CS186 Discussion #11

(Logging & Recovery)

Michelle Nguyen

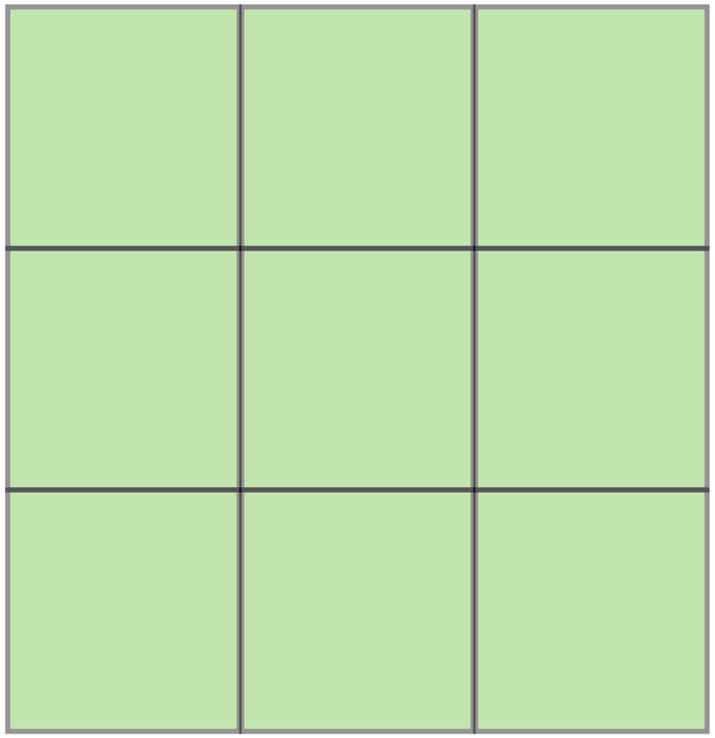
ACID

- Atomicity: either none or all instructions committed
- Consistency: database remains in consistent state
- Isolation: runs as if it is only transaction
- Durability: committed changes are never lost

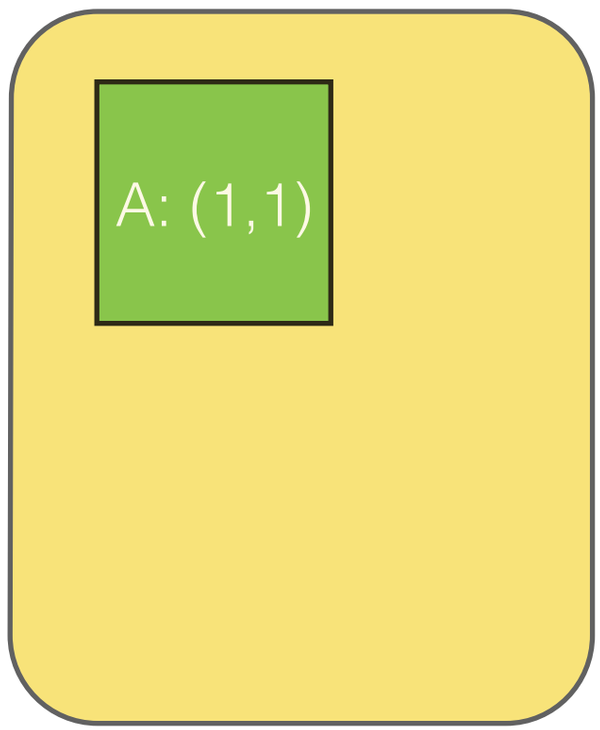
Buffer Policies

T1	R(A)	W(A)

Buffer Pool



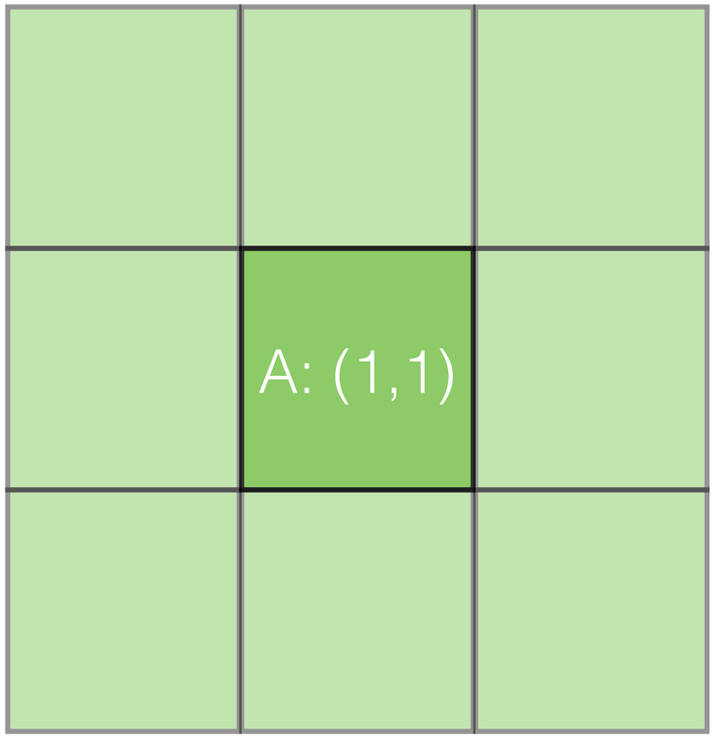
Disk



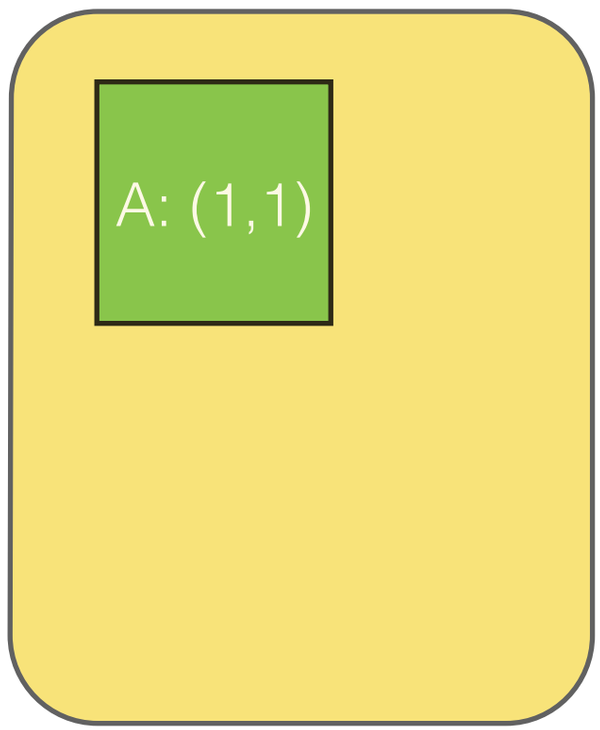
Buffer Policies

T1	R(A)	W(A)

Buffer Pool



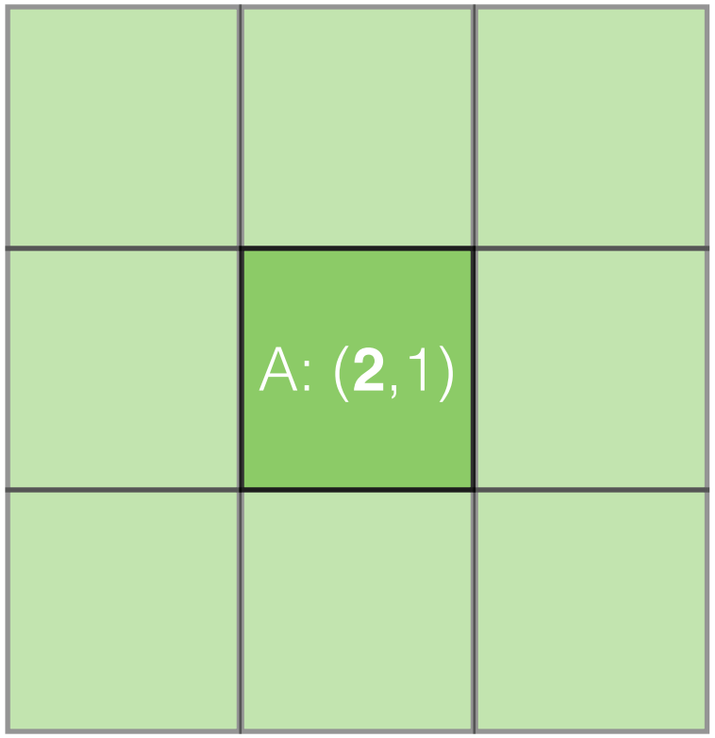
Disk



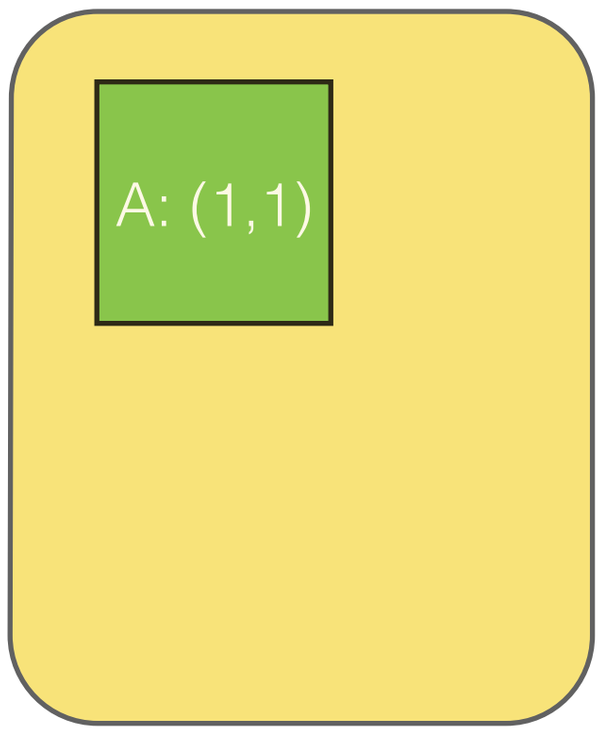
Buffer Policies

T1	R(A)	W(A)

