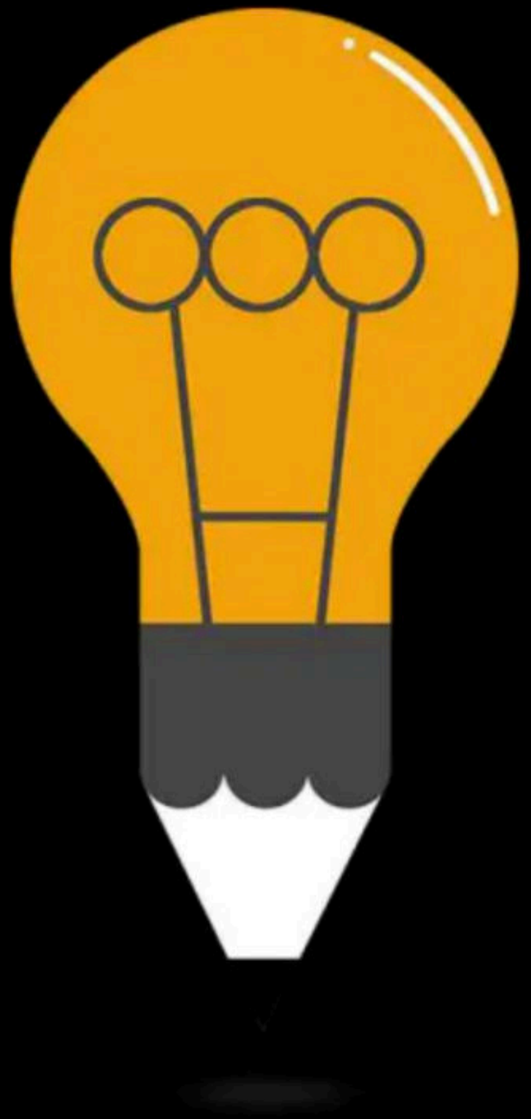
unacademy

Transaction & Concurrency Control: Part IV

Complete Course on Database Management System

Vishvadeep Gothi • Lesson 26 • July 21, 2023

unacademy



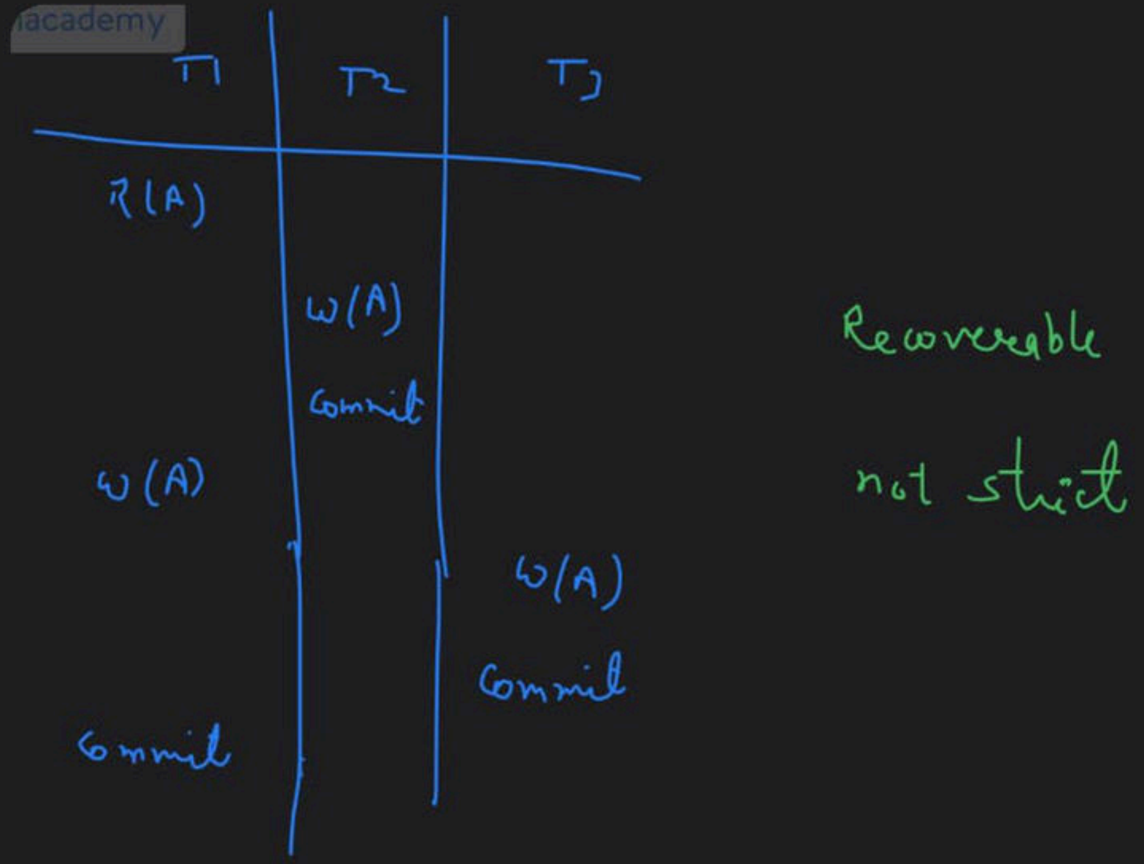
DBMS

Locking Protocols

By: Vishvadeep Gothi

▲ 1 • Asked by Vaishnavij...

