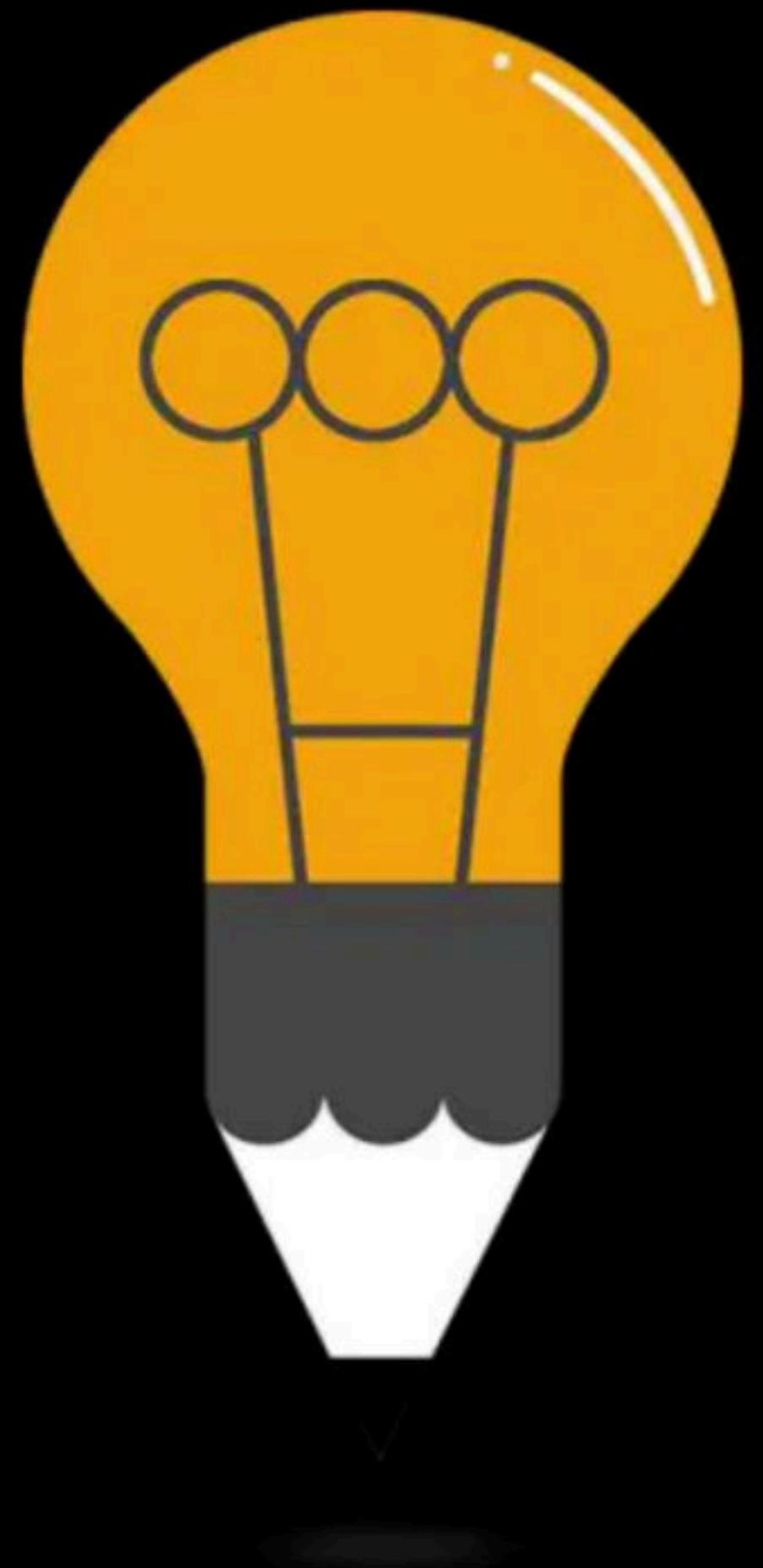




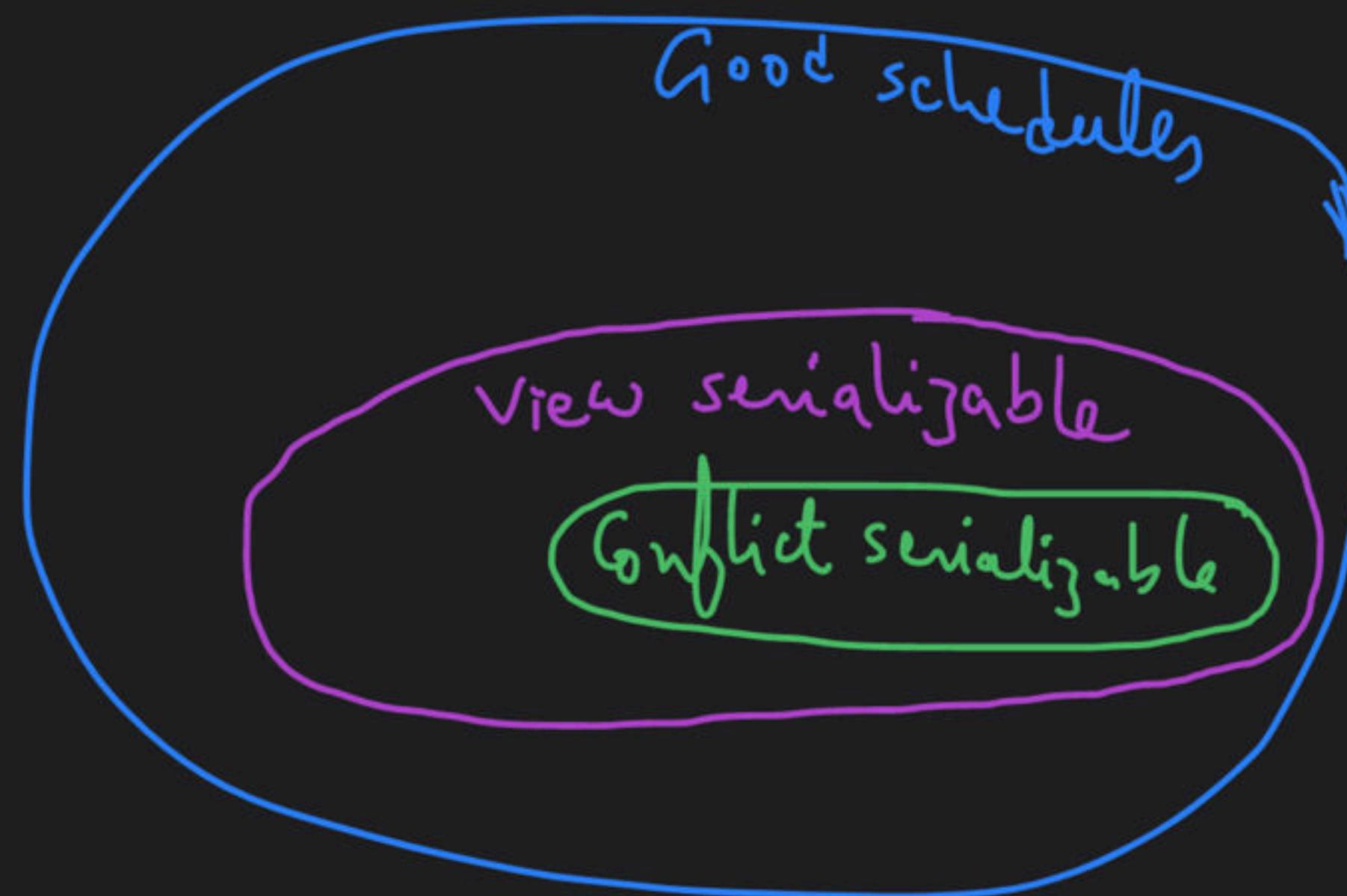
Transaction & Concurrency Control: Part III

