

# CSE 444: Database Inter

## Section 9:

Transactions – Recovery with AF

# Review in this section

Recovery for ARIES

Follows homework closely

# ARIES

- A popular protocol for UNDO-REDO logging
- **Steal (like UNDO)**
  - Changes by uncommitted transactions can be written to a dirty page is flushed
- **No-force (like REDO)**
  - Changes by committed transactions may not have been
- **Write-ahead logging:**
  - Any changes to a database object is first recorded in the log before the change to database object is written to disk, before the change to database object
  - A record of every change to the database is available with the log from a crash

# ARIES Data Structures

## ▪ Active Transactions Table

- Lists all active TXN's
- For each TXN: **lastLSN** = its most recent update

## ▪ Dirty Page Table

- Lists all dirty pages
- For each dirty page: **recoveryLSN** (**recLSN**) = first LSN that caused page to become dirty

## ▪ Write Ahead Log

- LSN, **prevLSN** = previous LSN for same txn

# ARIES Data Structure

Dirty page table

| pageID | recLSN |
|--------|--------|
|        |        |
|        |        |
|        |        |

LSN  
101

Log

| prevLSN | tID | pID | Log entry |
|---------|-----|-----|-----------|
|         |     |     |           |
|         |     |     |           |
|         |     |     |           |
|         |     |     |           |

Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
|         |         |        |
|         |         |        |

# Log Record “Types”

- **Update:** easy
- **Commit:** log-tail forced-written to disk, up to & including c (note that still no-force, the actual modified pages may not be much smaller cost)
- **Abort:** abort type log record is written + undo is initiated for transaction
- **End:** when a transaction is aborted or committed, some actions are performed, after that an end record is written
- **CLR:**  
Undoing updates (during abort or recovery from crash),  
for every update record undone, write a CLR (Compensation)

Example.

1.  $T_{1000}$  changes the value of **A** from “abc” to “def” on
2.  $T_{2000}$  changes the value of **B** from “hij” to “klm” on p
3.  $T_{2000}$  changes the value of **D** from “mnp” to “qrs” on
4.  $T_{1000}$  changes the value of **C** from “tuv” to “wxy” on
5.  $T_{2000}$  commits and the end log record is written
6.  $T_{1000}$  changes the value of **E** from “pq” to “rs” on pag
7. **P600** is flushed to disk
8. **Crash!!**

# ARIES Data Structure

## Dirty page table

| pageID | recLSN |
|--------|--------|
|        |        |
|        |        |
|        |        |

LSN  
101

## Log

| prevLSN | tID | pID | Log entry |
|---------|-----|-----|-----------|
|         |     |     |           |
|         |     |     |           |
|         |     |     |           |
|         |     |     |           |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
|         |         |        |
|         |         |        |

## Buffer Pool

**P500**

PageLSN= -

A = abc D = mnp

**P600**

PageLSN= -

B = hij

**P505**

PageLSN= -

C = tuv

**P700**

PageLSN= -

E = pq

**PageLSN= -**

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

First operation:

1.  $T_{1000}$  changes the value of A from “abc” to “def” o

Dirty page table

| pageID | recLSN |
|--------|--------|
|        |        |
|        |        |

LSN  
101

Log

| prevLSN | tID | pID | Log entry |
|---------|-----|-----|-----------|
|         |     |     |           |
|         |     |     |           |
|         |     |     |           |
|         |     |     |           |

Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
|         |         |        |

Buffer Pool

P500

PageLSN= -

A = abc D = mnp

P600

PageLSN= -

B = hij

P505

PageLSN= -

C = tuv

P700

PageLSN= -

E = pq

PageLSN= -

A = abc D = mnp

P505

PageLSN= -

C = tuv

Changes

1.  $T_{1000}$  changes the value of A from “abc” to “def” o

Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
|        |        |
|        |        |

Log

| LSN | prevLSN | tID   | pID  | Log entry              |
|-----|---------|-------|------|------------------------|
| 101 | -       | T1000 | P500 | Write A<br>“abc” -> “d |
|     |         |       |      |                        |
|     |         |       |      |                        |
|     |         |       |      |                        |

Transaction table

| transID    | lastLSN | status  |
|------------|---------|---------|
| $T_{1000}$ | 101     | Running |
|            |         |         |

Buffer Pool

P500  
PageLSN= 101

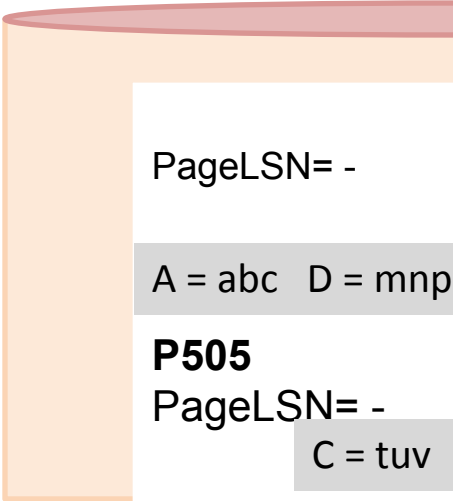
P600  
PageLSN= -

A = def D = mnp

B = hij

P505  
PageLSN= -  
C = tuv

P700  
PageLSN= -  
E = pq



# Comments

In hw4/lecture notes,

## 1. “Transaction T3 writes A”

If it is stored in LSN8 (say), and A is not written earlier, then log entry = **“Write A, A -> A8”**