Sir i have a doubt, in such case final commit is of T1 then final value of A will be the one written by T1?



Recoverable

not strict

unacademy

Locking Protocols

unacademy

What is Lock?

unacademy

Lock

How many locks?

unacademy

Lock

Locks for only write operation?

unacademy

Lock

1. Shared lock
2. Exclusive lock

unacademy

Lock

T1	T2
R(X)	
	W(X)

Lock: Busy Waiting

T1	T2
Lock_S(X)	
R(X)	
	Lock_Ex(X)
	W(X)
Unlock(X)	

Available	Mode

busy waiting

Lock_Ex(X)

W(X)

Unlock(X)

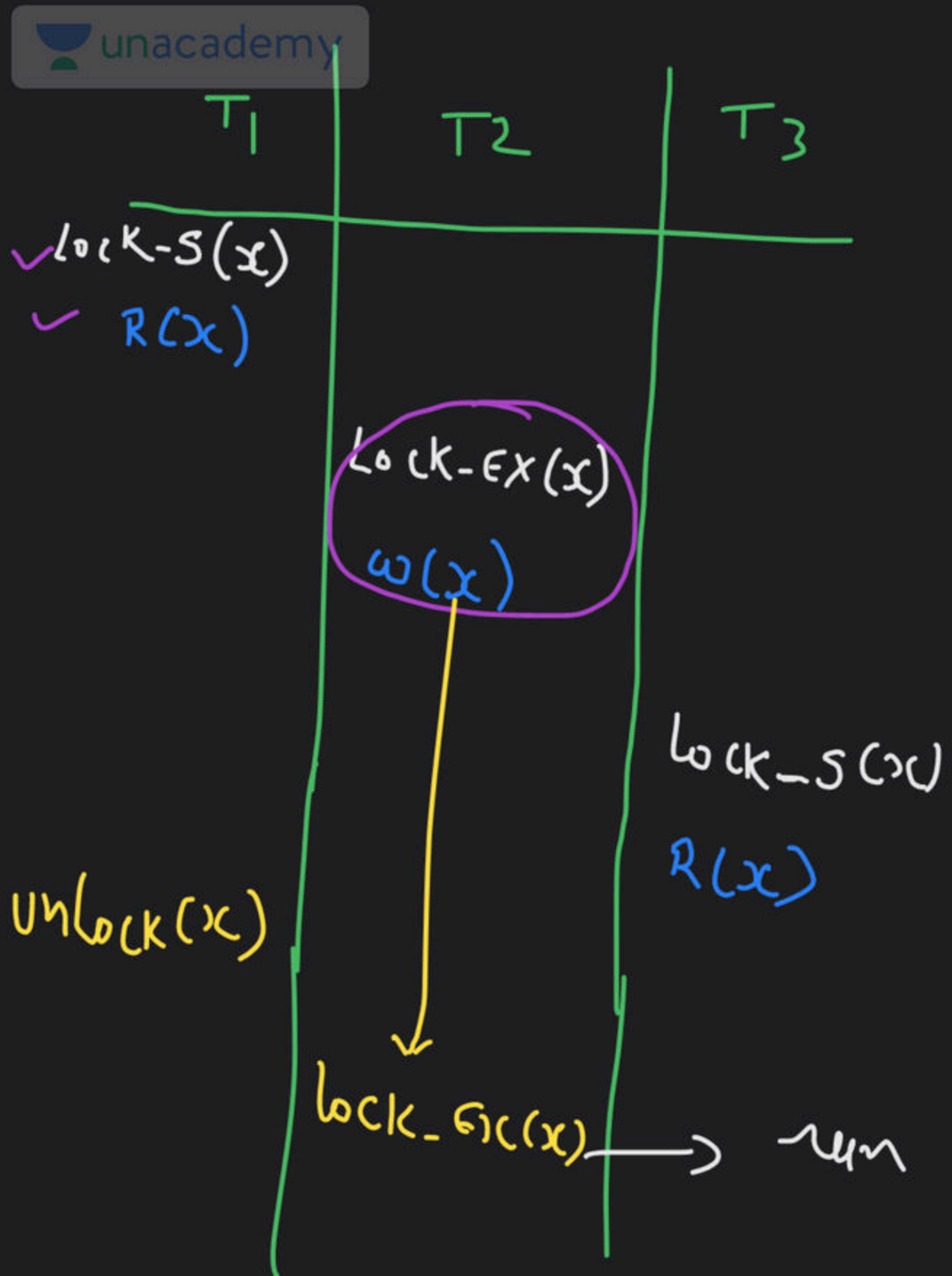
Lock: Blocked Transaction

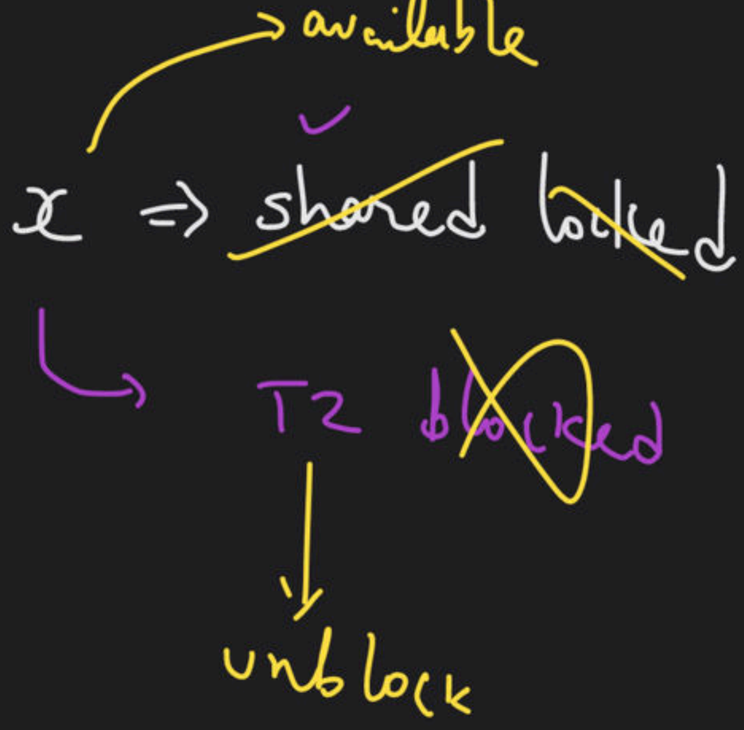
T1	T2
Lock_S(X)	
R(X)	
	Lock_Ex(X)
	W(X)
Unlock(X)	