Complete Course on Database Management System



DBMS **Recoverability & Locking Protocols**

By: Vishvadeep Gothi



CSC :-

T_1	T_2
$\omega(x)$	$\omega(x)$
$\omega(x)$	$\omega(x)$
$\omega(x)$	$\omega(x)$

T_1	T_2
$\omega(x)$	$\omega(x)$
$\omega(x)$	$\omega(x)$

not conflict serializable

but view serializable

$T_2 \rightarrow T_1$

no. of transactions = n

after graph is given

⇒ To find cycle
↓

Run-time complexity ⇒ $n + |E|$

$$\leq n^2$$

view serializability

assume to find that 2 schedules
are view equivalent or not.

Time ⇒ k

Run time comp. ⇒ $O(k * n!)$

↓
NP hard problem

Recoverability

Recoverability

Recoverable schedule:

When no any committed transaction should be rolled back.

Recoverable Schedule

T1	T2
R(X)	
X=X+2	
W(X)	
	R(X) 12
	X=X+3 15
	W(X)
	Commit
failed	

$$\begin{aligned}x &= t_0 & 10 \\&t_2 \\&15\end{aligned}$$

not-recoverable schedule

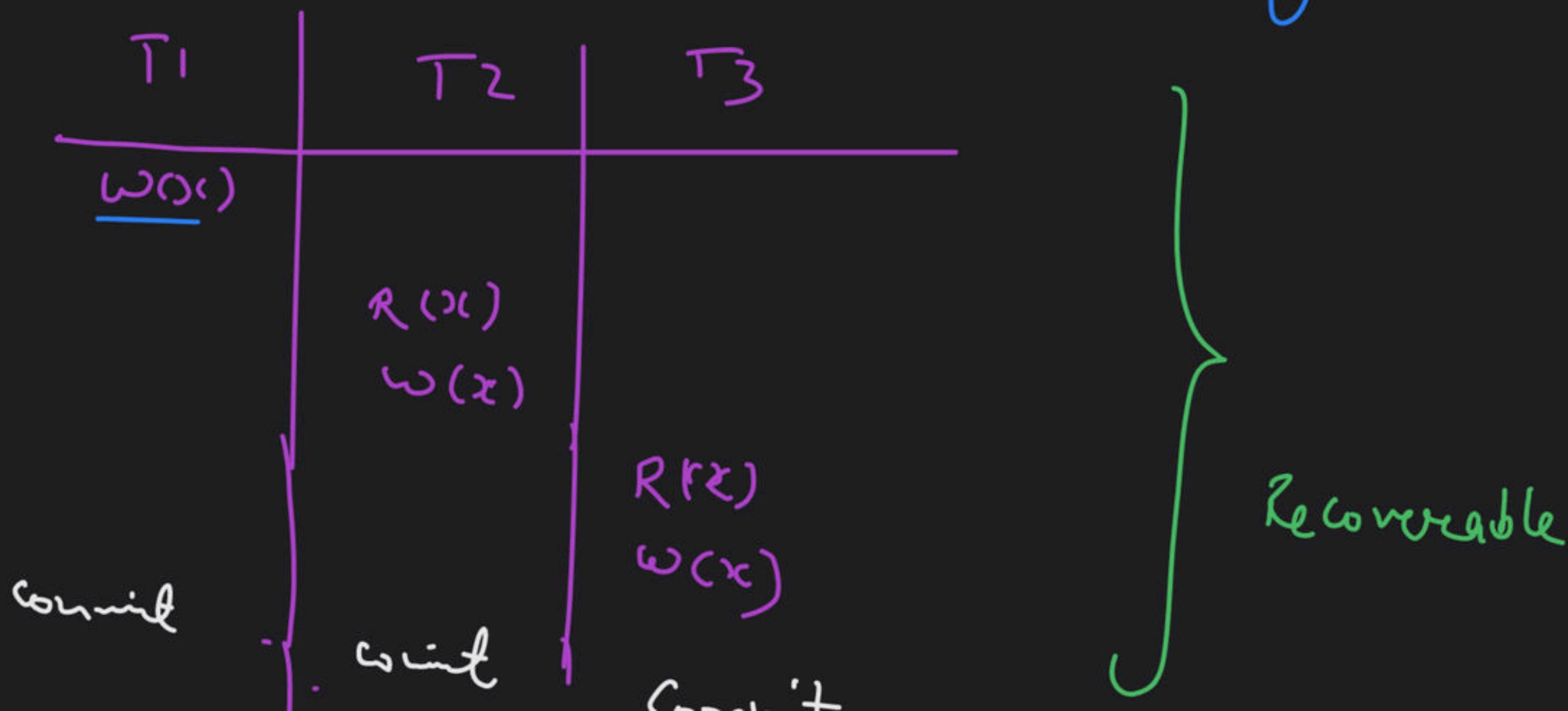
Recoverable Schedule

T1	T2
R(X)	
X=X+2	
W(X)	
	R(X)
	X=X+3
	W(X)
<i>failed</i> Commit	Commit

if T_1 failed just before } \Rightarrow not-recoverable
commit .

