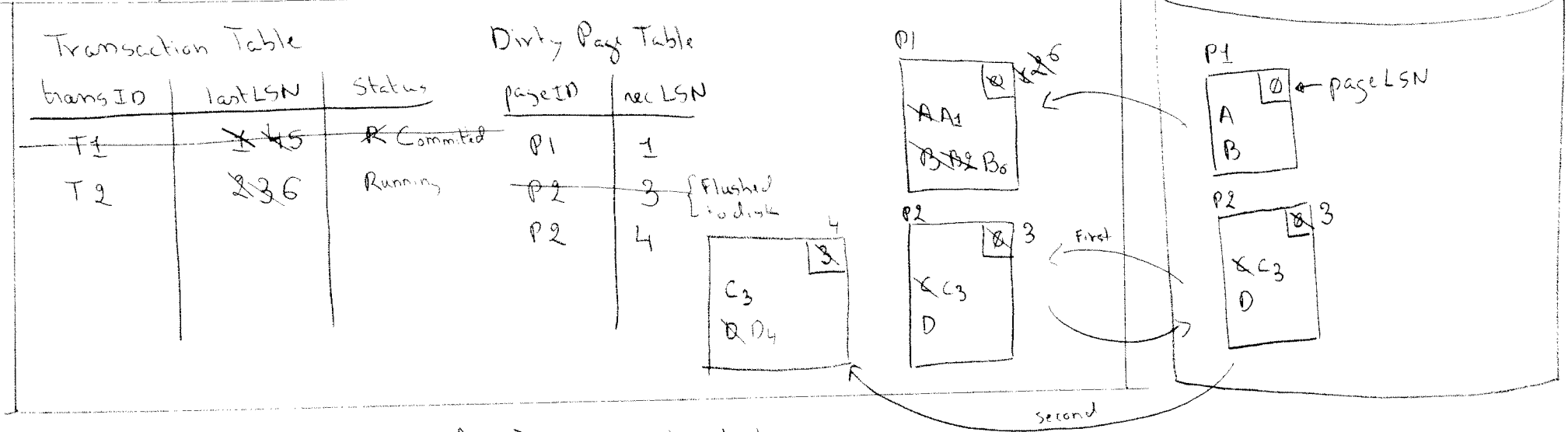


In memory



Log (tail in memory, rest on disk) Last after crash

LSN	1	2	3	4	5	6	7
transID	T1	T2	T2	T1	T1	T2	T1
prevLSN	-	-	2	1	4	3	5
type	Update	U	U	U	Commit	U	End
logentry	Write A A → A1	Write B B → B2	Write C C → C3	Write D D → D4	-	Write B B2 → B6	-
undoNextLSN (only for CLR)	-	-	-	-	-	-	-



② Analysis phase

Transaction Table

trans ID	last LSN	Status
T1	1 5	Unknown Committed
T2	2 3	U

Dirty Page Table

page ID	rec LSN
P1	1
P2	3

Analysis Rules

- END Record removes transaction from table
- Other records update last LSN
- Commit record changes status to C
- For redoable records update Dirty Page Table

③ Redo phase Repeating history

Start at first LSN smallest LSN in Dirty Page Table

For each redoable log record (update or CLR), redo if necessary

LSN1: ① check if P1 is in dirty page table YES

② check if rec LSN for P1 \leq LSN1 YES

③ load from disk & check if page LSN $<$ LSN1 Yes

Redo change

LSN2: Redo change

LSN3: No need to redo

LSN4: Redo

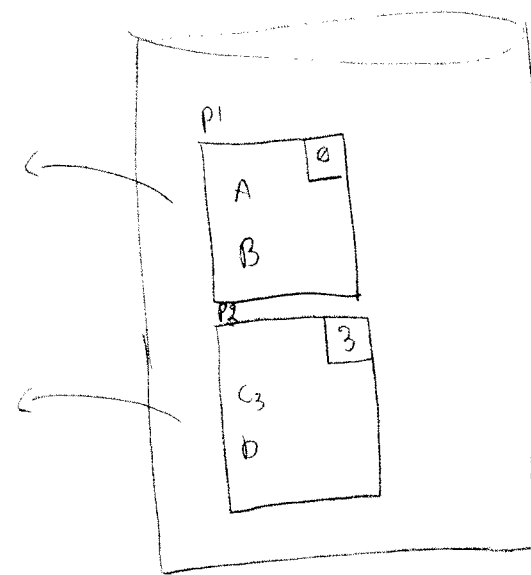
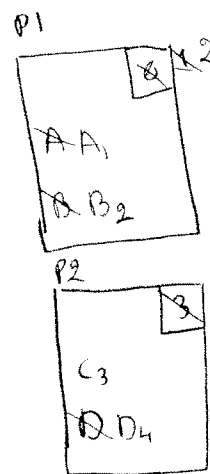
LSN5: Skip

System back in state as of time when log last flushed to disk

Now write an END type record for T1.

