

Recovery Management

→ Memory : Store data

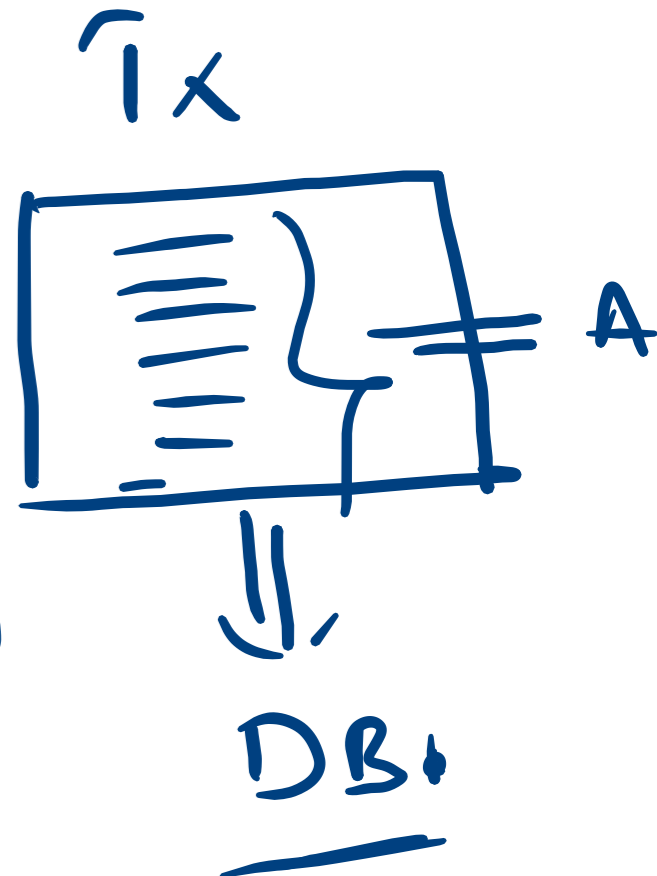
① Primary/main memory : volatile memory
(DB tr. operations)

② Secondary memory : Non volatile memory
(DB permanent)

③ Stable Storage : Theoretical

"data never lost"

RAID



Transaction failure:

Tx. Success: REDO ✓ (Next State)
Tx. failure: UNDO ✓ (Previous) } DB consistent

Causes of
Tx. failure

- ① System failure/crash : Power, n/w
- ② Transaction/System Error: division by zero, range overflow.
- ③ concurrency control enforcement: $\hat{T}_1 - \hat{T}_2 - \hat{T}_3$

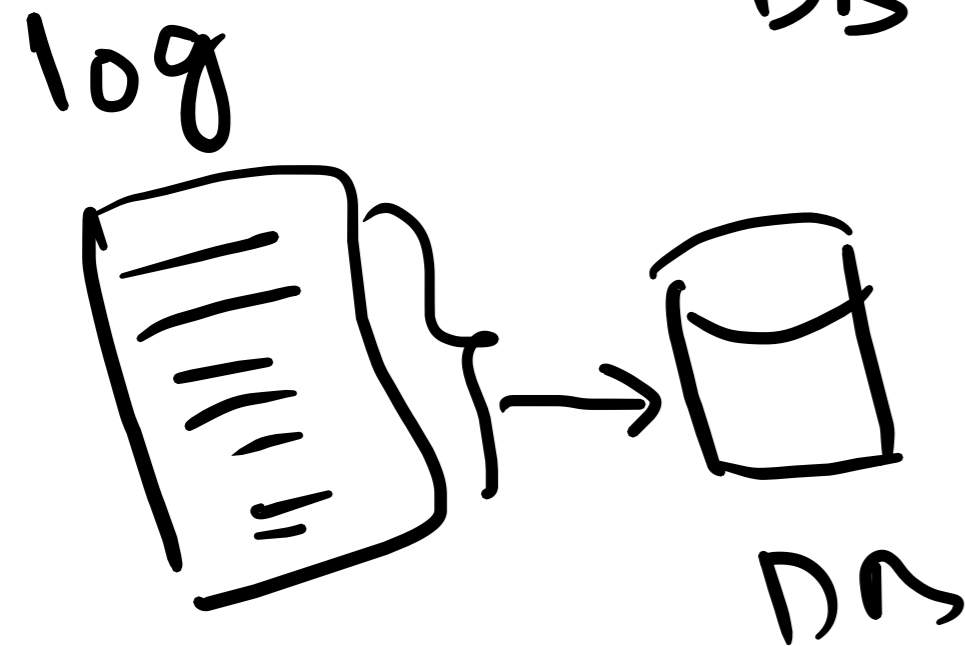
Hard
Recovery

- ④ Disk failure
- ⑤ Physical problem or Catastrophes

Recovery :

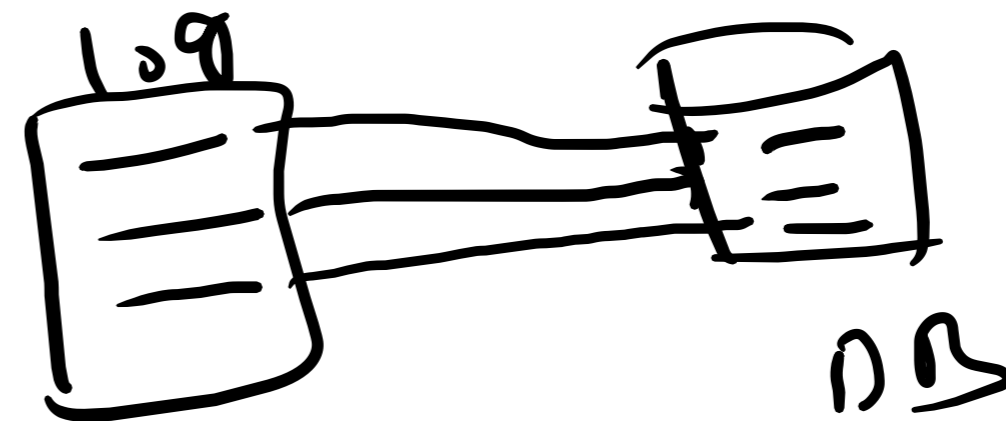
- 1) Shadow Paging ✓
- 2) Log Based Alg's.