Buffer Pool

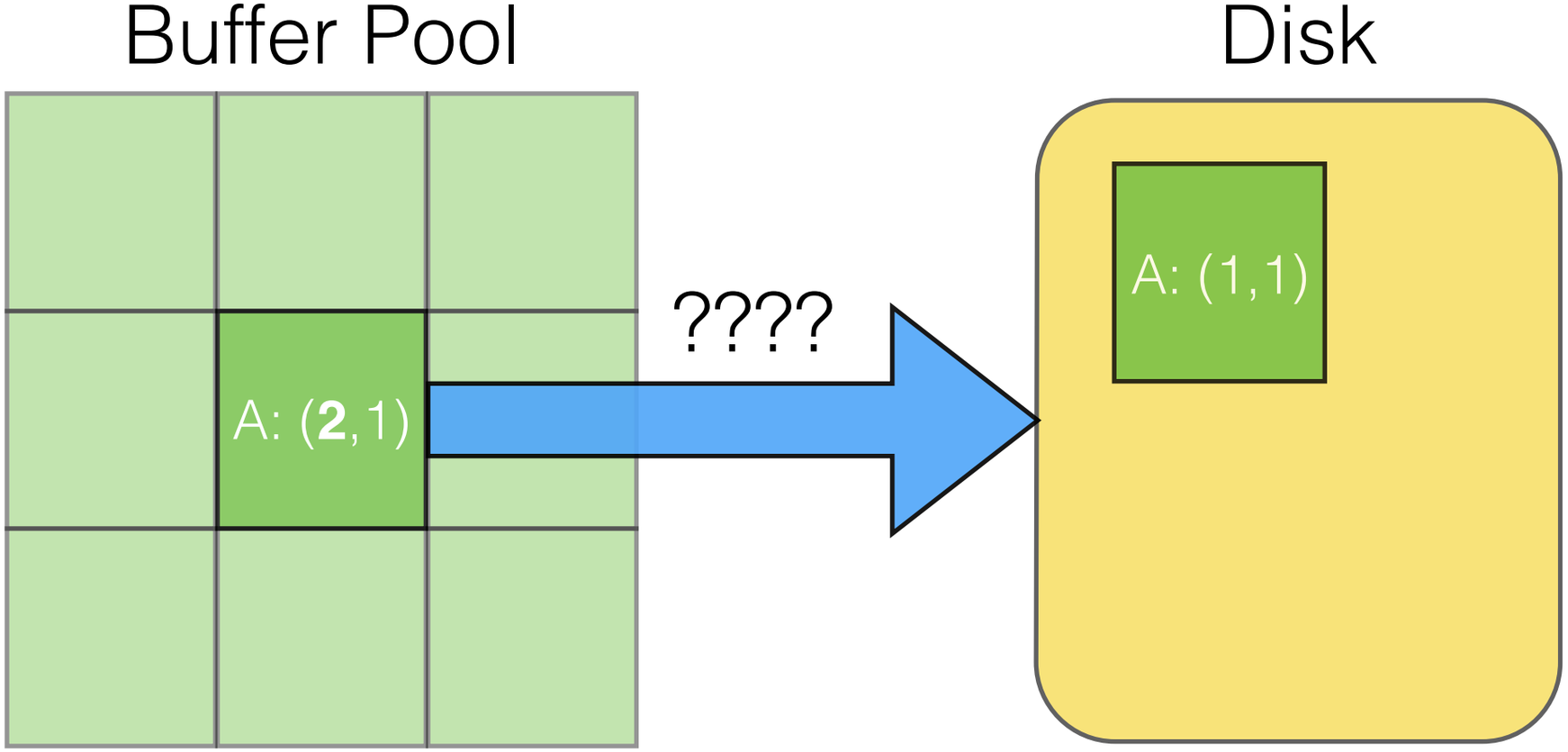


Disk



Buffer Policies

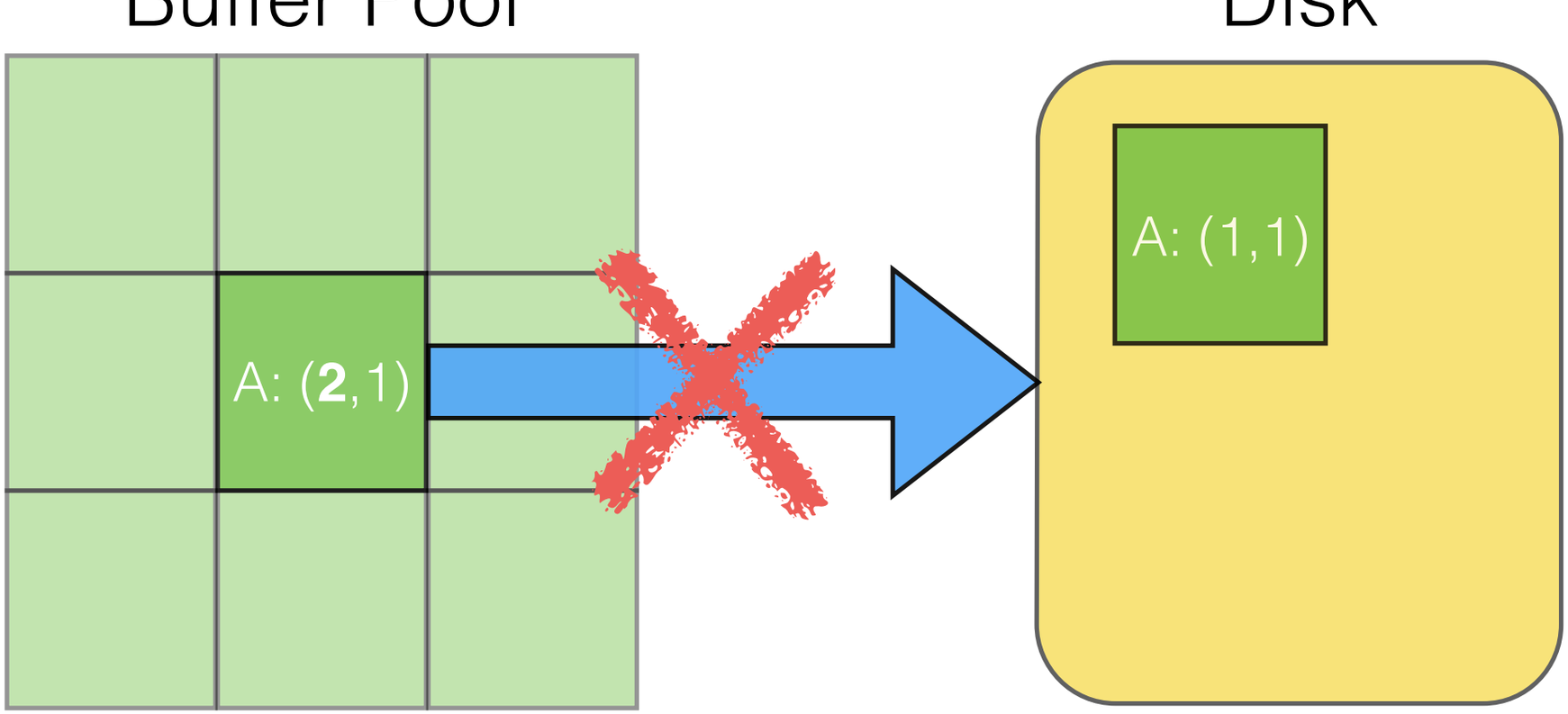
T1	R(A)	W(A)



Can we flush page A out to disk?

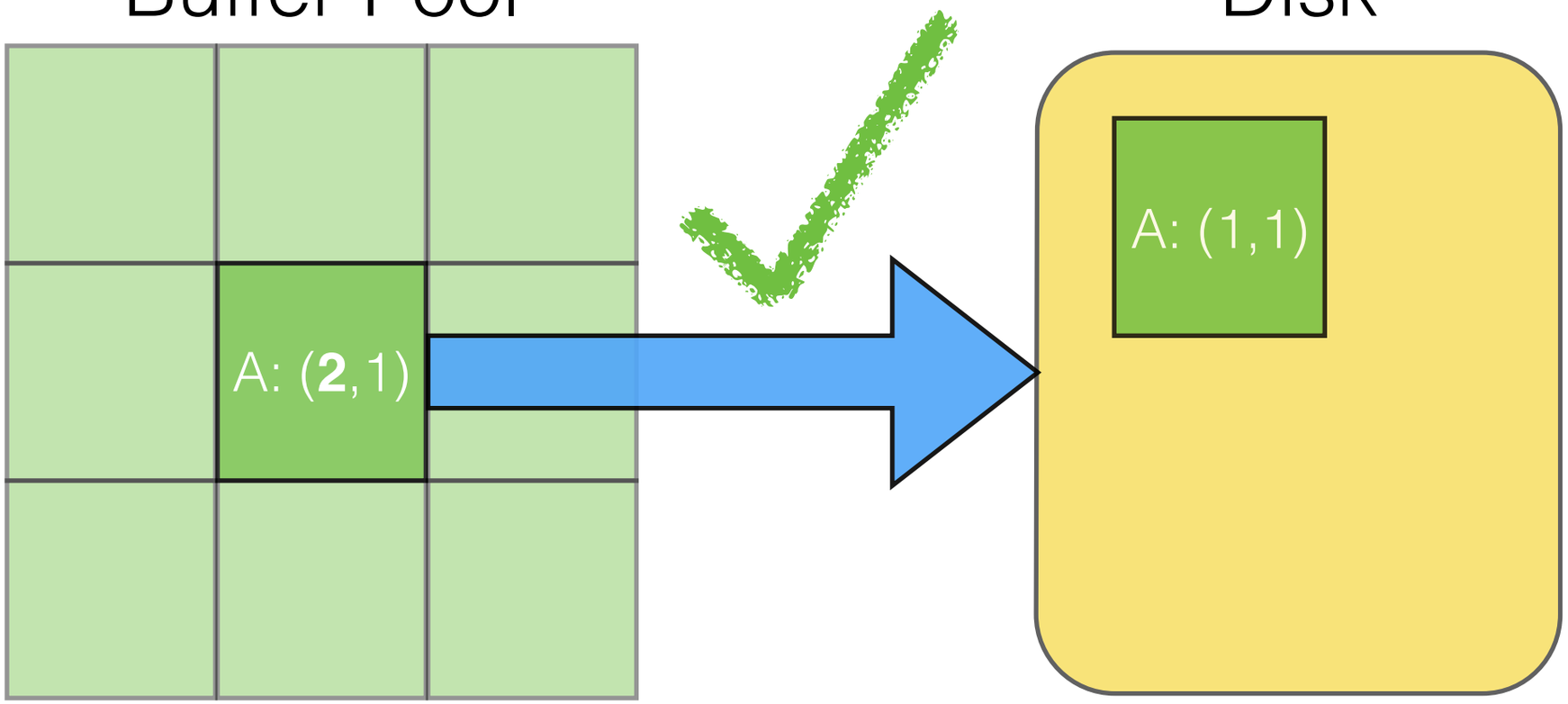
Buffer Policies

NO STEAL: Don't let system "steal" pages with uncommitted updates from buffer pool and write them to disk



