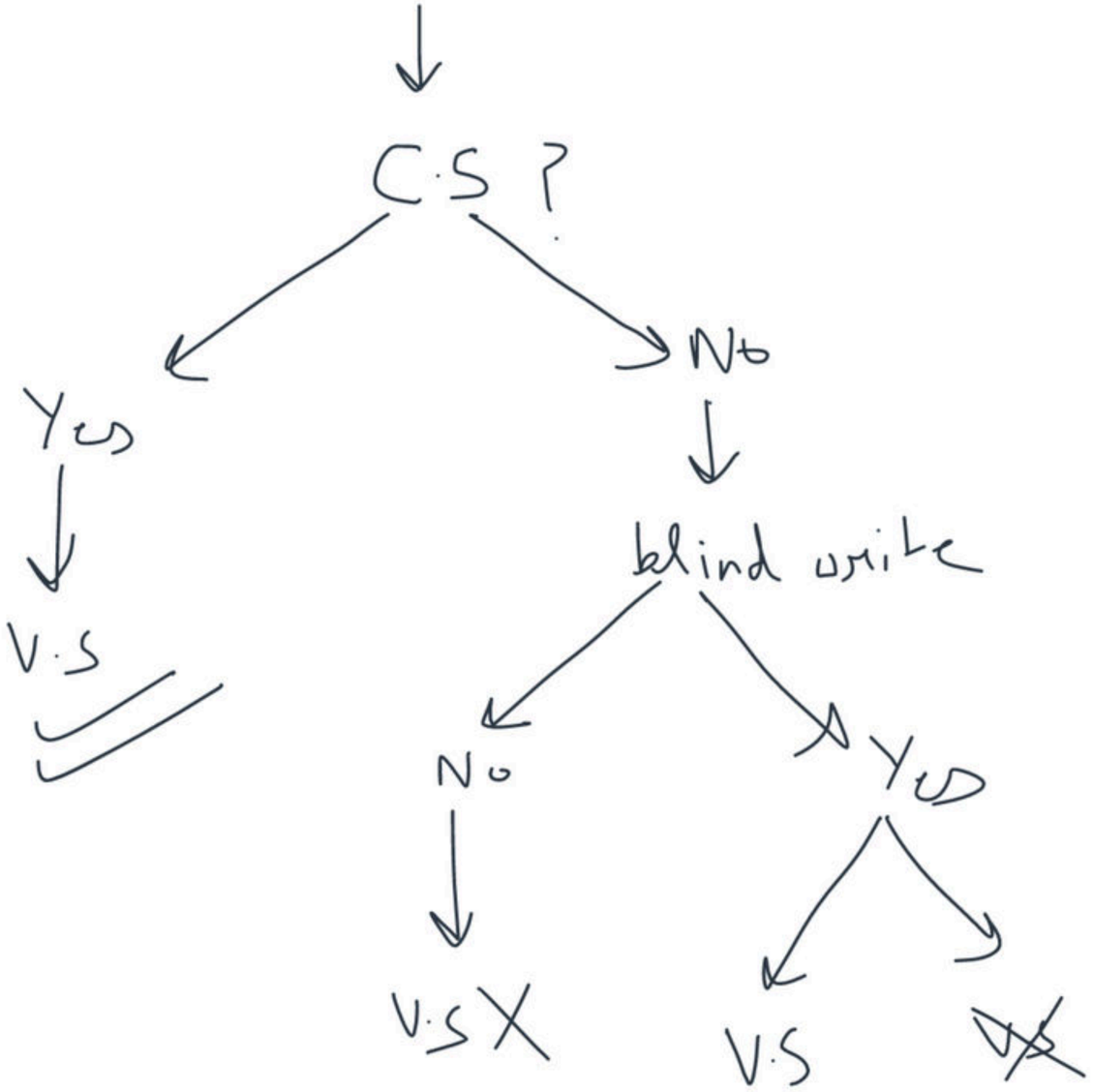
Sanchit Jain • Lesson 7 • Mar 24, 2022

VIEW SERIALIZABLE

- If a schedule is not conflict serializable, still it can be consistent, so let us study a weaker form of serializability called View serializability, and even if a schedule is view serializable still it can be consistent.

- If a schedule is conflict serializable then it will also be view serializable, so we must check view serializability only if a schedule is not conflict serializable.