2. Also, “**page id**” was not mentioned in log entries places (implied from the log entry)

It is ok if you do not write page Id in hw4

3. Note the “**status**” column in transaction table was omitted in some lecture 6 slides

Next:

2.  $T_{2000}$  changes the value of **B** from “hij” to “klm” on p

Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
|        |        |
|        |        |

Log

| LSN | prevLSN | tID   | pID  | Log entry              |
|-----|---------|-------|------|------------------------|
| 101 | -       | T1000 | P500 | Write A<br>“abc” -> “d |
|     |         |       |      |                        |
|     |         |       |      |                        |
|     |         |       |      |                        |

Transaction table

| transID    | lastLSN | status  |
|------------|---------|---------|
| $T_{1000}$ | 101     | Running |
|            |         |         |

Buffer Pool

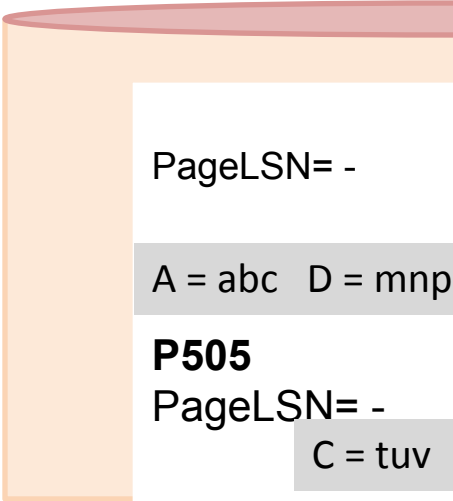
**P500**  
PageLSN= 101

**P600**  
PageLSN= -

A = def    D = mnp                      B = hij

**P505**  
PageLSN= -  
C = tuv

**P700**  
PageLSN= -  
E = pq



## Changes:

2.  $T_{2000}$  changes the value of **B** from “hij” to “klm” on p

### Dirty page table

| pageID      | recLSN     |
|-------------|------------|
| P500        | 101        |
| <b>P600</b> | <b>102</b> |
|             |            |

LSN

101

102

### Log

| prevLSN | tID        | pID         | Log entry                       |
|---------|------------|-------------|---------------------------------|
| -       | T1000      | P500        | Write A<br>“abc” -> “d          |
| -       | $T_{2000}$ | <b>P600</b> | <b>Write B</b><br>“hij” -> “klm |
|         |            |             |                                 |
|         |            |             |                                 |

### Transaction table

| transID                      | lastLSN    | status         |
|------------------------------|------------|----------------|
| $T_{1000}$                   | 101        | Running        |
| <b><math>T_{2000}</math></b> | <b>102</b> | <b>Running</b> |

### Buffer Pool

**P500**

PageLSN= 101

A = def D = mnp

**P505**

PageLSN= -

C = tuv

**P600**

**PageLSN= 102**

**B = klm**

**P700**

PageLSN= -

E = pq

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

Next:

3.  $T_{2000}$  changes the value of **D** from “mnp” to “qrs” on

### Dirty page table

| pageID      | recLSN     |
|-------------|------------|
| P500        | 101        |
| <b>P600</b> | <b>102</b> |
|             |            |

LSN

101

102

### Log

| prevLSN | tID        | pID         | Log entry                       |
|---------|------------|-------------|---------------------------------|
| -       | T1000      | P500        | Write A<br>“abc” -> “d          |
| -       | $T_{2000}$ | <b>P600</b> | <b>Write B</b><br>“hij” -> “klm |
|         |            |             |                                 |
|         |            |             |                                 |

### Transaction table

| transID                      | lastLSN    | status         |
|------------------------------|------------|----------------|
| $T_{1000}$                   | 101        | Running        |
| <b><math>T_{2000}</math></b> | <b>102</b> | <b>Running</b> |

### Buffer Pool

**P500**

PageLSN= 101

A = def D = mnp

**P505**

PageLSN= -

C = tuv

**P600**

**PageLSN= 102**

**B = klm**

**P700**

PageLSN= -

E = pq

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

## Changes:

3.  $T_{2000}$  changes the value of **D** from “mnp” to “qrs” on

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
|        |        |

### Transaction table

| transID    | lastLSN | status  |
|------------|---------|---------|
| $T_{1000}$ | 101     | Running |
| $T_{2000}$ | 103     | Running |

