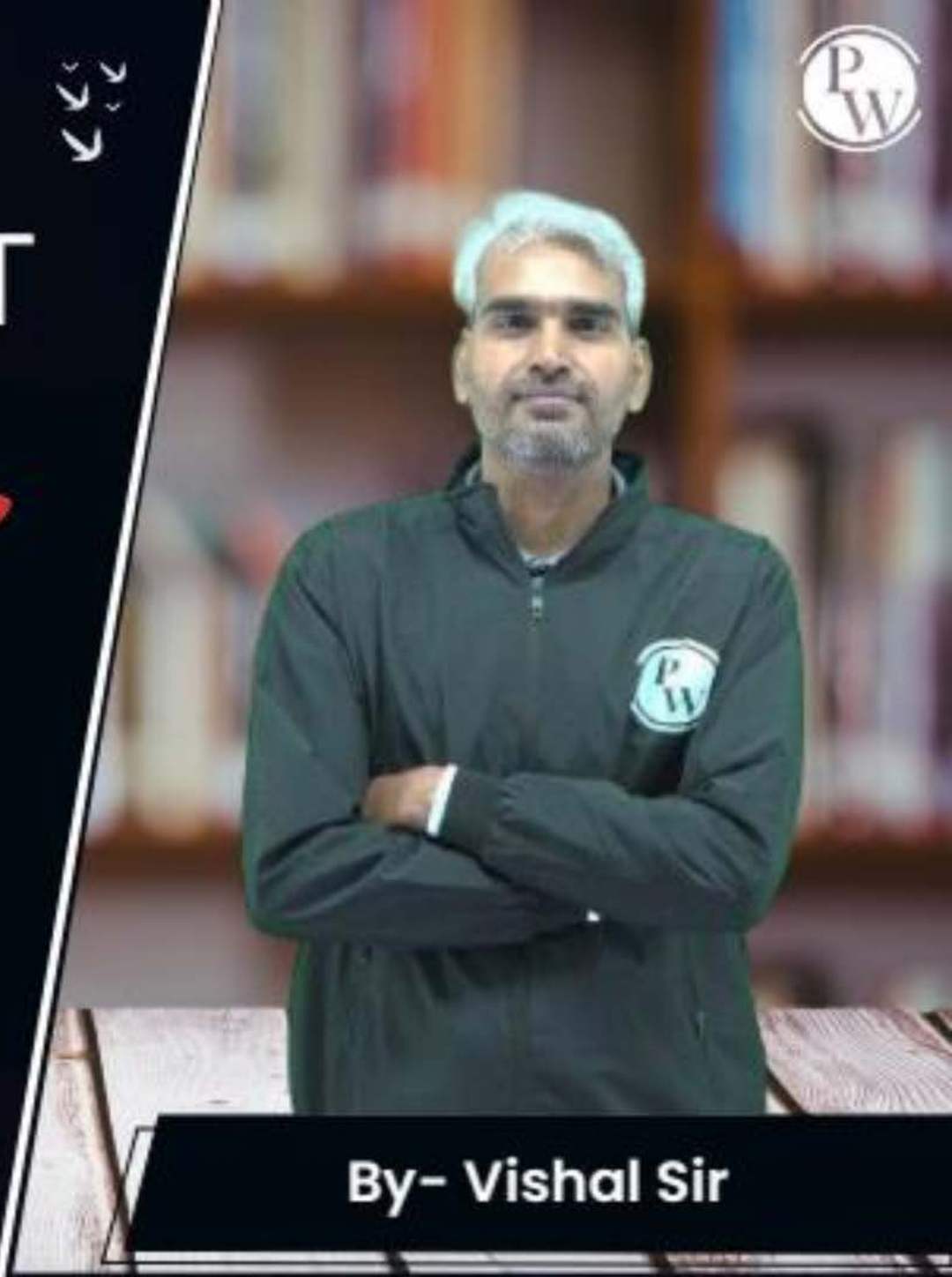
Computer Science & IT

Database Management System

Transaction & concurrency control

Lecture No. 04



By- Vishal Sir

Recap of Previous Lecture

- Topic: Equivalent schedules
- Topic: Serializable schedules

Slide

Topics to be Covered

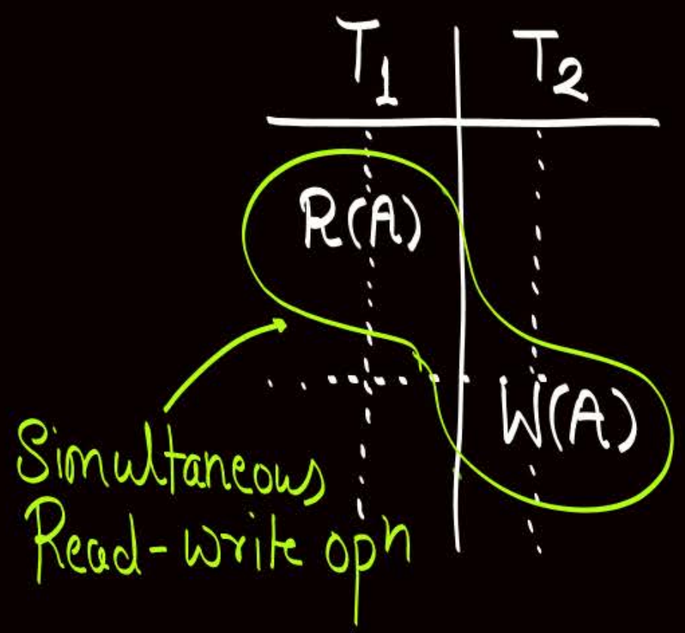
- Topic — RW problem
- Topic — WR problem
- Topic — WW problem
- Topic — Lost update problem
- Topic — Classification of schedules

Slide

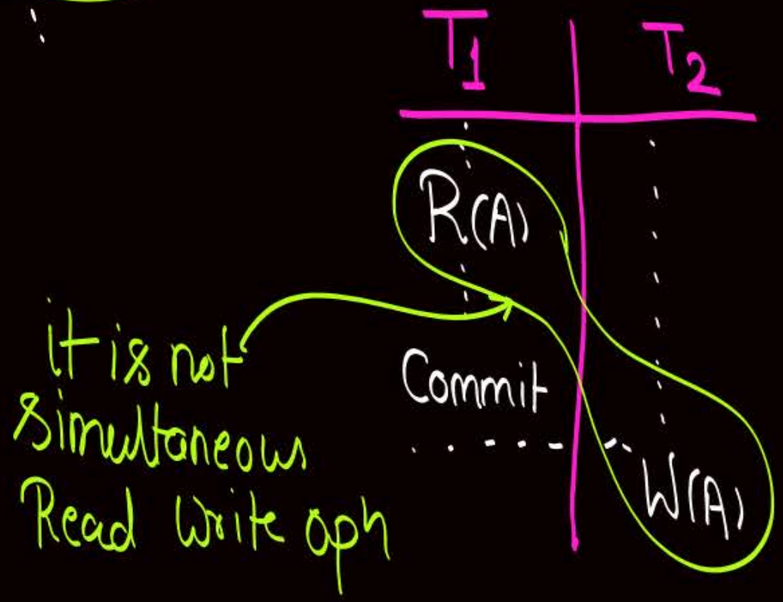
Topic : Problems because of concurrent execution