Deferred
DB modification



ix.

Immediate
DB modification.



Shadow Paging (DB Recovery)

update

Account: A

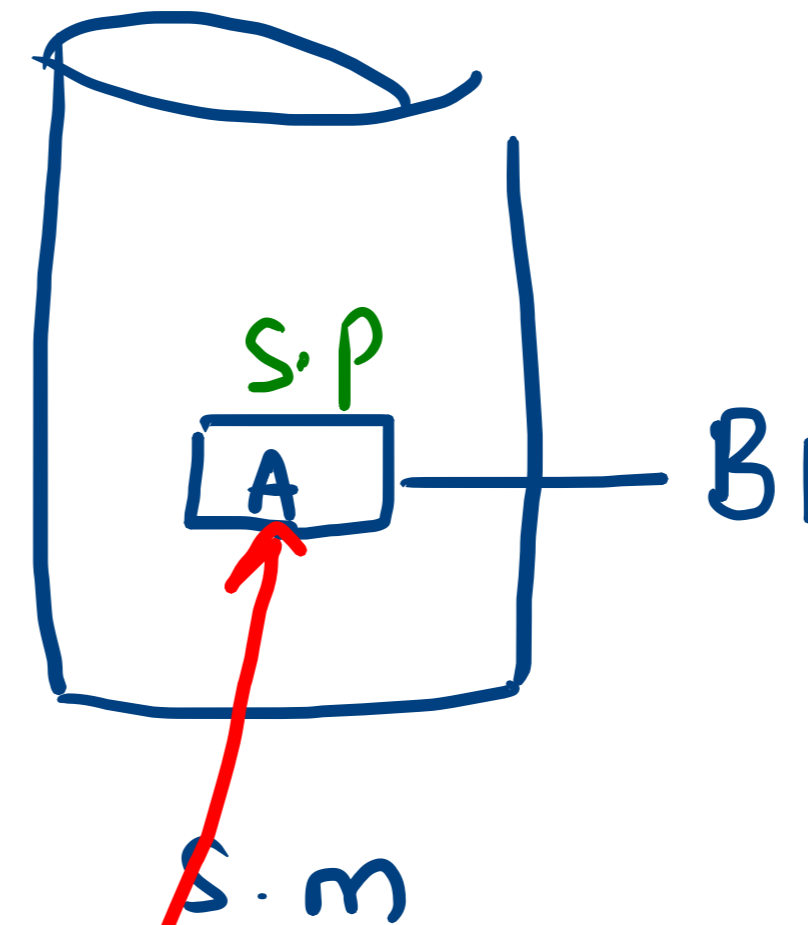
Pagetable

Shadow

Page

(A)

←→ initially
C.P & S.P both are same.



Tables
↓
os files
↓
Secondary memory (blocks)

~~X~~ (M/M)
Current
Page
(A)



