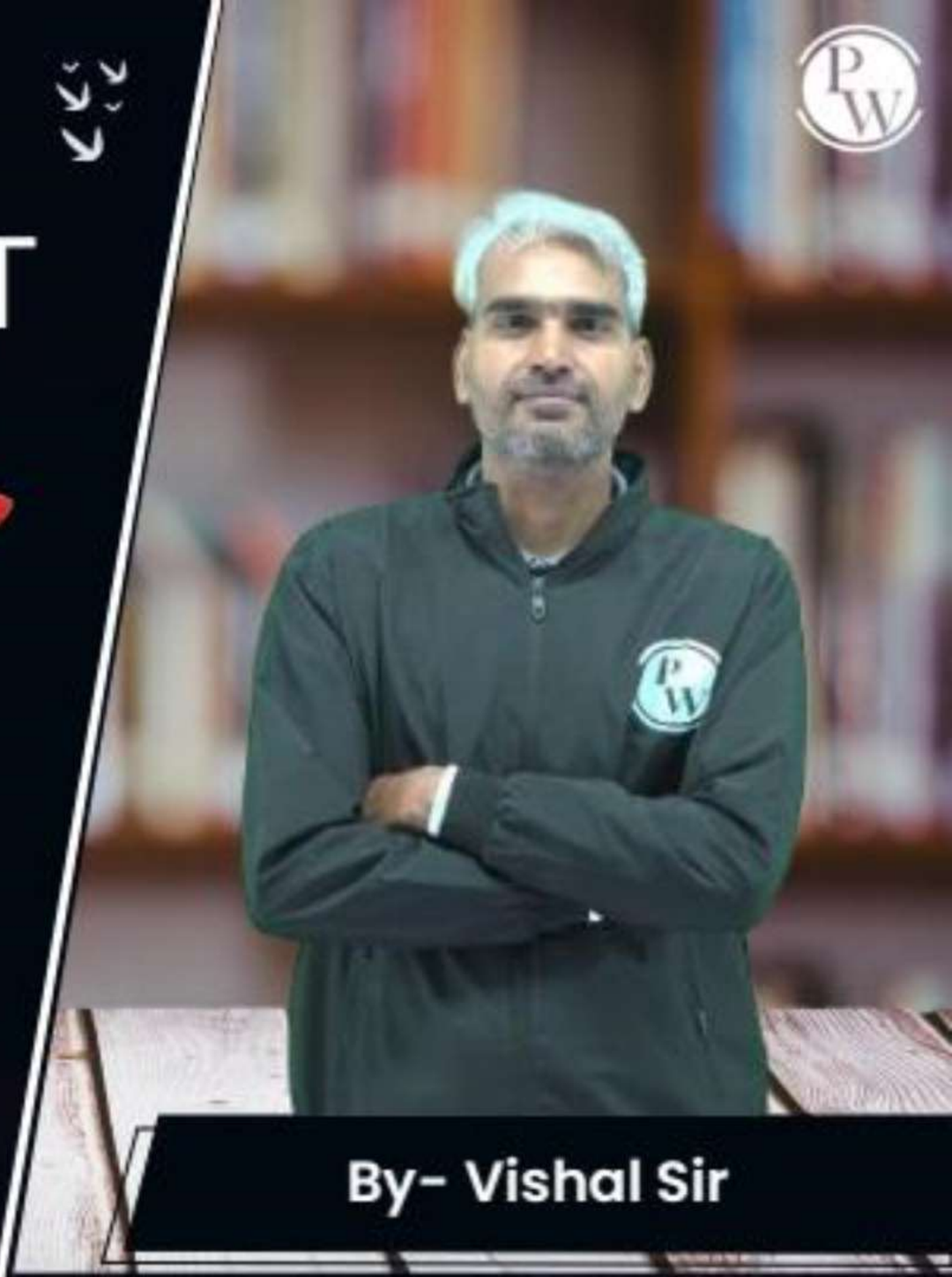
Computer Science & IT

Database Management System

Transaction & Concurrency control

Lecture No. 02



By- Vishal Sir

Recap of Previous Lecture

Topic — Domain relational calculus (TRC)

Topic — Transaction

Topics to be Covered

- Topic: ACID properties
- Topic: Atomicity
- Topic: Durability
- Topic: Isolation
- Topic: Consistency

Slide

Topic : ACID properties

- Atomicity
- Consistency
- Isolation
- Durability

for the integrity of database Transactions must follow ACID properties.