### Log

LSN

101

102

103

| prevLSN | tID        | pID  | Log entry                |
|---------|------------|------|--------------------------|
| -       | $T_{1000}$ | P500 | Write A<br>“abc” -> “d   |
| -       | $T_{2000}$ | P600 | Write B<br>“hij” -> “klm |
| 102     | $T_{2000}$ | P500 | Write D<br>“mnp” -> “q   |
|         |            |      |                          |

### Buffer Pool

**P500**

PageLSN= 103

A = def **D = qrs**

**P505**

PageLSN= -

C = tuv

**P600**

PageLSN= 102

B = klm

**P700**

PageLSN= -

E = pq

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

Next:

4.  $T_{1000}$  changes the value of **C** from “tuv” to “wxy” on

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
|        |        |

### Transaction table

| transID    | lastLSN | status  |
|------------|---------|---------|
| $T_{1000}$ | 101     | Running |
| $T_{2000}$ | 103     | Running |

### Log

LSN

101

102

103

| prevLSN | tID        | pID  | Log entry                |
|---------|------------|------|--------------------------|
| -       | $T_{1000}$ | P500 | Write A<br>“abc” -> “d   |
| -       | $T_{2000}$ | P600 | Write B<br>“hij” -> “klm |
| 102     | $T_{2000}$ | P500 | Write D<br>“mnp” -> “q   |
|         |            |      |                          |

### Buffer Pool

**P500**

PageLSN= 103

A = def **D = qrs**

**P505**

PageLSN= -

C = tuv

**P600**

PageLSN= 102

B = klm

**P700**

PageLSN= -

E = pq

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

Changes:

4.  $T_{1000}$  changes the value of **C** from “tuv” to “wxy” on

Dirty page table

| pageID      | recLSN     |
|-------------|------------|
| P500        | 101        |
| P600        | 102        |
| <b>P505</b> | <b>104</b> |

LSN

101

102

103

104

Log

| prevLSN    | tID                     | pID         | Log entry                                  |
|------------|-------------------------|-------------|--------------------------------------------|
| -          | T1000                   | P500        | Write A<br>“abc” -> “de                    |
| -          | T <sub>2000</sub>       | P600        | Write B<br>“hij” -> “klm”                  |
| 102        | T <sub>2000</sub>       | P500        | Write D<br>“mnp” -> “qrs                   |
| <b>101</b> | <b>T<sub>1000</sub></b> | <b>P505</b> | <b>Write C</b><br><b>“tuv” -&gt; “wxy”</b> |

Transaction table

| transID                 | lastLSN    | status         |
|-------------------------|------------|----------------|
| <b>T<sub>1000</sub></b> | <b>104</b> | <b>Running</b> |
| T <sub>2000</sub>       | 103        | Running        |

Buffer Pool

**P500**

PageLSN= 103

**P600**

PageLSN= 102

A = def D = qrs

B = klm

**P505**

**PageLSN= 104**

**C = wxy**

**P700**

PageLSN= -

E = pq

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

Next:

5.  $T_{2000}$  commits and the end log record is written

### Dirty page table

| pageID      | recLSN     |
|-------------|------------|
| P500        | 101        |
| P600        | 102        |
| <b>P505</b> | <b>104</b> |

### Transaction table

| transID                      | lastLSN    | status         |
|------------------------------|------------|----------------|
| <b><math>T_{1000}</math></b> | <b>104</b> | <b>Running</b> |
| $T_{2000}$                   | 103        | Running        |

### Log

LSN

101

102

103

104

| prevLSN    | tID                          | pID         | Log entry                                  |
|------------|------------------------------|-------------|--------------------------------------------|
| -          | $T_{1000}$                   | P500        | Write A<br>"abc" -> "de"                   |
| -          | $T_{2000}$                   | P600        | Write B<br>"hij" -> "klm"                  |
| 102        | $T_{2000}$                   | P500        | Write D<br>"mnp" -> "qrs"                  |
| <b>101</b> | <b><math>T_{1000}</math></b> | <b>P505</b> | <b>Write C</b><br><b>"tuv" -&gt; "wxy"</b> |

### Buffer Pool

**P500**

PageLSN= 103

A = def D = qrs

**P505**

**PageLSN= 104**

**C = wxy**

**P600**

PageLSN= 102

B = klm

**P700**

PageLSN= -

E = pq

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

Changes:

5.  $T_{2000}$  commits and the end log record is written

Log

Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |

LSN

101

102

103

104

105

106

| prevLSN | tID        | pID  | Log entry                 |
|---------|------------|------|---------------------------|
| -       | T1000      | P500 | Write A<br>"abc" -> "de"  |
| -       | $T_{2000}$ | P600 | Write B<br>"hij" -> "klm" |
| 102     | $T_{2000}$ | P500 | Write D<br>"mnp" -> "qrs" |
| 101     | $T_{1000}$ | P505 | Write C<br>"tuv" -> "wxy" |
| 103     | $T_{2000}$ |      |                           |
| 105     | $T_{2000}$ |      |                           |