Buffer Policies

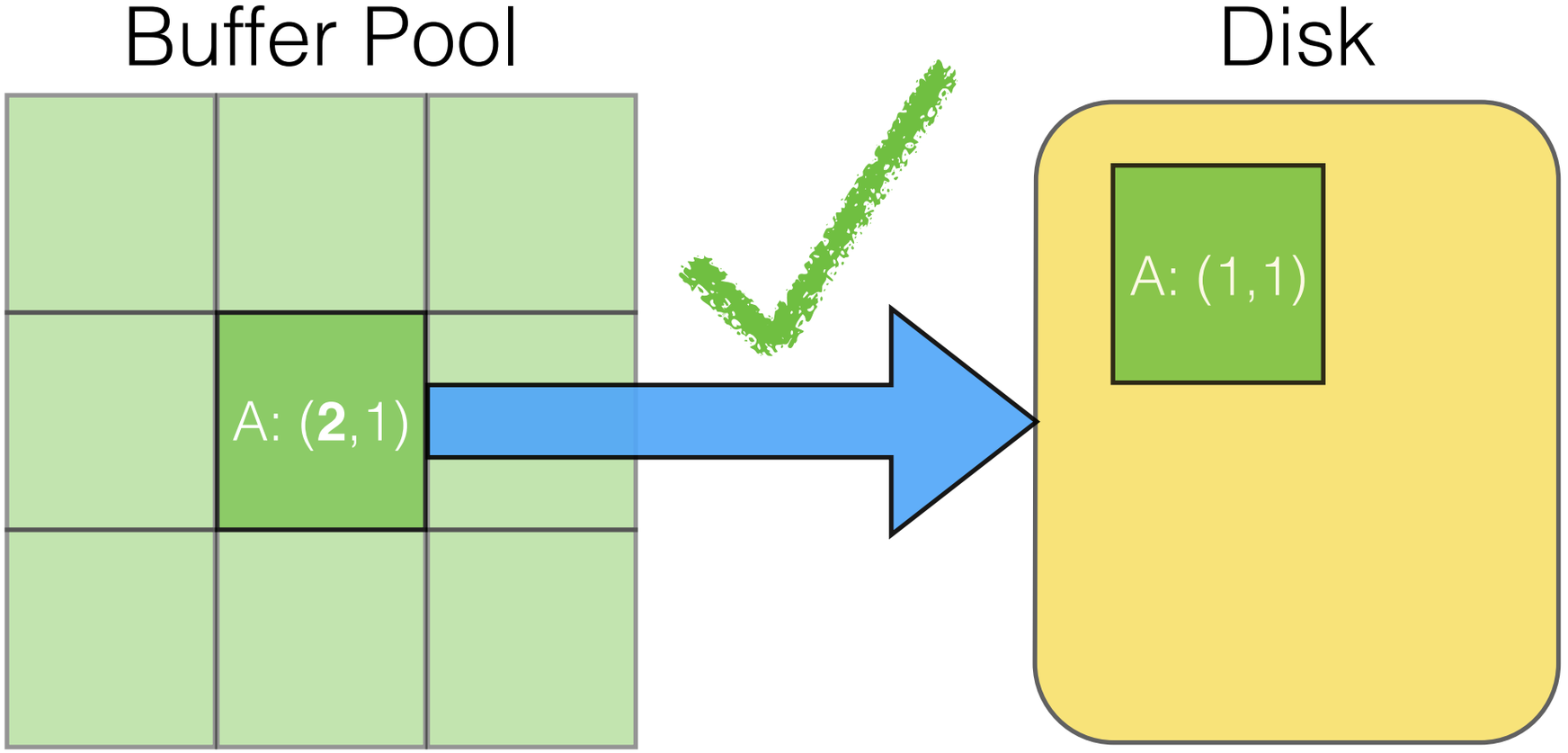
T1	R(A)	W(A)	**COMMIT**



Only flush data out to disk after commit.

Buffer Policies

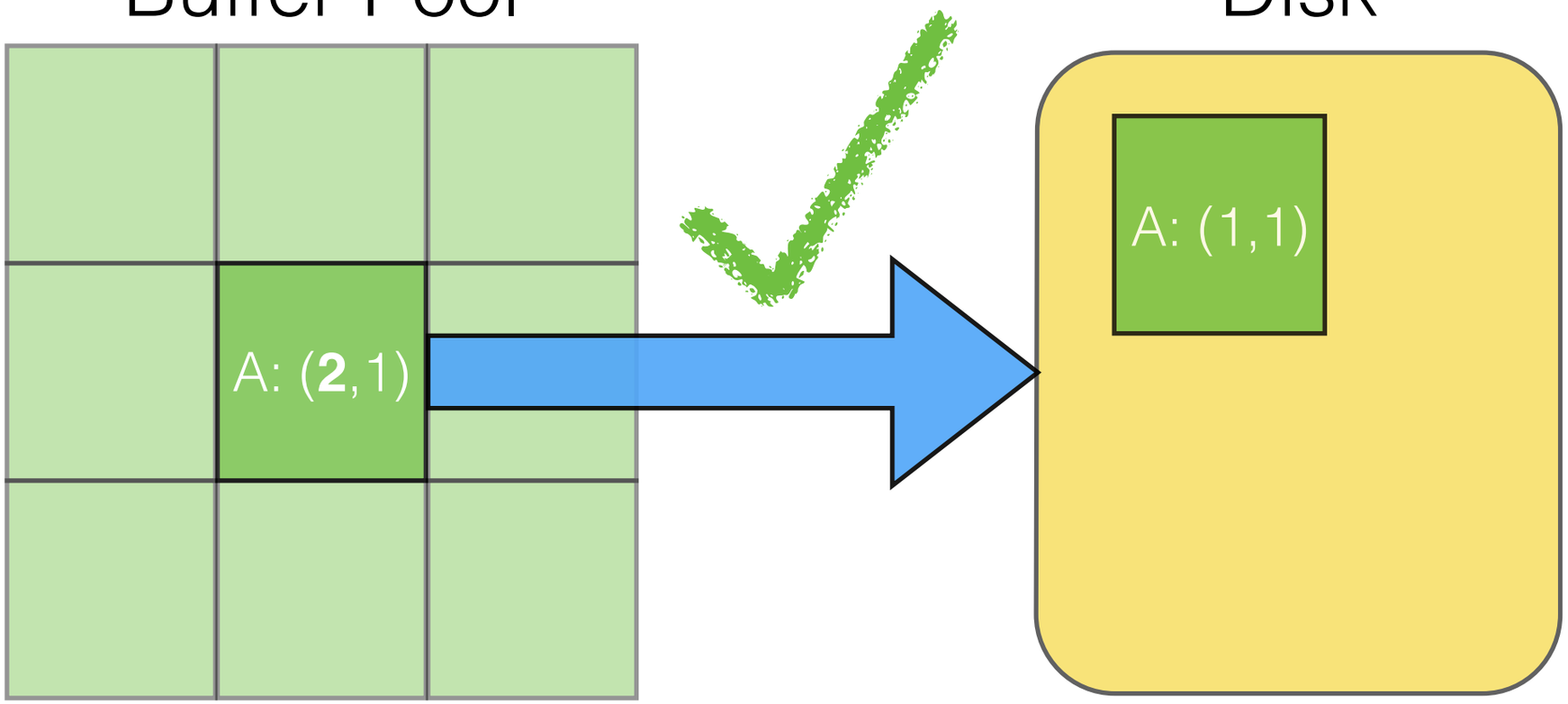
T1	R(A)	W(A)



What if we want to flush uncommitted data to disk?

Buffer Policies

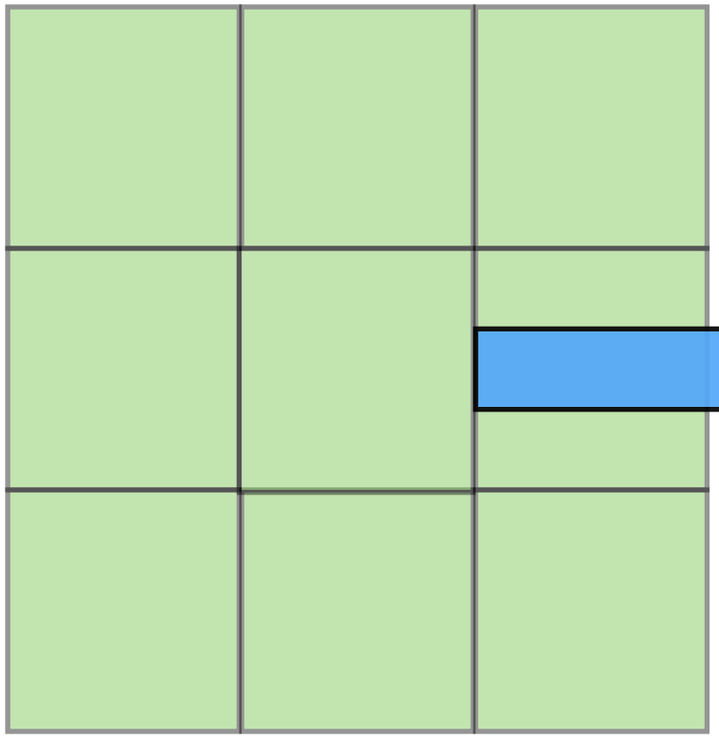
STEAL: Allow uncommitted data in disk



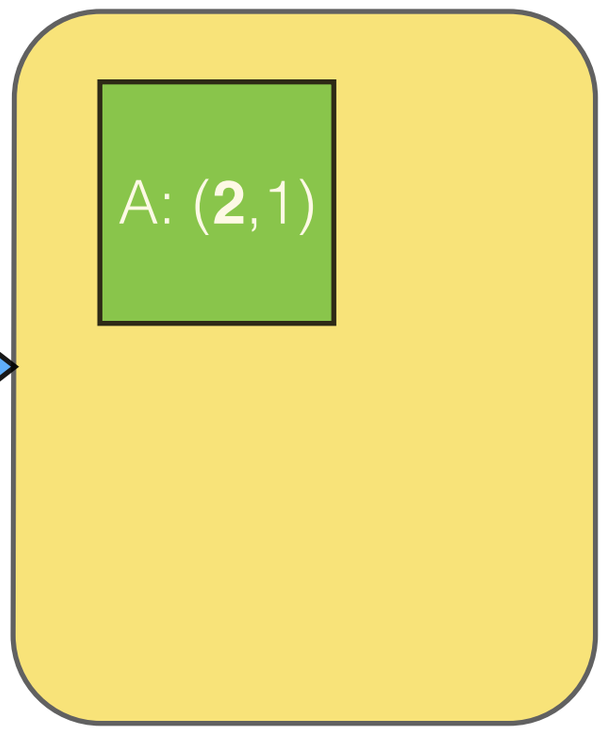
Buffer Policies

T1	R(A)	W(A)

Buffer Pool



Disk

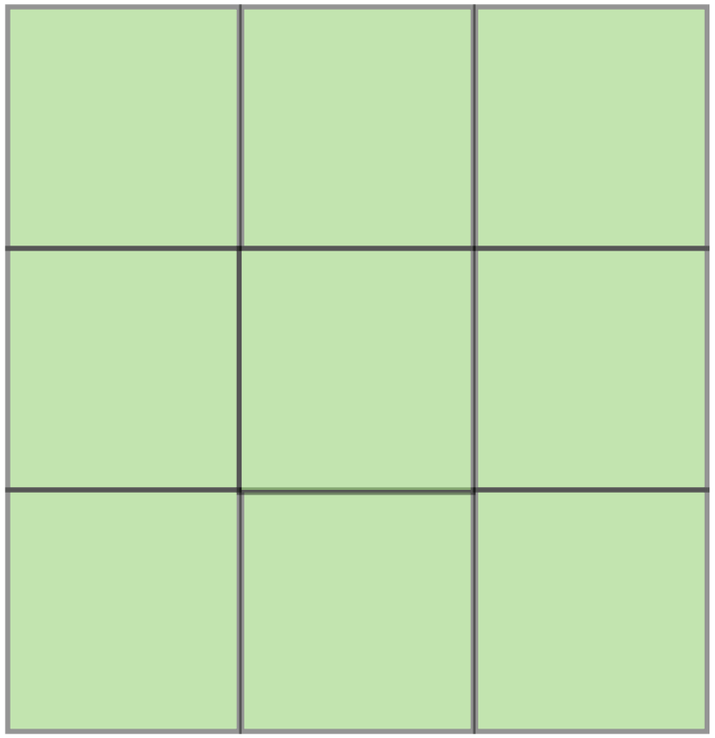


Buffer Policies

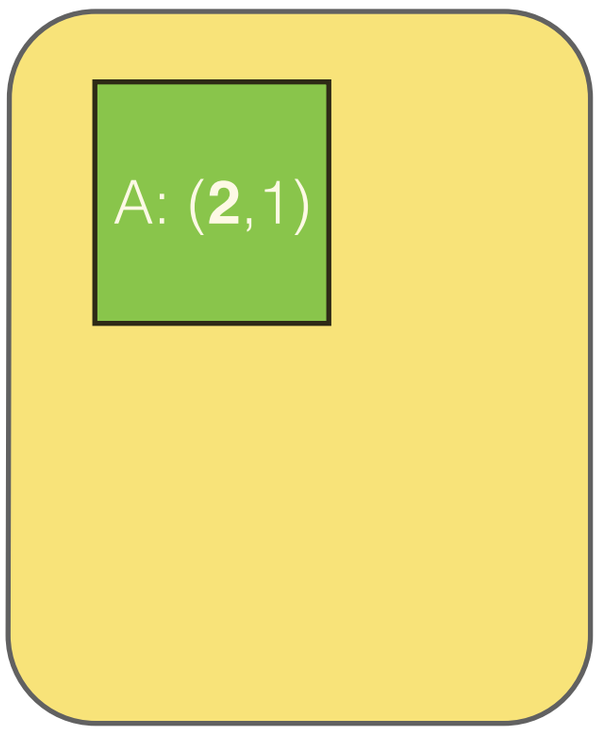
T1	R(A)	W(A)