To make recoverable schedule \Rightarrow

Seq. of commit, should be same as sequence
of reading dirty values



Recoverable Schedule

T1	T2
R(X)	
X=X+2	
W(X)	
	R(X)
	X=X+3
	W(X)
Commit	
	Commit

Recoverable

Recoverable Schedule

1. Cascadeless Recoverable Rollback
2. Cascading Recoverable Rollback
3. Strict Recoverable Rollback

Cascading Recoverable Rollback

T1	T2	T3
R(X)		
X=X+2		
W(X)		
	R(X)	
	X=X+3	
	W(X)	
		R(X)
		X=X+4
		W(X)
Commit		
	Commit	
		Commit

if T_1 fails just before commit
then T_1, T_2, T_3 rolled back

if T_2 fails just before its
commit, then T_2, T_3 rolled back

Cascadeless Recoverable Rollback

T1	T2	T3
R(X)		
X=X+2		
W(X)		
Commit		
	R(X)	
	X=X+3	
	W(X)	
	Commit	
		R(X)
		X=X+4
		W(X)
		Commit

→ When a transaction fails
only that transaction will be
rolled-back.

Question

T1	T2	T3
W(X)		
	W(Y)	
	R(X)	
		R(Y)
Commit	<u>Failed</u>	Commit
	Commit	Then T3 will be

not-recoverable

required to rollback.

Question

T1	T2	T3
R(X)		
W(X)	R(X)	
	W(X)	
Commit	Commit	Rollback

cascading recoverable rollback

Question

T1	T2
R(A)	
W(A)	
	W(A)
commit	
	R(A)
	commit

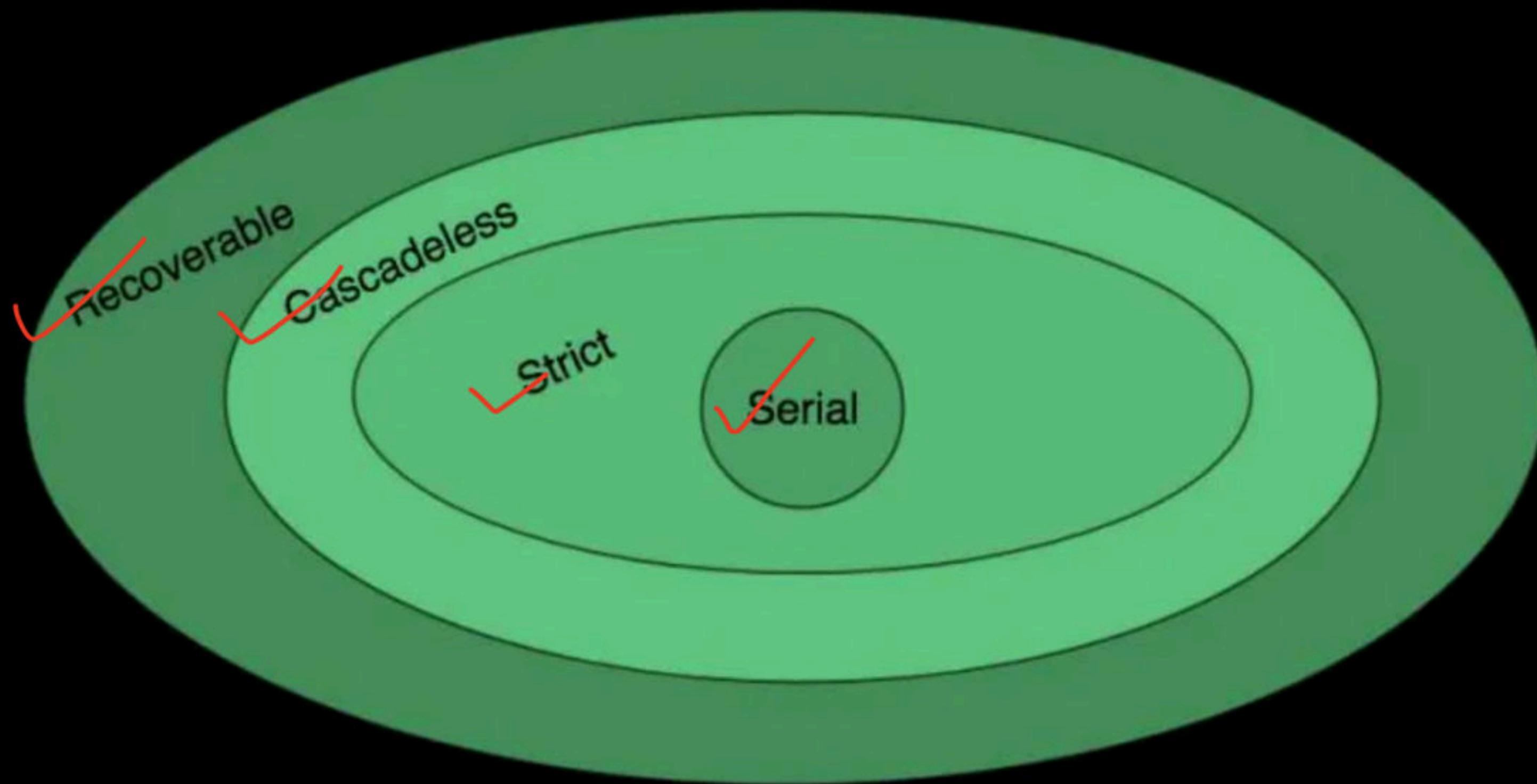
Recoverable
Consistency

Strict Recoverable Rollback

T1	T2
R(A)	
	R(A)
W(A)	
commit	
	W(A)
	R(A)
	commit

Dirty read and write can be done in another transaction only after commit of first transaction.