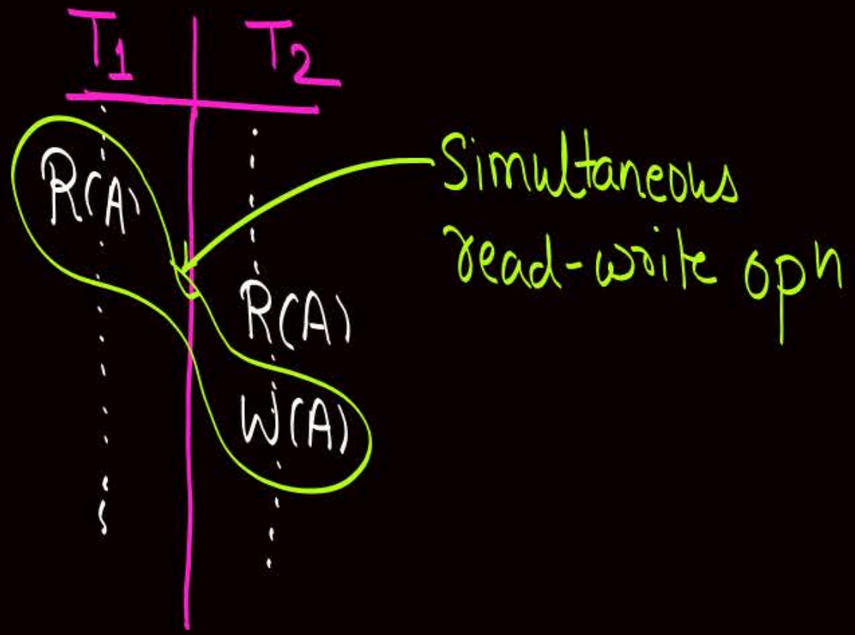
1. R W Problem / Read-Write Problem / Write after Read problem
   (Sequence of oph)
2. W R Problem / Write–Read Problem / Read after Write problem / Dirty Read Problem
3. W W Problem / Write-Write Problem / Write-after-Write problem
4. Lost update problem

RW Problem / Read-Write Problem / Write after Read Problem



If transaction T_2 updates the dataitem 'A' which is already read by an uncommitted transaction T_1, then that sequence of oph is called simultaneous Read-write oph.





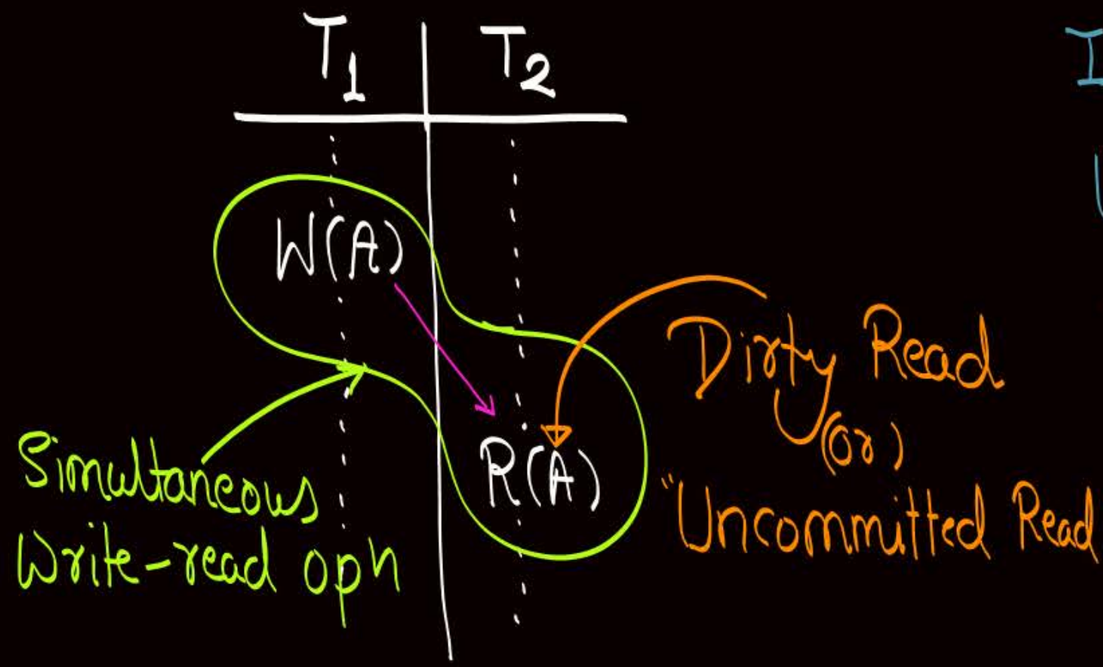
Topic : RW problem

A schedule is said to have RW problem if and only if,

① Simultaneous Read-write oph should exist

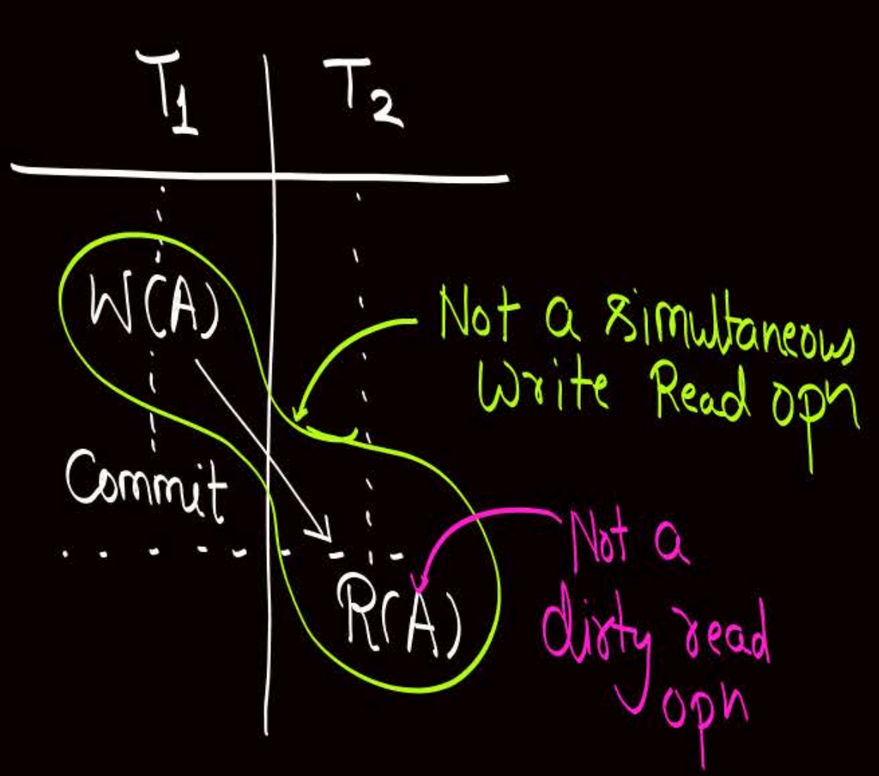
and ② Schedule must be a non-serializable schedule