1. Atomicity
2. Durability
3. Isolation
4. Consistency

} order of discussion

Topic : Atomicity

Atomicity states that either Execute all operations of a transaction or none of them

There are two problems :

- All operations of a transaction can not be executed at once, they all will be executed in a sequence one after another
- If we start executing the operations of a transaction then we can not guarantee that the transaction will not fail before its completion

Topic : Atomicity

If we start the execution of operations of a transaction, and if transaction fails before its Completion then we say that atomicity is violated

Consider the following transaction,

Transaction: Transfer Rs 500/- from account A to account B.

Let, initial amount in account A = 1000
initial amount in account B = 0

Transaction

Read(A)

A = A-500

W(A)

R(B)

B = B+500

W(B)

Not important from database point of view

Transaction failed at any point before its completion then atomicity is violated

There may be various reasons for the failure of a transaction,

1. Power failure
2. Software Crash
3. Hardware Crash
4. Any natural Calamity
5. Concurrency Control Component of DBMS or operating System may terminate(kill) any transaction if needed

:

etc

Note :→ To ensure atomicity, if a transaction fails before its completion then "undo" the operations performed by that transaction until its failure Point { then behaviour will be same as if none of the operation of that transaction is Executed }

Note :- To ensure atomicity "Recovery Management Component" is used

Topic : Recovery management component

- Recovery management Component is responsible for "undo" and "redo" operation
  - "undo" w.r.t Failed transaction
  - "redo" w.r.t. Committed transaction
- The process of undoing the operations of a transaction is called "rollbacking" of that transaction.
- For the purpose of "undo" and "redo" Operation "transaction log" is maintained by recovery Management Component

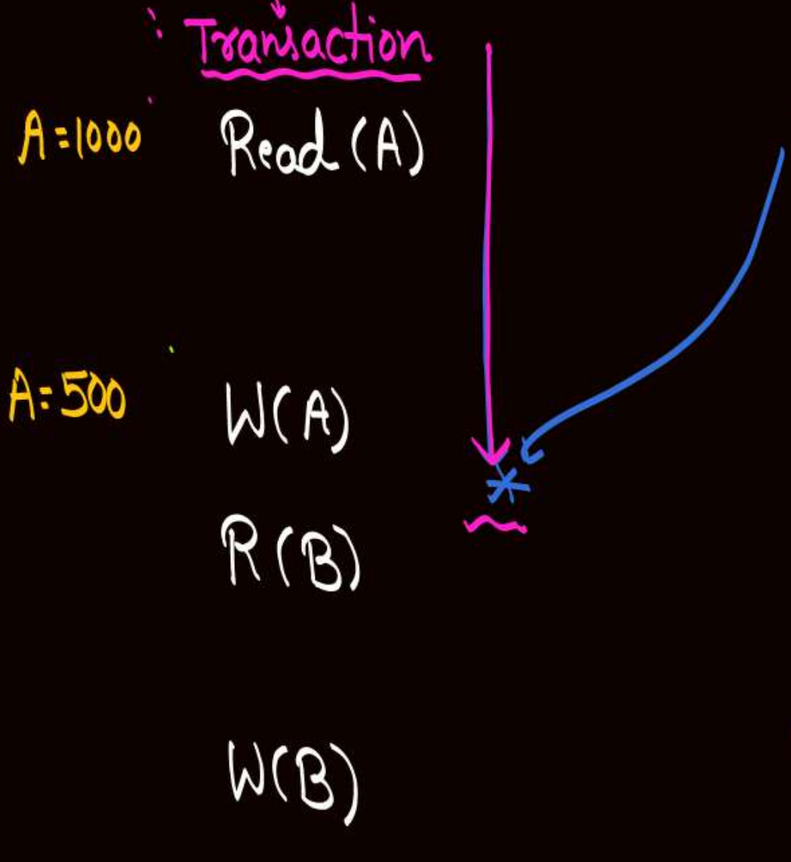
Topic : Transaction log

- Transaction log is used to record the activities performed by transactions.

Consider the following transaction,

Transaction: Transfer Rs 500/- from account A to account B.

Let, initial amount in account A = 1000
initial amount in account B = 0



Transaction failed at any point before its completion then atomicity is violated

Topic : Transaction log

- Transaction log is used to record the activities Peformed by transactions.
- Transaction log w.r.t above eg: {i.e. Transfer 500/- from A to B}

If transaction fails then recovery management Component can update the values of the dataitem in the database by old Values Present in the transaction log Corresponding to those dataitems

Transaction Log

A.old = 1000
A.new = 500

* Transaction log is managed by recovery management Component
  - We can recover from a failure only if transaction log is not lost, i.e. transaction log is stored in secondary memory.
  - Either we can use seperate transaction for every transaction, or we can use a unified transaction log for all ongoing transactions.