Recoverable Schedules



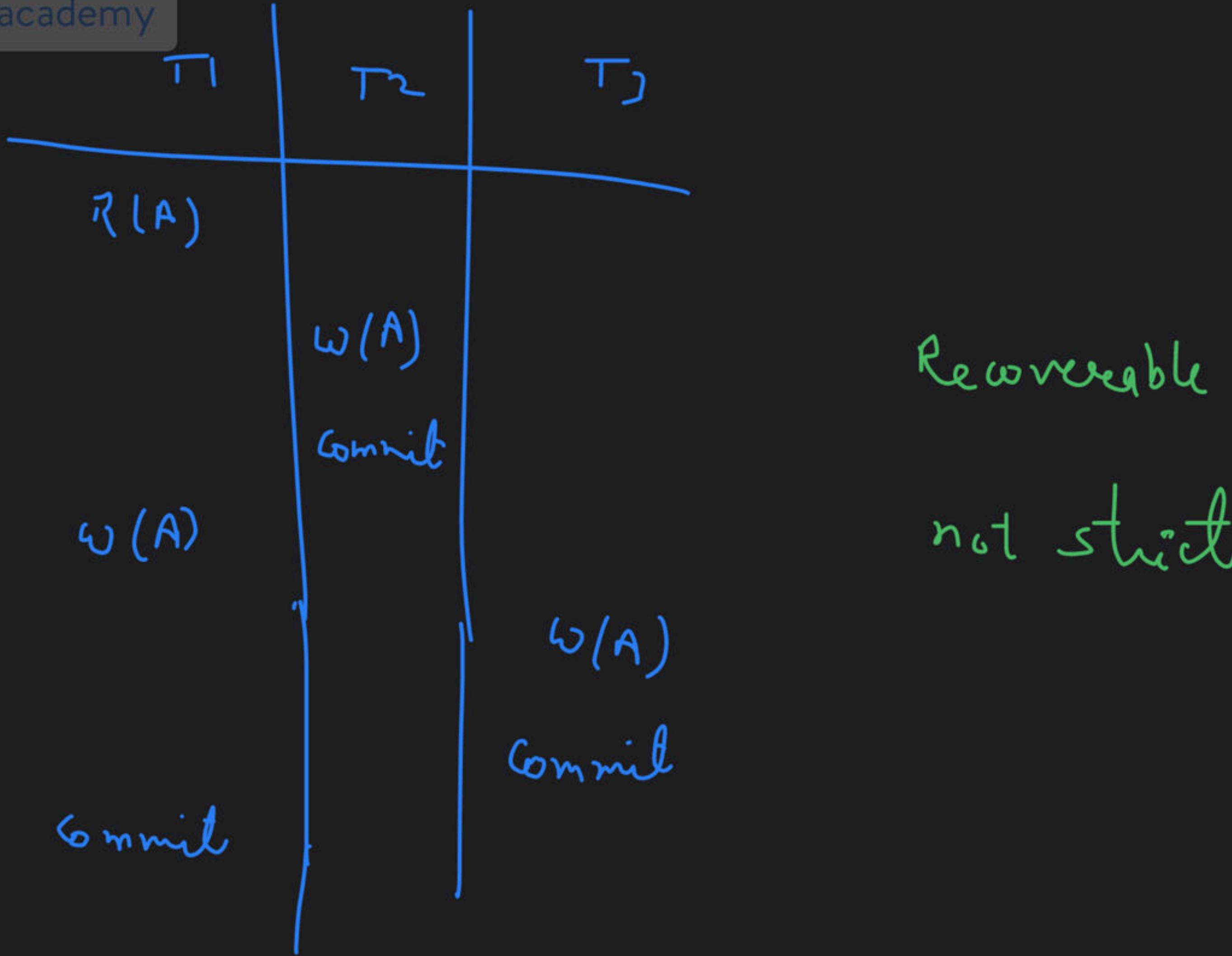
Question

Consider the following schedule:

S:R1(A), W2(A), Commit2, W1(A), W3(A), Commit3, Commit1

Which of the following is true?

- (A) The schedule is view serializable schedule and strict recoverable schedule
- (B) The schedule is non-serializable schedule and strict recoverable schedule
- (C) The schedule is non-serializable schedule and is not strict recoverable schedule.
- (D) The Schedule is serializable schedule and is not strict recoverable schedule