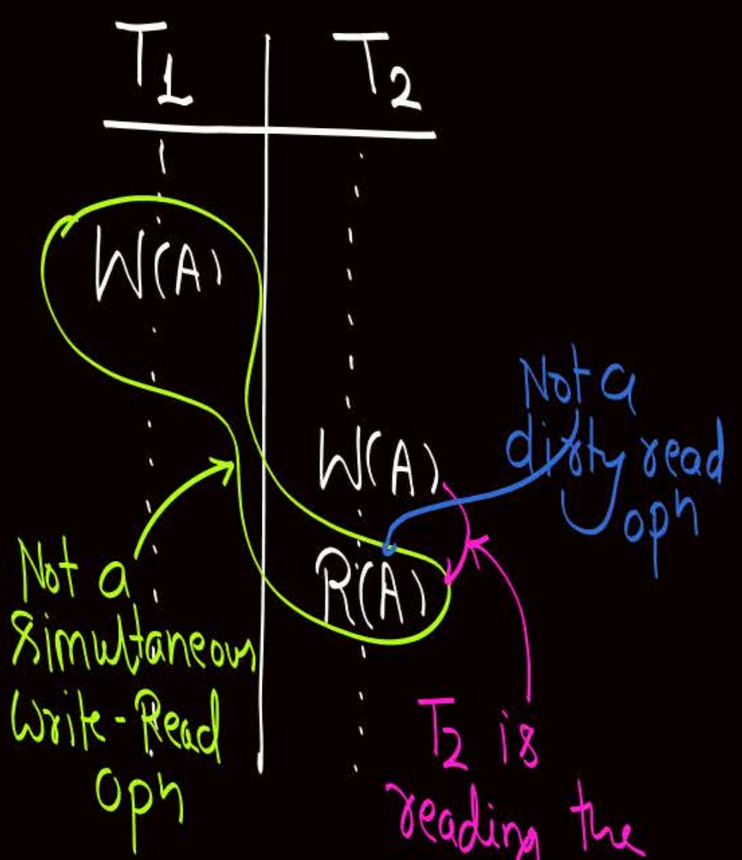
WR Problem / Write-Read Problem / Read after Write Problem
(or)
Dirty Read Problem

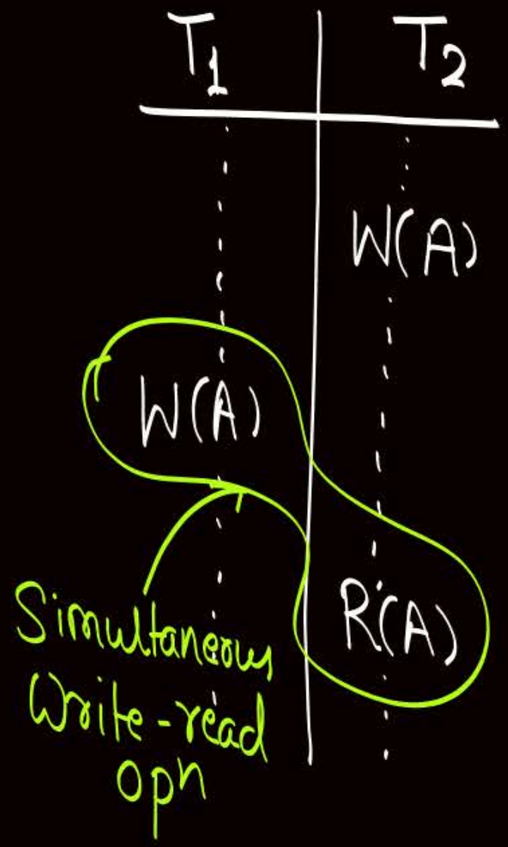


If transaction T_2 reads the value updated by an uncommitted transaction T_1, then that sequence of operation is called "Simultaneous write-read oph" and that read oph is called "dirty read" or "uncommitted read"

WR Problem / Write-Read Problem / Read after Write Problem (or) Dirty Read Problem







Topic : WR problem

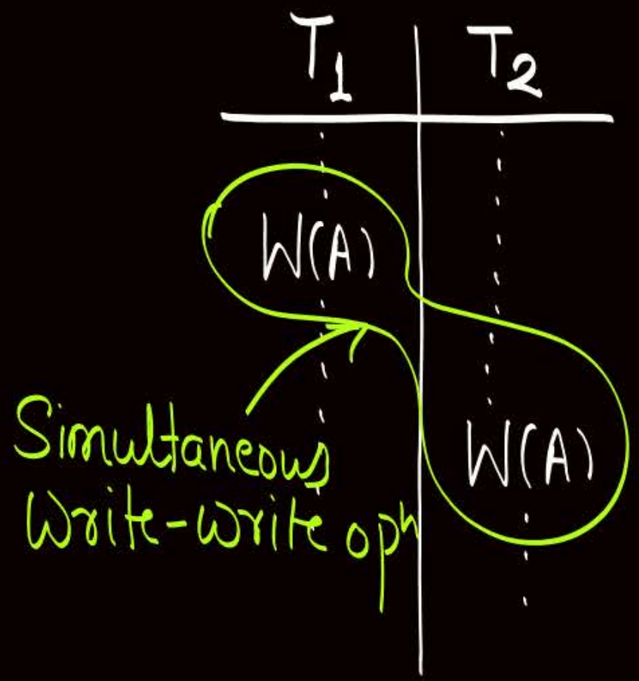
/ Read after Write Problem / Dirty Read problem

* A schedule is said to have WR problem if and only if,

① Simultaneous Write-Read op^{n} should exist
{i.e. Uncommitted Read op^{n} must exist}

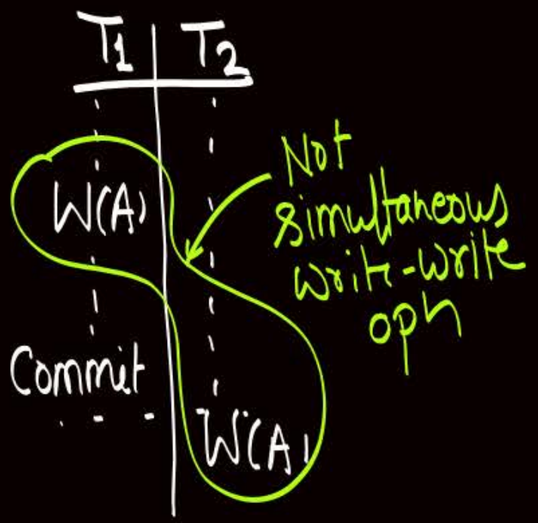
and ② Schedule must be a non-serializable schedule.

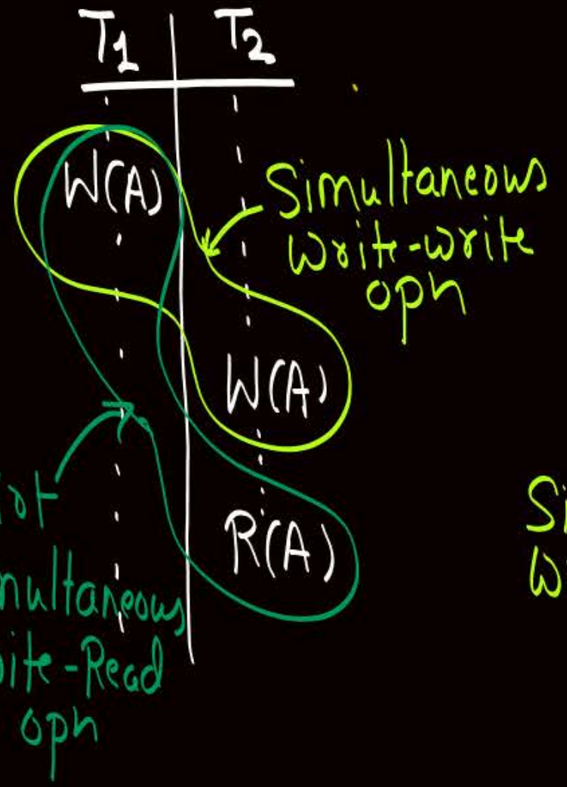
WW Problem / Write-Write Problem / Write after Write Problem