blocked

Available	Mode	Blocked Transaction
~~0~~ 1	~~shared~~	T2

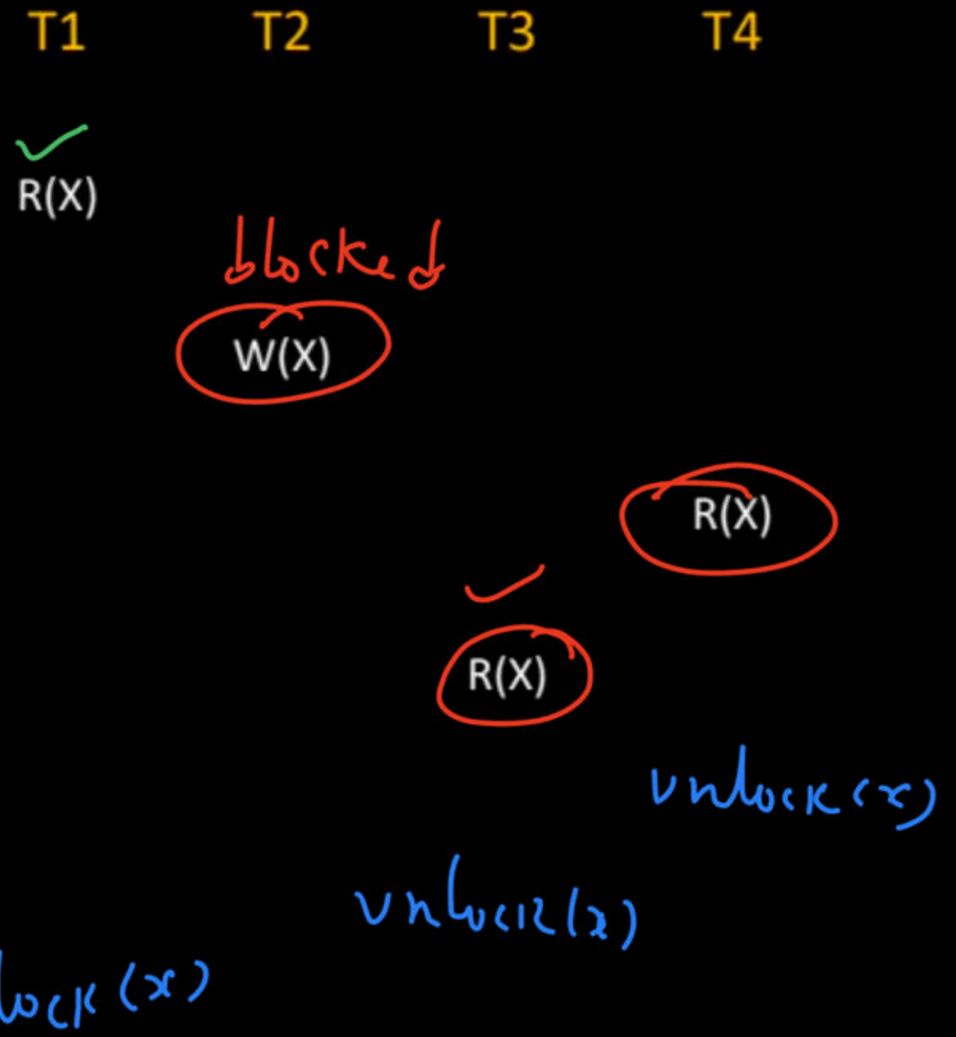
unblock





T3 ⇒ shared
T2 ⇒ Exc.
→ both there

Lock: Multiple Shared Locks



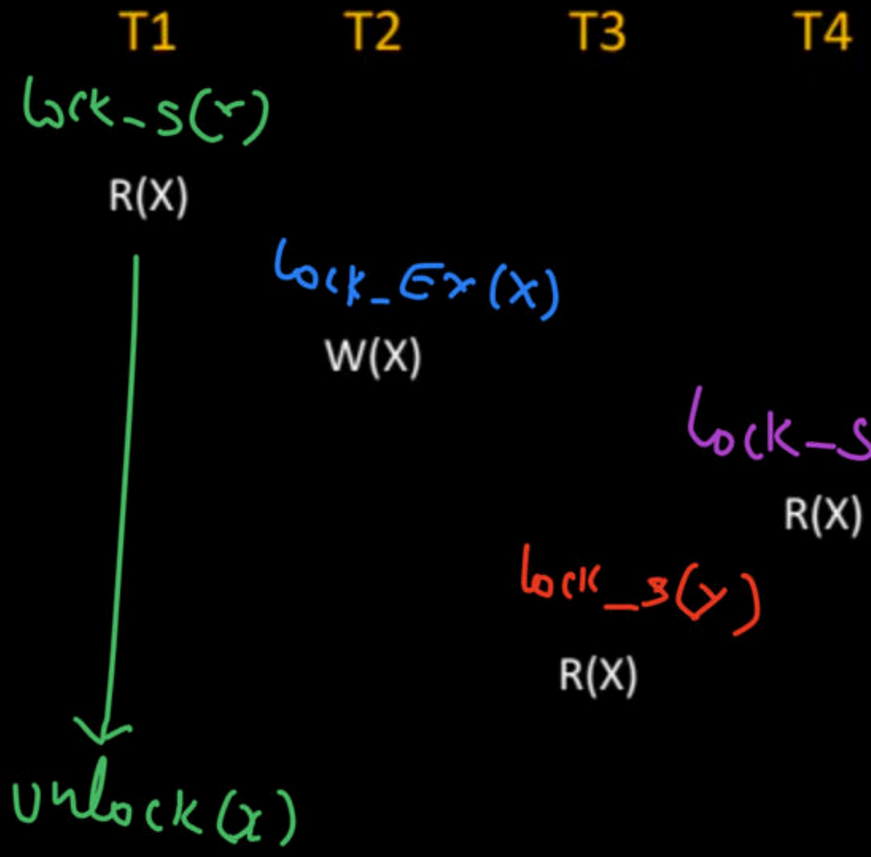
Available	Mode	Count	Blocked Transaction
	shared	1 2 3 2 1 0	T2

unblock

Blocked transactn is unblocked only when count becomes 0 for shared locks.

Here is new transactions keep asking for shared locks on x, then T2 may starve.

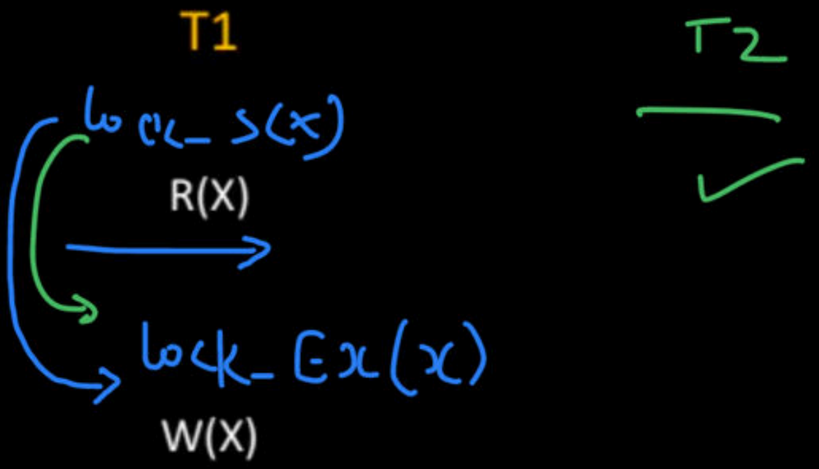
Multiple Shared Locks without Starvation



Available	Mode	Count	Blocked Transaction
~~0~~ 1	shared	1	T2 | T4 | T3

lock_S(x) ⇒ if no any blocked transaction on x then allowed o/w blocked.