CRASH

Buffer Pool



Disk

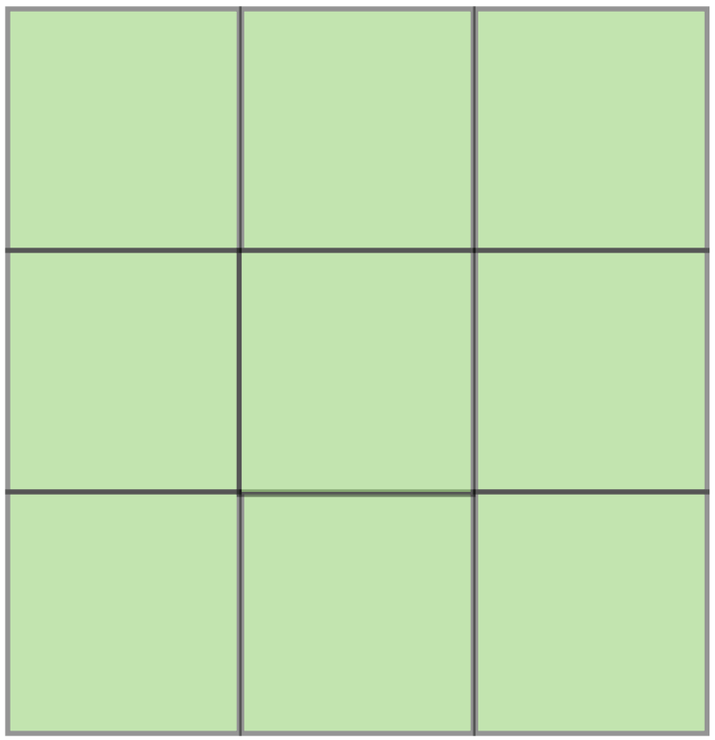


Buffer Policies

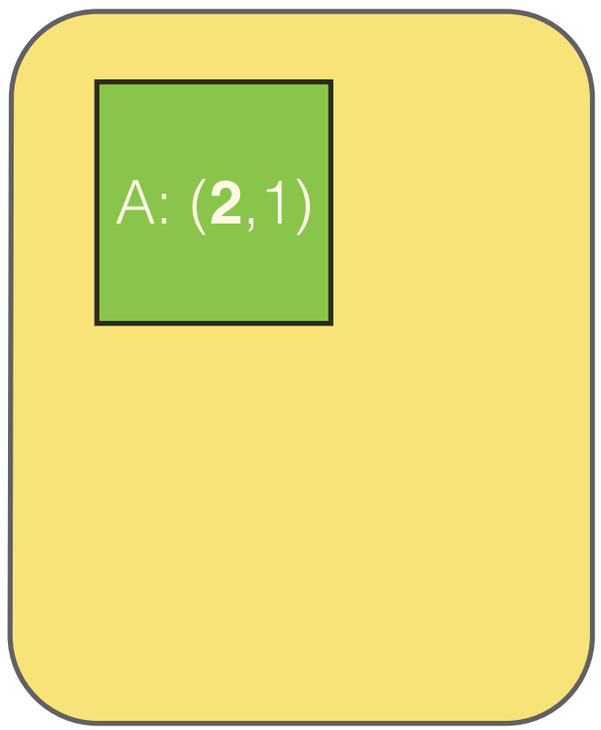
T1	R(A)	W(A)

CRASH

Buffer Pool



Disk



Could threaten atomicity! Need to **undo** actions.

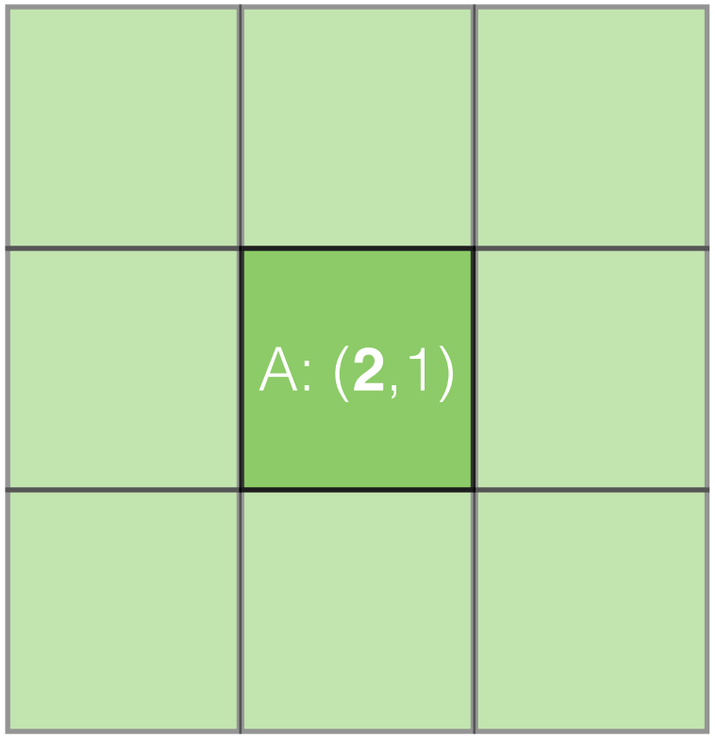
Buffer Policies

- **NO STEAL**: Don't let system "steal" pages with uncommitted updates from buffer pool and write them to disk
- **STEAL**:
  - Allow uncommitted data in disk
  - Requires UNDO to preserve atomicity

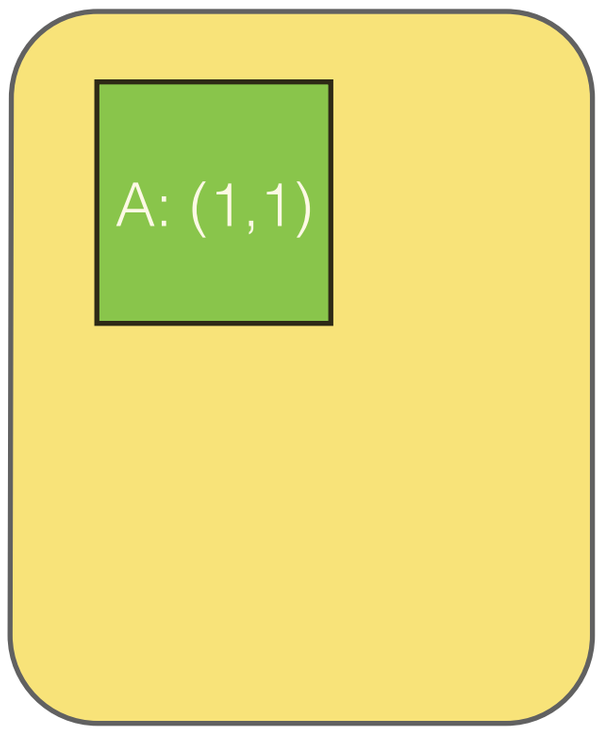
Buffer Policies

T1	R(A)	W(A)	**COMMIT**

Buffer Pool



Disk

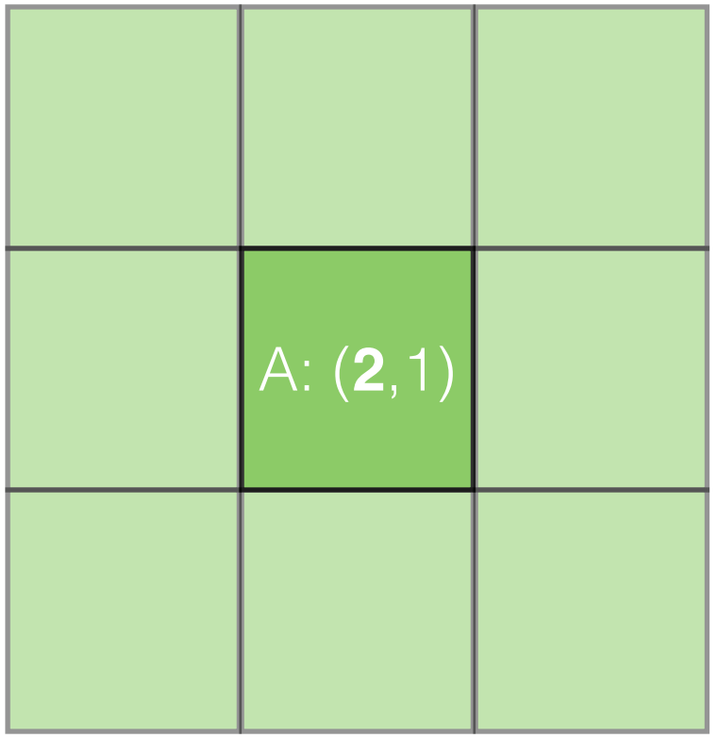


Can pages be flushed to disk **after** commits?