C.P

S.P

A C.P

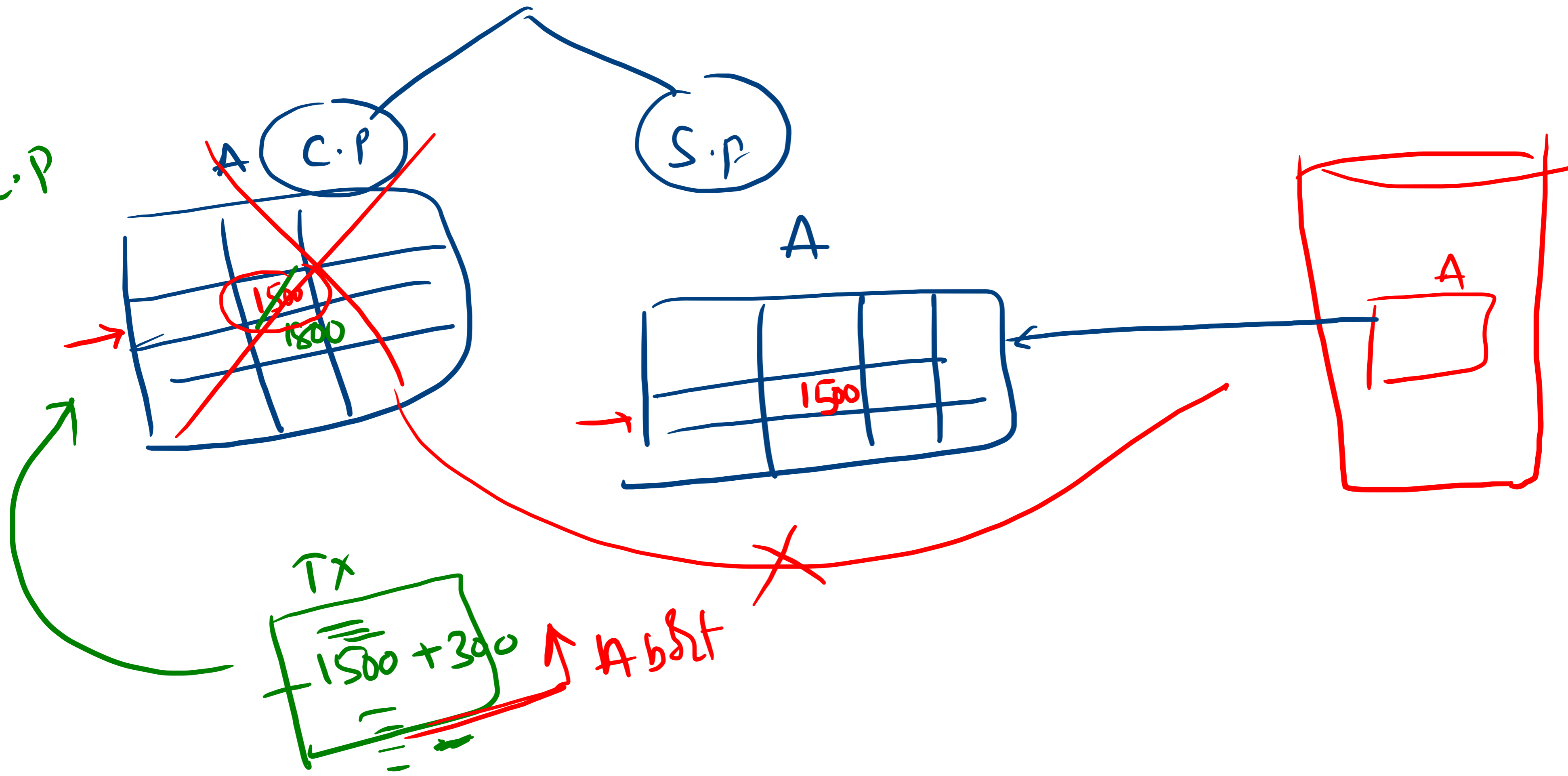
A

A

Tx