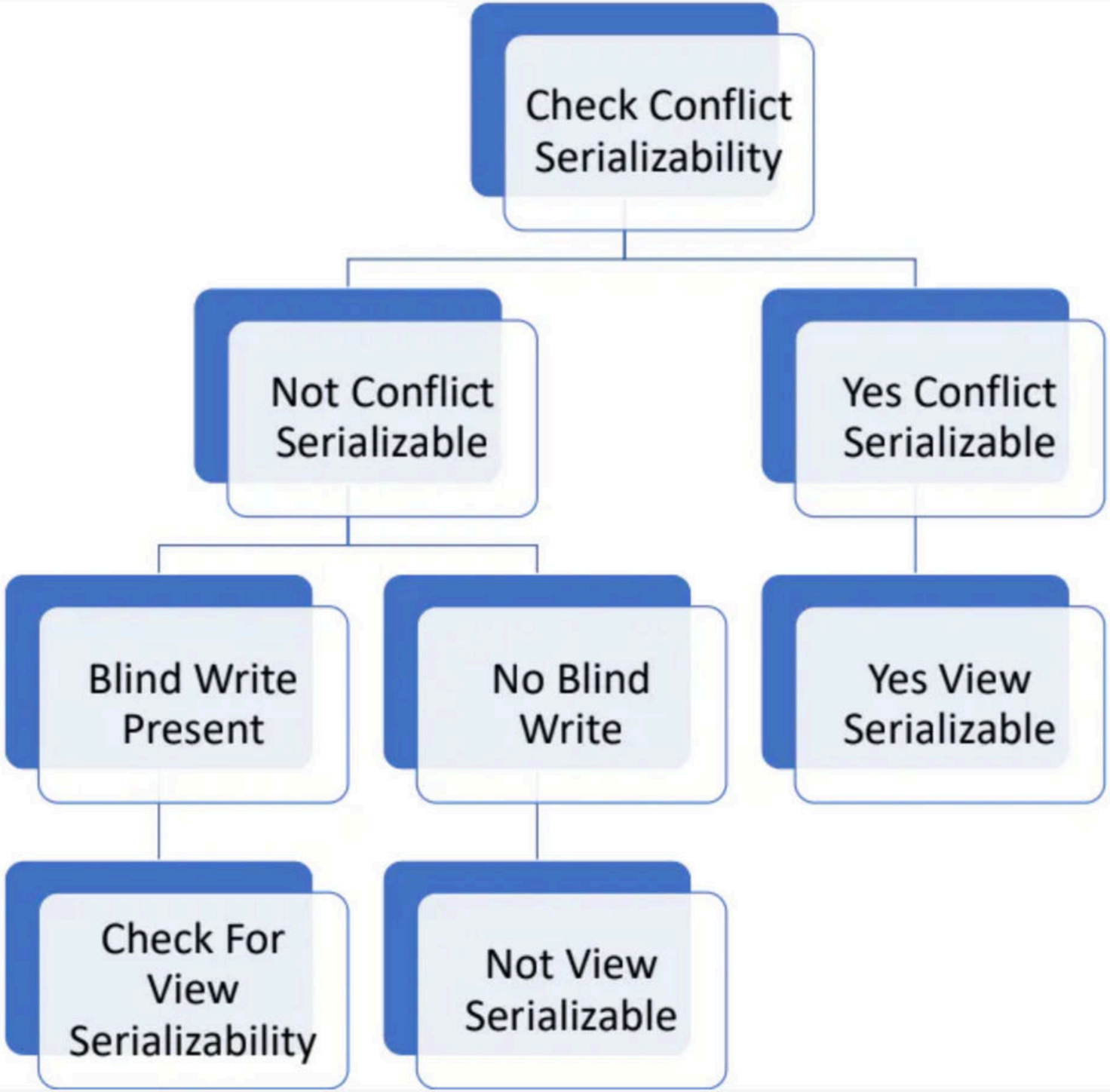



C.S ?
Yes
No
V.S
blind write
No
Yes
V.S X
V.S

- If a schedule is not conflict serializable then it must have at least one blind write to be eligible for view serializable.

- i.e. if a schedule is not conflict serializable and it does not contain any blind write then it can never be view serializable, but if not conflict serializable and have blind write then may or may not be view serializable.

- If a schedule is not conflict serializable and if there exist a blind write.
- First tabulate all serial schedules possible. Then check one by one whether given schedule is view equivalent to any of the serial schedule.
- If yes then schedule is view serializable otherwise not.


Check Conflict Serializability
Not Conflict Serializable
Yes Conflict Serializable
Blind Write Present
No Blind Write
Yes View Serializable
Check For View Serializability
Not View Serializable

- Two schedules S and S' are view equivalent, if they satisfy following conditions –
  - For each data item Q, if the transaction T_i reads the initial value of Q in schedule S , then then the transaction T_i must, in schedule S' ,also read the initial value of Q.
  - If a transaction T_i in schedule S reads any data item Q, which is updated by transaction T_j, then a transaction T_i must in schedule S' also read data item Q updated by transaction T_j in schedule S'.
  - For each data item Q, the transaction (if any) that performs the final write(Q) operation in schedule S, then the same transaction must also perform final write(Q) in schedule S'.