Locks: Upgrade



unacademy

Locks: Downgrade

T1

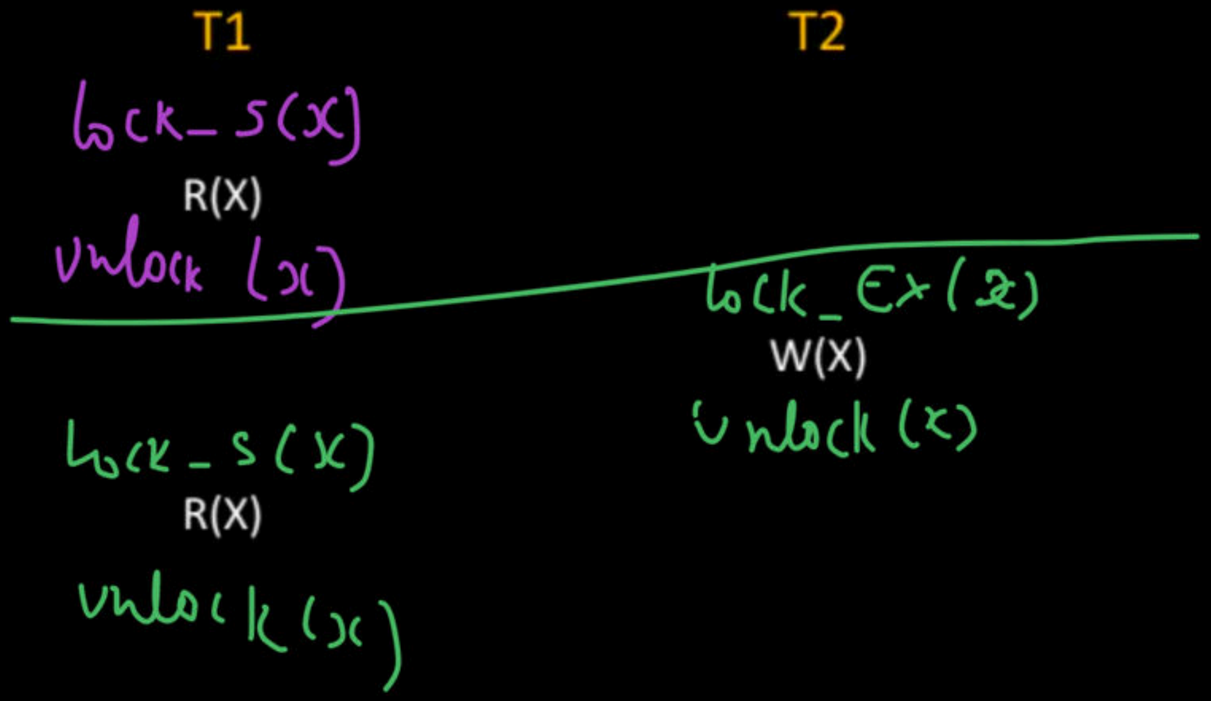
lock_ex(x)

W(X)

lock_s(x)

R(X)

Problem with Locking Mechanism



unrepeatable read

unacademy

Locking Protocols ⇒ 2-phase locking Protocol (2PL)