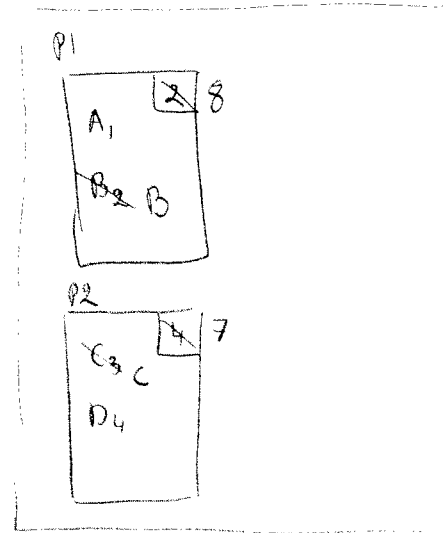
Notice that LSN=8 for this record.

And remove from transaction table



Undo phase Need to Undo 12
 Notice no need to write about log record

SN	6	7	8	9
ans ID	T1	T2	T2	T2
LSN	5	-	-	8
type	E	CLR	CLR	E
genby	-	Undo T2 LSN 3	Undo T2 LSN 2	-
relNextLSN	-	2	-	-



Start undo at LSN=3

FOR LSN=3, Write CLR

(Notice that we did not write about log record)

FOR LSN=2, Write CLR

Write an End log record.

Second crash

① What if we crash again now? IF we did not write anything to disk, we redo analysis/undo/redo exactly in the same way.

② IF we flushed log to disk except for T2's end log record then →

Analysis	T1	5	6
	T2	8	4

Removed thanks to end record
 Same dirty page table

Redo same as before until LSN 5
 LSN 6 do nothing
 LSN 7 if we previously flushed page to disk then do nothing else apply
 LSN 8 same

Undo Finds only CLR without undo point so write only an END record

③ IF we had flushed end log record also then transactions table would have been empty and we would not undo anything

④ IF we crashed before second CLR
 Analysis & Redo same as ②
 Add second CLR & END record

2) Checkpoints: 3 steps

Step 1: begin-checkpoint record is written

Step 2: end-checkpoint record is constructed with current content of transaction table and dirty page table. The record is appended to the log.

Step 3: LSN of begin-checkpoint is written in special place

} called a
fuzzy checkpoint

In our example, imagine that we checkpointed after flushing P2 to disk and before $W_1[D]$.

Then after analysis, we would have:

Redo would still start at LSN 1
but we would know that LSN 3
did not need to be redone without
loading P2 from disk.

Dirty Page Table	
P1	1
P2	4

And if we had also flushed P1 to disk before the checkpoint we would get the following table after analysis.

Dirty Page Table	
P2	4

Redo would start at LSN=4

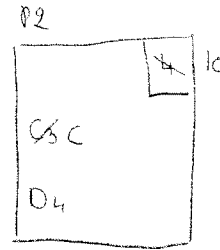
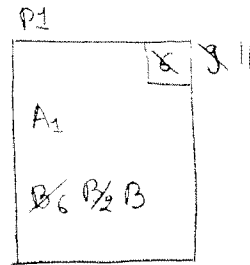
(6) Finally imagine that instead of writing, T2 aborts ... W2 [13] → 12 done → committing → Abort T2

Transaction Table

	transID	lastLSN	Status
insert →	T1	5	Committed
	T2	8 11	Running Aborted

Dirty Page Table

pageID	nextLSN
P1	1
P2	4



Log

LSN	6	7	8	9	10	11	12
transID	T2	T1	T2	T2	T2	T2	T2
prevLSN	3	5	6	-	-	-	11
type	U	End	Abort	CLR	CLR	CLR	End
logentry	was B P2 → B6	-	-	Undo T2 LSN 6	Undo T2 LSN 3	Undo T2 LSN 2	-
undoNextLSN	-	-	-	3	2	-	-