S'

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

A B

S''

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

- Complexity wise finding the schedule is view serializable or not is a **NP- complete** problem.

View Serializable

3! = 6

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

Schedule A		
T3	T4	T6
~~read(Q)~~		
	write(Q)	
write(Q)		
		~~write(Q)~~

T_3	T_4	T_6
r(Q)		
w(Q)		

T_3	T_4	T_6
	w(Q)	
		w(Q)
r(Q)		
w(Q)		

T_3	T_4	T_6
	w(Q)	
		~~w(Q)~~

100,000!

3 4 6
3 6 4
4 3 6
4 6 3
6 3 4
6 4 3

View Serializable

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

Schedule A				Serial schedule <T3,T4,T6>		
T3	T4	T6		T3	T4	T6
read(Q)				read(Q)		
	write(Q)			write(Q)		
write(Q)					write(Q)	
		write(Q)				write(Q)

- ***BLIND WRITES***- In the above example, transaction T_4 and T_6 perform write operation on data item Q without accessing (reading the data item), such updation without knowing/accessing previous value of data item, are called Blind updation or **BLIND WRITE.**

Break

Q A Schedule that is not conflict serializable and contains at least one blind write then the schedule is

a) Always view serializable

b) Always non-serializable

c) May be View serializable

d) None of the above

Q Which of the following statements (s) is/are TRUE?

S_1: All view serializable schedules are also conflict serializable.

S_2: All conflict serializable schedules are also view serializable.

S_3: If a schedule is not conflict serializable then it is not view serializable

S_4: If a schedule is not view serializable then it is not conflict serializable.

a) S_1 and S_2 only

b) S_2 and S_3 only

c) S_2 and S_4 only

d) S_1 and S_3 only

Q Consider the following schedule 'S' with three transactions.

S: $R_1(B); R_3(C); R_1(A); W_2(A); W_1(A), W_2(B); W_3(A); W_1(B); W_3(B), W_3(C)$

Which of the following is TRUE with respect to the above schedule?

a) It is conflict serializable with sequence $[T_1, T_2\ T_3]$

b) It is conflict serializable with sequence $[T_2, T_1\ T_3]$

c) It is view serializable but not conflict serializable

d) It is neither conflict serializable nor view serializable

Q Consider the following schedule 'S' with three transactions.

S: $R_1(B); R_3(C); R_1(A); W_2(A); W_1(A), W_2(B); W_3(A); W_1(B); W_3(B), W_3(C)$

Which of the following is TRUE with respect to the above schedule?

a) It is conflict serializable with sequence $[T_1, T_2\ T_3]$

b) It is conflict serializable with sequence $[T_2, T_1\ T_3]$

c) It is view serializable but not conflict serializable

d) It is neither conflict serializable nor view serializable

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

Q The following schedule S is having 4 transactions and is executed concurrently. The order of their operations is given below.

S: R_1 (x); R_2 (y); W_2 (x); W_3 (z); R_4 (z); R_3 (x); W_3 (y); W_1 (x); W_2 (y) W_3 (x); R_4(x); W_4 (y); $commit_1$; $commit_2$; $commit_3$; $commit_4$;

The schedule is

a) Conflict serializable with sequence [T_1 T_2 T_3 T_4]

b) Conflict serializable with sequence [T_2 T_1 T_3 T_4]

c) View serializable but not conflict serializable

d) Neither conflict serializable nor view serializable

Q The following schedule S is having 4 transactions and is executed concurrently. The order of their operations is given below.

S: R_1 (x); R_2 (y); W_2 (x); W_3 (z); R_4 (z); R_3 (x); W_3 (y); W_1 (x); W_2 (y) W_3 (x); R_4(x); W_4 (y); $commit_1$; $commit_2$; $commit_3$; $commit_4$;

The schedule is

a) Conflict serializable with sequence [T_1 T_2 T_3 T_4]

b) Conflict serializable with sequence [T_2 T_1 T_3 T_4]

c) View serializable but not conflict serializable

d) Neither conflict serializable nor view serializable

S			
T_1	T_2	T_3	T_4
R(X)			
	R(Y)		
	W(X)		
		W(Z)	
			R(Z)
		R(X)	
		W(Y)	
W(X)			
	W(Y)		
		W(X)	
			R(X)
			W(Y)
C			
	C		
		C	
			C

Break

NON- RECOVERABLE SCHEDULE

- A schedule in which for each pair of transaction T_i and T_j, such that if T_j reads a data item previously written by T_i, then the commit or abort operation of T_i appears before T_j. Such a schedule is called Non- Recoverable schedule.