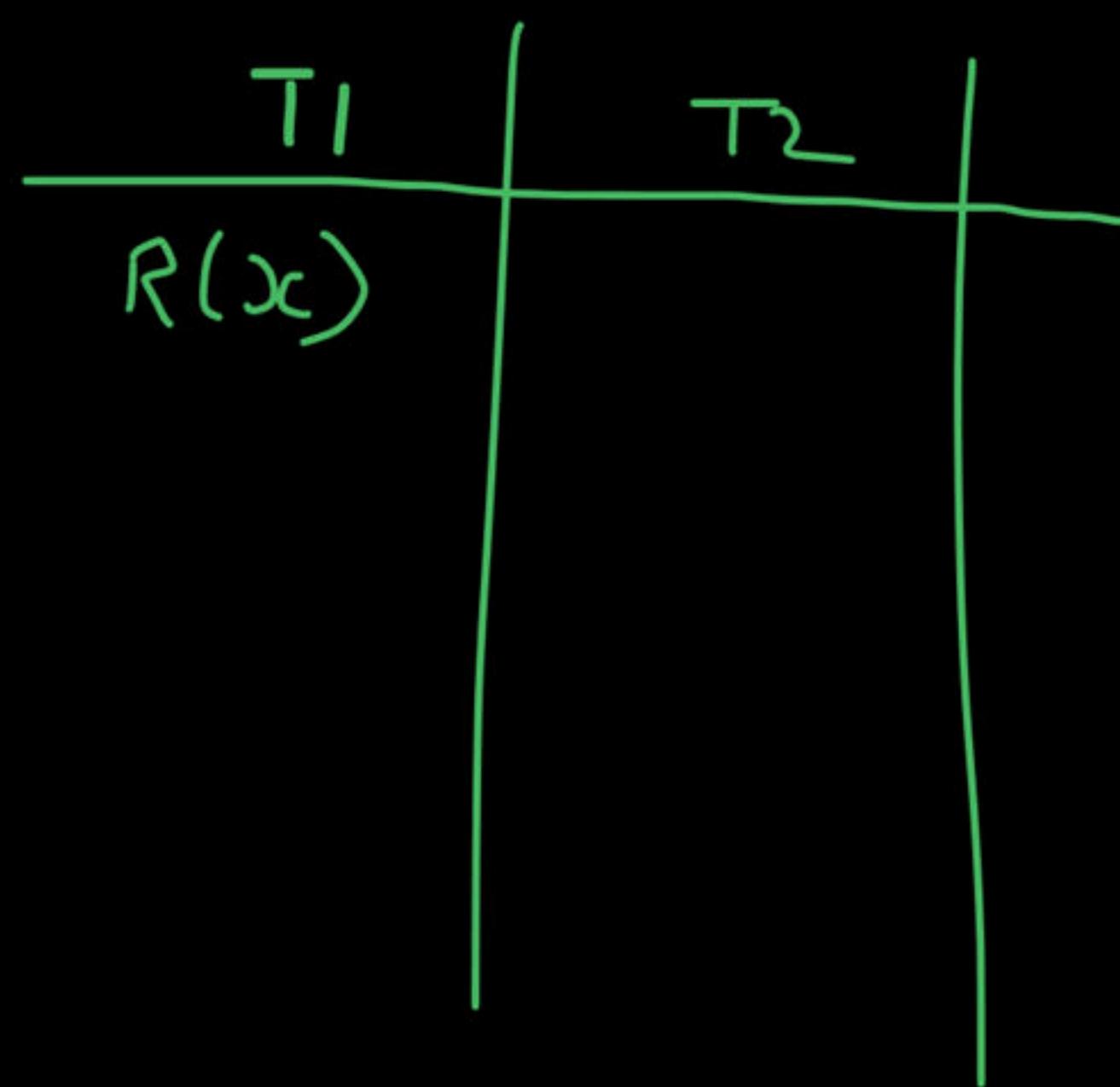


Renewable

not strict

Concerning Questions

1. Future transaction statements are predictable?
2. Which transaction will run which statement in what sequence is known?



Practically

~~strict~~ and view serializability is not implemented *practically*
conflict

Locking Protocols

What is Lock?

T1

lock (A)

R (A)

unlock (A)

