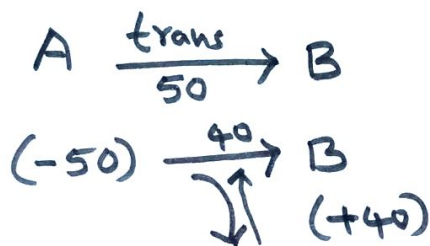


Transaction management

Consistency & Isolation



user program's logic

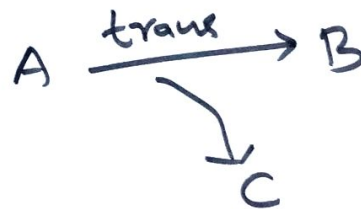
T1 & T2 Concurrently
 $\left. \begin{array}{l} T1; T2 \\ T2; T1 \end{array} \right\}$ should be equivalent.

ACID properties

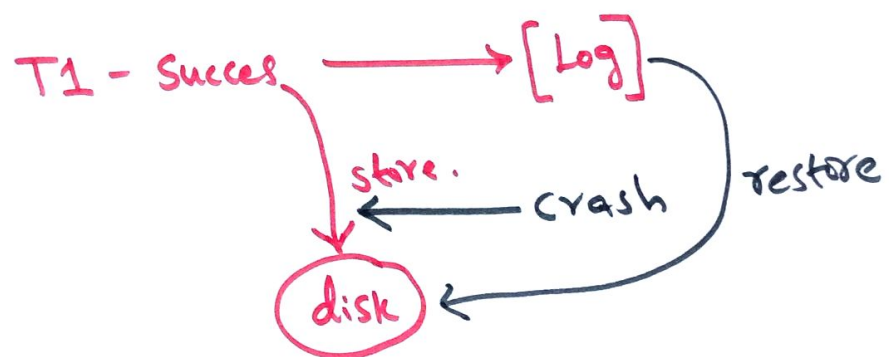
Atomicity & Durability

Transaction incomplete

- Aborted
- system crash
- read unexpected data.
unable access disk.



[log] - Used remember and restore.



Transactions & Schedules

Transaction: List of actions.

Actions: read, write of database object.

O - object, T - transaction.

$R_T(O)$ - action of T is read object O

$W_T(O)$ - action of T is write object O .

Transaction must specify: **Commit**
Abort

Schedule (read, write, abort, commit)

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

Serializability

T1; T2

T2; T1.

T1 & T2 Conflict each other:

→ WR

→ RW

→ WW

T1 : transfer Rs. 100 from A to B ^{200 300}
 T2 : increment both A & B by 6 %

T1; T2

100 A - 100 } → A: 106
 400 B + 100 } → B: 424

T2; T1

A: 212 } ⇒ A: 112
 B: 318 } ⇒ B: 418

T1	T2
A: 200 R(A)	
A: 100 W(A)	
	R(A) A: 100
	W(A) A: 106 ←
	R(B) B: 300
	W(B) B: 318
	commit
B: 318	R(B)
B: 418	W(B)
	commit

Unrepeatable reads (RW Confut)

A - No. of copies available for a book.

integrity constrain $A \geq 0$.

T : places an order, first read A
then check $A > 0$
then decrement 1.

T1: R(A) and sees $A=1$

T2: R(A) and sees $A=1$

W(A), decrement by 1, $A=0$
commit

T1: W(A) and decrement by 1, $A=-1$

Overwriting Uncommitted Data (WW conflict)

A & B are two employees, their salaries must be kept equal.

T₁: sets A and B salaries Rs. 1000

T₂: sets A and B salaries Rs. 2000

T₂ : W(A)

T₁ : W(B)

T₂ : W(B) - commit

T₁ : W(A) - commit

A	B
2000	1000
1000	2000 ←

↑

Schedule Involving Aborted Transactions

T1	T2
R(A) W(A)	R(A) W(A) R(B) W(B) Commit
Aborted	

T1: Deduct Rs. 100 from A
T2: Read A & B
add 6%
Commit.

T1: Aborted.

strict Two-phase Locking
(strict 2PL)

Unrecoverable schedule.

Strict 2PL :

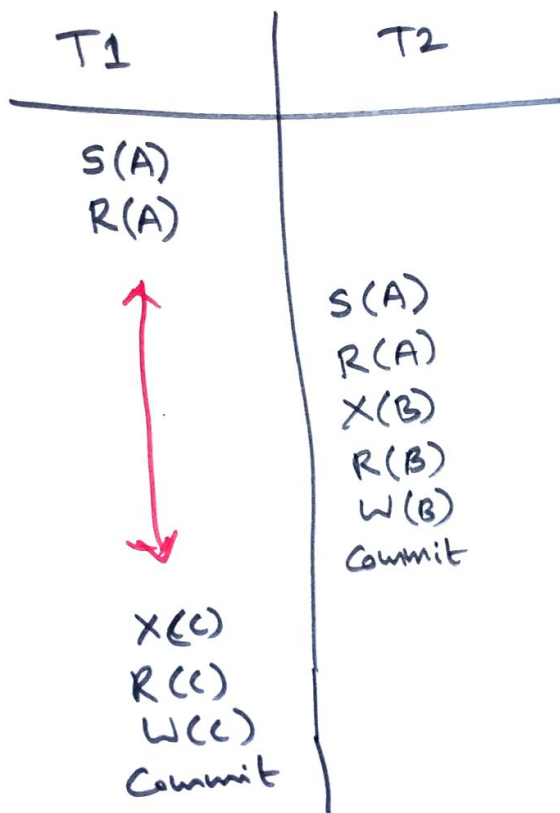
Lock $\left\{ \begin{array}{l} \text{shared lock } S_T(o) - \text{read} \\ \text{exclusive lock } X_T(o) - \text{write, read} \end{array} \right.$

Ex:

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

T1	T2
X(A) R(A) W(A)	Request X(A) Not allowed T2 suspend.
X(B) R(B) W(B) Commit	
	X(A) R(A) W(A) X(B) R(B) W(B) Commit

Ex:



Deadlock