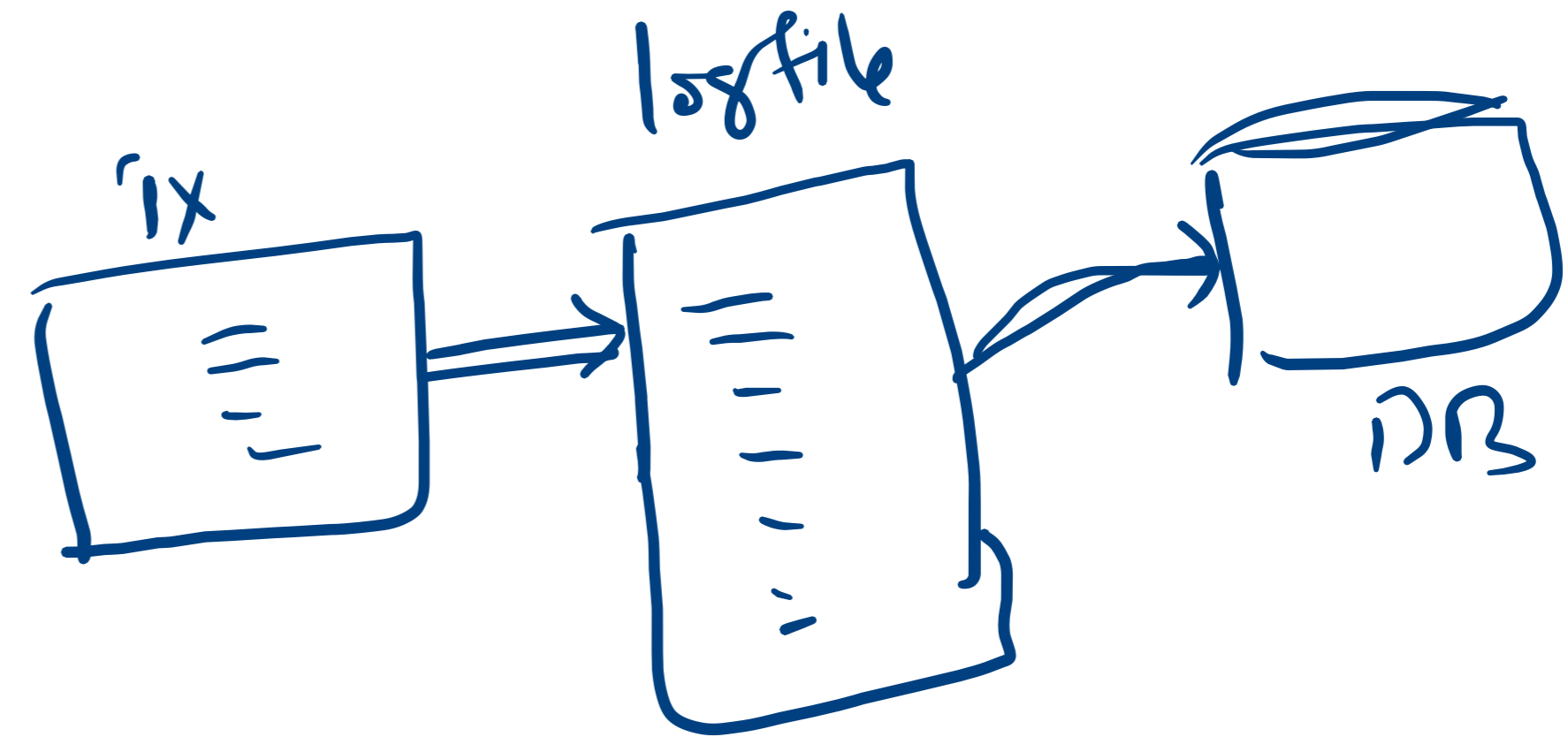
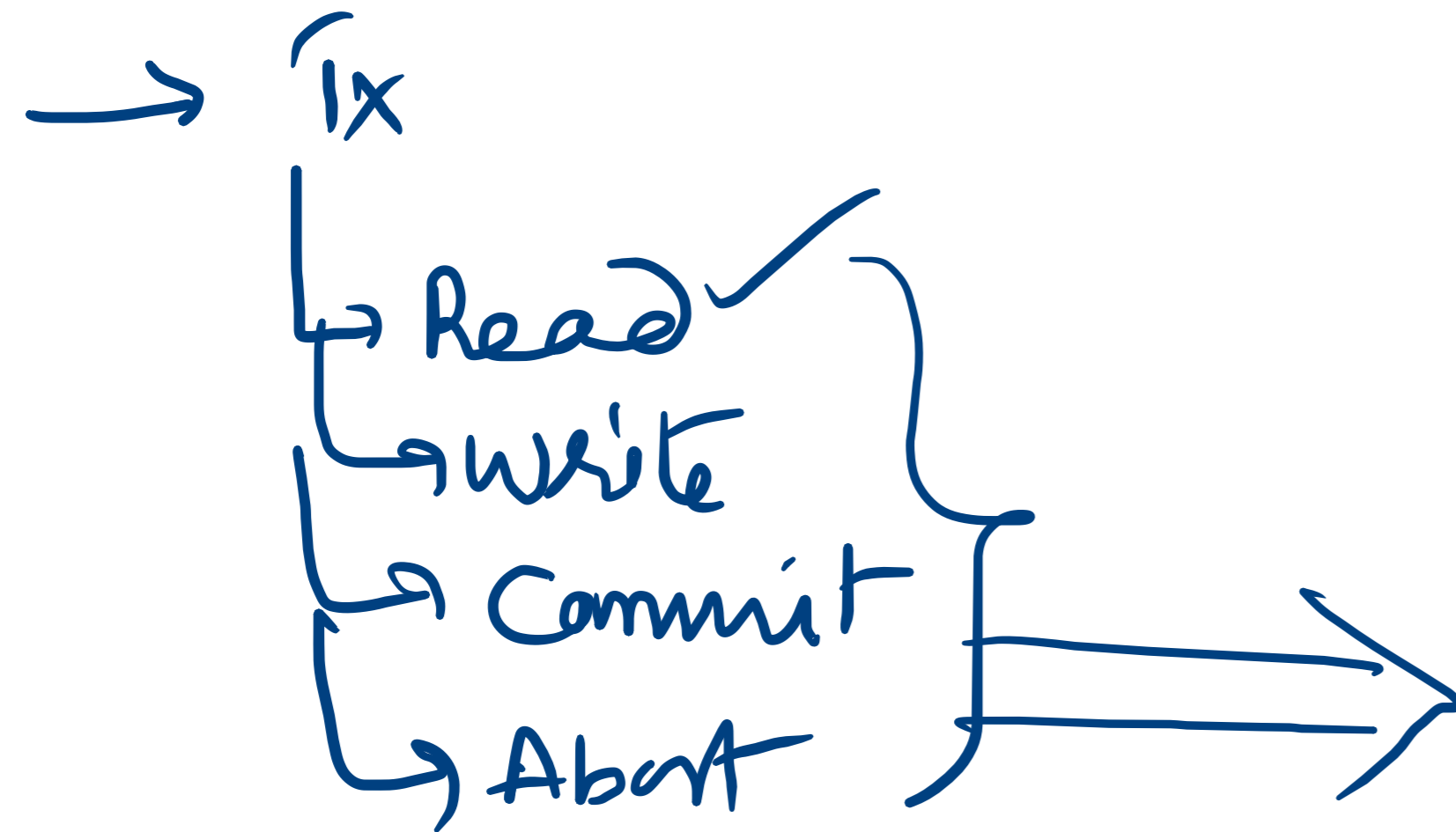
$1500 + 300$