Transaction table

| transID    | lastLSN | status    |
|------------|---------|-----------|
| $T_{1000}$ | 104     | Running   |
| $T_{2000}$ | 103     | Committed |

Buffer Pool

P500

PageLSN= 103

P600

PageLSN= 102

A = def D = qrs

B = klm

P505

PageLSN= 104

C = wxy

P700

PageLSN= -

E = pq

PageLSN= -

A = abc D = mnp

P505

PageLSN= -

C = tuv

Changes:

5.  $T_{2000}$  commits and the end log record is written

Log

Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |

T2000 removed from transaction table

Transaction table

| transID                          | lastLSN        | status               |
|----------------------------------|----------------|----------------------|
| $T_{1000}$                       | 104            | Running              |
| <del><math>T_{2000}</math></del> | <del>103</del> | <del>Committed</del> |

| LSN | prevLSN | tID        | pID  | Log entry                 |
|-----|---------|------------|------|---------------------------|
| 101 | -       | $T_{1000}$ | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | $T_{1000}$ | P600 | Write B<br>"hij" -> "klm" |
| 103 | -       | $T_{2000}$ | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | $T_{1000}$ | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | $T_{2000}$ |      |                           |
| 106 | 105     | $T_{2000}$ |      |                           |

Buffer Pool

P500

PageLSN= 103

A = def D = qrs

P505

PageLSN= 104

C = wxy

P600

PageLSN= 102

B = klm

P700

PageLSN= -

E = pq

PageLSN= -

A = abc D = mnp

P505

PageLSN= -

C = tuv

## Changes:

5.  $T_{2000}$  commits and the end log record is written to the **Log**

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |

### Transaction table

| transID    | lastLSN | status    |
|------------|---------|-----------|
| $T_{1000}$ | 104     | Running   |
| $T_{2000}$ | 103     | Committed |

| LSN | prevLSN | tID        | pID  | Log entry                 |
|-----|---------|------------|------|---------------------------|
| 101 | -       | $T_{1000}$ | P500 | Write A<br>"abc" -> "def" |
| 102 | -       | $T_{2000}$ | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | $T_{2000}$ | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | $T_{1000}$ | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | $T_{2000}$ |      |                           |
| 106 | 105     | $T_{2000}$ |      |                           |

### Buffer Pool

**P500**

PageLSN= 103

A = def D = qrs

**P505**

PageLSN= 104

C = wxy

**P600**

PageLSN= 102

B = klm

**P700**

PageLSN= -

E = pq

Log written

Note: no force  
pages changed

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

Whenever a transaction commits,  
log is flushed to the disk == the log-tail is written to

**NOTE:**

1. The “Commit” record is required to be flushed (i.e., the log-tail is written to and including that commit record)
2. The “End” record is not required to be flushed, i.e., we are only assuming that it has been flushed as well (we should have a good example while doing recovery 😊)

Next:

6.  $T_{1000}$  changes the value of **E** from “pq” to “rs” c  
Log

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |

### Transaction table

| transID    | lastLSN | status  |
|------------|---------|---------|
| $T_{1000}$ | 104     | Running |
|            |         |         |

| LSN | prevLSN | tID        | pID  | Log entry                 |
|-----|---------|------------|------|---------------------------|
| 101 | -       | $T_{1000}$ | P500 | Write A<br>“abc” -> “de   |
| 102 | -       | $T_{2000}$ | P600 | Write B<br>“hij” -> “klm” |
| 103 | 102     | $T_{2000}$ | P500 | Write D<br>“mnp” -> “qrs  |
| 104 | 101     | $T_{1000}$ | P505 | Write C<br>“tuv” -> “wxy  |
| 105 | 103     | $T_{2000}$ |      |                           |
| 106 | 105     | $T_{2000}$ |      |                           |

### Buffer Pool

**P500**

PageLSN= 103

A = def D = qrs

**P505**

PageLSN= 104

C = wxy

**P600**

PageLSN= 102

B = klm

**P700**

PageLSN= -

E = pq

**P500**

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

## Changes:

6.  $T_{1000}$  changes the value of E from “pq” to “rs”

Log

### Dirty page table

| pageID      | recLSN     |
|-------------|------------|
| P500        | 101        |
| P600        | 102        |
| P505        | 104        |
| <b>P700</b> | <b>107</b> |