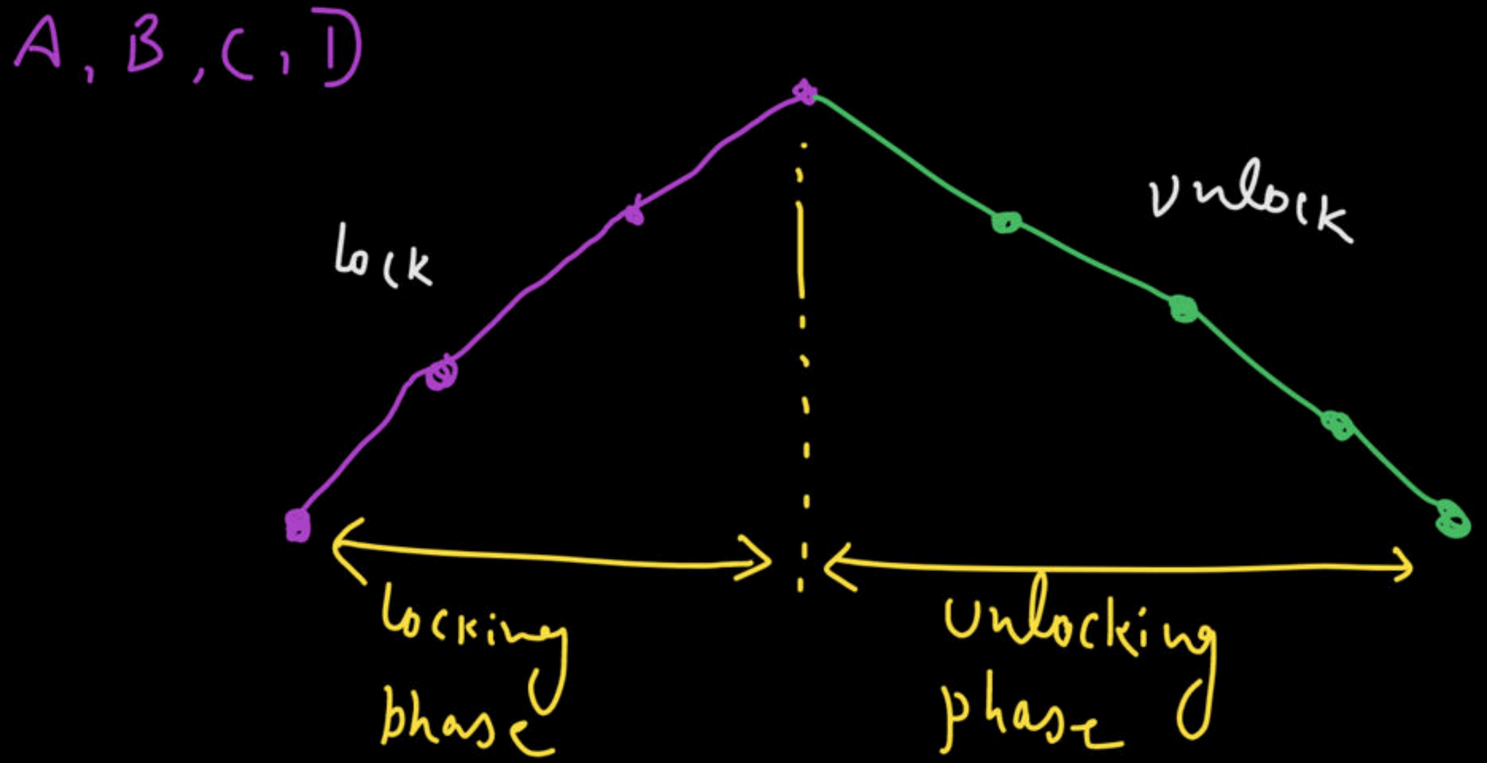
systematic locking mechanism

1. Basic 2PL
2. Strict 2PL
3. Rigorous 2PL
4. Conservative 2PL

Basic 2 Phase Locking Protocol

Systematic Locking mechanism

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



unacademy

T1

lock_S(x)

lock_Ex(Y)

R(x)

w(y)

unlock (x)

R(y)

unlock (y)