Abdt



→ Log Based Recovery

Tx. operations are stored in log file



Log Records : operations

① $[T_1, \text{start}]$

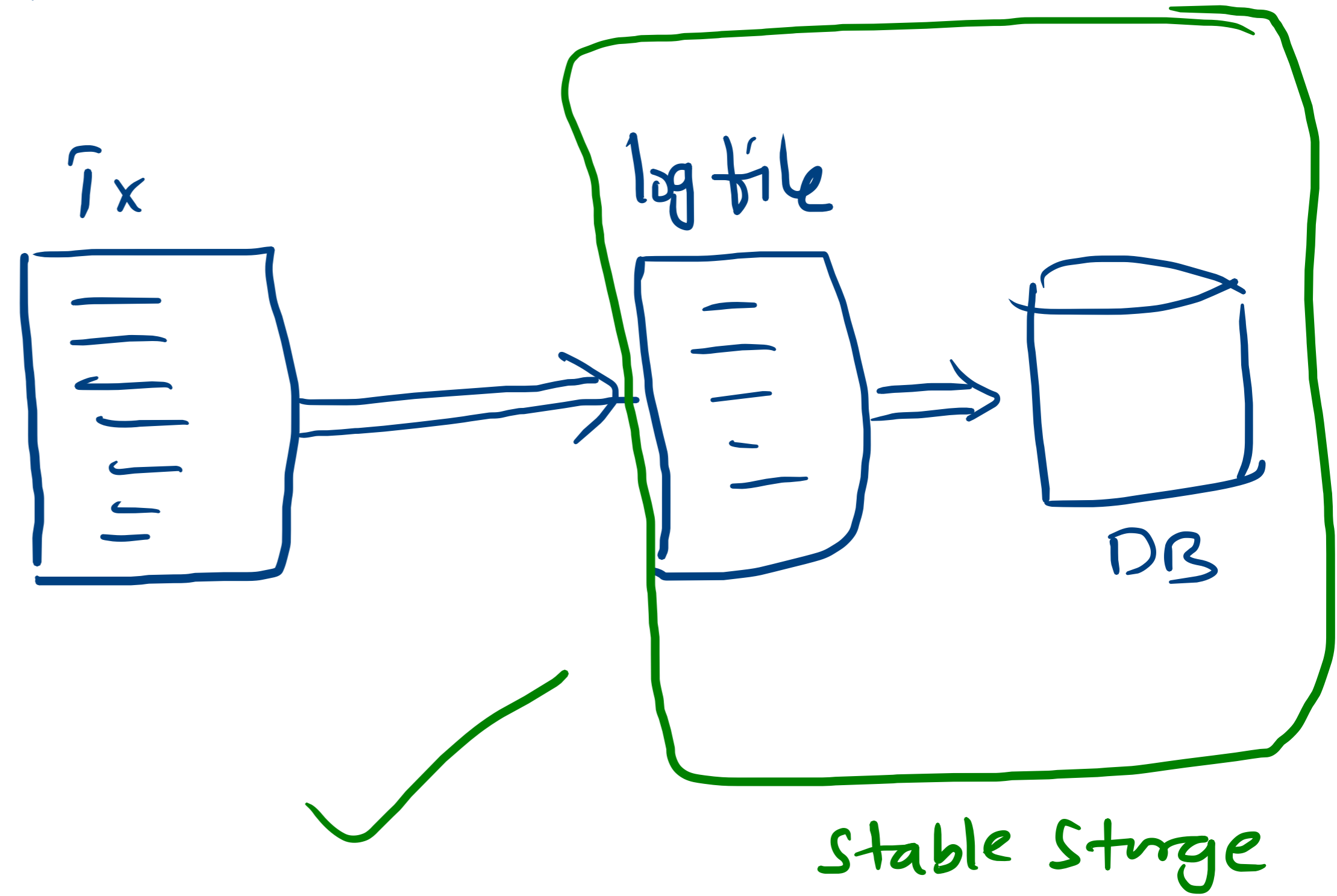
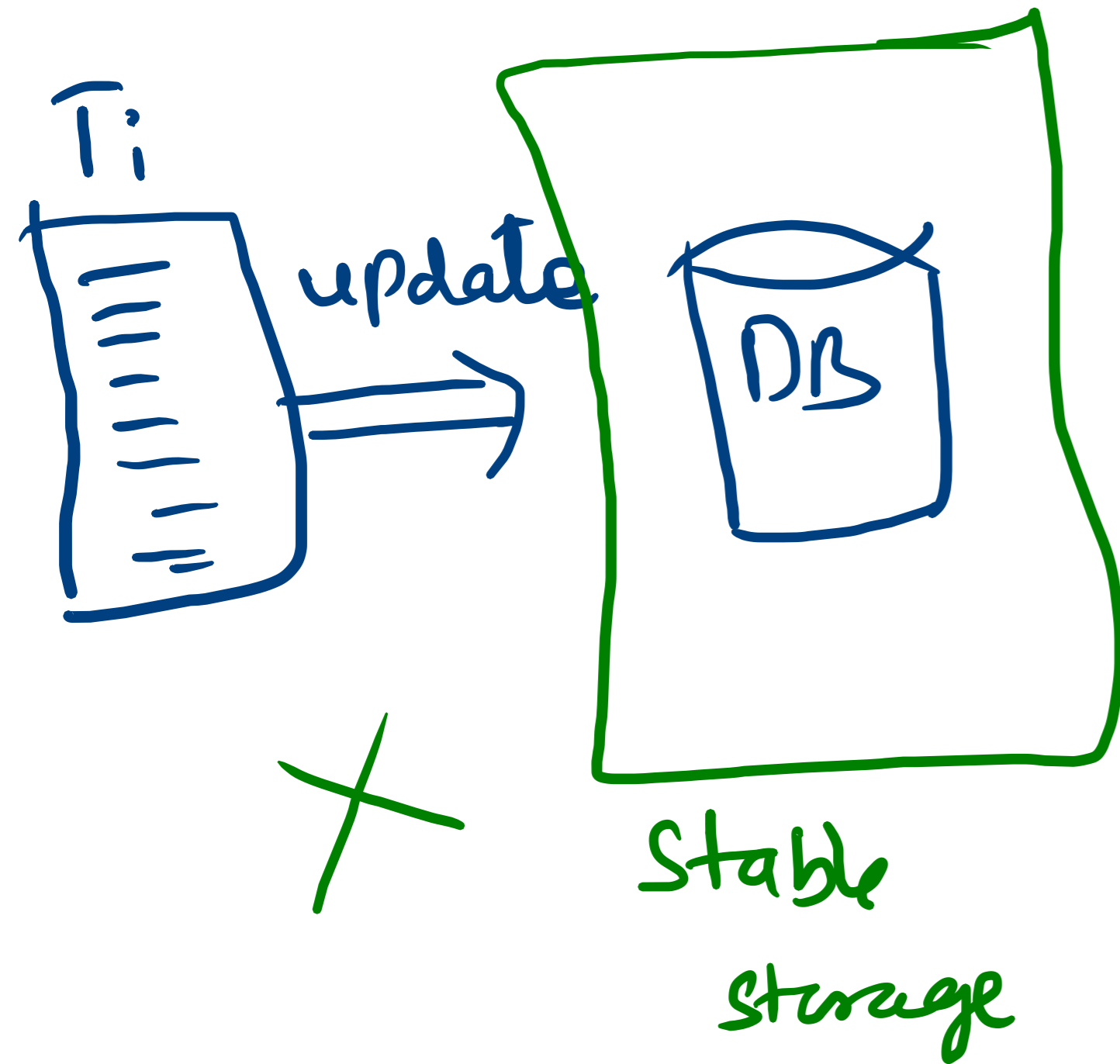
② $[\underline{T}_1, \underline{X}, \underline{\text{old}}, \underline{\text{new}}] \leftarrow \text{write operations}$

③ $[T_1, \text{Commit}]$

④ $[T_1, \text{Abort}]$

Write Ahead Log Strategy: (WAL)

Log is written before any update is made to the database.



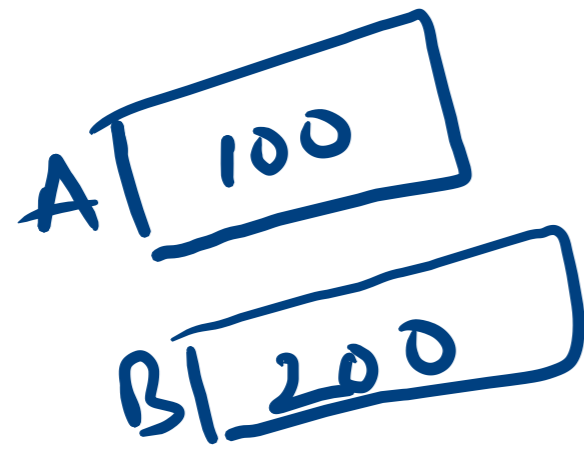
① Deferred Database Modification:

- Transaction operations do not immediately update the physical database.
- Only transaction log is updated
- Database is physically updated only after the transaction reaches commit point.
- Also known as NO-UNDO/REDO Alg.