T2

lock \Rightarrow 0

1

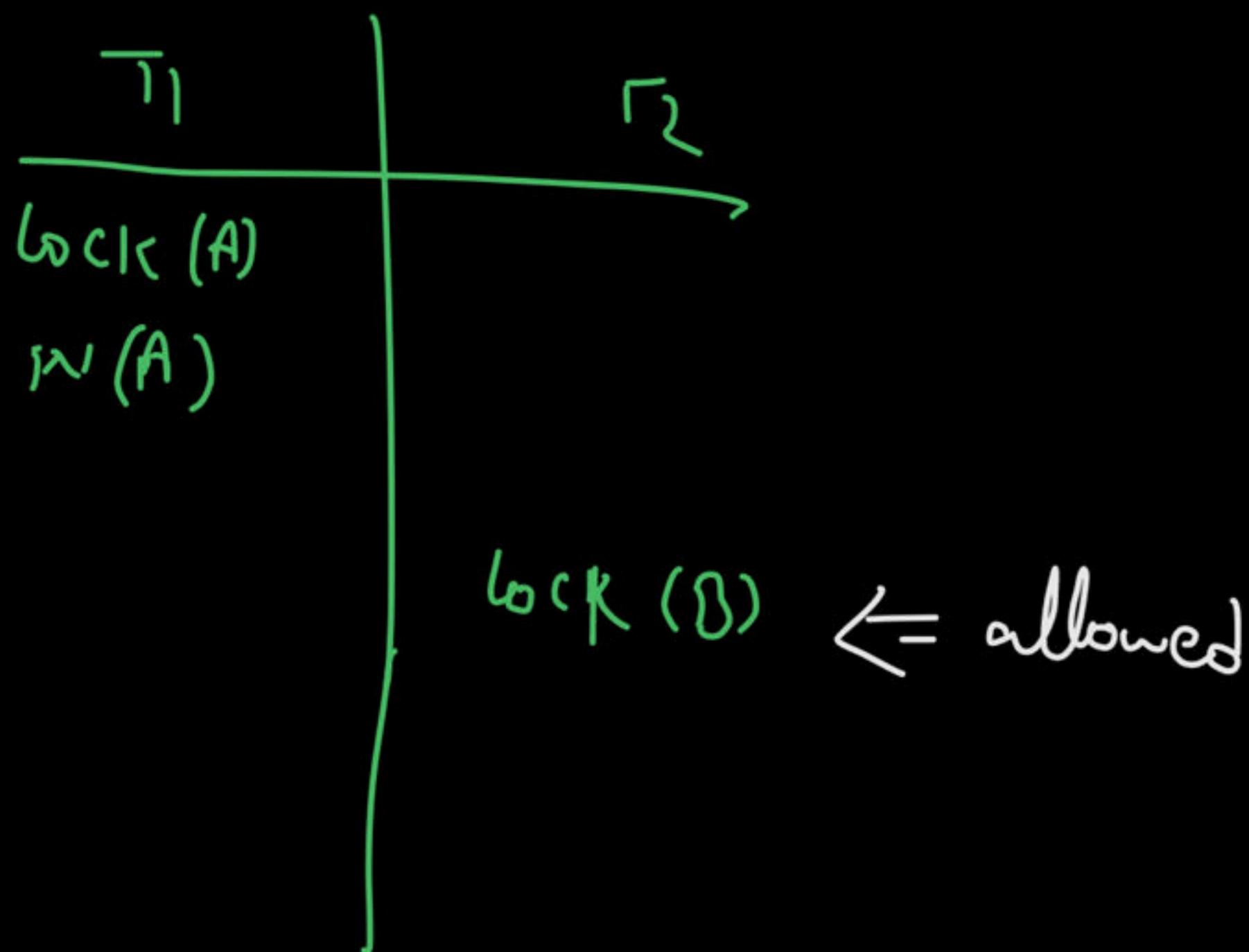
available

not-available

Lock

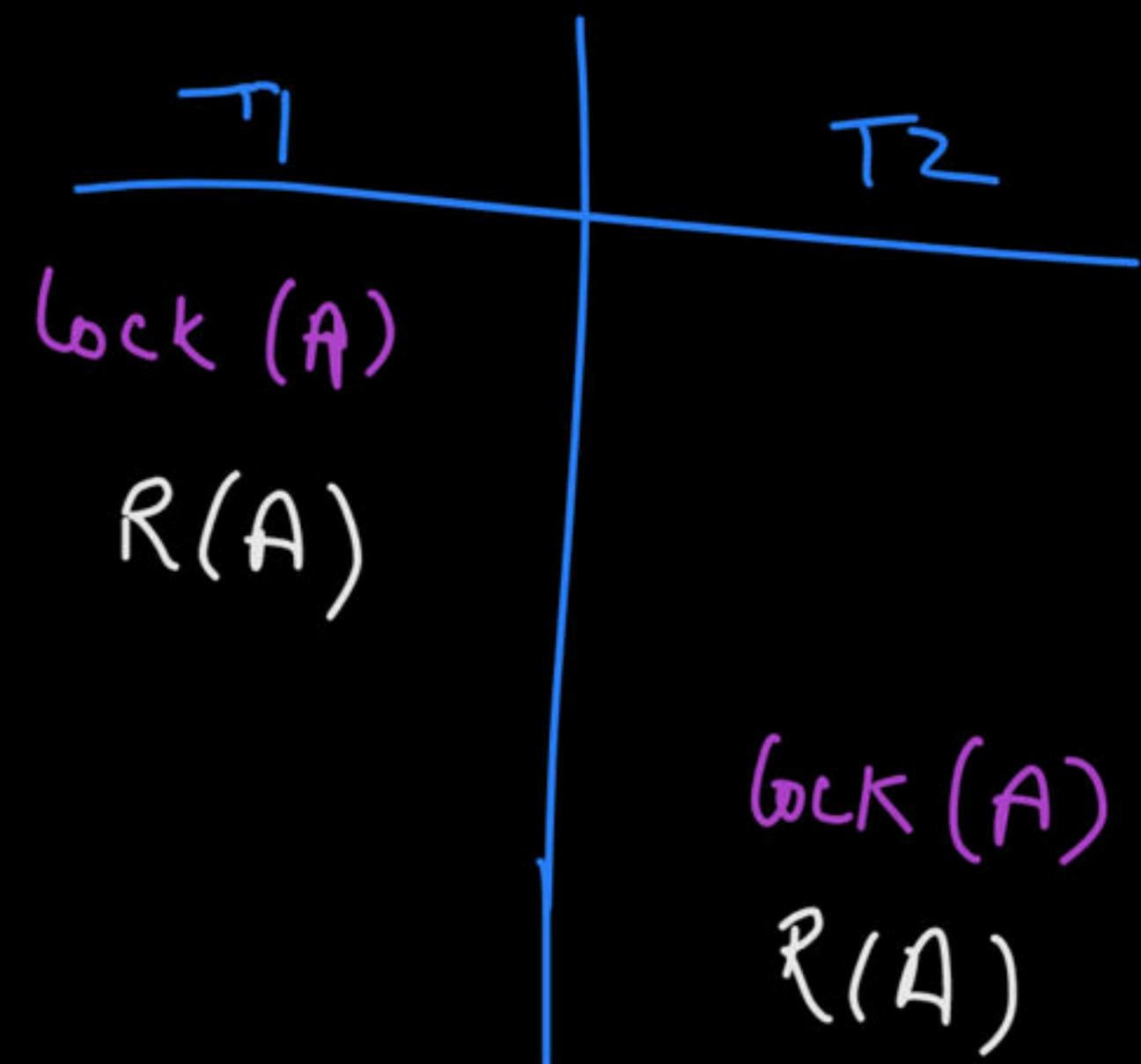
How many locks?

one lock for each data item



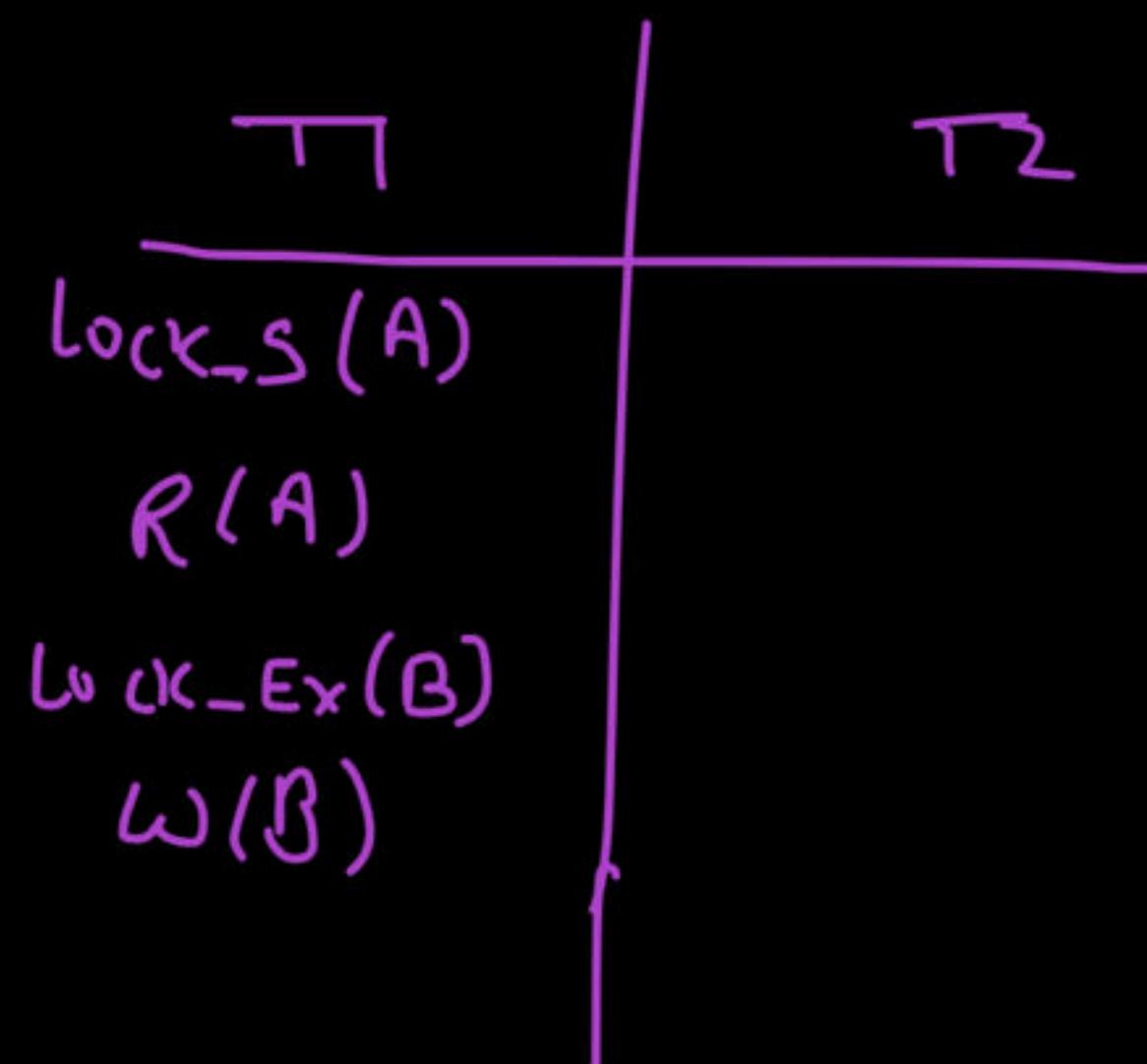
Lock

Locks for only write operation?