Topic : Durability

* Durability says that we should be able to recover under maximum cases of failure
  - For the purpose of recovery transaction log is required
    - ∴ Main objective of durability is to safe guard transaction log
  - Transaction log is stored in secondary storage.
    - ∴ for durability secondary storage design must be more robust

For more information you can read about RAID architecture.

RAID

→ Redundant array of independent disk:

- RAID-0
- RAID-1
- RAID-2
- etc.

Topic : Isolation

- Isolation states that if two or more transactions are executing Concurrently, then they all must be unaware of each-other.

- Consider two transactions

T_1: Transfer Rs 500/- from account A to B

T_2: Read Current amount in 'A' then in 'B'

Not important from Database point of view

A : A - 500

B : B + 500

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

T_2
R(A)
R(B)

Topic : Schedule

Time ordered sequence of operations of two or more transaction is called schedule

Topic : Schedule

Time ordered sequence of operations of two or more transaction is called schedule

- T_1: R(A), W(A), R(B), W(B)
  T_2: R(A), R(B)

$R_i(A)$: Read of data item 'A' by transaction T_i

$W_j(B)$: Write of dataitem B by transaction T_j

Schedule "S_1"

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

time in increasing order

Schedule S_2

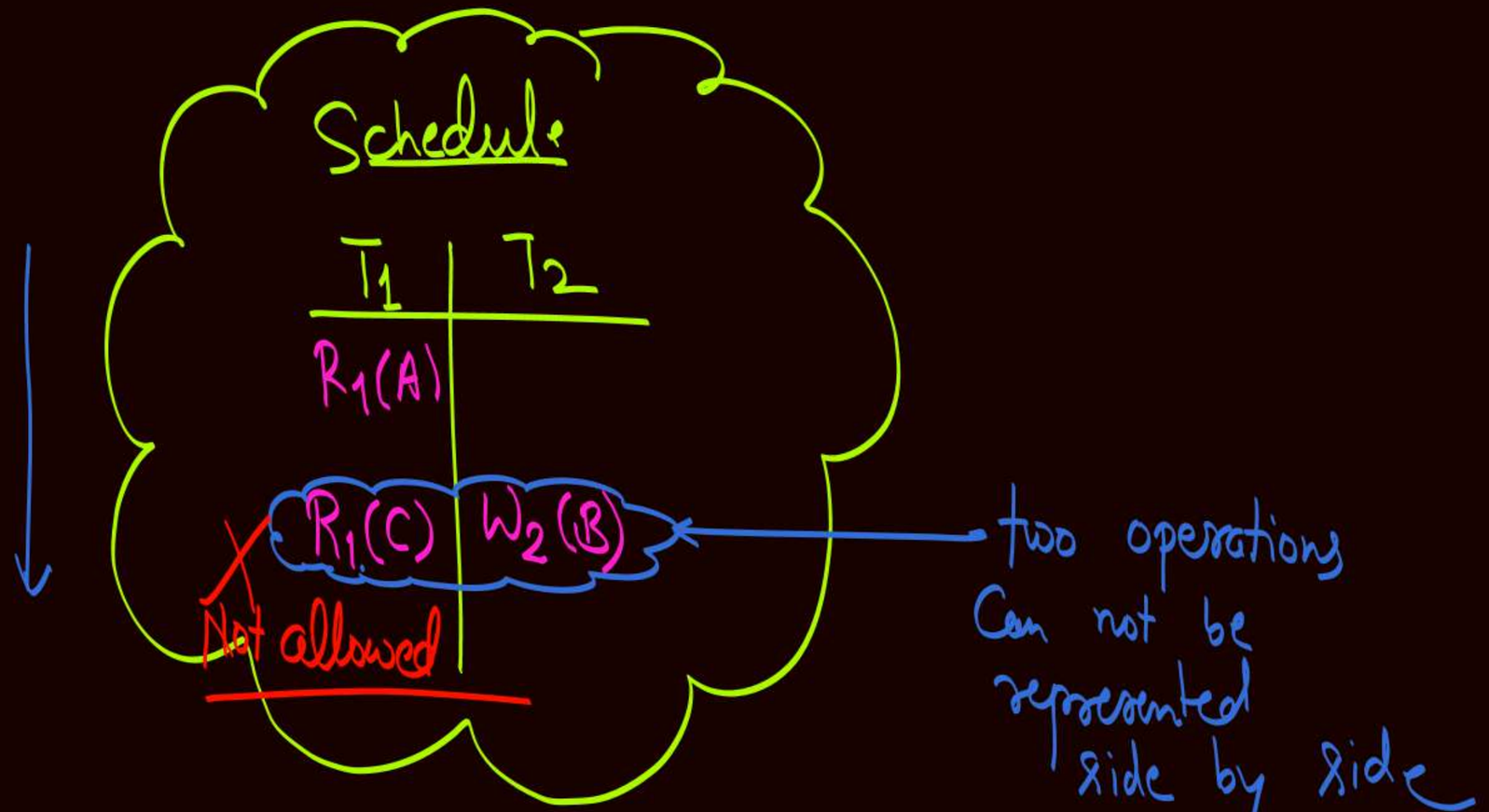
T_1	T_2
$R_1(A)$	
$W_1(A)$	
	$R_2(A)$
$R_1(B)$	
$W_1(B)$	
	$R_2(B)$