Example:

$\hat{T}_x \rightarrow \log \rightarrow DR$

Log



$\frac{T_1}{R(A)}$

$A = A + 100$

$w(A)$

$R(B)$

$B = B + 200$

$w(B)$

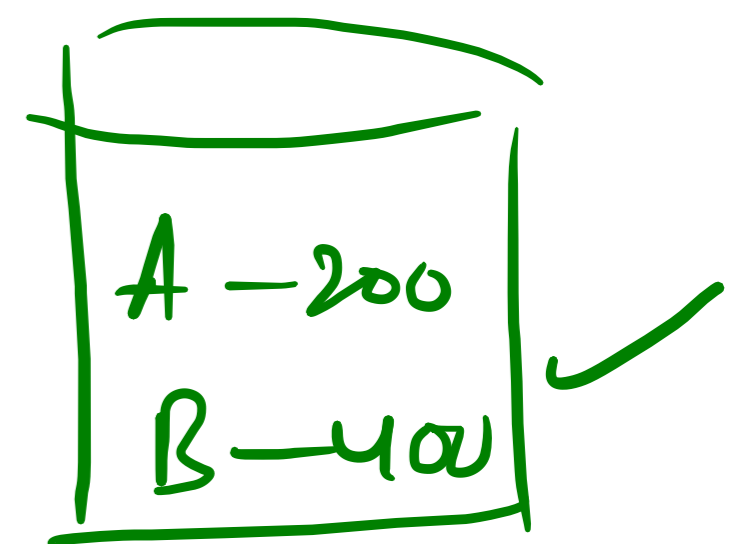
Commit

$(T_1, \text{start}) \checkmark$

$(T_1, \underline{A}, \underline{200}) \checkmark$

$(T_1, \underline{B}, \underline{400}) \checkmark$

$(\underline{T_1}, \text{Commit}) \checkmark$



Eg:

Log

(T₁, start) ✓

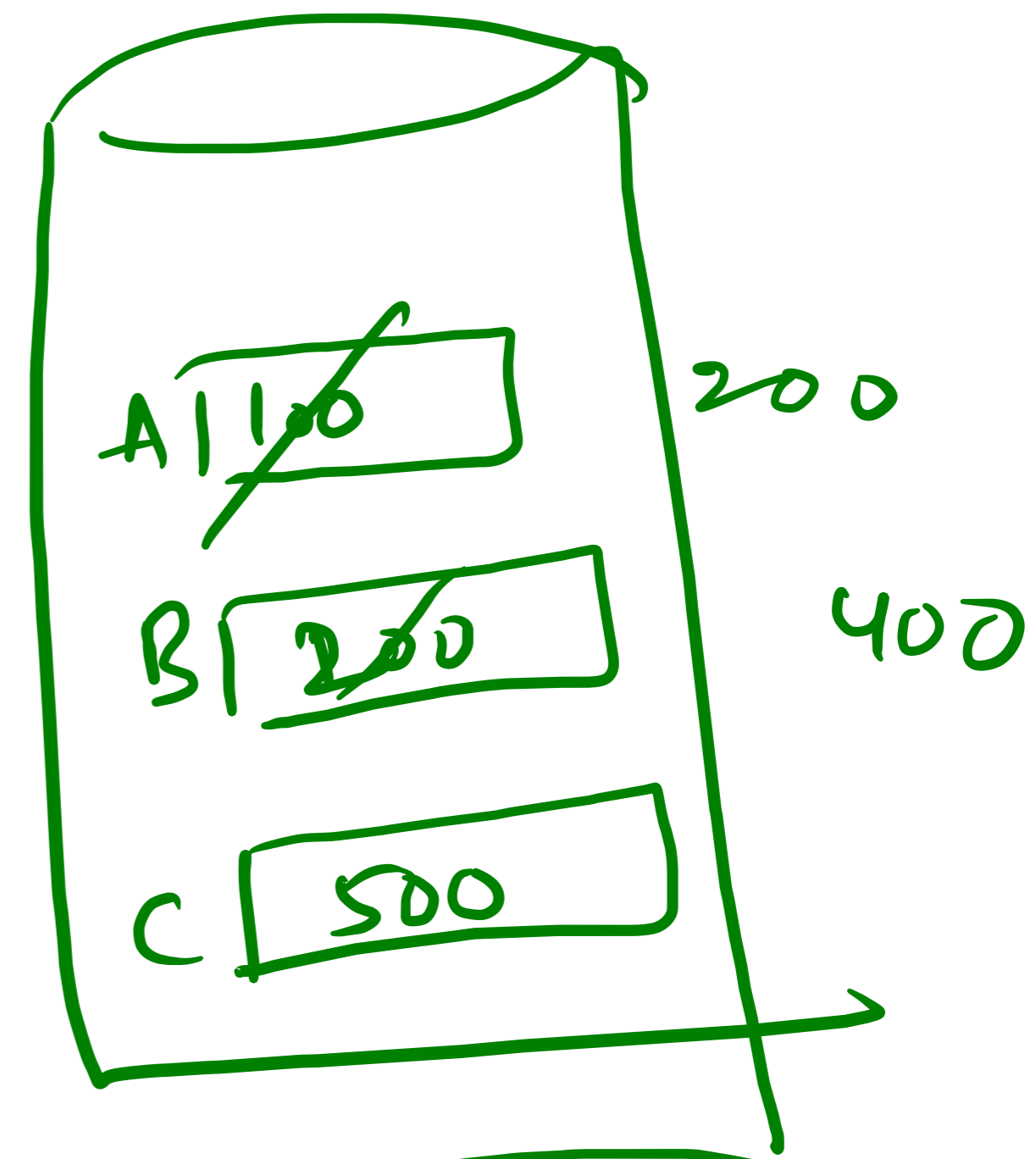
(T₁, A, 200)

(T₁, B, 400)

(T₂, start) ✗

(T₁, Commit)

→ (T₂, C, 1000) ✗



Tx. fail — UNDO

Tx. success — REDO

② Immediate Database Modification

- Database immediately updated by transaction operations during the execution of transaction before it reaches commit point.
- In case of transaction abort UNDO operations needs to be done to restore the database to its consistent state.
- Also called (UNDO/No-REDO) alg. ✓

Example

\hat{T}_1 A | 100
 B | 200

R(A)
A = A + 100

W(A)

R(B)

B = B + 200

w(B)