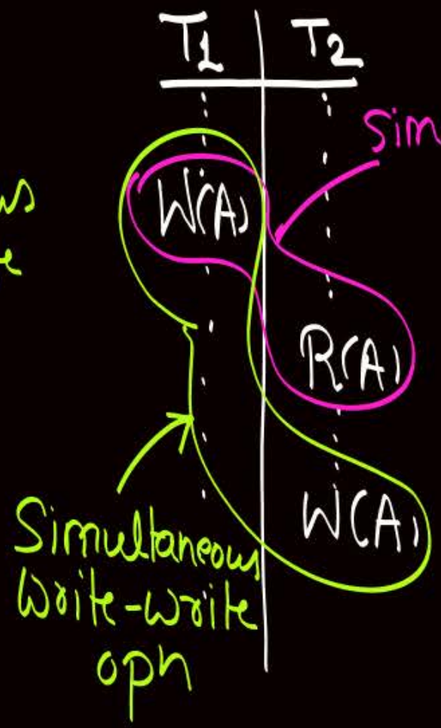


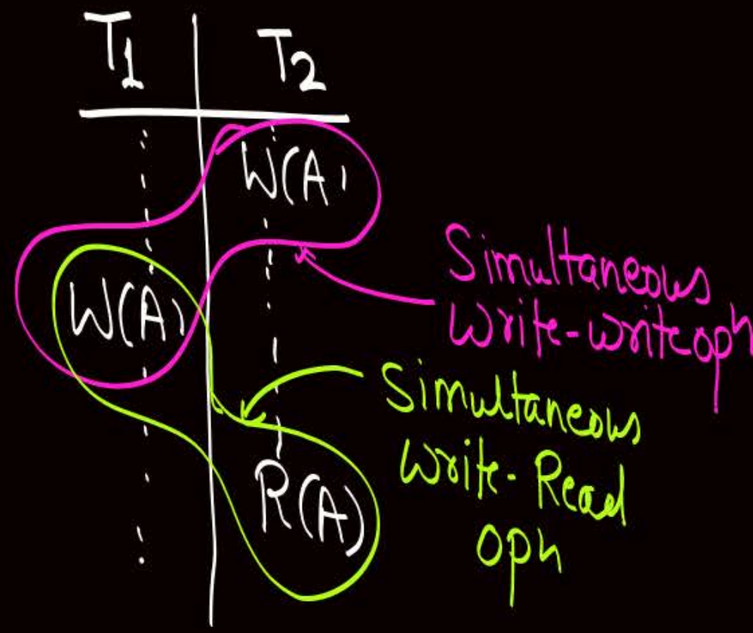
If transaction T_2 updates the dataitem 'A' which is already updated by an uncommitted transaction T_1, then that sequence of operation is called Simultaneous write-write oph.

WW Problem / Write-Write Problem / Write after Write Problem









Topic : WW problem

- A schedule is said to have WW problem if and only if,

① Simultaneous Write-Write oph should exist

and ② Schedule must be a non-serializable schedule

Topic : Problems because of concurrent execution

1. RW Problem / Read-Write Problem / Write after Read problem
2. WR problem / Write–Read Problem / Read after Write problem / Dirty Read Problem
3. WW Problem / Write-Write Problem / Write-after-Write problem
4. Lost update problem

← Lost update problem is possible with Serializable as well as non-Serializable Schedules

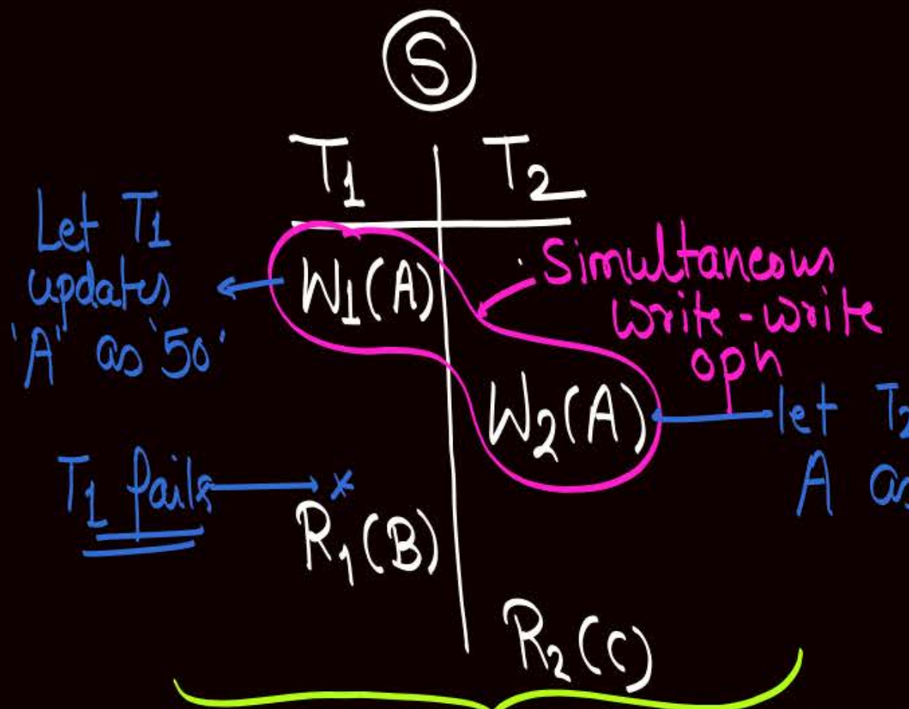
These problems are possible only if Schedule is non-Serializable

Note:- If schedule is serializable schedule, then none of RW Problem, WR problem or WW problem can exist in the schedule, but lost update problem is still possible

Topic : Lost update problem

A schedule is said to have lost-update problem if and only if simultaneous Write-Write opⁿ exists in the schedule. {does not matter whether the schedule is serializable or non-serializable}