S	
T_1	T_2
R(X)	
W(X)	
	R(X)
	C
C	

C

RECOVERABLE SCHEDULE

- A schedule in which for each pair of transaction T_i and T_j, such that if T_j reads a data item previously written by T_i, then the commit or abort of T_i must appear before T_j. Such a schedule is called Recoverable schedule.

S	
T_1	T_2
R(X)	
W(X)	
	R(X)
C	
	C

Q Consider the following schedule S of transactions T_1, T_2, T_3, T_4 **(GATE - 2014) (2 Marks)**
Which one of the following statements is CORRECT?

(A) S is conflict-serializable but not recoverable
(B) S is not conflict-serializable but is recoverable
(C) S is both conflict-serializable and recoverable
(D) S is neither conflict-serializable nor is it recoverable

T1	T2	T3	T4
	Reads(X)		
		Writes(X) Commit	
Writes(X) Commit			
	Writes(Y) Reads(Z) Commit		
			Reads(X) Reads(Y) Commit

Q.24 Let S be the following schedule of operations of three transactions T_1, T_2 and T_3 in a relational database system:

(GATE-2021)

$$R_2(Y),\ R_1(X),\ R_3(Z),\ R_1(Y),\ W_1(X),\ R_2(Z),\ W_2(Y),\ R_3(X),\ W_3(Z)$$

Consider the statements P and Q below:

P : S is conflict-serializable.

Q : If T_3 commits before T_1 finishes, then S is recoverable.

Which one of the following choices is correct?

(a) P is true and Q is false.

(b) Both P and Q are false.

(c) Both P and Q are true.

(d) P is false and Q is true.

Break

CASCADING ROLLBACK

- It is a phenomenon, in which a single transaction failure leads to a series of transaction rollbacks, is called cascading rollback. Even if the schedule is recoverable the, the commit of transaction may lead lot of transaction to rollback.

- Cascading rollback is undesirable, since it leads to undoing of a significant amount of work. Uncommitted reads are not allowed in cascade less schedule.

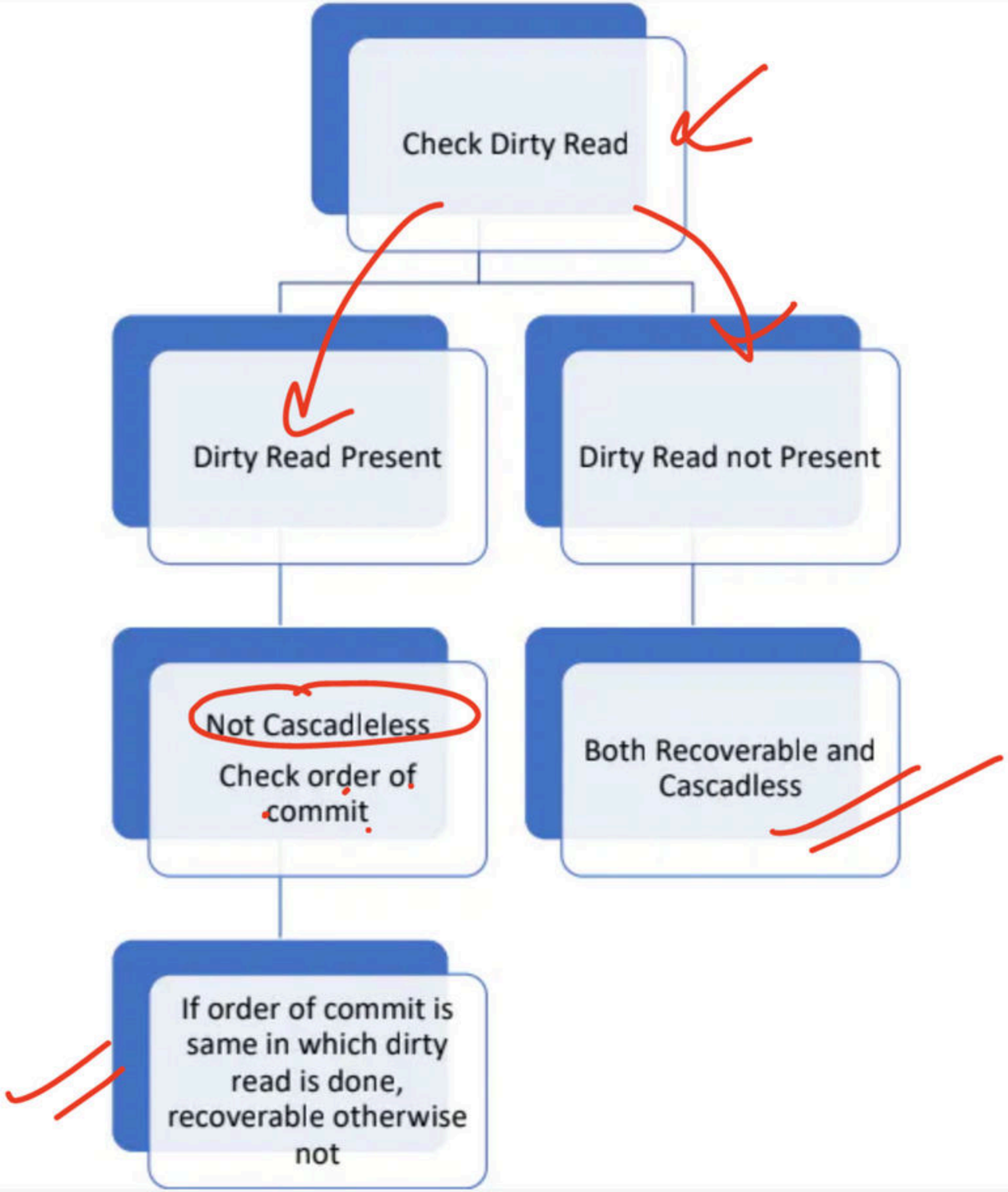
S		
T_1	T_2	T_3
R(X)		
W(X)		
	R(X)	
	W(X)	
		R(X)
C		
	C	
		C