correct acc. to basic 2PL

T1

lock_S(x)

R(x)

lock_Ex(y)

w(y)

unlock (x)

unlock (y)

lock_Ex(z)

w(z)

unlock(z)

lock_(~) not allowed after unlock.

Basic 2 Phase Locking Protocol

S1

T1	T2
lock_S(x)	
R(X)	
	lock_Ex(x) ← blocked
	W(X)
R(X)	
unlock(x)	
	lock_Ex(x)
	w(x)
	Unlock(x)

S1

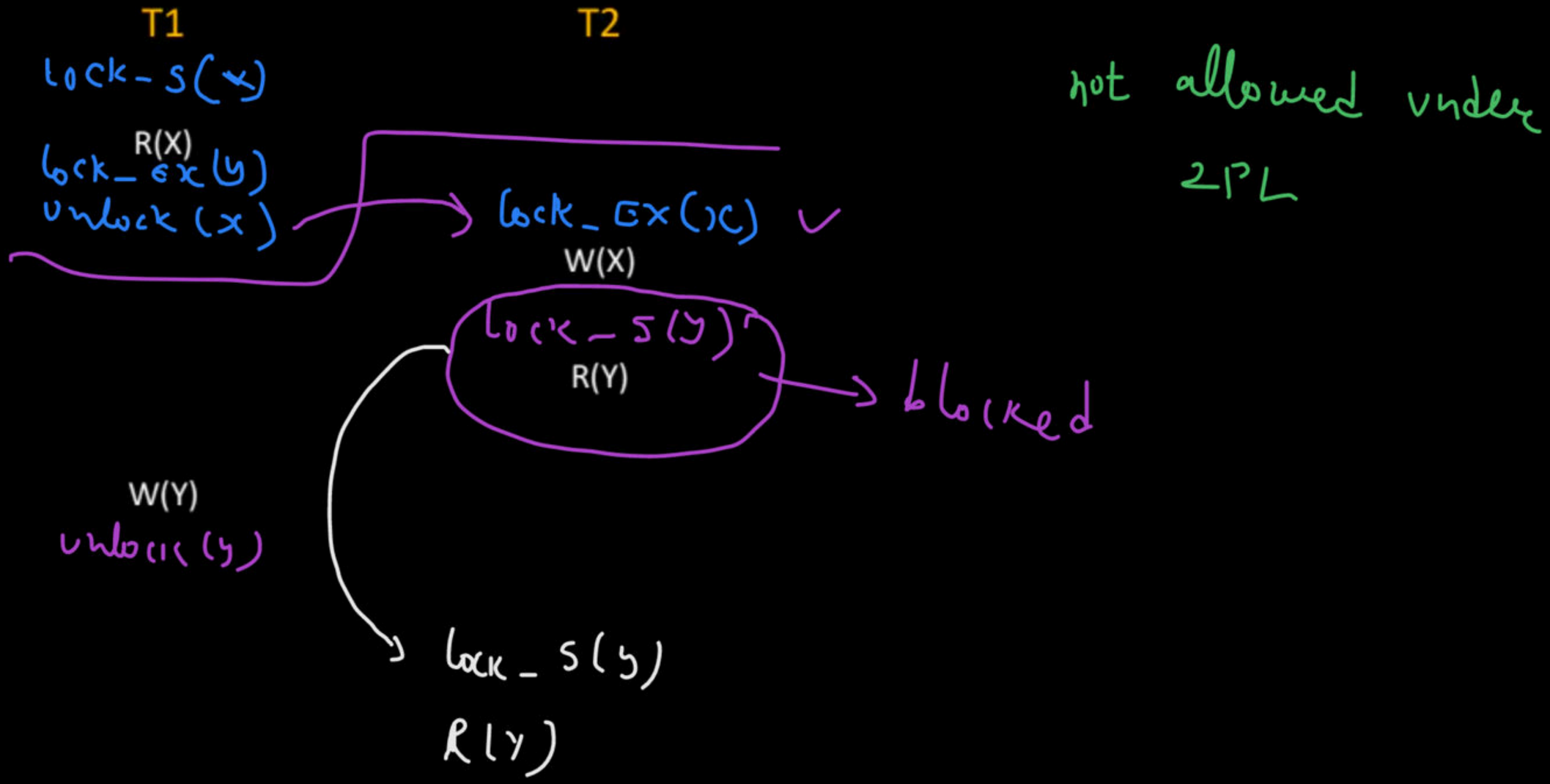
T1	T2
R(x)	
	w(x)
R(x)	