Lost update problem



Schedule 'S' is a serializable schedule And simultaneous write-write opn exist

Let initially the value of dataitem 'A' in the database is A = 100

S: $W_1(A)$, $W_2(A)$, $R_1(B)$, $R_2(C)$

initially A = ~~100~~ ~~50~~ 200

Transaction log

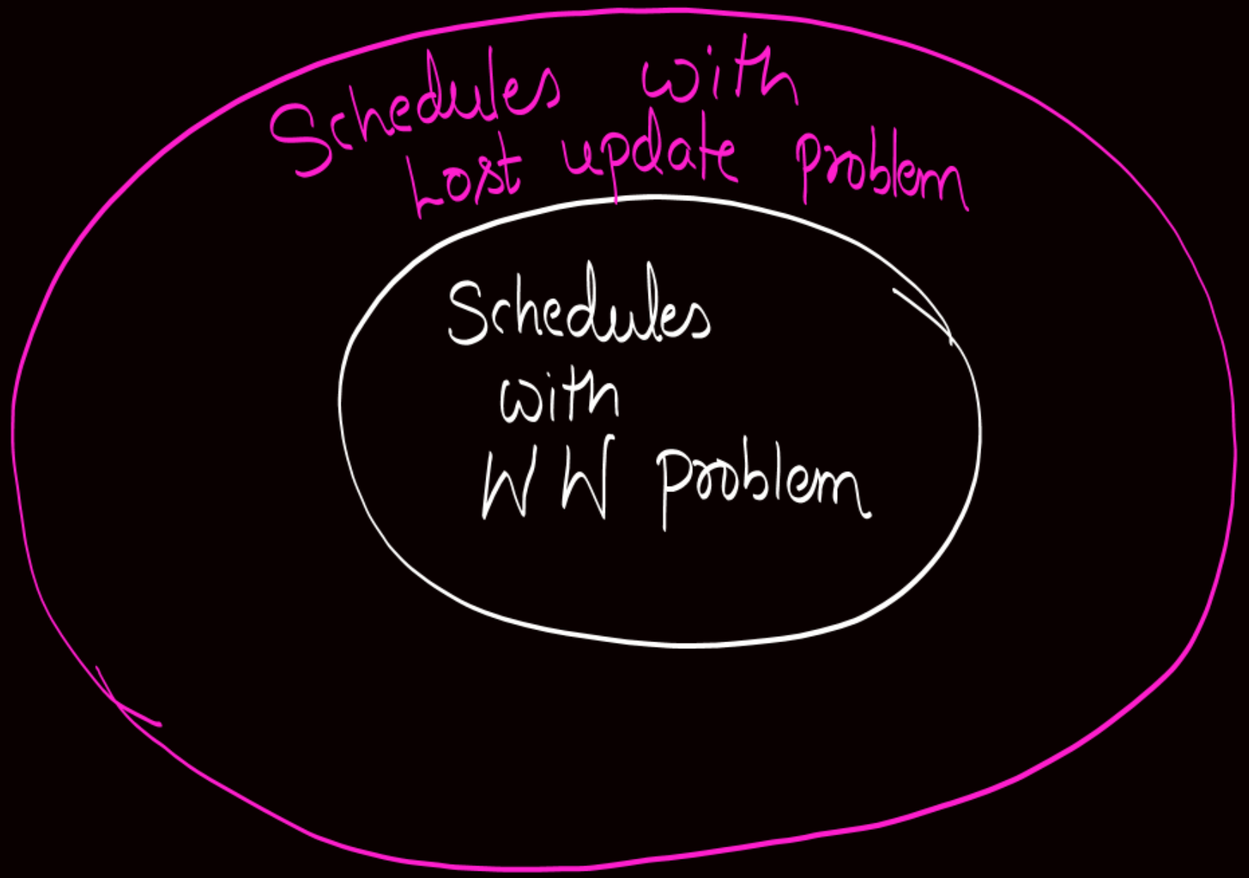
Start T_1.
$(T_1, A, 100, 50)$
Start T_2
$(T_2, A, 50, 200)$
* T_1 fails

If T_1 fails then we will roll-back T_1
∴ we will update the value of dataitem A in the database by 100 w.r.t

T_2 still thinks that value of 'A' in the database = 200 but it is not

i.e; Update of T_2 is lost

Similar problem can exist in a non-serializable schedule as well (eg: Just change $R_1(B)$ by $R_1(A)$)


Schedules with
Lost update problem
Schedules
with
W W problem

Topic : Classification of schedule

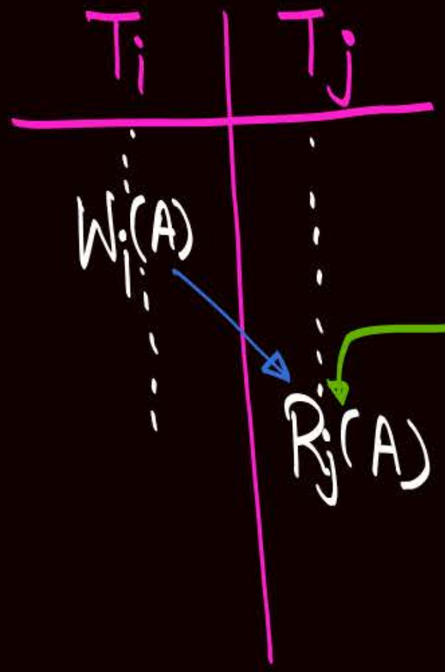
Classification based on Recoverability

1. Irrecoverable schedule
2. Recoverable schedule
3. Cascadeless rollback recoverable schedule
4. Strict recoverable schedule

Classification based on Serializability

1. Conflict serializable schedule
2. View serializable schedule

Note :- If transaction T_j reads the value written by an uncommitted transaction T_i, then transaction T_j is dependent on transaction T_i.



T_j reads the value of dataitem 'A' written by T_i
∴ T_j is dependent on T_i

If transaction T_j is dependent on T_i, and for some reason if we rollback transaction T_i, then we must rollback transaction T_j as well

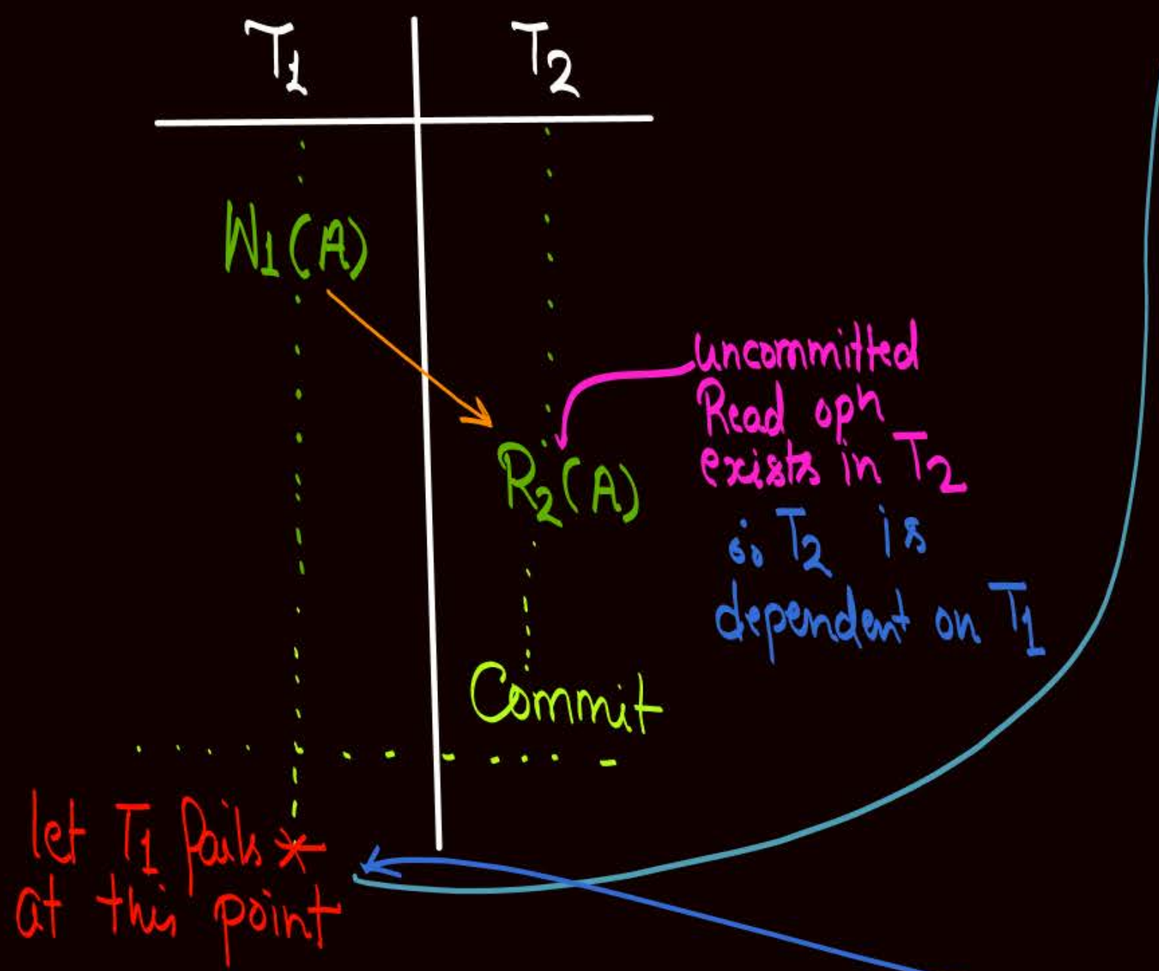
We can rollback the transaction only if it has not executed its Commit op^n successfully