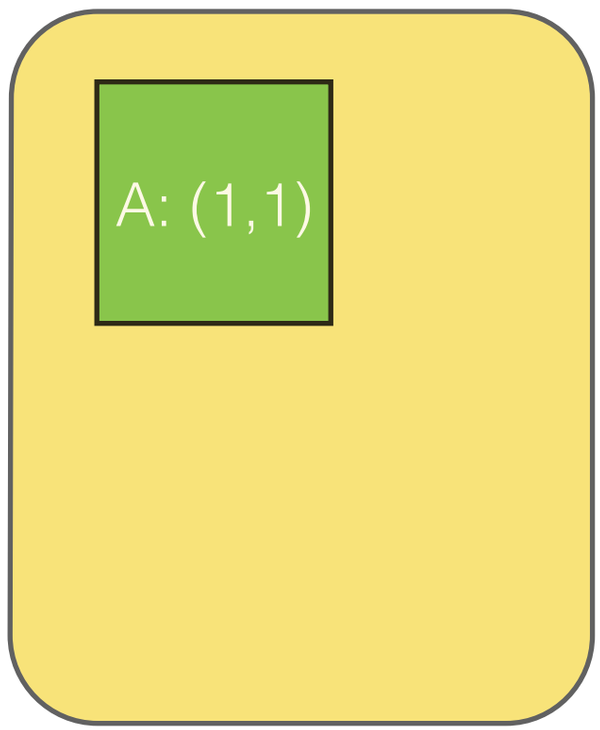
Buffer Policies

T1	R(A)	W(A)	**COMMIT**

Buffer Pool

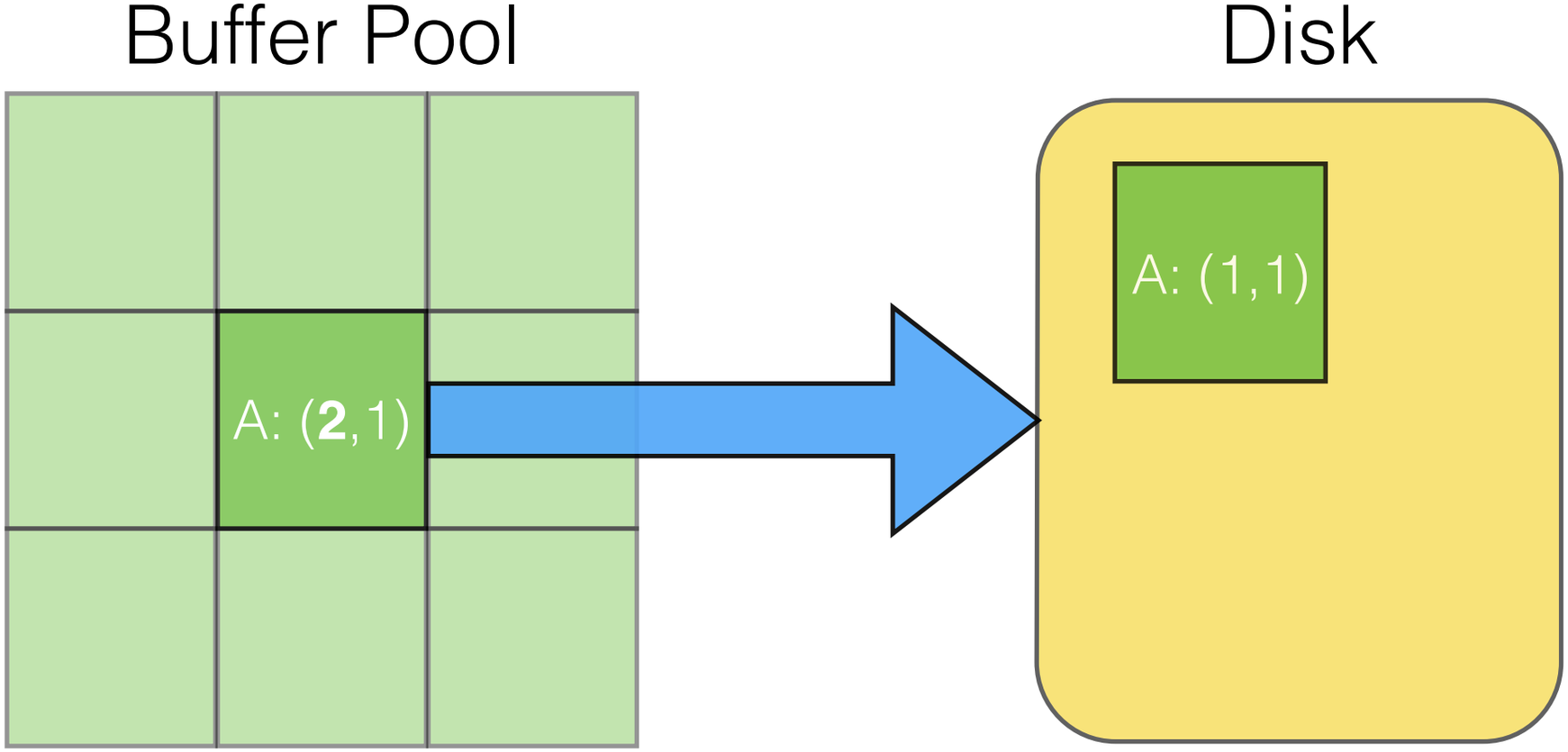


Disk



Buffer Policies

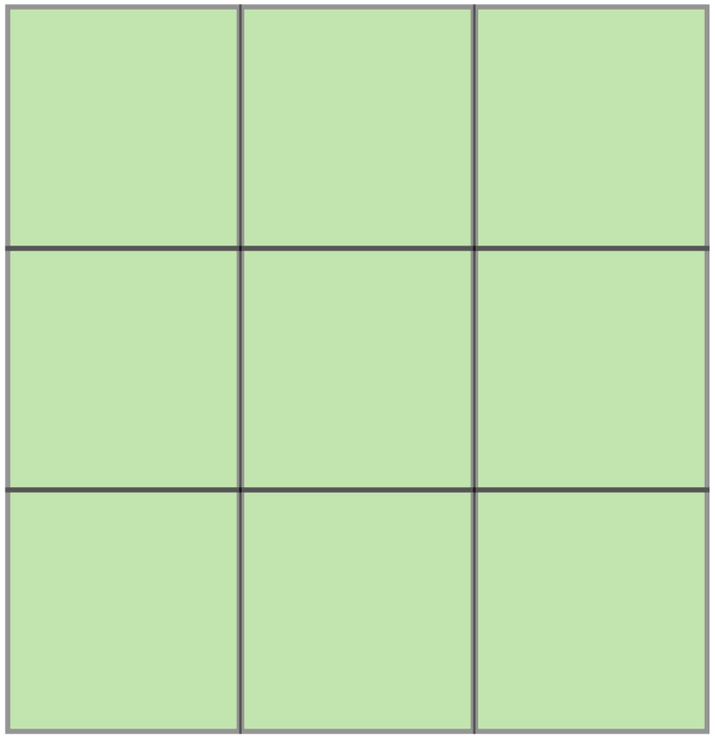
T1	R(A)	W(A)	**COMMIT**



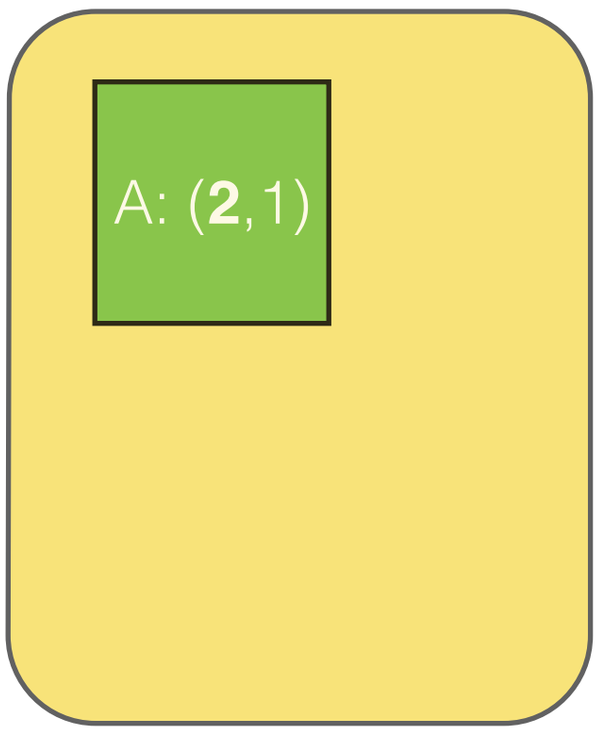
Buffer Policies

T1	R(A)	W(A)	**COMMIT**

Buffer Pool



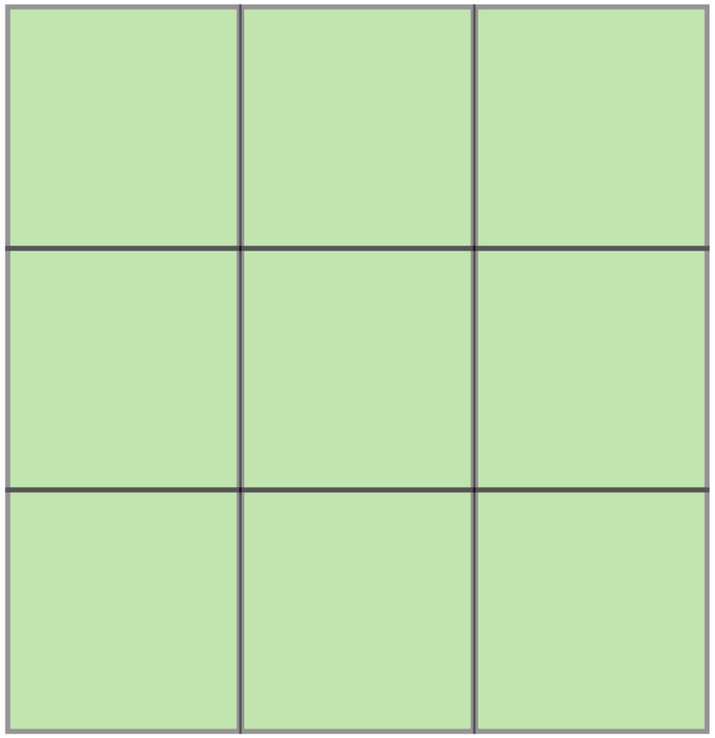
Disk



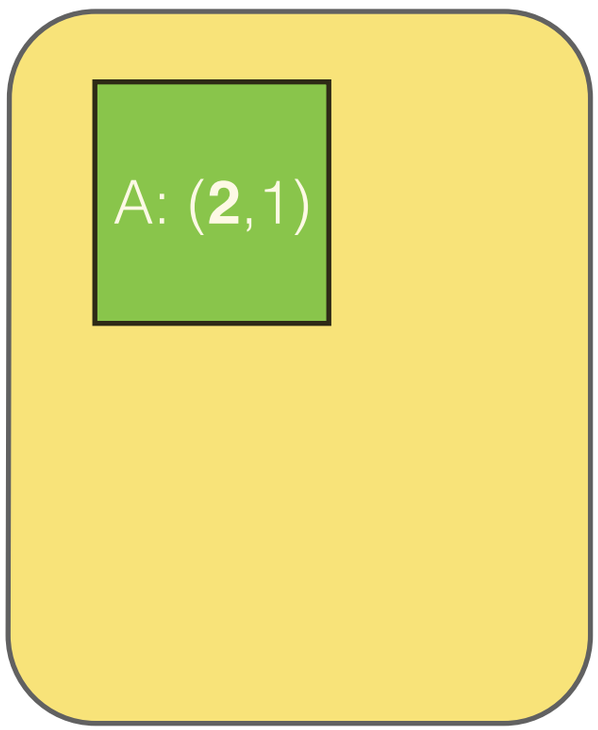
Buffer Policies

T1	R(A)	W(A)	**COMMIT**

Buffer Pool

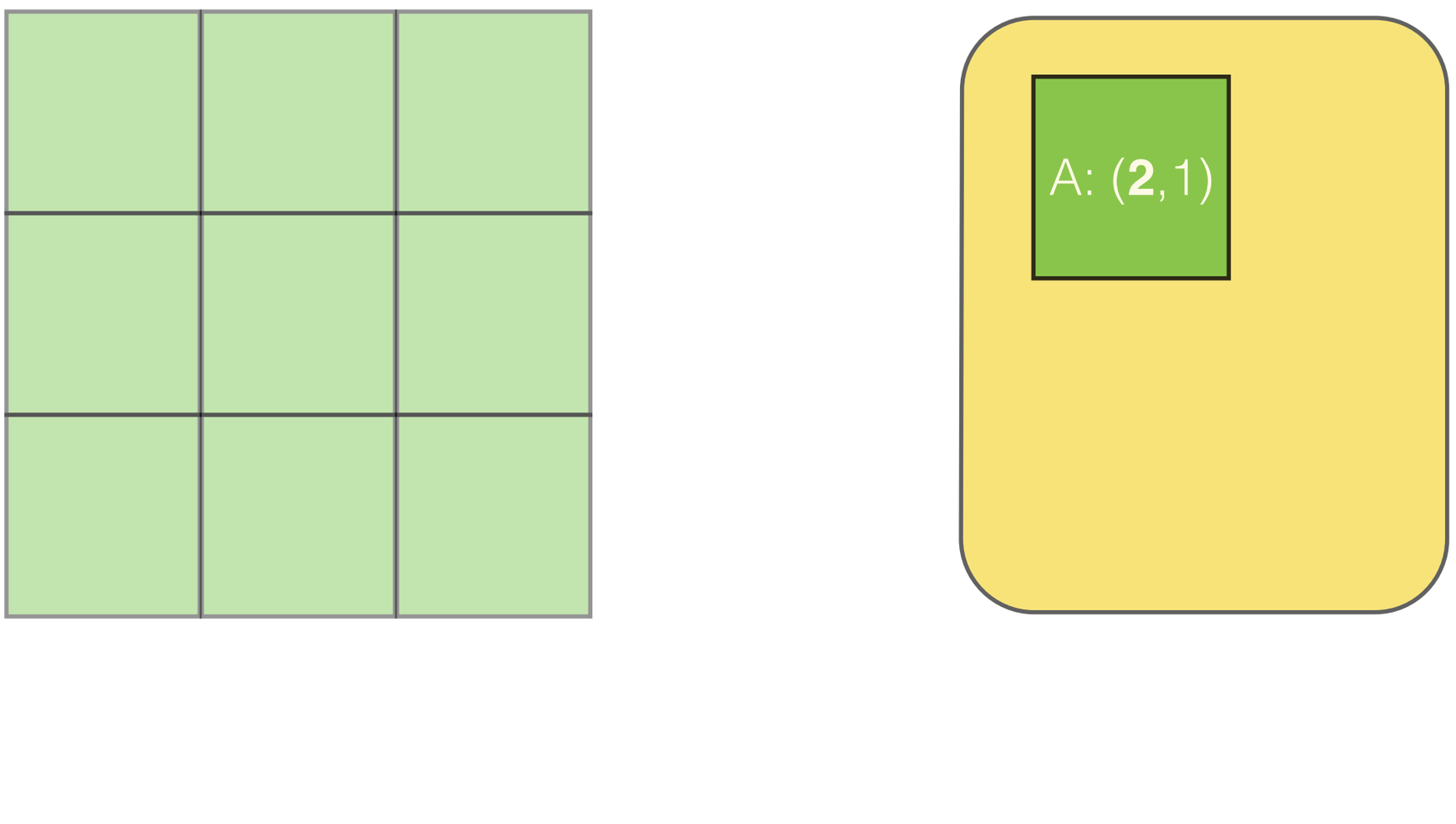


Disk



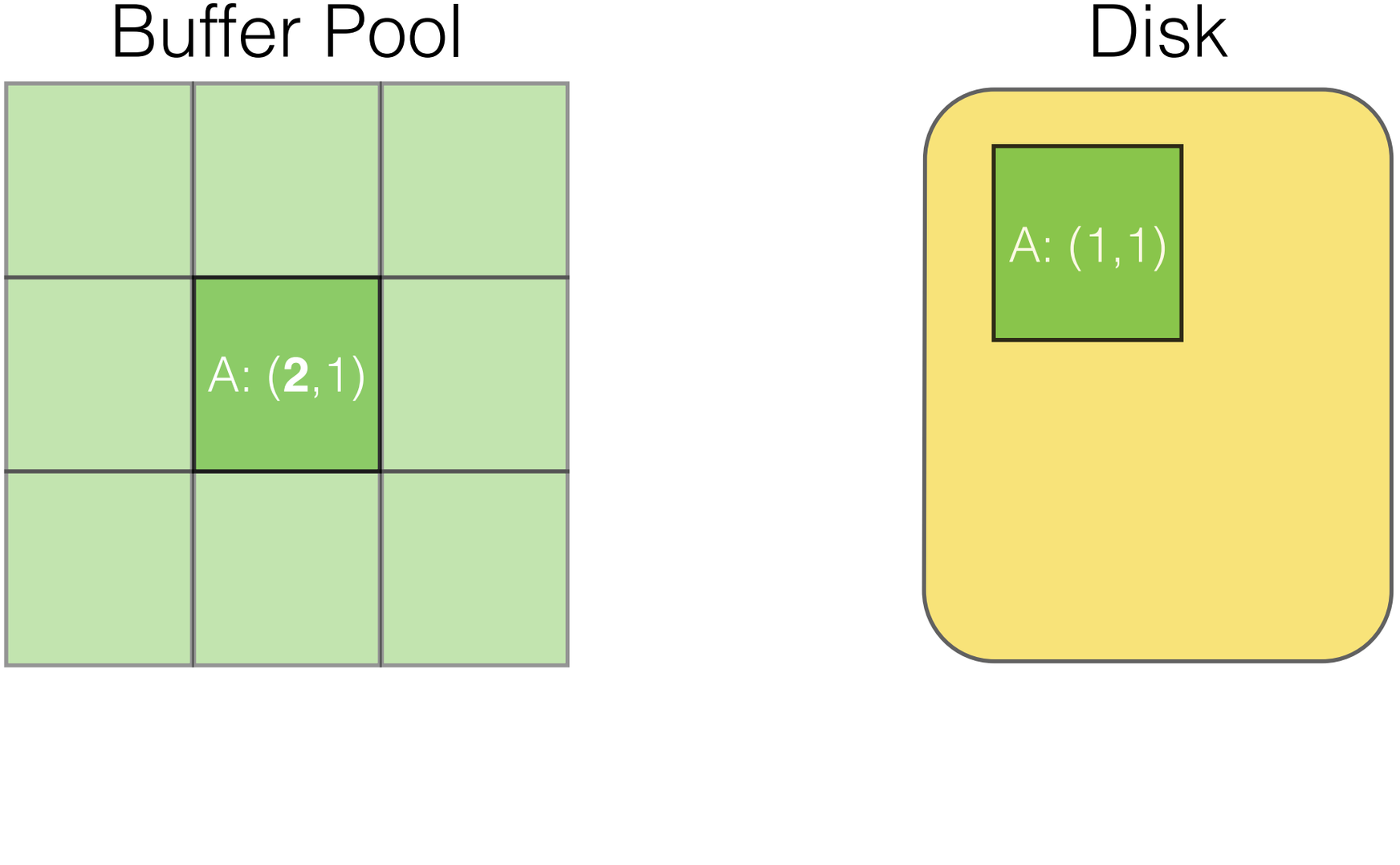
Buffer Policies

FORCE: “Force” buffer manager to write dirty pages to disk before committing



Buffer Policies

NO FORCE: Allow commits before updates are written to disk

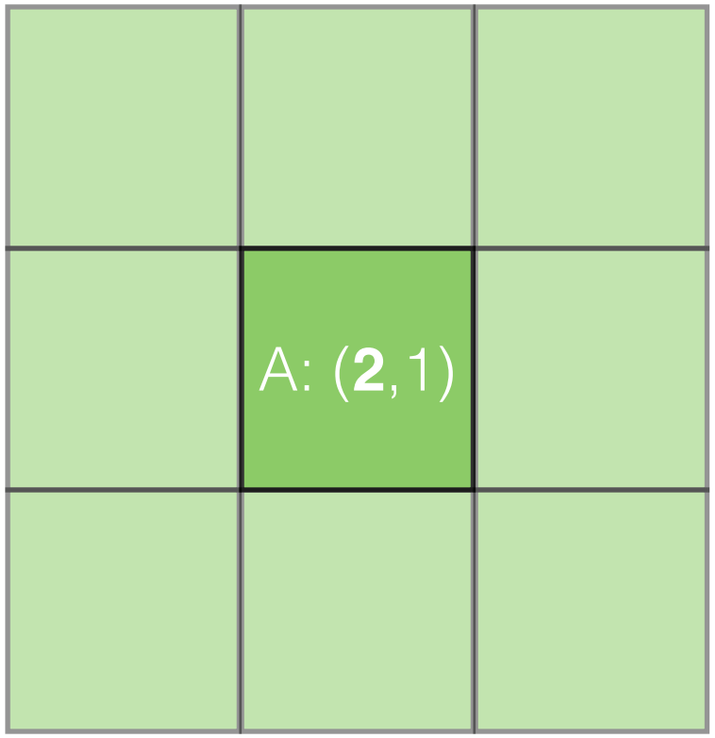


Buffer Policies

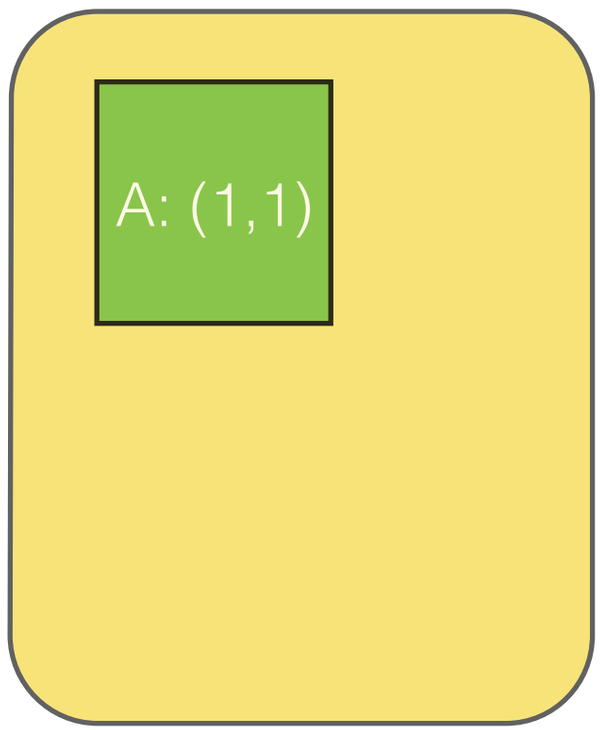
T1	R(A)	W(A)	**COMMIT**

CRASH

Buffer Pool



Disk

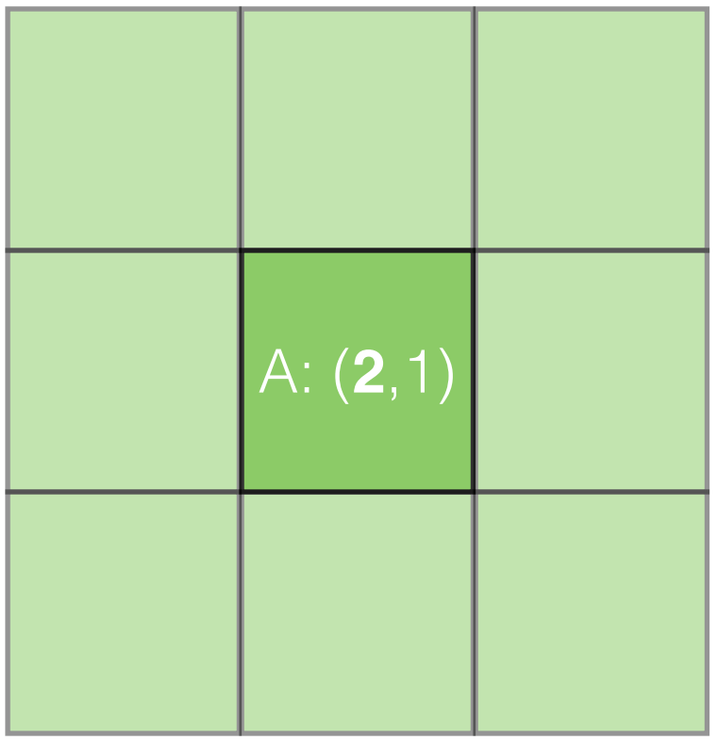


Buffer Policies

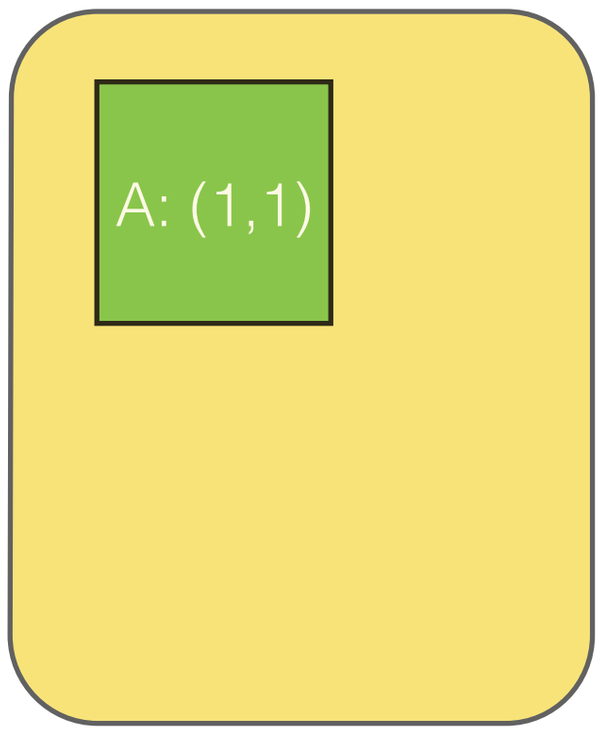
T1	R(A)	W(A)	**COMMIT**

CRASH

Buffer Pool



Disk



Could threaten durability! Need to **redo** actions.

Buffer Policies

- **FORCE**: “Force” buffer manager to write dirty pages to disk before committing

- **NO FORCE**:
  - Allow commits before updates are written to disk
  - Requires REDO to preserve durability

Buffer Policies

- **NO STEAL/FORCE**
  - Guarantees atomicity and durability
  - Slow at commit time
- **STEAL/NO FORCE**
  - No guarantees about atomicity and durability
  - Much faster
  - Use write ahead logging!

Write-Ahead Logging

- Log everything:
  - Starts
  - Updates
  - Commits
  - Aborts

LSN	Log	prevLSN
10	T1 Start	null
20	T1 writes P5	10
30	T1 writes P1	20
40	T1 Commit	30
50	T1 End	40

- Force log record for updates before updated data written to disk

- Transaction not committed until all logs on disk

Log Record

<LSN, pageID, offset, old data, new data, prevLSN>

20	Update: T1 writes P5	10

- LSN (“Log Sequence Number”): globally increasing ID for log records

- prevLSN: LSN of the last operation for this xact

Transaction Table

- Tells which xacts are currently running
- Contains:
  - XID: Transaction ID
  - Status: Running/Committing/Aborting
  - lastLSN: most recent LSN written by xact

⋮

50	T1 writes P5	40
60	T2 Commit	30
70	T3 Abort	20
80	T4 Commit	10
90	T4 End	80

⋮

50	T1 writes P5	40
60	T2 Commit	30
70	T3 Abort	20
80	T4 Commit	10
90	T4 End	80

Transaction Table @ Time 40

XID	State	LastLSN
1	Running	40
2	Running	30
3	Running	20
4	Running	10

⋮

50	T1 writes P5	40
60	T2 Commit	30
70	T3 Abort	20
80	T4 Commit	10
90	T4 End	80

Transaction Table

XID	State	LastLSN
1	Running	50
2	Running	30
3	Running	20
4	Running	10

⋮

50	T1 writes P5	40