Lock

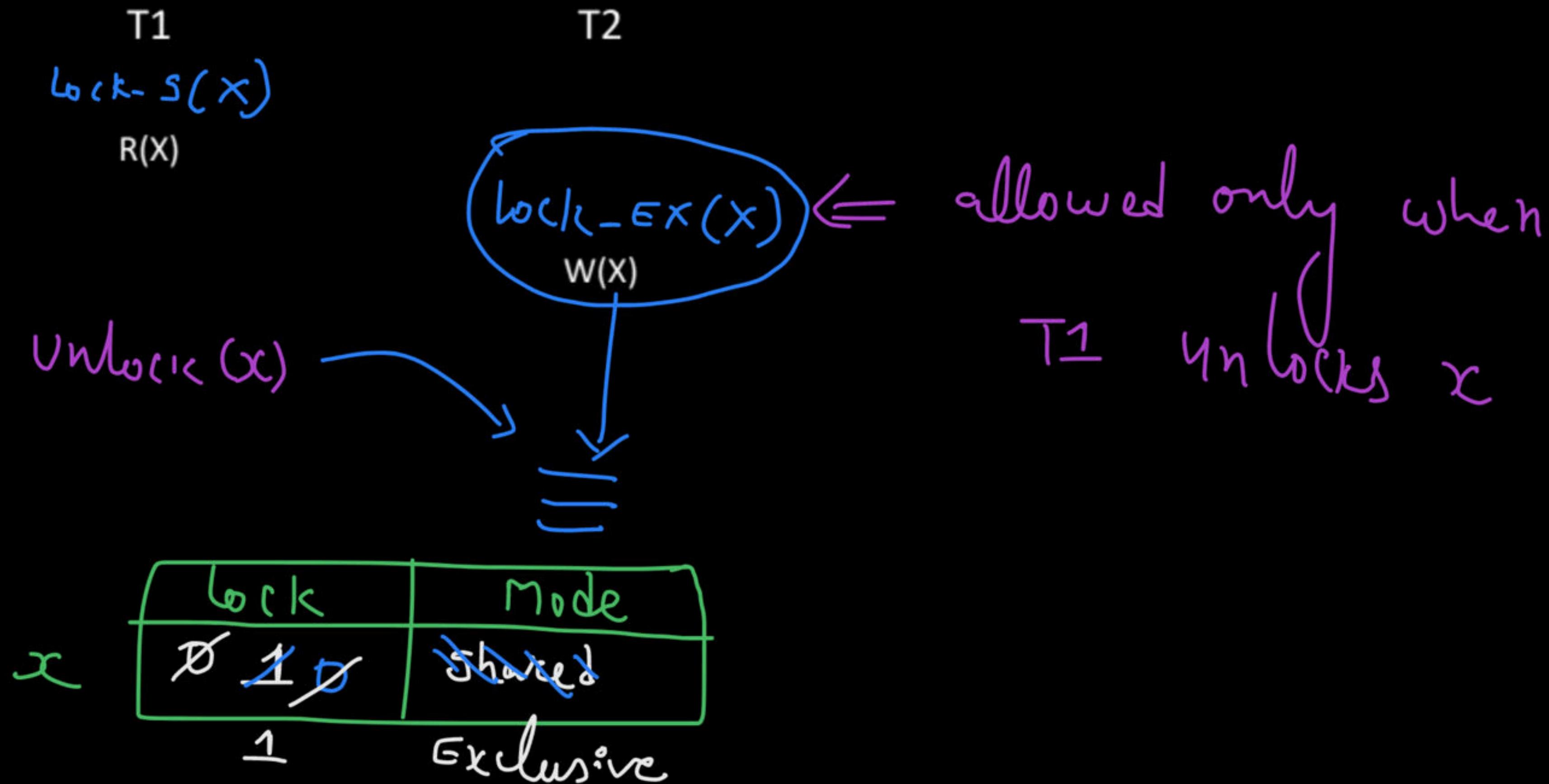
1. Shared lock \Rightarrow for only read operations
2. Exclusive lock \Rightarrow for read/write both



Compatibility

	shared	Exclusive
shared	✓	✗
Exclusive	✗	✗

Lock



Lock: Busy Waiting

T1
Lock_S(X)
R(X)
Unlock(X)

T2

Available	Mode

~~Lock_Ex(X)
W(X)~~ → busy waiting for exclusive lock on x.



Lock: Blocked Transaction

T1
Lock_S(X)
R(X)
Unlock(X)

T2
Lock_Ex(X)
W(X)

Available	Mode	Blocked Transaction

Lock: Multiple Shared Locks

T1 T2 T3 T4

R(X)

W(X)

R(X)

R(X)

Available	Mode	Count	Blocked Transaction

Multiple Shared Locks without Starvation

T1 T2 T3 T4

R(X)

W(X)

R(X)

R(X)

Available	Mode	Count	Blocked Transaction

Locks: Upgrade

T1

R(X)

W(X)

Locks: Downgrade

T1

W(X)

R(X)

Problem with Locking Mechanism

T1

T2

R(X)

W(X)

R(X)

Basic 2 Phase Locking Protocol

Systematic Locking mechanism

Once unlock done, a transaction is not allowed to lock any database item.

Basic 2 Phase Locking Protocol

T1

T2

R(X)

W(X)

R(X)

Basic 2 Phase Locking Protocol

T1

T2

R(X)

W(X)

R(Y)

W(Y)

Basic 2 Phase Locking Protocol

Every schedule which is allowed under basic 2PL, is conflict serializable also.