We can not rollback a Committed transaction

Topic : Irrecoverable schedule

- It is not possible to rollback a committed transaction.
- If there arise a situation when we are required to rollback a committed transaction, then it will not be possible to rollback that transaction. And hence we will not be able to recover from that failure, such schedules are called irrecoverable schedule

Irrecoverable Schedule



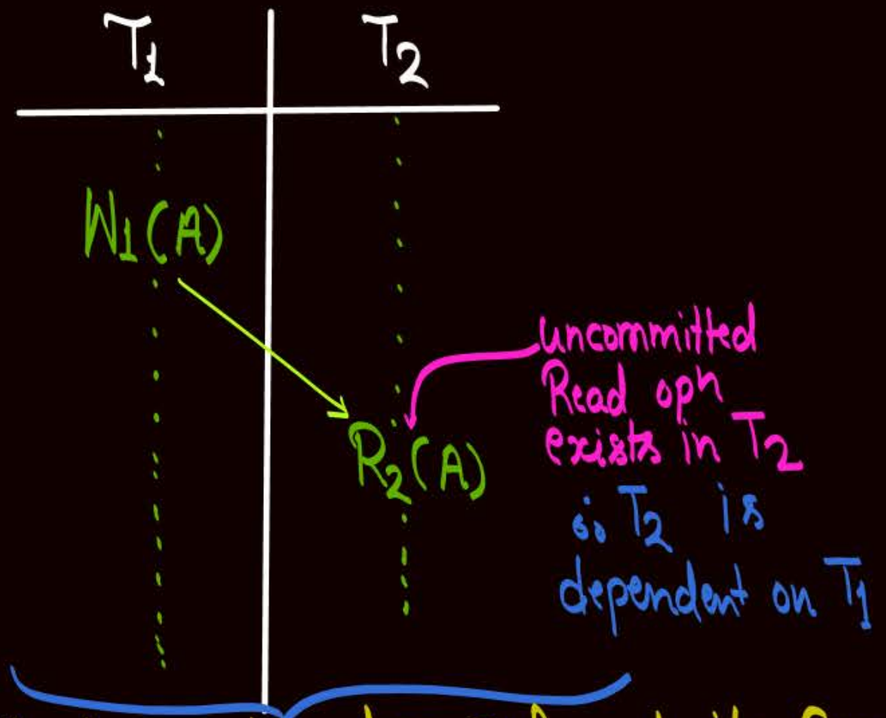
T1 failed, ∴ We will rollback T1.
Because of rollback of T1, we need to rollback all the transactions that are dependent on T1
{ in our eg: T2 is dependent on T1.
∴ We must rollback T2 as well }

But T2 has already Executed its commit oph, ∴ We can not rollback T2

Hence we can not recover from this failure

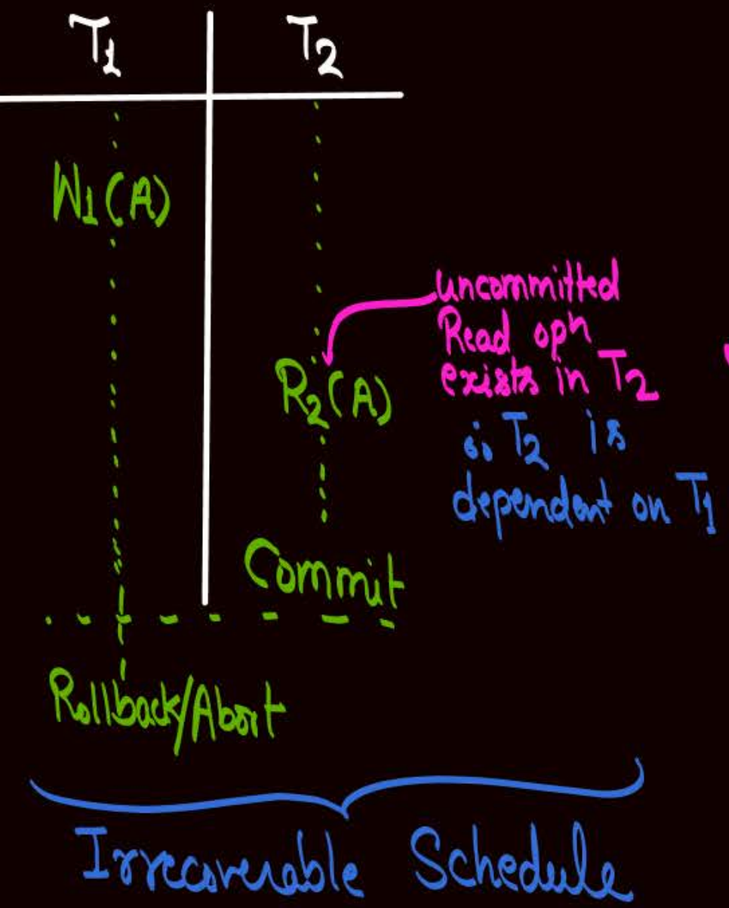
Hence schedule is irrecoverable schedule