CASCADELESS SCHEDULE

- To avoid cascading rollback, cascade less schedule are used.

- A schedule in which for each pair of transactions T_i and T_j, such that if T_j reads a data item previously written by T_i then the commit or abort of T_i must appear before read operation of T_j. Such a schedule is called cascade less schedule.

S		
T_1	T_2	T_3
R(X)		
W(X)		
C		
	R(X)	
	W(X)	
	~~C~~	
		R(X)
		C


Check Dirty Read
Dirty Read Present
Dirty Read not Present
Not Cascadleless
Check order of commit
Both Recoverable and Cascadless
If order of commit is same in which dirty read is done, recoverable otherwise not

Q Consider the following partial Schedule S involving two transactions T_1 and T_2. Only the read and the write operations have been shown. The read operation on data item P is denoted by read(P) and the write operation on data item P is denoted by write(P).

Suppose that the transaction T_1 fails immediately after time instance 9. Which one of the following statements is correct? **(GATE-2015) (2 Marks)**

(A) T_2 must be aborted and then both T_1 and T_2 must be re-started to ensure transaction atomicity

(B) Schedule S is non-recoverable and cannot ensure transaction atomicity

(C) Only T_2 must be aborted and then re-started to ensure transaction atomicity

(D) Schedule S is recoverable and can ensure atomicity and nothing else needs to be done

Time	Transaction-id	
	T1	*T2*
1	*read(A)*	
2	*write(A)*	
3		*read(C)*
4		*write(C)*
5		*read(B)*
6		*write(B)*
7		*read(A)*
8		*commit*
9	*read(B)*	