Run actually like:-

T1	T2
R(x)	
R(x)	
	w(x)

Given schedule S1 is not allowed under basic 2PL, because it does not run same as given.

Basic 2 Phase Locking Protocol



not allowed under 2PL

T1	T2
lock_S(x)	
R(x)	
	lock_S(y)
	R(y)
	Unlock(y)
lock-Ex(y)	
w(y)	
Unlock(x)	
Unlock(y)	

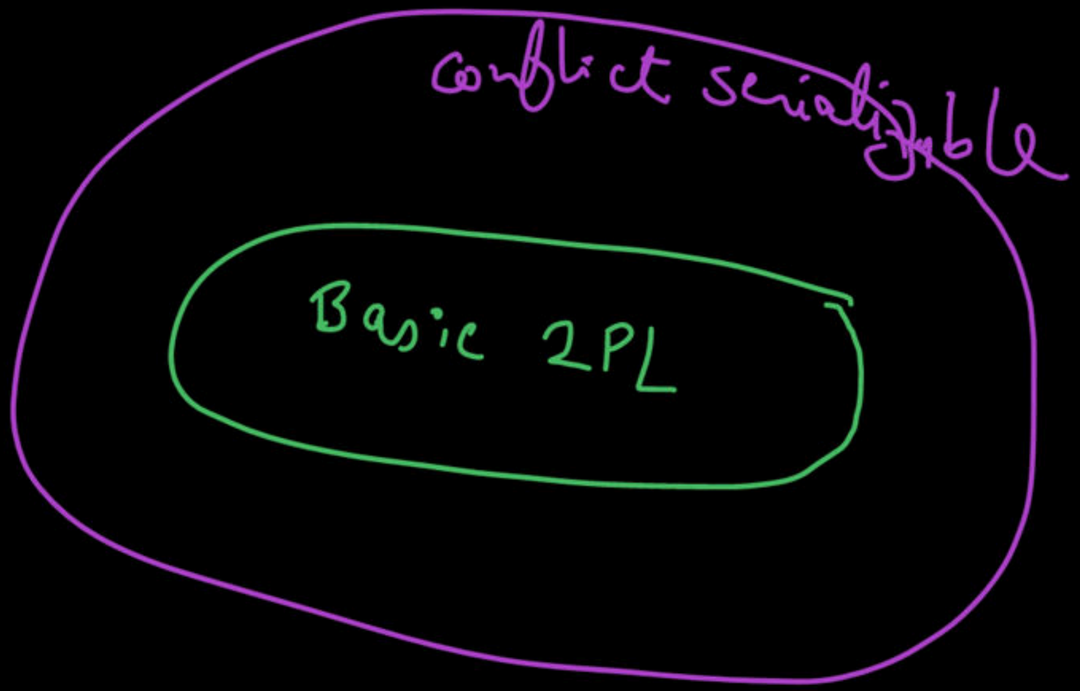
allowed under

2PL

unacademy

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

unacademy

Basic 2 Phase Locking Protocol

Can we acquire lock with single instruction?

unacademy

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

unacademy

Strict Schedule

Every Write operation should be ended with commit before other transaction performs read or write on the same DB item.

Strict Schedule

Every Write operation should be ended with commit before other transaction performs read or write on the same DB item.

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

Strict Schedule

Every Write operation should be ended with commit before other transaction performs read or write on the same DB item.

T1	T2	T3	T4
R(X)			
	W(X)		
Commit			
	Commit		
		W(X)	
			W(X)

unacademy

Strict 2PL

Exclusive lock should not be release until commit

unacademy

Strict 2PL

Exclusive lock should be released after commit

T1

Lock_X(A)

W(X)

Commit

Unlock(X)

unacademy

Strict 2PL

So Strict 2PL allows only strict schedules

T1

Lock_X(A)

W(X)

Commit

Unlock(X)

unacademy

Rigorous 2PL

Every lock should be released after commit

unacademy

Rigorous 2PL

Every lock should be released after commit

T1

Lock(A)

W(X) or R(X)

Commit

Unlock(X)

unacademy

Rigorous 2PL

Every lock should be released after commit

unacademy

Rigorous 2PL

Is it allowed under Basic 2PL?

unacademy

Rigorous 2PL

Strict & Rigorous 2PL don’t have dirty read

unacademy

Rigorous 2PL

Is it allowed under Basic 2PL?