### Transaction table

| transID    | lastLSN    | status         |
|------------|------------|----------------|
| $T_{1000}$ | <b>107</b> | <b>Running</b> |
|            |            |                |

| LSN | prevLSN | tID        | pID  | Log entry                 |
|-----|---------|------------|------|---------------------------|
| 101 | -       | $T_{1000}$ | P500 | Write A<br>“abc” -> “de   |
| 102 | -       | $T_{2000}$ | P600 | Write B<br>“hij” -> “klm” |
| 103 | 102     | $T_{2000}$ | P500 | Write D<br>“mnp” -> “qrs  |
| 104 | 101     | $T_{1000}$ | P505 | Write C<br>“tuv” -> “wxy  |
| 105 | 103     | $T_{2000}$ |      |                           |
| 106 | 105     | $T_{2000}$ |      |                           |
| 107 | 104     | $T_{1000}$ | P700 | Write E<br>“pq” -> “rs”   |

### Buffer Pool

**P500**

PageLSN= 103

A = def D = qrs

**P505**

PageLSN= 104

C = wxy

**P600**

PageLSN= 102

B = klm

**P700**

**PageLSN= 107**

**E = rs**

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

Next:

7. Page P600 is flushed to disk

Log

Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
| P700   | 107    |

Transaction table

| transID           | lastLSN | status  |
|-------------------|---------|---------|
| T <sub>1000</sub> | 107     | Running |
|                   |         |         |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T <sub>1000</sub> | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
| 107 | 104     | T <sub>1000</sub> | P700 | Write E<br>"pq" -> "rs"   |

Buffer Pool

P500

PageLSN= 103

P600

PageLSN= 102

A = def D = qrs

B = klm

P505

PageLSN= 104

C = wxy

P700

PageLSN= 107

E = rs

PageLSN= -

A = abc D = mnp

P505

PageLSN= -

C = tuv

Next:

## 7. Page P600 is flushed to disk – Step 1

### Dirty page table

| pageID          | recLSN         |
|-----------------|----------------|
| P500            | 101            |
| <del>P600</del> | <del>102</del> |
| P505            | 104            |
| P700            | 107            |

### Transaction table

| transID           | lastLSN | status  |
|-------------------|---------|---------|
| T <sub>1000</sub> | 107     | Running |
|                   |         |         |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T <sub>1000</sub> | P500 | Write A<br>“abc” -> “de   |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>“hij” -> “klm” |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>“mnp” -> “qrs  |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>“tuv” -> “wxy  |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
| 107 | 104     | T <sub>1000</sub> | P700 | Write E<br>“pq” -> “rs”   |

### Buffer Pool

**P500**

PageLSN= 103

A = def D = qrs

**P505**

PageLSN= 104

C = wxy

**P600**

PageLSN= 102

B = klm

**P700**

PageLSN= 107

E = rs

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

Next:

## 7. Page P600 is flushed to disk – Step 2

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P505   | 104    |
| P700   | 107    |

### Transaction table

| transID           | lastLSN | status  |
|-------------------|---------|---------|
| T <sub>1000</sub> | 107     | Running |
|                   |         |         |

| LSN | prevLSN | tID               | pID  | Log entry                |
|-----|---------|-------------------|------|--------------------------|
| 101 | -       | T <sub>1000</sub> | P500 | Write A<br>“abc” -> “d   |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>“hij” -> “klm |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>“mnp” -> “q   |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>“tuv” -> “w   |
| 105 | 103     | T <sub>2000</sub> |      |                          |
| 106 | 105     | T <sub>2000</sub> |      |                          |
| 107 | 104     | T <sub>1000</sub> | P700 | Write E<br>“pq” -> “rs”  |

### Buffer Pool

**P500**

PageLSN= 103

A = def D = qrs

**P505**

PageLSN= 104

C = wxy

**P600**

PageLSN= 102

B = klm

**P700**

PageLSN= 107

E = rs

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

**NOTE:** Write Ahead Log – Keep track of all changes

1. All LSNs changing that page must be written to disk
  2. In this case it is okay, since the last log record involved P600 while P600 is being flushed
  3. When a page is written, we need to ensure that all log records up to the lastLSN of the last transaction that ever wrote that page are on disk
- (Log is always written to disk in order, i.e. we can't have some log entries in between!)**

## 8. CRASH!!



8. Crash!! ---- These are gone from memory

Dirty page table

|      |
|------|
| P700 |
|------|

Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
|---------|---------|--------|

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "d"   |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "q"   |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "w"   |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
| 107 |         |                   |      |                           |

Buffer Pool

PageLSN= 103

A = def

P505

PageLSN=

C = wxy

N= 107  
E = rs

PageLSN= -

A = abc D = mnp

P505

PageLSN= -

C = tuv

# Three Recovery Phases of

- Analysis

- Reconstructs Dirty page table (for Redo) + Undo Table (for Undo, active transactions at crash)

- Redo

- Restores the database state at the time of crash by repeatedly Updates + CLR

- Undo

- Undoes the actions of uncommitted transactions
- Only updates can be undone, No CLR is even needed

# About Checkpointing

- This example has no checkpointing == Ch at the beginning (like hw 5 !)
- Analysis phase in the recovery starts with Page table and empty Transaction Table
  - With checkpointing the latest copies of these t be read from disk from the last checkpoint