Break

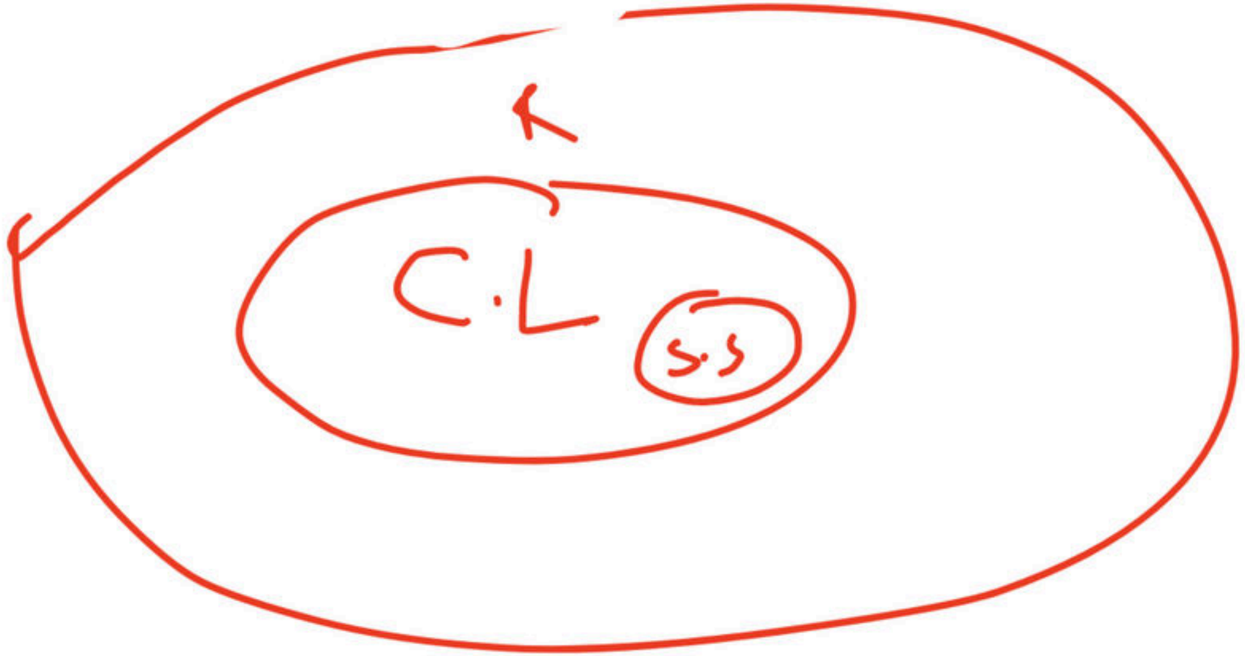
Strict Schedule

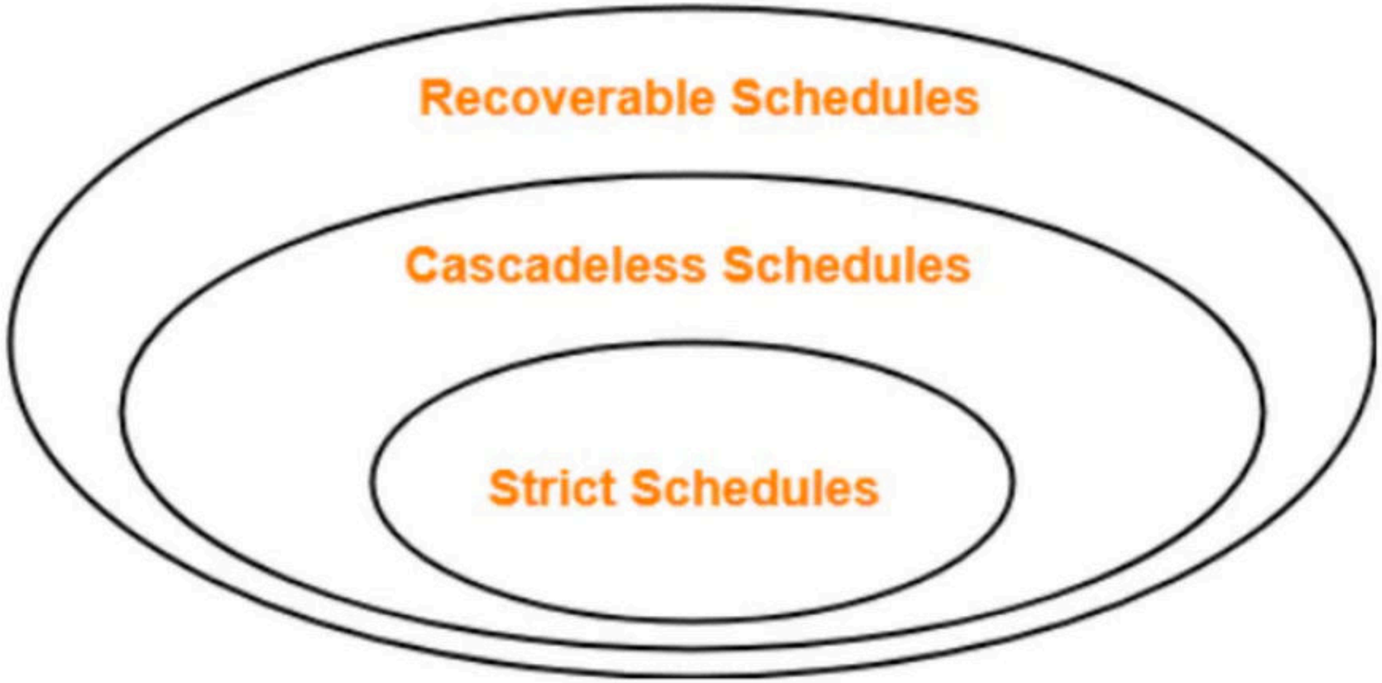
- A schedule in which for each pair of transactions T_i and T_j, such that if T_j reads a data item previously written by T_i then the commit or abort of T_i must appear before read and write operation of T_j.