S_2: $R_1(A)$, $W_1(A)$, $R_2(A)$, $R_1(B)$, $W_1(B)$, $R_2(B)$

S_3: $R_1(A)$, $W_1(A)$, $R_2(A)$, $R_2(B)$, $R_1(B)$, $W_1(B)$

Schedule S_3

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


Schedule
T1
T2
R1(A)
R1(C)
W2(B)
Not allowed
two operations
Can not be
represented
side by side

Topic : Schedule

There are two types of Schedules

1. Serial Schedule { one after another }
2. Concurrent Schedule { interleaved }

Topic : Serial Schedule

It states that. Start the execution of a new transaction only after the Complete execution of Previously started transaction

eg:- T_1: $R_1(A)$ $W_1(A)$ $R_1(B)$ $W_1(B)$

T_2: $R_2(A)$ $R_2(B)$

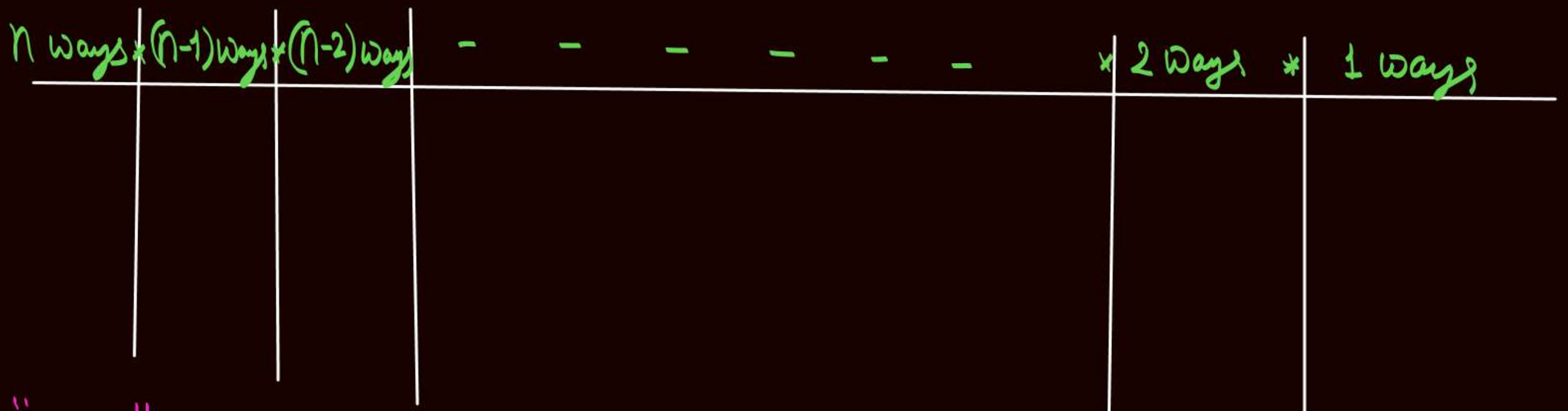
① Serial schedule T_1 then T_2 $(T_1 \rightarrow T_2)$

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

② Serial schedule T_2 then T_1 $(T_2 \rightarrow T_1)$

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

Note:- With 'n' transactions how many different serial schedules are possible



"n!" different serial schedules are possible with 'n' transactions

Note:-

Serial schedules will always satisfy the isolation condition {Because operations of transactions are not executing in interleaved manner, i.e. there is no reason that isolation is dis-satisfied}

Note: Serial schedule will always satisfy the isolation condition. But throughput will be very low with serial schedules.

throughput → No. of transactions Completed per unit of time

2 mins Summary

- Topic: ACID properties
- Topic: Atomicity
- Topic: Durability
- Topic: Isolation
- Topic: Consistency

THANK - YOU

Slide