# Analysis Phase

- Reconstruct (conservatively) Dirty Pages and Transaction Table
- Read the log from the last checkpoint
- Actual pages are not used

# Analysis Phase

## Log

### Dirty page table

| pageID | recLSN |
|--------|--------|
|        |        |
|        |        |
|        |        |
|        |        |

### Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
|         |         |        |
|         |         |        |
|         |         |        |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

### Buffer Pool



Will not show the buffer pool from the next slide

#### P500

PageLSN= -

A = abc D = mnp

#### P505

PageLSN= -

C = tuv

# Analysis Phase

## Log

### Dirty page table

| pageID      | recLSN     |
|-------------|------------|
| <b>P500</b> | <b>101</b> |
|             |            |
|             |            |
|             |            |

### Transaction table

| transID      | lastLSN    | status               |
|--------------|------------|----------------------|
| <b>T1000</b> | <b>101</b> | <b>U=</b><br>Unknown |
|              |            |                      |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

**P500**  
PageLSN= -  
A = abc D = mnp

**P600**  
PageLSN=102  
B = klm

**P505**  
PageLSN= -  
C = tuv

**P700**  
PageLSN= -  
E = pq

Disk

In hw5, you can also write "progress" instead of "Unknown"

# Analysis Phase

## Log

### Dirty page table

| pageID      | recLSN     |
|-------------|------------|
| P500        | 101        |
| <b>P600</b> | <b>102</b> |
|             |            |
|             |            |

### Transaction table

| transID      | lastLSN    | status   |
|--------------|------------|----------|
| T1000        | 101        | U        |
| <b>T2000</b> | <b>102</b> | <b>U</b> |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

**P500** PageLSN= -      **P600** PageLSN=102

A = abc   D = mnp      B = klm

**P505** PageLSN= -      **P700** PageLSN= -

C = tuv      E = pq

**Disk**

# Analysis Phase

## Log

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
|        |        |
|        |        |

### Transaction table

| transID      | lastLSN    | status   |
|--------------|------------|----------|
| T1000        | 101        | U        |
| <b>T2000</b> | <b>103</b> | <b>U</b> |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

**P500**  
PageLSN= -

A = abc D = mnp

**P505**  
PageLSN= -

C = tuv

**P600**  
PageLSN=102

B = klm

**P700**  
PageLSN= -

E = pq

Disk

# Analysis Phase

## Log

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

### Transaction table

| transID      | lastLSN    | status   |
|--------------|------------|----------|
| <b>T1000</b> | <b>104</b> | <b>U</b> |
| T2000        | 103        | U        |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

**P500** PageLSN= -      **P600** PageLSN=102

A = abc   D = mnp      B = klm

**P505** PageLSN= -      **P700** PageLSN= -

C = tuv      E = pq

**Disk**

# Analysis Phase

## Log

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

### Transaction table

| transID      | lastLSN    | status   |
|--------------|------------|----------|
| T1000        | 104        | U        |
| <b>T2000</b> | <b>105</b> | <b>C</b> |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

#### P500

PageLSN= -

A = abc D = mnp

#### P505

PageLSN= -

C = tuv

#### P600

PageLSN=102

B = klm

#### P700

PageLSN= -

E = pq

Disk

Write A or Abort if you see  
instead

# Analysis Phase

## Log

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

### Transaction table

| transID          | lastLSN        | status       |
|------------------|----------------|--------------|
| T1000            | 104            | U            |
| <del>T2000</del> | <del>105</del> | <del>C</del> |
|                  |                |              |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

#### P500

PageLSN= -

A = abc D = mnp

#### P505

PageLSN= -

C = tuv

#### P600

PageLSN=102

B = klm

#### P700

PageLSN= -

E = pq

Disk

Remove entry from Transaction table  
you see an End record (both Uncommitted and Committed transaction)

# Analysis Phase

## Log

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

### Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

#### P500

PageLSN= -

A = abc D = mnp

#### P505

PageLSN= -

C = tuv

#### P600

PageLSN=102

B = klm

#### P700

PageLSN= -

E = pq

Disk

Already written to disk,  
but reappears

## Compare with Dirty Table and Transaction Table right

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P505   | 104    |
| P700   | 107    |

### Transaction table

| transID           | lastLSN | status  |
|-------------------|---------|---------|
| T <sub>1000</sub> | 107     | Running |
|                   |         |         |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T <sub>1000</sub> | P500 | Write A<br>"abc" -> "d"   |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
| 107 | 104     | T <sub>1000</sub> | P700 | Write E<br>"pq" -> "rs"   |

Buffer Pool

**P500**

PageLSN= 103

A = def D = qrs

**P505**

PageLSN= 104

C = tuv

**P700**

PageLSN= 107

E = rs

Lost update during crash,  
but write ahead log, so  
safe!