S_1	
T_1	T_2
R(a)	
W(a)	
	W(a)
C	
	R(a)
	C

S_2	
T_1	T_2
R(a)	
W(a)	
C	
	W(a)
	R(a)
	C

S_3	
T_1	T_2
R(a)	
	R(b)
W(a)	
	W(b)
C	
	R(a)
	C


K
C.L
S.S


Recoverable Schedules
Cascadeless Schedules
Strict Schedules

Q Consider the following database schedule with two transactions, T_1 and T_2. $S = r_2(X); r_1(X); r_2(Y); w_1(X); r_1(Y); w_2(X); a_1; a_2$. where $r_i(Z)$ denotes a read operation by transaction T_i on a variable Z, $w_i(Z)$ denotes a write operation by T_i on a variable Z and a_i denotes an abort by transaction T_i. Which one of the following statements about the above schedule is **TRUE**? **(GATE-2016) (1 Marks)**

(a) S is non-recoverable

(b) S is recoverable, but has a cascading abort

(c) S does not have a cascading abort

(d) S is strict

Q Consider the following database schedule with two transactions, T_1 and T_2. $S = r_2(X); r_1(X); r_2(Y); w_1(X); r_1(Y); w_2(X); a_1; a_2$. where $r_i(Z)$ denotes a read operation by transaction T_i on a variable Z, $w_i(Z)$ denotes a write operation by T_i on a variable Z and a_i denotes an abort by transaction T_i. Which one of the following statements about the above schedule is **TRUE**? **(GATE-2016) (1 Marks)**

(a) ~~S is non-recoverable~~ 14

(b) ~~S is recoverable, but has a cascading abort~~ 48

(c) S does not have a cascading abort 26

(d) ~~S is strict~~ 11

S	
T_1	T_2
	R(x)
R(X)	
	R(Y)
W(X)	
R(Y)	
	W(X)
a_1	
	a_2

Q Consider the following schedules:

S_1: $R_1(x)$ $W_1(x)$ $R_1(y)$ $R_2(x)$ $W_2(x)$ C_2, C_1;

S_2: $R_2(x)$ $W_2(x)$ $R_1(y)$ $R_1(x)$ $W_2(x)$ C_2, C_1;

Which of the following is true?

a) Both S_1 and S_2 are recoverable

b) S_1 is recoverable but S_2 is not

c) S_2 is recoverable but S_1 is not

d) Both schedule are not Recoverable

Q Consider the following schedules:

S_1: $R_1(x)\ W_1(x)\ R_1(y)\ R_2(x)\ W_2(x)\ C_2,\ C_1;$

S_1	
T_1	T_2
R(x)	
W(x)	
R(y)	
	R(x)
	W(x)
	C
C	

S_2: $R_2(x)\ W_2(x)\ R_1(y)\ R_1(x)\ W_2(x)\ C_2,\ C_1;$

S_2	
T_1	T_2
	R(x)
	W(x)
R(y)	
R(x)	
	W(x)
	C
C	

Which of the following is true?

a) Both S_1 and S_2 are recoverable

b) S_1 is recoverable but S_2 is not

c) S_2 is recoverable but S_1 is not

d) Both schedule are not Recoverable

Q Consider the following schedule 'S'.

S: $r_1(X); r_2(Z); r_3(X); r_1(Z); r_2(Y); r_3(Y); w_1(X); c_1; w_2(Z); w_3(Y); w_2(Y); c_3; c_2;$

The schedule 'S' is

a) Recoverable

b) Cascade less

c) recoverable and cascade-less

d) None

Q Consider the following schedule 'S'.

S: $r_1(X); r_2(Z); r_3(X); r_1(Z); r_2(Y); r_3(Y); w_1(X); c_1; w_2(Z); w_3(Y); w_2(Y); c_3; c_2;$

The schedule 'S' is

a) Recoverable

b) Cascade less

c) recoverable and cascade-less

d) None

S		
T_1	T_2	T_3
R(X)		
	R(Z)	
		R(X)
R(Z)		
	R(Y)	
		R(Y)
W(X)		
C		
	W(Z)	
		W(Y)
	W(Y)	
		C
	C	

Q Consider the given schedule

$R_1(X)$, $R_2(Z)$, $R_1(Z)$, $R_3(X)$, $R_3(Y)$, $W_1(X)$, $Commit_1$, $W_3(Y)$, $Commit_3$, $R_2(Y)$, $W_2(Z)$, $W_2(Y)$, $Commit_2$. The given schedule is

a) Recoverable only

b) Cascade less only

c) recoverable and cascade-less

d) None

Q Consider the given schedule

$R_1(X)$, $R_2(Z)$, $R_1(Z)$, $R_3(X)$, $R_3(Y)$, $W_1(X)$, $Commit_1$, $W_3(Y)$, $Commit_3$, $R_2(Y)$, $W_2(Z)$, $W_2(Y)$, $Commit_2$. The given schedule is

a) Recoverable only

b) Cascade less only

c) recoverable and cascade-less

d) None

S		
T_1	T_2	T_3
R(X)		
	R(Z)	
R(Z)		
		R(X)
		R(Y)
W(X)		
C		
		W(Y)
		C
	R(Y)	
	W(Z)	
	W(Y)	
	C	

Q Consider the following schedule

S; $R_2(x)$, $w_2(x)$, $R_3(y)$, $R_1(x)$, $R_1(y)$, $w_1(x)$, $w_3(y)$, $R_3(x)$, $R_1(y)$, C_3, C_2, C_1;