60	T2 Commit	30
70	T3 Abort	20
80	T4 Commit	10
90	T4 End	80

Transaction Table

XID	State	LastLSN
1	Running	50
2	Committing	60
3	Running	20
4	Running	10

⋮

50	T1 writes P5	40
60	T2 Commit	30
70	T3 Abort	20
80	T4 Commit	10
90	T4 End	80

Transaction Table

XID	State	LastLSN
1	Running	50
2	Committing	60
3	Aborting	70
4	Running	10

⋮

50	T1 writes P5	40
60	T2 Commit	30
70	T3 Abort	20
80	T4 Commit	10
90	T4 End	80

Transaction Table

XID	State	LastLSN
1	Running	50
2	Committing	60
3	Aborting	70
4	Committing	80

⋮

50	T1 writes P5	40
60	T2 Commit	30
70	T3 Abort	20
80	T4 Commit	10
90	T4 End	80

Transaction Table

XID	State	LastLSN
1	Running	50
2	Committing	60
3	Aborting	70

Dirty Page Table

- Tells which buffer pages are dirty
- Contains:
  - PageID
  - recLSN: LSN of first update that dirtied this page

Page ID	recLSN
1	10
5	30
6	40

Normal Execution of Xact

- Series of reads and writes followed by commit or abort

- Commit: Flush logs to disk

- Abort: Undo all of xact's changes
  - Get lastLSN of xact and follow chain of prevLSNs
  - Write a CLR ("compensation log record") for each UNDO

Example Abort

LSN	Log	prevLSN
20	T1 writes P5	null
30	T1 writes P6	20
40	T1 Abort	30

Example Abort

LSN	Log	prevLSN
20	T1 writes P5	null
30	T1 writes P6	20
40	T1 Abort	30

Example Abort

LSN	Log	prevLSN
20	T1 writes P5	null
30	T1 writes P6	20
40	T1 Abort	30
50	CLR: Undo T1 LSN=30, undoNextLSN=20	40

Example Abort

LSN	Log	prevLSN
20	T1 writes P5	null
30	T1 writes P6	20
40	T1 Abort	30
50	CLR: Undo T1 LSN=30, undoNextLSN=20	40
60	CLR: Undo T1 LSN=20, undoNextLSN=null	50

Example Abort

LSN	Log	prevLSN
20	T1 writes P5	null
30	T1 writes P6	20
40	T1 Abort	30
50	CLR: Undo T1 LSN=30, undoNextLSN=20	40
60	CLR: Undo T1 LSN=20, undoNextLSN=null	50
70	T1 End	60

Checkpoints

- Occasionally create checkpoints to minimize recovery time

- begin_checkpoint: Indicates when checkpt began

- end_checkpoint:
  - Record contains current xact table and dirty page table
  - Accurate only as of time of begin_checkpoint

- Store LSN of most recent checkpoint

ARIES

- Find failed and committed xacts since checkpoint
- Re-apply changes made by committed xacts
- Undo changes made by failed xacts

Analyze - DPT

- Rebuilding dirty page table:
  - Start from checkpoint DPT
  - Add new entry for every dirtied page
    - recLSN = LSN
- Create conservative approximation of DPT
  - Entries may have already been flushed

LSN	Log	prevLSN
30	T1 writes P5	10
40	T1 writes P6	30
50	T2 writes P1	20

Dirty Page Table @ Checkpoint

Page ID	recLSN
1	10

LSN	Log	prevLSN
30	T1 writes P5	10
40	T1 writes P6	30
50	T2 writes P1	20

Dirty Page Table

Page ID	recLSN
1	10
5	30

LSN	Log	prevLSN
30	T1 writes P5	10
