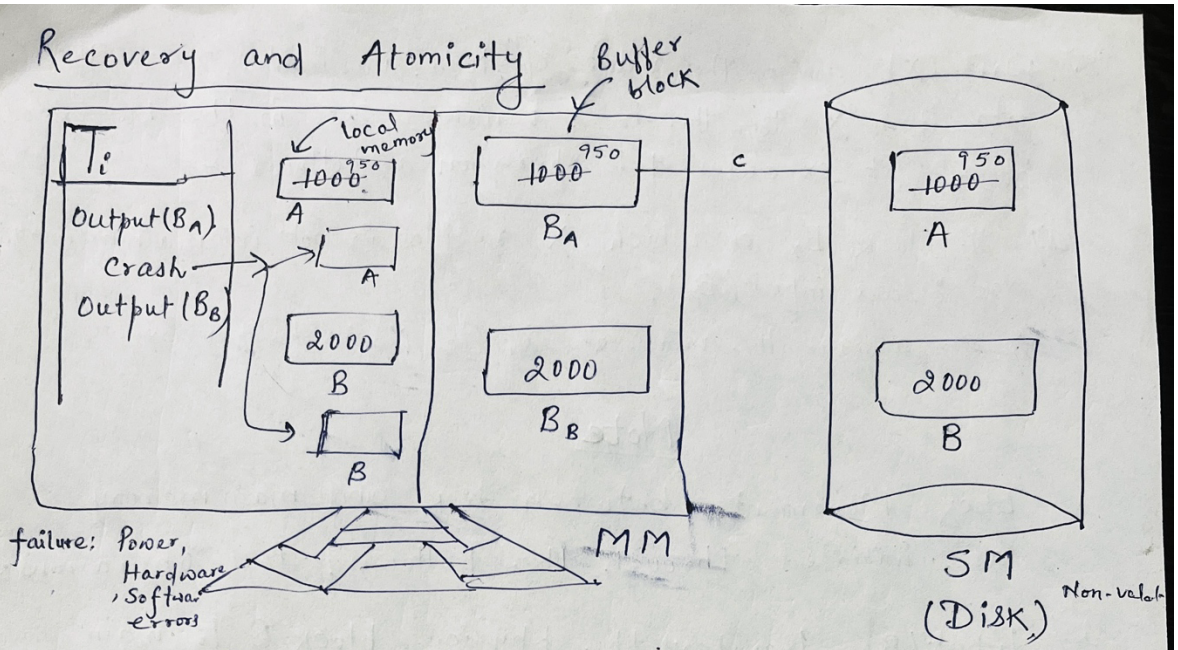
Recovery and Atomicity

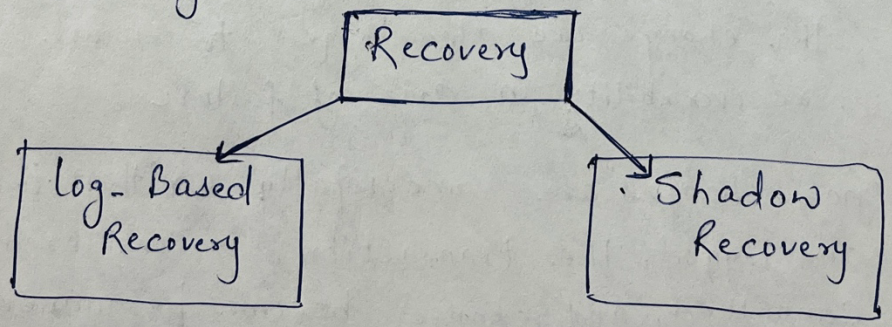


Recovery Procedure

i) Reexecute T_i : Output(B_A) 900, Output(B_B) 2050 — Inconsistent

ii) Do not Reexecute T_i : A : 950, B : 2000 — Inconsistent

So, Atomicity is our goal.



2.) Write(X) assigns the value of local variable x_i to ~~the~~ data item X ~~to the local variable x_i~~ in the buffer block. It executes this operation as follows:-

a) If block B_X on which X resides is not in main memory, it issues input(B_X).

b) It assigns the value of x_i to X in buffer B_X

Note

Block Movements between disk and main memory are initiated through the following two operations:-

1.) Input(B) transfers the physical block B to main memory.

2.) Output(B) transfers the buffer block B to the disk, and replaces the appropriate physical block there.

If the DBMS uses a logging mechanism for transaction management (eg. write-Ahead logging), the changes are also logged to ensure recoverability in case of failure.

- Once changes are successfully written to disk and logged, the transaction is considered committed, and changes become permanent.