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

unacademy

Rigorous 2PL

Strict & Rigorous 2PL have only recoverable schedules (cascadeless)

unacademy

2PL

Basics 2PL may allow non-recoverable schedules

unacademy

2PL

Basics 2PL may allow non-recoverable schedules

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

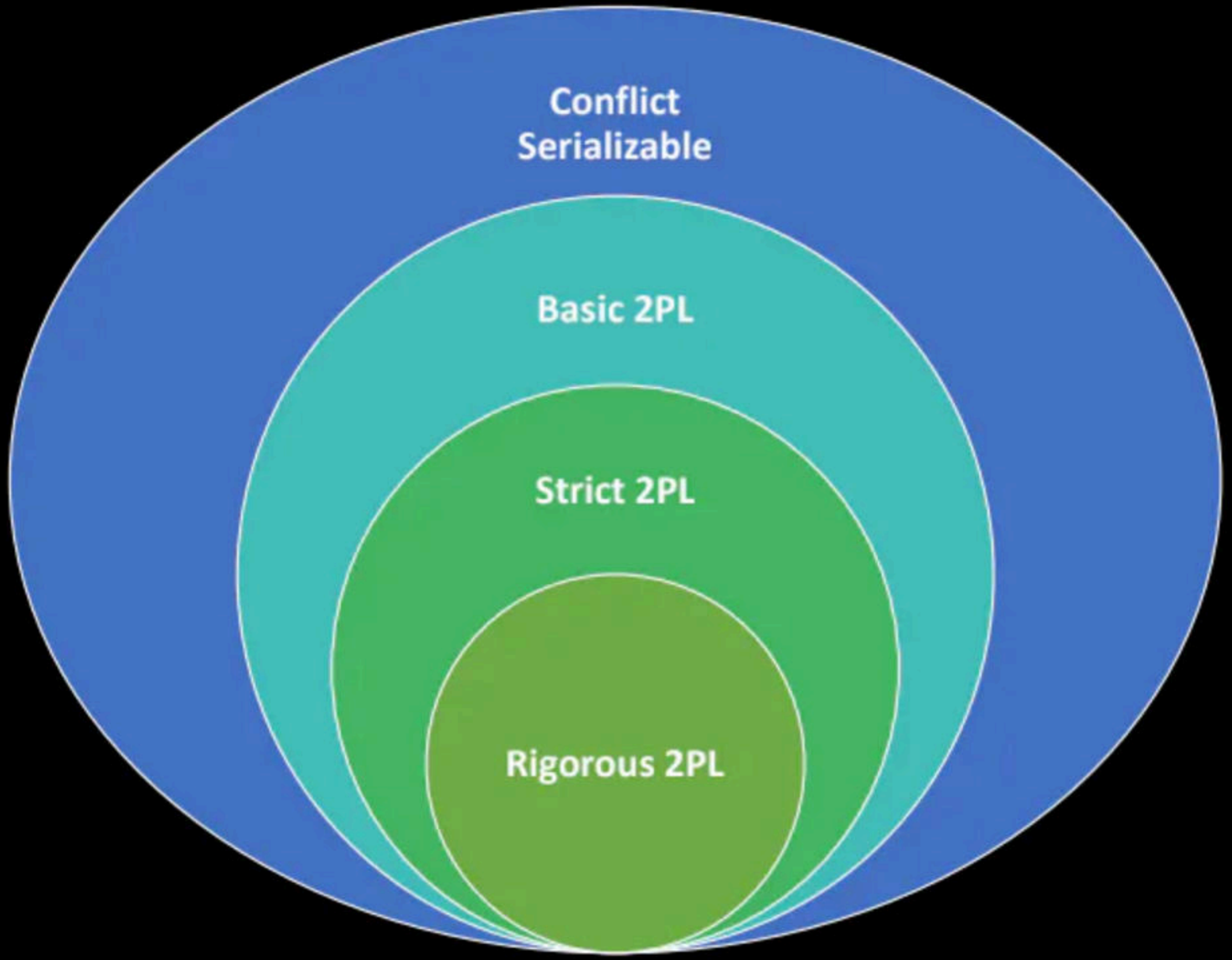
unacademy

Conservative 2PL (Static 2PL)

Lock all the items before the Transaction begins execution by predeclaring its read-set and write-set

unacademy

2PL



unacademy

2PL

T1	T2	T3
	R(A)	
		W(A)
W(B)		
	R(B)	
	W(C)	

unacademy

Happy Learning.!

unacademy

▲ 1 • Asked by Shreyas

serializable hai ki nahi,Agar t1->t2 chalta hu toh t1 mein joh write(A) waha mujhe konsi value write krni hogi joh given qn mein maine write ki thi ya first time A pe write wali ?

T1	T2
Read(A) 20	
A = A − 10	
10	
	Read(A) 10
	Temp = 0.2*A 2
	Write(A)
	Read(B) 15
Write(A)	
Read(B)	
B = B + 10	
Write(B)	
	A 15 2
	B = B + Temp
	Write(B)

A = ~~20~~ ~~10~~ 10

B = ~~5~~ ~~15~~ 5

A = ~~20~~ ~~10~~ 2

B = ~~5~~ ~~15~~ 17