40	T1 writes P6	30
50	T2 writes P1	20

Dirty Page Table

Page ID	recLSN
1	10
5	30
6	40

Analyze - Xact Table

- Rebuilding xact table:
  - Remove xacts when you see END
  - Add/change xact states and lastLSNs as you go
  - Table will be precisely correct to last log flush before crash

30	T1 writes P6	20
40	T1 Abort	30
50	CLR: Undo T1 LSN=30, undoNextLSN=20	40
60	CLR: Undo T1 LSN=20, undoNextLSN=null	50
70	T2 End	10

Transaction Table @ Checkpoint

XID	State	LastLSN
1	Running	20
2	Committing	10

30	T1 writes P6	20
40	T1 Abort	30
50	CLR: Undo T1 LSN=30, undoNextLSN=20	40
60	CLR: Undo T1 LSN=20, undoNextLSN=null	50
70	T2 End	10

Transaction Table

XID	State	LastLSN
1	Running	30
2	Committing	10

30	T1 writes P6	20
40	T1 Abort	30
50	CLR: Undo T1 LSN=30, undoNextLSN=20	40
60	CLR: Undo T1 LSN=20, undoNextLSN=null	50
70	T2 End	10

Transaction Table

XID	State	LastLSN
1	Aborting	40
2	Committing	10

30	T1 writes P6	20
40	T1 Abort	30
50	CLR: Undo T1 LSN=30, undoNextLSN=20	40
60	CLR: Undo T1 LSN=20, undoNextLSN=null	50
70	T2 End	10

Transaction Table

XID	State	LastLSN
1	Aborting	50
2	Committing	10

30	T1 writes P6	20
40	T1 Abort	30
50	CLR: Undo T1 LSN=30, undoNextLSN=20	40
60	CLR: Undo T1 LSN=20, undoNextLSN=null	50
70	T2 End	10

Transaction Table

XID	State	LastLSN
1	Aborting	60
2	Committing	10

30	T1 writes P6	20
40	T1 Abort	30
50	CLR: Undo T1 LSN=30, undoNextLSN=20	40
60	CLR: Undo T1 LSN=20, undoNextLSN=null	50
70	T2 End	10

Transaction Table

XID	State	LastLSN
1	Aborting	60

Worksheet #1a,b

The log record at LSN 60 says that transaction 2 updated page 5. Was this update to page 5 successfully written to disk?

Transaction Table			**Dirty Page Table**	
Transaction	**lastLSN**	**Status**	**PageID**	**recLSN**
T1	70	Running	P5	50
T2	60	Running	P1	40
T3	30	Running		
T4	50	Running		

The log record at LSN 60 says that transaction 2 updated page 5. Was this update to page 5 successfully written to disk?

Transaction Table			**Dirty Page Table**	
Transaction	**lastLSN**	**Status**	**PageID**	**recLSN**
T1	70	Running	P5	50
T2	60	Running	P1	40
T3	30	Running		
T4	50	Running		

Update at LSN 60 MAY have been written to disk. The page was not yet flushed at the time of the checkpoint, but may have flushed later, because of the NO FORCE policy.

The log record at LSN 70 says that transaction 1 updated page 2. Was this update to page 2 successfully written to disk?

Transaction Table			**Dirty Page Table**	
Transaction	**lastLSN**	**Status**	**PageID**	**recLSN**
T1	70	Running	P5	50
T2	60	Running	P1	40
T3	30	Running		
T4	50	Running		

The log record at LSN 70 says that transaction 1 updated page 2. Was this update to page 2 successfully written to disk?