~~Commit~~

Abort

(T_1, start) ✓

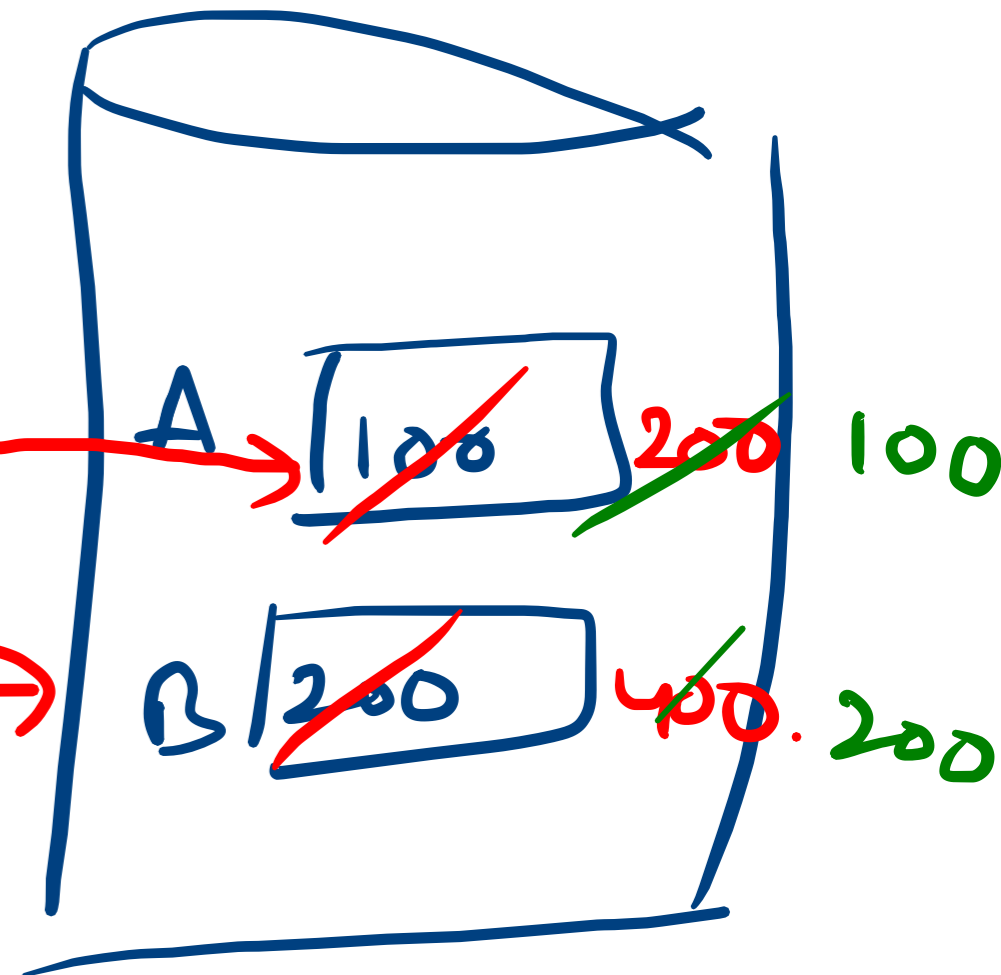
$(T_1, A, 100, 200)$ ✓

$(T_1, B, 200, 400)$ ✓

(T_1, Abort)