Irrecoverable Schedule

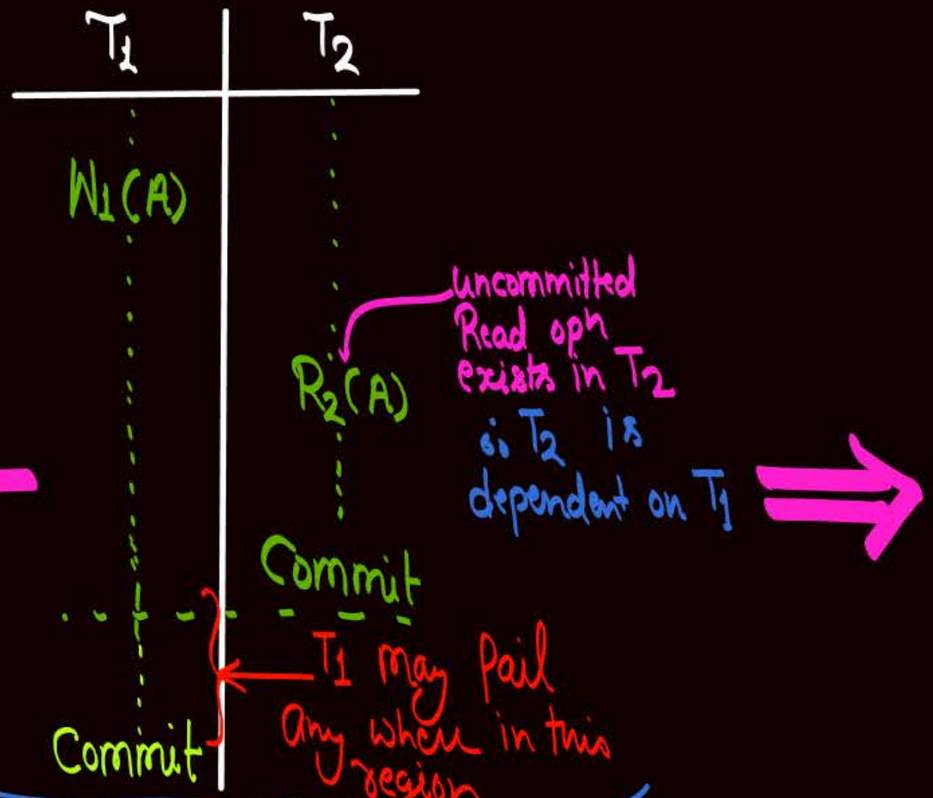


- No transaction has performed its Commit oph
  so If any failure occur, then we can rollback both transaction.
  i.e, we will be able to recover from failure
  Hence, Schedule is recoverable schedule