Transaction Table			**Dirty Page Table**	
Transaction	**lastLSN**	**Status**	**PageID**	**recLSN**
T1	70	Running	P5	50
T2	60	Running	P1	40
T3	30	Running		
T4	50	Running		

Update at LSN 70 was flushed to disk because P2 was not in the dirty page table at the time of the checkpoint.

Transaction Table

XID	LastLSN	Status
T1	70	Running
T2	60	Running
T3	30	Running
T4	50	Running

Dirty Page Table

Page ID	recLSN
P5	50
P1	40

Transaction Table

XID	LastLSN	Status
T1	90	Running
T2	60	Running
T3	30	Running
T4	50	Running

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90

Transaction Table

XID	LastLSN	Status
T1	90	Running
T2	110	Running
T3	30	Running
T4	50	Running

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90

Transaction Table

XID	LastLSN	Status
T1	90	Running
T2	120	Committing
T3	30	Running
T4	50	Running

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90

Transaction Table

XID	LastLSN	Status
T1	90	Running
T2	120	Committing
T3	30	Running
T4	130	Running

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90

Transaction Table

XID	LastLSN	Status
T1	90	Running
T3	30	Running
T4	130	Running

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90

Transaction Table

XID	LastLSN	Status
T1	90	Running
T3	30	Running
T4	150	Aborting

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90

Transaction Table

XID	LastLSN	Status
T1	90	Running
T3	30	Running
T4	150	Aborting
T5	160	Running

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

Transaction Table

XID	LastLSN	Status
T1	90	Running
T3	30	Running
T4	180	Aborting
T5	160	Running

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

REDO

- Redo changes that didn’t make it out to disk
- Start at the smallest recLSN in DPT
- Redo each log record or CLR except if:
  - Affected page is not in DPT
  - Affected page is in DPT, but:
    - recLSN > LSN, or
    - pageLSN (in DB) ≥ LSN

LSN	Log	prevLSN
30	T1 writes P5	10
40	T1 writes P6	30
50	T2 writes P1	20
60	T1 writes P1	40

Dirty Page Table

Page ID	recLSN
1	60
6	40

LSN	Log	prevLSN
30	T1 writes P5	10
40	T1 writes P6	30
50	T2 writes P1	20
60	T1 writes P1	40

Dirty Page Table

Page ID	recLSN
1	60
6	40

REDO: 40

LSN	Log	prevLSN
30	T1 writes P5	10
40	T1 writes P6	30
50	T2 writes P1	20
60	T1 writes P1	40

Dirty Page Table

Page ID	recLSN
1	60
6	40

REDO: 40

LSN	Log	prevLSN
30	T1 writes P5	10
40	T1 writes P6	30
50	T2 writes P1	20
60	T1 writes P1	40

Dirty Page Table

Page ID	recLSN
1	60
6	40

REDO: 40, 60

Worksheet #1c

At which LSN in the log should redo begin? Which log records will be redone (list their LSNs)? All other log records will be skipped.

Start at the smallest recLSN in DPT: 40

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO:

Start at the smallest recLSN in DPT: 40

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO: 40

Start at the smallest recLSN in DPT: 40

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO: 40, 50

Start at the smallest recLSN in DPT: 40

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO: 40, 50, 60

Start at the smallest recLSN in DPT: 40

recLSN > LSN

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO: 40, 50, 60

Start at the smallest recLSN in DPT: 40

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO: 40, 50, 60, 90

Start at the smallest recLSN in DPT: 40

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO: 40, 50, 60, 90, 110

Start at the smallest recLSN in DPT: 40

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO: 40, 50, 60, 90, 110, 130

Start at the smallest recLSN in DPT: 40

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO: 40, 50, 60, 90, 110, 130, 160

Start at the smallest recLSN in DPT: 40

Dirty Page Table

Page ID	recLSN
P5	50
P1	40
P3	90
P2	160

LSN	Record	prevLSN
30	update: T3 writes P5	null
40	update: T4 writes P1	null
50	update: T4 writes P5	40
60	update: T2 writes P5	null
70	update: T1 writes P2	null
80	begin checkpoint	-
90	update: T1 writes P3	70
100	end checkpoint	-
110	update: T2 writes P3	60
120	T2 commit	110
130	update: T4 writes P1	50
140	T2 end	120
150	T4 abort	130
160	update: T5 writes P2	null
180	CLR: undo T4 LSN 130	150

REDO: 40, 50, 60, 90, 110, 130, 160, 180

UNDO

- Undo changes by unfinished “loser” transactions

XID	State	LastLSN
1	Running	60
2	Running	80

UNDO

- ToUndo = {lastLSN of all Xacts in Xact Table}
- while ToUndo not empty:
  - Choose largest LSN in ToUndo (most recent)
  - If LSN is an update record:
    - UNDO, write CLR, and add prevLSN to ToUndo.
  - If LSN is a CLR and undoNextLSN != null:
    - Add undoNextLSN to ToUndo
  - If LSN is a CLR and undoNextLSN == null:
    - Write END

UNDO

- Undo changes by unfinished “loser” transactions

XID	State	LastLSN
1	Running	60
2	Running	80

toUndo: 60, 80

UNDO Update

toUndo: 60, 80

LSN	Log	prevLSN
50	T1 writes P5	40
60	T1 writes P6	50
70	T2 writes P1	30
80	T2 writes P1	70

UNDO Update

toUndo: 60, **80**

LSN	Log	prevLSN
50	T1 writes P5	40
60	T1 writes P6	50
70	T2 writes P1	30
80	T2 writes P1	70

UNDO Update

toUndo: 60, 70

LSN	Log	prevLSN
50	T1 writes P5	40
60	T1 writes P6	50
70	T2 writes P1	30
80	T2 writes P1	70
90	CLR: Undo T2 LSN=80, undoNextLSN=70	80

UNDO CLR

toUndo: 60, 80

LSN	Log	prevLSN
50	T1 writes P5	40
60	T2 writes P6	50
70	T2 Abort	30
80	CLR: Undo T2 LSN=60, undoNextLSN=50	70

UNDO CLR

toUndo: 60, **80**

LSN	Log	prevLSN
50	T1 writes P5	40
60	T2 writes P6	50
70	T2 Abort	30
80	CLR: Undo T2 LSN=60, undoNextLSN=50	70

UNDO CLR

toUndo: 60, 50

LSN	Log	prevLSN
50	T1 writes P5	40
60	T2 writes P6	50
70	T2 Abort	30
80	CLR: Undo T2 LSN=60, undoNextLSN=50	70

UNDO CLR

toUndo: 60, 80

LSN	Log	prevLSN
50	T1 writes P5	40
60	T2 writes P6	null
70	T2 Abort	30
80	CLR: Undo T2 LSN=60, undoNextLSN=null	70

UNDO CLR

toUndo: 60, **80**

LSN	Log	prevLSN
50	T1 writes P5	40
60	T2 writes P6	null
70	T2 Abort	30
80	CLR: Undo T2 LSN=60, undoNextLSN=null	70

UNDO CLR

toUndo: 60

LSN	Log	prevLSN
50	T1 writes P5	40
60	T2 writes P6	null
70	T2 Abort	30
80	CLR: Undo T2 LSN=60, undoNextLSN=null	70
90	T2 End	80

Worksheet #2

During Analysis, what log records are read? What are the contents of the transaction table and the dirty page table at the end of the analysis stage?

During Analysis, what log records are read? What are the contents of the transaction table and the dirty page table at the end of the analysis stage?

- All records since last checkpoint are read.

Transaction Table

XID	LastLSN	Status

Dirty Page Table

Page ID	recLSN

Transaction Table

XID	LastLSN	Status
T1	10	Running

Dirty Page Table

Page ID	recLSN
P1	10

Transaction Table

XID	LastLSN	Status
T1	10	Running
T2	20	Running

Dirty Page Table

Page ID	recLSN
P1	10
P3	20

Transaction Table

XID	LastLSN	Status
T1	30	Committing
T2	20	Running

Dirty Page Table

Page ID	recLSN
P1	10
P3	20

Transaction Table

XID	LastLSN	Status
T1	30	Committing
T2	20	Running
T3	40	Running

Dirty Page Table

Page ID	recLSN
P1	10
P3	20
P4	40

Transaction Table

XID	LastLSN	Status
T1	30	Committing
T2	50	Running
T3	40	Running

Dirty Page Table

Page ID	recLSN
P1	10
P3	20
P4	40

Transaction Table

XID	LastLSN	Status
T2	50	Running
T3	40	Running

Dirty Page Table

Page ID	recLSN
P1	10
P3	20
P4	40

Transaction Table

XID	LastLSN	Status
T2	50	Running
T3	70	Running

Dirty Page Table

Page ID	recLSN
P1	10
P3	20
P4	40
P2	70

Transaction Table

XID	LastLSN	Status
T2	80	Aborting
T3	70	Running

Dirty Page Table

Page ID	recLSN
P1	10
P3	20
P4	40
P2	70

During Redo, what log records are read? What data pages are read? What operations are redone (assuming no updates made it out to disk before the crash)?

During Redo, what log records are read? What data pages are read? What operations are redone (assuming no updates made it out to disk before the crash)?

- Read all log records after 10 (smallest recLSN in DPT)

- Read all pages in DPT

- Assuming no updates made it to disk, all updates and CLR's are redone.
  - LSN's: 10, 20, 40, 50, 70.

During Undo, what log records are read? What operations are undone? Show any new log records that are written for CLR's. Start at LSN 100.

During Undo, what log records are read? What operations are undone? Show any new log records that are written for CLR's. Start at LSN 100.

ToUndo: 80, 70

During Undo, what log records are read? What operations are undone? Show any new log records that are written for CLR's. Start at LSN 100.

ToUndo: 70, 50

Read: 80

During Undo, what log records are read? What operations are undone? Show any new log records that are written for CLR's. Start at LSN 100.

ToUndo: 50, 40

LSN	Log Record
100	CLR T3 LSN = 70; undoNextLSN = 40

Read: 80, 70
Undone: 70

During Undo, what log records are read? What operations are undone? Show any new log records that are written for CLR's. Start at LSN 100.

ToUndo: 40, 20

LSN	Log Record
100	CLR T3 LSN = 70; undoNextLSN = 40
110	CLR T2 LSN = 50; undoNextLSN = 20

Read: 80, 70, 50

Undone: 70, 50

During Undo, what log records are read? What operations are undone? Show any new log records that are written for CLR's. Start at LSN 100.

ToUndo: 20

LSN	Log Record
100	CLR T3 LSN = 70; undoNextLSN = 40
110	CLR T2 LSN = 50; undoNextLSN = 20
120	CLR T3 LSN = 40; undoNextLSN = null
130	T3 End

Read: 80, 70, 50, 40

Undone: 70, 50, 40

During Undo, what log records are read? What operations are undone? Show any new log records that are written for CLR's. Start at LSN 100.

ToUndo:

LSN	Log Record
100	CLR T3 LSN = 70; undoNextLSN = 40
110	CLR T2 LSN = 50; undoNextLSN = 20
120	CLR T3 LSN = 40; undoNextLSN = null
130	T3 End
140	CLR T2 LSN = 20; undoNextLSN = null
150	T2 End

Read: 80, 70, 50, 40, 20

Undone: 70, 50, 40, 20