**P505**

PageLSN= -

C = tuv

# REDO Phase

- “Repeating history” (including actions by transactions that aborted in the undo phase)
- Work with the Dirty Page Table
- Find the smallest recLSN in the dirty page table = FirstLSN
- Redo the “**Update/CLR**” action, unless (in this order)
  - Affected page is not in the dirty page table
  - Or, recLSN > LSN being checked (i.e. the page was dirtied later than the update)
  - Or, pageLSN >= LSN being checked (i.e. LSN still not at most recent)
- End/Commit/Abort LSNs are “skipped”
- In HW, write “Redone” or “Skipped” for each LSN

# Analysis Phase

## Log

### Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

### Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

**P500**  
PageLSN= -

A = abc D = mnp

**P505**  
PageLSN= -

C = tuv

**P600**  
PageLSN=102

B = klm

**P700**  
PageLSN= -

E = pq

Disk

# REDO Phase: find firstLSN

Log

## Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

**P500**  
PageLSN= -  
A = abc D = mnp

**P600**  
PageLSN=102  
B = klm

**P505**  
PageLSN= -  
C = tuv

**P700**  
PageLSN= -  
E = pq

Disk

# REDO Phase

## Log

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

## Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

## Buffer Pool

PageLSN= "-" to **101**

**A = def** D = mnp

**P500**

PageLSN= -

A = abc D = mnp

**P505**

PageLSN= -

C = tuv

**P600**

PageLSN=102

B = klm

**P700**

PageLSN= -

E = pq

**Disk**

- Affected page is not in the
- Else, recLSN > LSN being ch
- Else, pageLSN >= LSN bein
- **REDO**

# REDO Phase

## Log

## Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

## Buffer Pool

PageLSN= 101

A = def D = mnp

PageLSN= 102

B = klm

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

## Disk

**P500**

PageLSN= -

A = abc D = mnp

**P600**

PageLSN=102

B = klm

**P505**

PageLSN= -

C = tuv

**P700**

PageLSN= -

E = pq

- Affected page is not in the
- Else, recLSN > LSN being ch
- Else, pageLSN >= LSN bein
- **NO REDO = SKIPPED**

# REDO Phase

## Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

## Buffer Pool

PageLSN= 101 to **103**

A = def **D = qrs**

B = klm

## Log

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

## Disk

**P500**

PageLSN= -

A = abc D = mnp

**P600**

PageLSN=102

B = klm

**P505**

PageLSN= -

C = tuv

**P700**

PageLSN= -

E = pq

- Affected page is not in the
- Else, recLSN > LSN being ch
- Else, pageLSN >= LSN bein
- **REDO**

# REDO Phase

## Log

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

## Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

## Buffer Pool

PageLSN= 103

A = def D = qrs

B = klm

**P505**

PageLSN="" to **104**

**C = wxy**

**P500**

PageLSN= -

A = abc D = mnp

**P600**

PageLSN=102

B = klm

**P505**

PageLSN= -

C = tuv

**P700**

PageLSN= -

E = pq

**Disk**

- Affected page is not in the
- Else, recLSN > LSN being ch
- Else, pageLSN >= LSN bein
- **REDO**

# UNDO Phase

- Work with the Transaction table in the analysis
  - “Loser transactions” must be undone
  - Changes during undo phases are written (C is not repeated at the time of repeated rest)
- Scan backward
- Maintain a set ToUndo
  - Initialize to lastLSNs of all “U” transactions a Transaction Table
  - undo the “largest LSN” in ToUndo at each st latest one in bottom-up order)

# UNDO Phase

## Log

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
|     |         |                   |      |                           |

ToUNDO = {104}

Disk

## Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

## Buffer Pool

PageLSN= 103

A = def D = qrs

B = klm

**P505**

PageLSN= 104

C =wxy

PageLSN= -

A = abc D = mnp

B = klm

**P505**

PageLSN= -

**P700**

PageLSN= -

C = tuv

E = pq

# CLR

- CLR is added so that no “Undo” action is undone
- If a CLR is encountered during UNDO, the next LSN goes to the LSN in UndoNextLSN

## Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

## Buffer Pool

PageLSN= 103

A = def D = qrs

B = klm

**P505**

**PageLSN= 107**

**C =tuv**

PageLSN= -

A = abc D = mnp

B = klm

**P505**

PageLSN= -

C = tuv

**P700**

PageLSN= -

E = pq

# UNDO Phase

## Log

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
| 107 |         | T <sub>1000</sub> |      | UndoT <sub>1000</sub> LSN |

- A CLR is written
- PageLSN = LSN (CLR)
- Value of C is undone

**Disk**

ToUNDO = {10

# UNDO Phase

## Log

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
| 107 |         | T <sub>1000</sub> |      | UndoT <sub>1000</sub> L   |
| 108 |         | T <sub>1000</sub> |      | UndoT <sub>1000</sub> L   |

## Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

## Buffer Pool

PageLSN= 108

A = abc D = qrs

B = klm

P505

PageLSN= 107

C = tuv

PageLSN= -

A = abc D = mnp

B = klm

P505

PageLSN= -

P700

PageLSN= -

C = tuv

E = pq

Disk

ToUNDO = {}

# UNDO Phase

## Log

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
| 107 |         | T <sub>1000</sub> |      | UndoT <sub>1000</sub> L   |

## Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

## Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

## Buffer Pool

PageLSN= 103

A = def D = qrs

B = klm

**P505**

PageLSN= 107

C =tuv

PageLSN= -

A = abc D = mnp

B = klm

**P505**

PageLSN= -

C = tuv

**P700**

PageLSN= -

E = pq

Disk

ToUNDO = {10

Dirty page table

| pageID | recLSN |
|--------|--------|
| P500   | 101    |
| P600   | 102    |
| P505   | 104    |
|        |        |

Transaction table

| transID | lastLSN | status |
|---------|---------|--------|
| T1000   | 104     | U      |

Buffer Pool

PageLSN= 108

A = abc D = qrs

B = klm

P505

PageLSN= 107

C =tuv

PageLSN= -

A = abc D = mnp

B = klm

P505

PageLSN= -

P700

PageLSN= -

C = tuv

E = pq

UNDO Phase

Log

| LSN | prevLSN | tID               | pID  | Log entry                 |
|-----|---------|-------------------|------|---------------------------|
| 101 | -       | T1000             | P500 | Write A<br>"abc" -> "de"  |
| 102 | -       | T <sub>2000</sub> | P600 | Write B<br>"hij" -> "klm" |
| 103 | 102     | T <sub>2000</sub> | P500 | Write D<br>"mnp" -> "qrs" |
| 104 | 101     | T <sub>1000</sub> | P505 | Write C<br>"tuv" -> "wxy" |
| 105 | 103     | T <sub>2000</sub> |      |                           |
| 106 | 105     | T <sub>2000</sub> |      |                           |
| 107 |         | T <sub>1000</sub> |      | UndoT <sub>1000</sub> LSN |
| 108 |         | T <sub>1000</sub> |      | UndoT <sub>1000</sub> LSN |
| 109 |         | T <sub>1000</sub> |      |                           |

Disk

Write an END record  
(explicitly mentioned)  
the aborted transact

# What happens if T abort

1. Write an “abort” log record for T
  - Like “commit”
  - Also the status in Transaction table from “Running”
- 2 Follow “prevLSN” to undo all updates by T
  - Like the “UNDO” phase
  - Undo page content in buffer pool, pageLSN change LSN(CLR)
  - Write the CLR records
    - Log entry like “Undo T3 LSN5”
    - No prevLSN, but undoNextLSN field present
    - Until undoNextLSN is null
3. Write an “End” log record for T

# Handling Crashes during

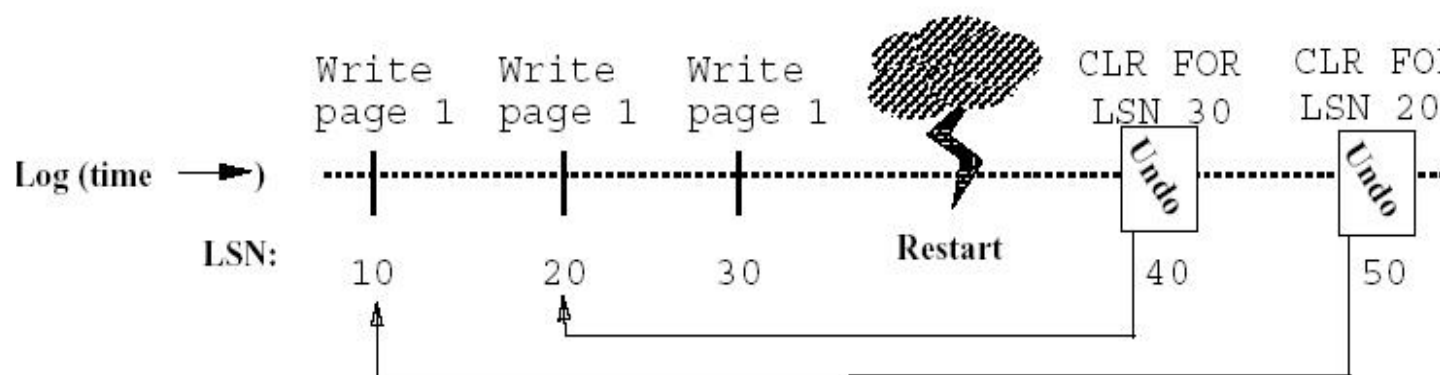


Figure 4: The Use of CLR's for UNDO

[Figure 4 from Franklin97]

In general, for every single crash (even for crash during Analysis phases), start again with Analysis

If some CLR records are written to disk during an UNDO phase happens (e.g. here LSN 40, 50 are written to disk before the crash), then the next UNDO phase will skip undoing those CLR's.