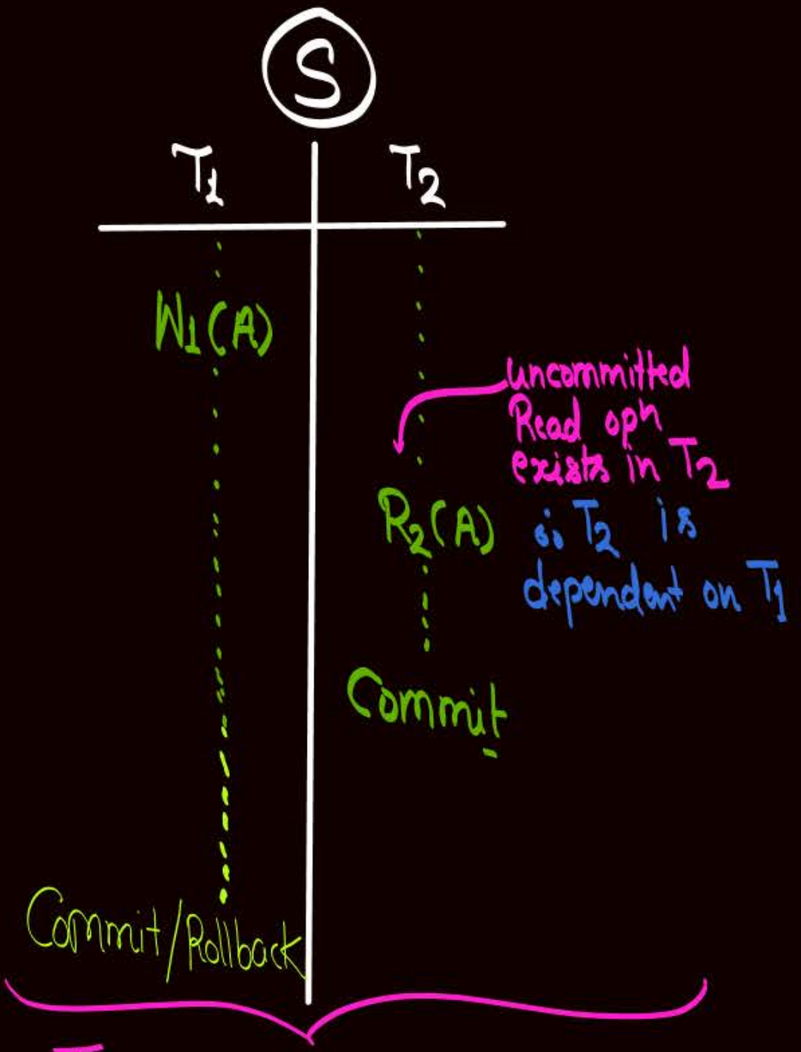
Irrecoverable Schedule



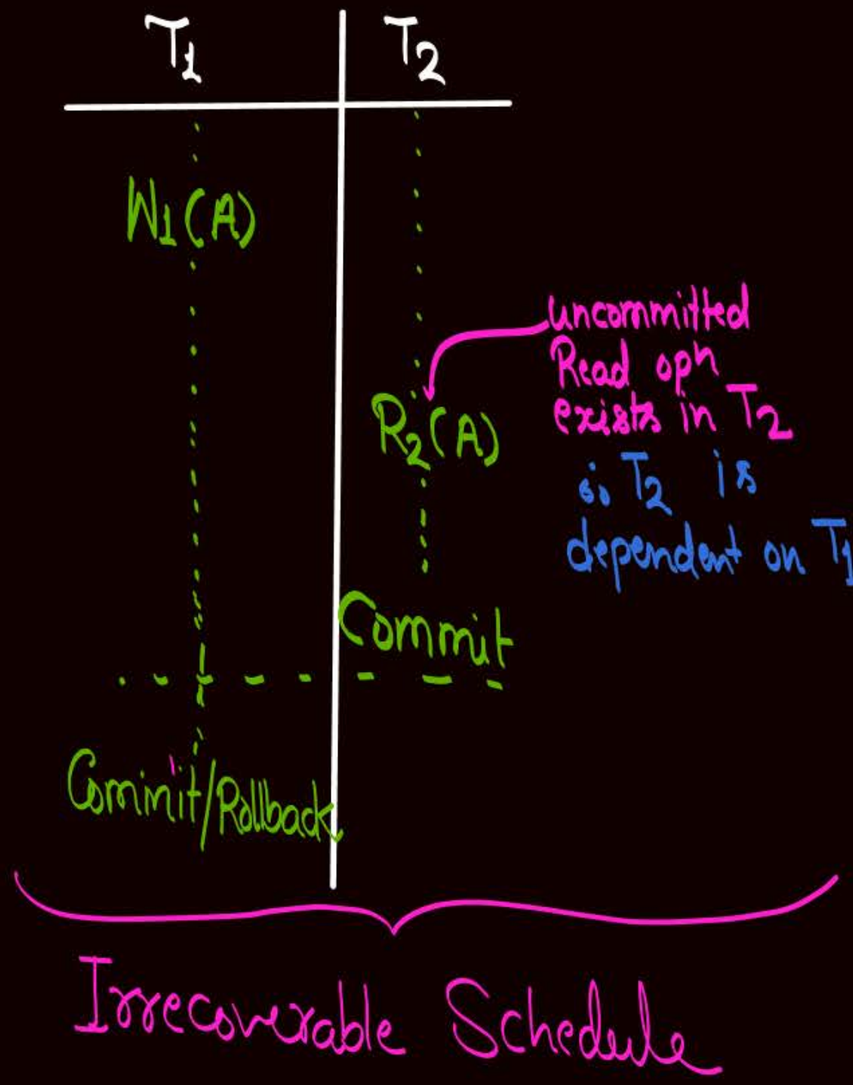
+



⇒



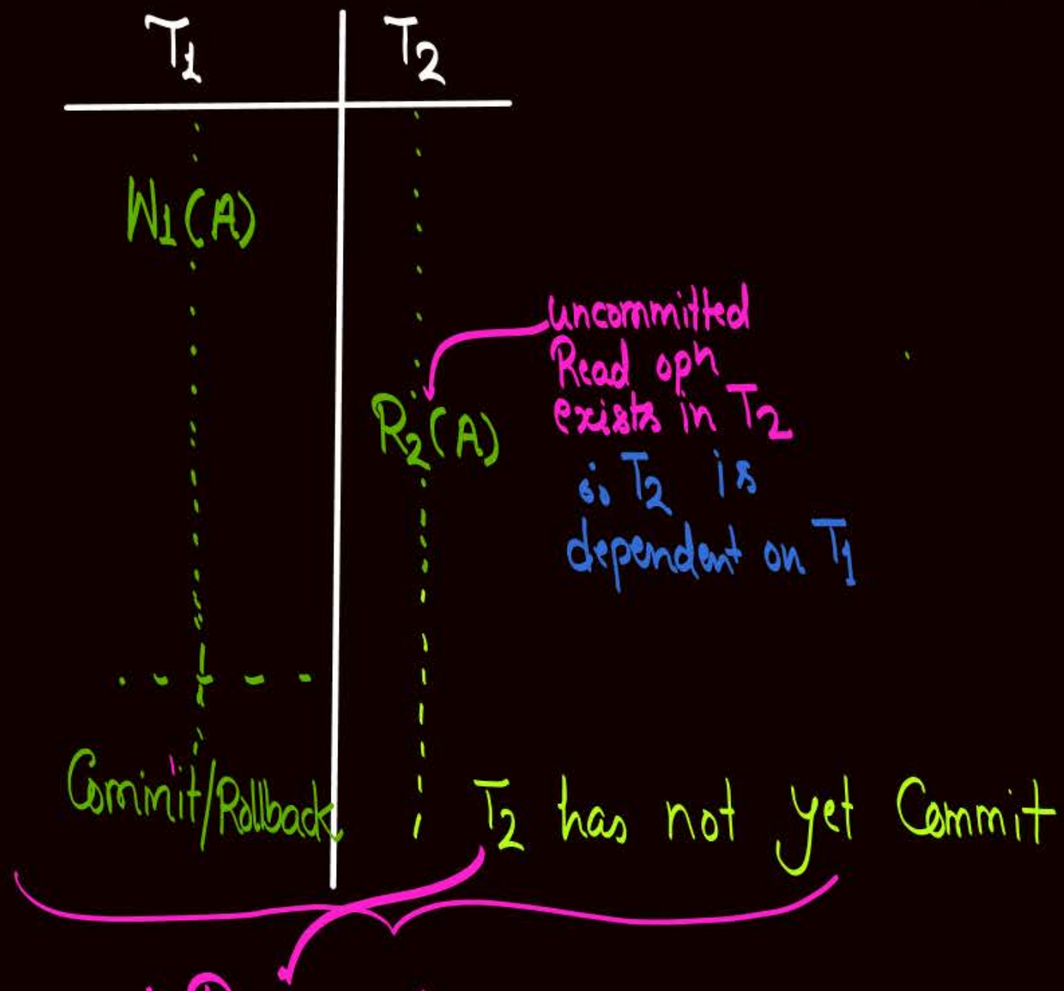
Irrecoverable Schedule



→ Let two transactions T_1 & T_2 exists in a schedule such that T_2 is dependent on T_1,

If transaction T_2 {i.e. dependent transaction} Commit before the commit/Rollback of transaction T_1 {ie, transaction on which T_2 depends}, then such schedule is called "Irrecoverable Schedule"

Irrecoverable Schedule



→ Let two transactions T_1 & T_2 exists in a schedule such that T_2 is dependent on T_1,

If transaction T_2 {i.e. dependent transaction} Commit before the commit/Rollback of transaction T_1 {ie, transaction on which T_2 depends}, then such schedule is called "Irrecoverable Schedule"

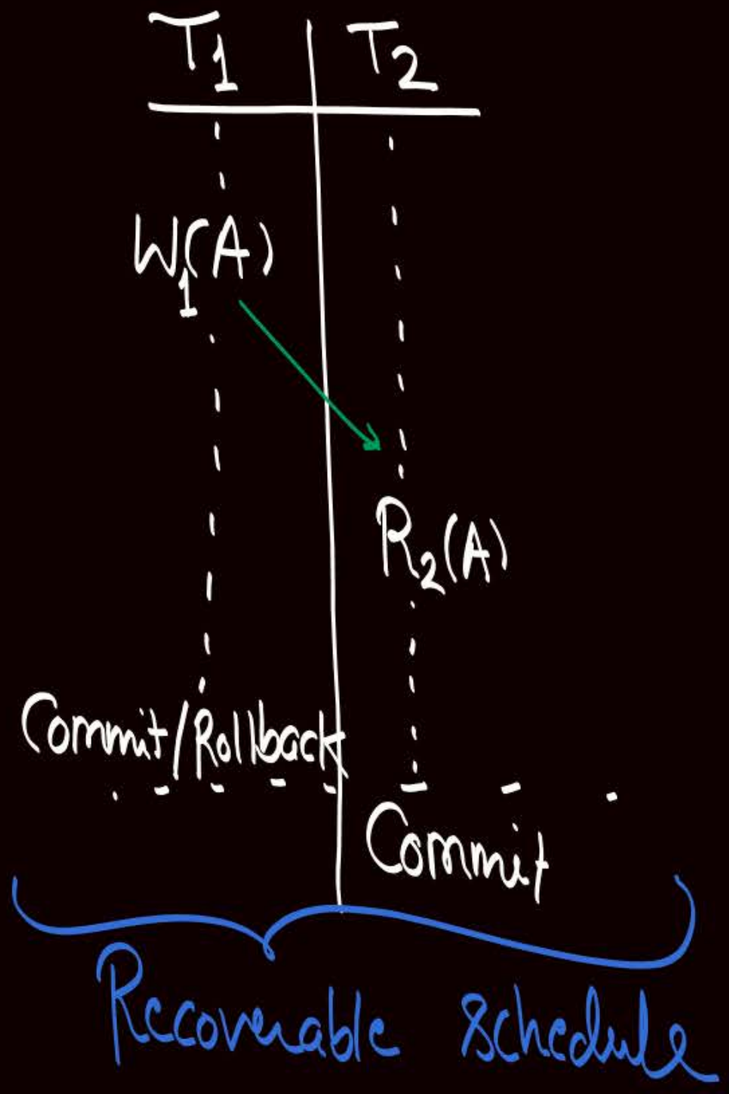
Irrecoverable Schedule

Uncommitted read oph are necessary (but not sufficient) for a schedule to be irrecoverable Schedule.

i.e, ① If uncommitted read oph does not exist in a schedule then that schedule is always recoverable

and ② If uncommitted read oph exists in a schedule then that schedule may or may not be an irrecoverable schedule.

Recoverable Schedule



For a schedule to be called recoverable schedule,

Either Uncommitted read oph should not exist in the schedule

Or if any uncommitted read oph exist, then Commit oph of dependent transaction must appear after the Commit/Rollback of the transaction on which it depends

2 mins Summary

- Topic: RW problem
- Topic: WR problem
- Topic: WW problem
- Topic: Lost update problem
- Topic: Classification of schedules

Slide

THANK - YOU

Slide