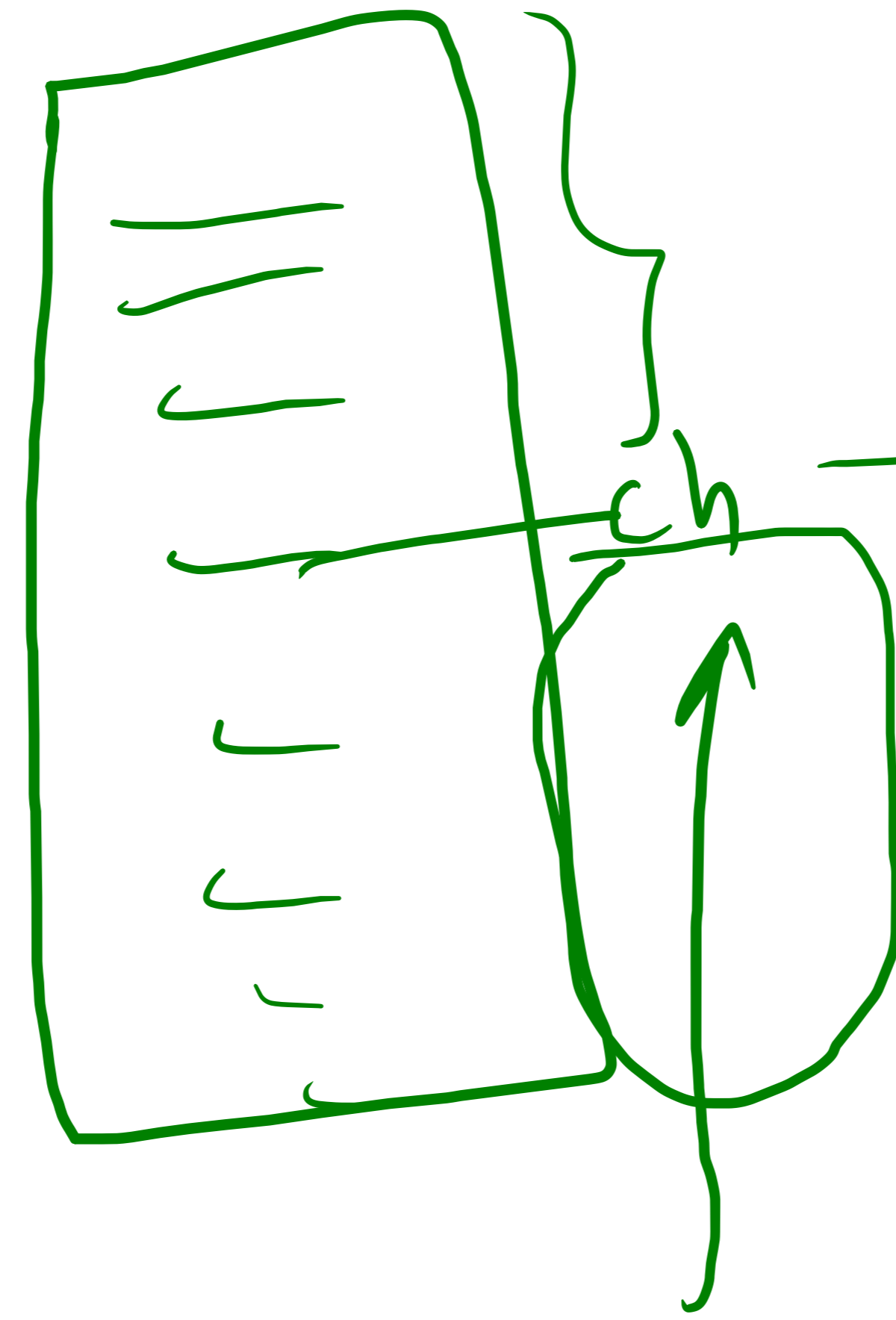
UNDO

No-REDO



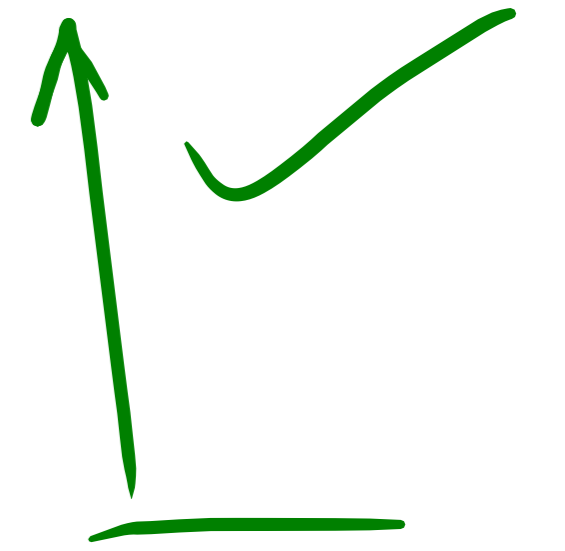
Check point:

108

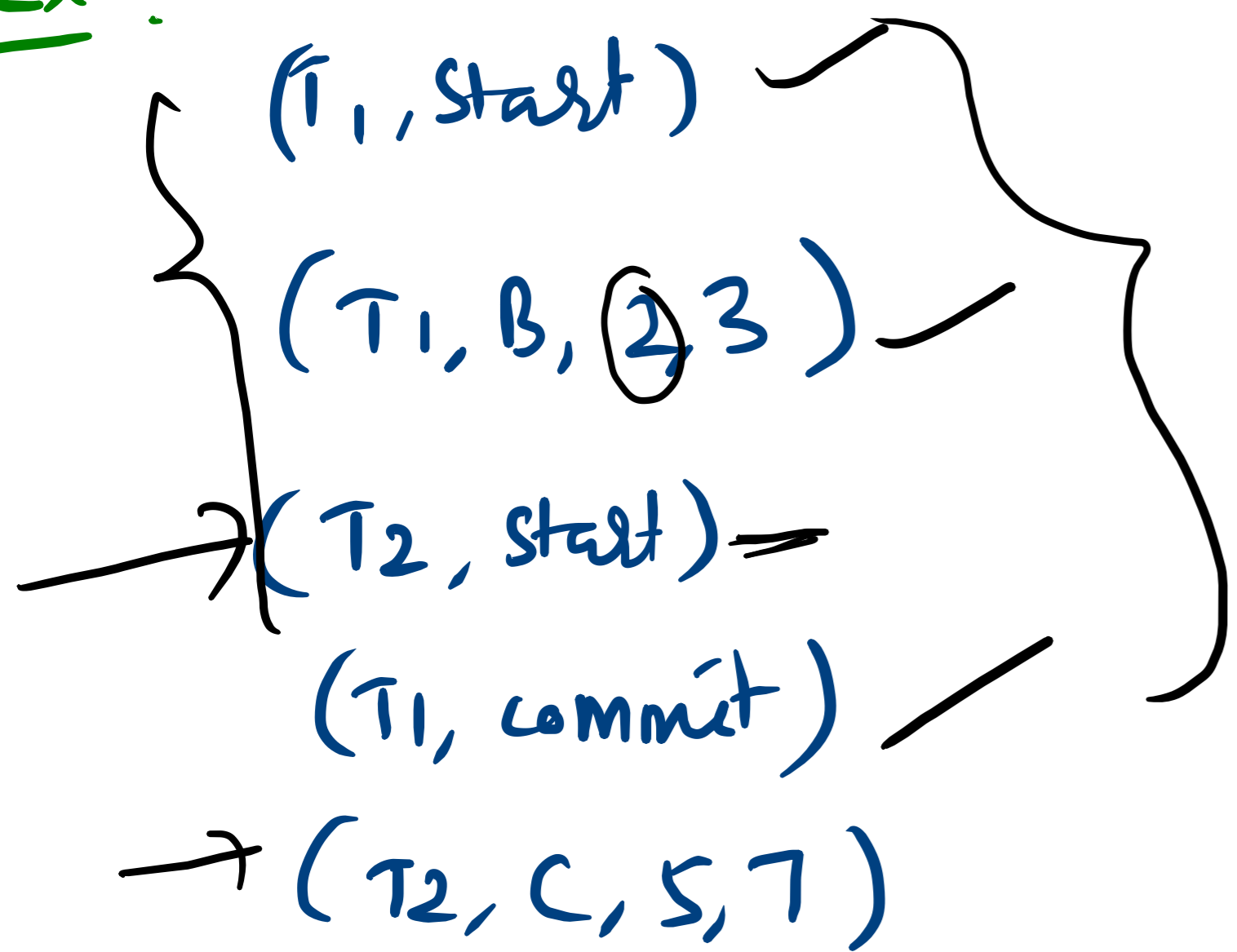


|||||

check point



Ex :



T_1 fail
 T_1 success

Undo-list : $\{T_4, T_2\}$

Redo-list : (T_3)