The above schedule is

a) Recoverable but not cascade less

b) Recoverable and cascade less but not strict

c) Recoverable, cascade less and also strict

d) Not recoverable

Q Consider the following schedule

S: $R_2(x)$, $w_2(x)$, $R_3(y)$, $R_1(x)$, $R_1(y)$, $w_1(x)$, $w_3(y)$, $R_3(x)$, $R_1(y)$, C_3, C_2, C_1;

The above schedule is

a) Recoverable but not cascade less

b) Recoverable and cascade less but not strict

c) Recoverable, cascade less and also strict

d) Not recoverable

S		
T_1	T_2	T_3
	R(X)	
	W(X)	
		R(Y)
R(X)		
R(Y)		
W(X)		
		W(Y)
		R(X)
R(Y)		
		C
	C	
C		

Q Which of the following statement is/are correct

a) Every view serializable schedule is conflict serializable

b) Every strict schedule is conflict serializable

c) Every conflict serializable schedule is cascade less

d) None of the above

Break

Q Two concurrent executing transactions T1 and T2 are allowed to update same stock item say 'A' in an uncontrolled manner. In such scenario. Following problems may occur:

(a) Dirty read problem
(b) Lost update problem
(c) Transaction failure
(d) Inconsistent database state

Which of the following option is correct if database system has no concurrency module and allows concurrent execution of above two transactions? **[Asked in L&T Infotech (LTI) 2021]**

A) (a), (b) and (c) only
B) (c) and (d) only
C) (a) and (b) only
D) (a), (b) and (d) only

Ans : D

Solution

Problems that can occur during concurrent execution of two transactions in an uncontrolled manner:

a) Lost update problem: Lost update problem occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of some database item incorrect.

b) Dirty read problem: It occurs when a transaction is allowed to retrieve or update a record that has been updated by another transaction. Because it has not yet been committed, there is always a possibility that it will never be committed but rather rollback.

c) Inconsistent database state: It occurs when a transaction reads several values, but a second transaction updates some of these values during the execution of first. This causes an inconsistency. For example: if one transaction is calculating an aggregate summary function on a number of records while other transactions are updating some of records.

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

(start, T_4);
(write, T_4, y, 2, 3);
(start, T_1);
(commit, T_4);
(write, T_1, z, 5, 7);
(checkpoint);
(start, T_2);
(write, T_2, x, 1, 9);
(commit, T_2);
(start, T_3);
(write, T_3, 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) (2 Marks)**

(A) Undo: T_3, T_1; Redo: T_2

(B) Undo: T_3, T_1; Redo: T_2, T_4

(C) Undo: none; Redo: T_2, T_4, T_3; T_1

(D) Undo: T_3, T_1, T_4; Redo: T_2

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

(start, T_4);
(write, T_4, y, 2, 3);
(start, T_1);
(commit, T_4);
(write, T_1, z, 5, 7);
(checkpoint);
(start, T_2);
(write, T_2, x, 1, 9);
(commit, T_2);
(start, T_3);
(write, T_3, z, 7, 2);

S			
T_1	T_2	T_3	T_4
			START
			(write, y, 2, 3)
START			
			COMMIT
(write, z, 5, 7)			
(checkpoint)	(checkpoint)	(checkpoint)	(checkpoint)
	START		
	(write, x, 1, 9)		
	COMMIT		
		START	
		(write, 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) (2 Marks)**

(A) Undo: T_3, T_1; Redo: T_2
(B) Undo: T_3, T_1; Redo: T_2, T_4
(C) Undo: none; Redo: T_2, T_4, T_3; T_1
(D) Undo: T_3, T_1, T_4; Redo: T_2

Q.18 Suppose a database system crashes again while recovering from a previous crash. Assume checkpointing is not done by the database either during the transactions or during recovery.

(Gate-2021) (1 Marks)

Which of the following statements is/are correct?

(a) The same undo and redo list will be used while recovering again.

(b) The system cannot recover any further.

(c) The database will become inconsistent.

(d) All the transactions that are already undone and redone will not be recovered again.