Basic 2 Phase Locking Protocol

T1

T2

R(X)

W(Y)

W(X)

R(Y)

Basic 2 Phase Locking Protocol

Suffers from deadlock

Basic 2 Phase Locking Protocol

Can we acquire lock with single instruction?

Basic 2 Phase Locking Protocol

Starvation of small transactions due to large transaction

Basic 2 Phase Locking Protocol

Starvation of large transaction due to small transactions

Happy Learning.!



▲ 1 • Asked by Saloni

sir esme statement 2 bta do na ek baar

RDBMS_PYQ_Discussion_Part_3.pdf

File | C:/Users/DELL/Desktop/dbms%20pyq/RDBMS_PYQ_Discussion_Part_3_with_anno.pdf

Draw Read aloud

43 of 58

Question GATE-2019

Let the set of functional dependencies $F = \{QR \rightarrow S, R \rightarrow P, S \rightarrow Q\}$ hold on a relation schema $X = \{P, Q, R, S\}$. X is not in BCNF. Suppose X is decomposed into two schemas Y and Z , where $Y = \{PR\}$ and $Z = \{Q, S\}$.

Consider the two statements given below:

I. Both Y and Z are in BCNF
II. Decomposition of X into Y and Z is dependency preserving and lossless

Which of the above statements is/are correct?

A. Both I and II B. I only C. II only D. Neither I nor II

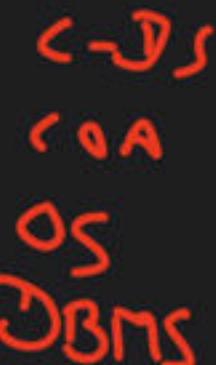
29°C Mostly cloudy

Search

Dell

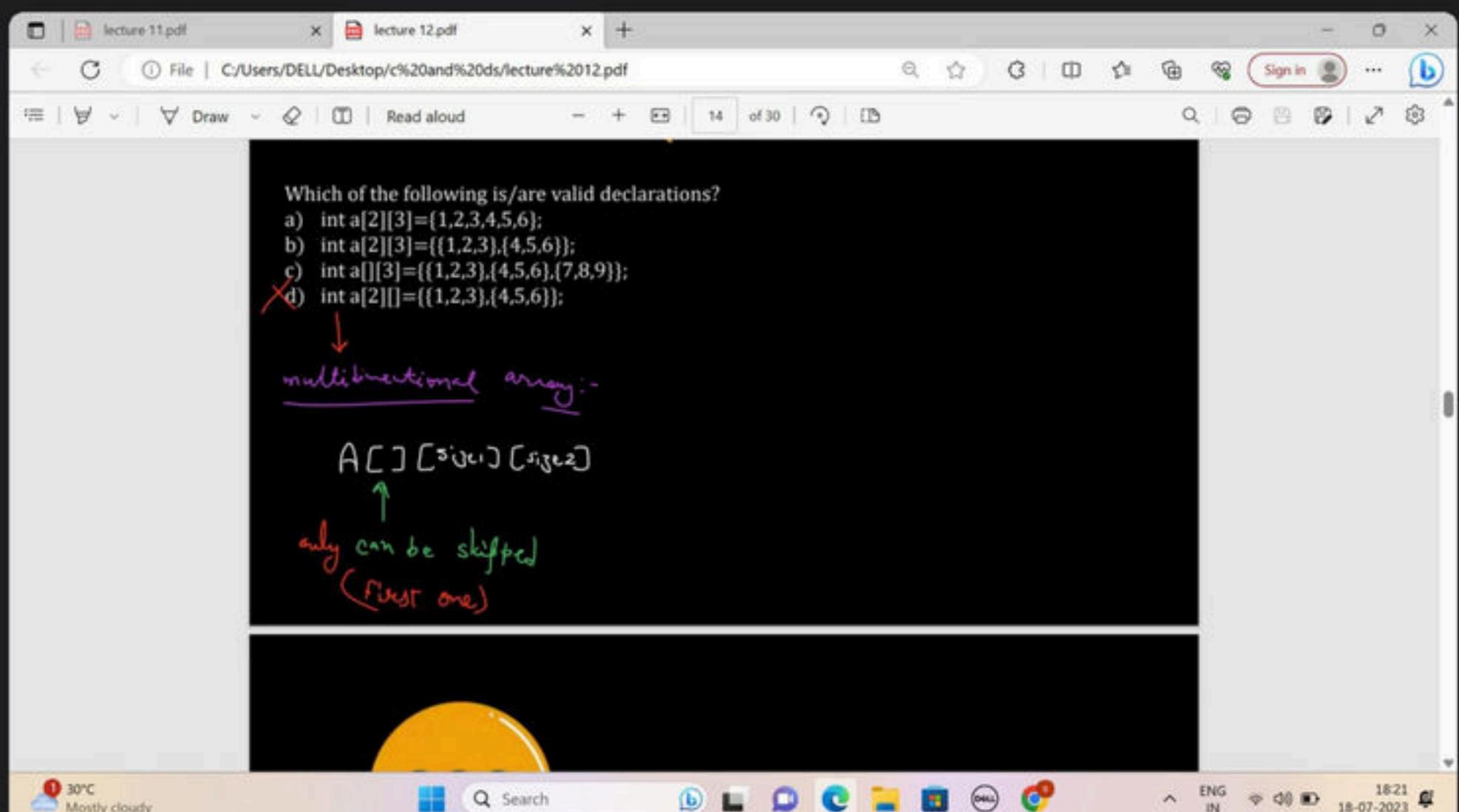
ENG IN

09:48 19-07-2023



▲ 1 • Asked by Saloni

Please help me with this doubt



Which of the following is/are valid declarations?

- a) int a[2][3]={1,2,3,4,5,6};
- b) int a[2][3]={{{1,2,3},{4,5,6}}};
- c) int a[][3]={{1,2,3},{4,5,6},{7,8,9}};
- d) int a[2][]={{1,2,3},{4,5,6}};

~~multidimensional array :-~~

A [] [size1] [size2]

only can be skipped
(first one)

A [] [] ✓

A [2] [] ✗

A [2] [3] [] ✗

A [2] [] [4] ✗

▲ 1 • Asked by Saloni

Please help me with this doubt