Actual Process

(1) write (A)
(2) commit (A) → logged written on disk
(3) write to Disk → data permanent on disk

Let

S1:

T1	T2
R(X)	
R(Y)	
	R(X)
	R(Y)
	W(Y)
W(X)	

→ not Serializable

S2:

T1	T2
(R(X))	
	R(X)
	R(Y)
	W(Y)
R(Y)	
W(X)	

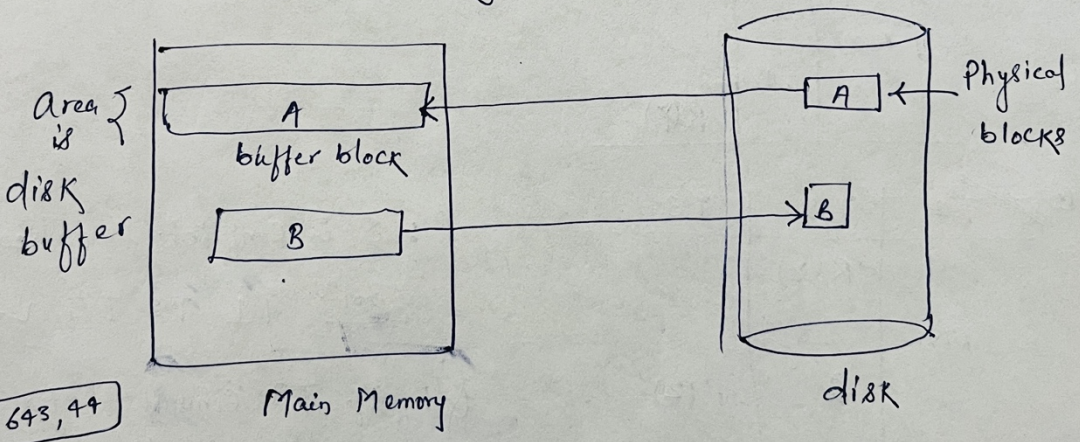
Serializable

T2 T1 ✓

Recovery System

- RAID
- Disk or Magnetic tapes are electromechanical] Non volatile

Volatile are mostly based on chips.

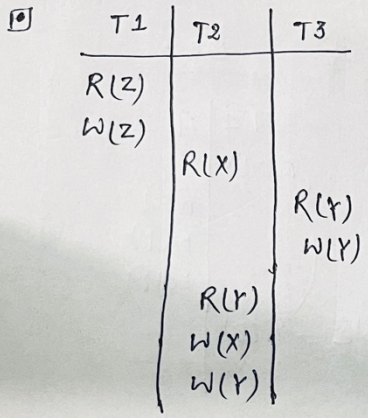


Page 643, 44

1) Read (X) assigns the value of data item X to the local variable x_i, and it executes this operations only as follows: —

a) If Block B_x on which X resides is not in the main memory, it issues input(B_x).

b) It assigns to x_i the value of X from the buffer block.



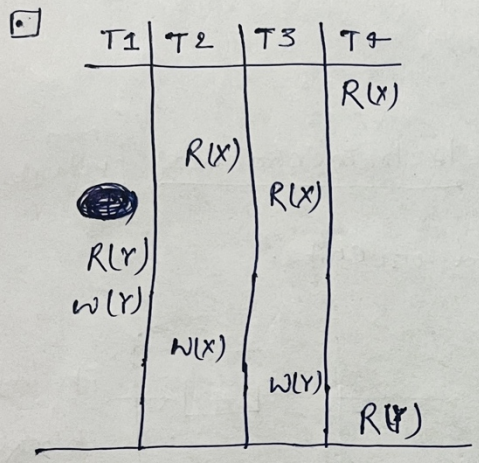
Conflict equivalent

A) T1 T2 T3 ✗

B) T1 T3 T2 ✓

C) T3 T2 T1 ✓

D) T3 T1 T2 ✓



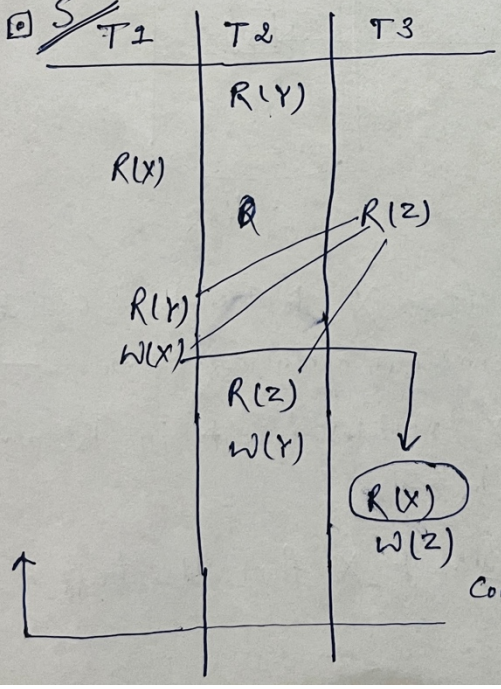
Conflict Equivalent –

A) T1 T3 T4 T2 ✓

B) T1 T4 T3 T2

C) T4 T1 T3 T2

D) T3 T1 T4 T2



P: S is conflict serializable

Sol": T1 T2 T3 ✓

Q: If T3 commits before T1 finishes, then S is recoverable.

Sol": Irrecoverable because T3 is reading X which is written by T1

P is true Q is false

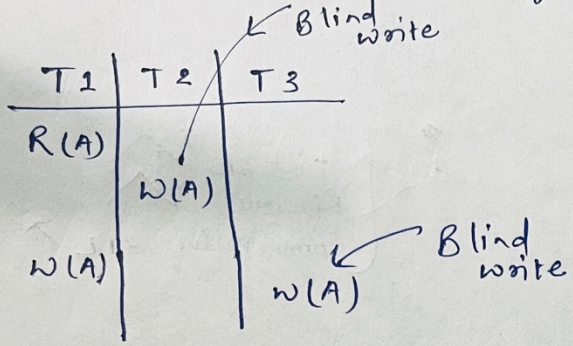
Transaction Concurrency Control

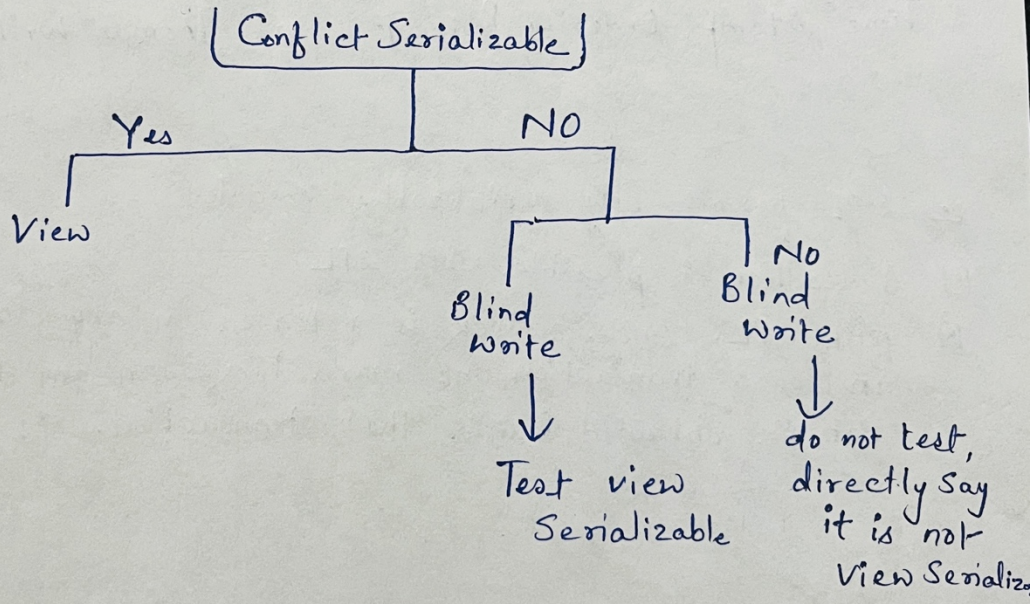
[Conflict Serializable Schedule] ⟷ (vs) [View Serializable Schedule]

if (Blind Write)

⇓

means directly writing without Reading.





DBMS

1) Irrecoverable

T_1	T_2
R(A)	
W(A)	
⋮	R(A)
	Commit

failed

2) Recoverable

T_1	T_2
W(A)	
	R(A)
C/R	
	C

1) Relational Algebra is procedural Query lang. and Relational calculus is a non-procedural Query lang.

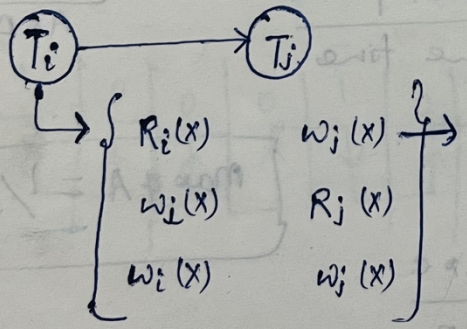
3) Cascadeless Recoverable

T_1	T_2
W(A)	
C/R	
	R(A)
	C

4) Strict recoverable

T_1	T_2
W(A)	
C/R	
	R(A) / W(A)

Conflict Serialisability



Cyclic then not Conflict SS.

Acyclic then Conflict SS.

View Serialisability

1) Initial read

2) Final updation

3) Write followed by read