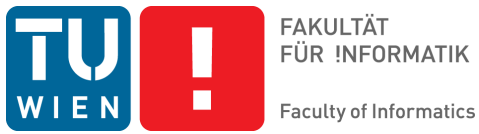
# Data Base Systems

## VU 184.686, WS 2021

## Recovery

Institute of Logic and Computation, TU Wien



# Acknowledgements

The slides are based on the slides (in German) of Sebastian Skritek.

The content is based on Chapter 10 of
(Kemper, Eickler: Datenbanksysteme – Eine Einführung).
Many examples and illustrations are taken from there.

For related literature in English see Chapter 18 of
(Ramakrishnan, Gehrke: Database Management Systems).

# Recovery (Error Handling and Logging)

1. Memory Management
2. Error Categories
3. Taken System Configuration
4. Logging
5. Recovery after an Error
6. Checkpoints

# Overview

# Overview

# Storage Hierarchy

available storage media typically form a hierarchy

- trade-off:
  - fast and expensive vs. slow and cheap
  - fast and volatile vs. slow and persistent
  - fast and small vs. slow and big
- "access gap" between the stages

# Storage Hierarchy

available storage media typically form a hierarchy

- trade-off:
  - fast and expensive vs. slow and cheap
  - fast and volatile vs. slow and persistent
  - fast and small vs. slow and big
- "access gap" between the stages

(simplified) assumption in the course: Two-Stage Storage Hierarchy

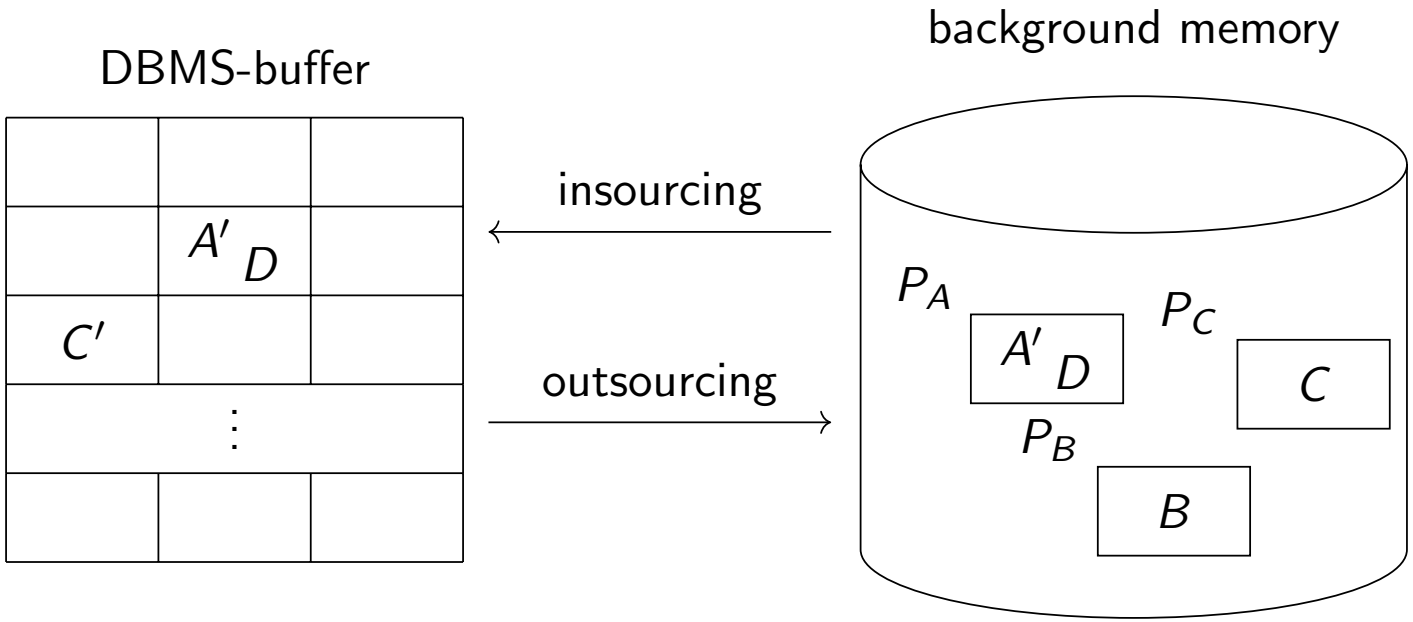
- database buffer (fast access, volatile)
- background memory (slow access, persistent)

# Memory Organization: Pages

- memory split in pages
- dataset are represented by pages
  - simplified assumption in course: date $A$ on page $P_A$
- granularity level in which the data is moved between buffer and background memory

# Memory Organization: Pages

- memory split in pages
- dataset are represented by pages
  - simplified assumption in course: date $A$ on page $P_A$
- granularity level in which the data is moved between buffer and background memory



# Overview

# Replacement of Buffer Pages

- no outsourcing during access (FIX)
- after that based on chosen strategy
- *dirty* pages: modified pages in buffer

# Replacement of Buffer Pages

- no outsourcing during access (FIX)
- after that based on chosen strategy
- *dirty* pages: modified pages in buffer

**strategy** for the replacement of buffer pages:

*steal* every non-fixed page is in principle a candidate for outsourcing

¬*steal* pages that are modified by a still active transaction may not be outsourced

# Outsourcing of Buffer Pages

**strategy** for the outsourcing of modified buffer pages after a successful transaction:

*force* modified pages are outsourced at the end of the transaction

¬force outsourcing at the end of a transaction is not forced, *dirty pages* may stay in buffer

# Overview

# Introducing Strategy

**update-in-place** ("direct" introducing strategy):

- every page has exactly one "place" in the background memory
- a page is copied at this place in the outsourcing process
- the old state of the page is overwritten

# Introducing Strategy

**update-in-place** ("direct" introducing strategy):

- every page has exactly one "place" in the background memory
- a page is copied at this place in the outsourcing process
- the old state of the page is overwritten

**twin-block-method** ("indirect" introducing strategy):

- for every page two copies in the background memory + bit indicating current copy
- page is copied at current copy during outsourcing

# Introducing Strategy

**update-in-place** ("direct" introducing strategy):

- every page has exactly one "place" in the background memory
- a page is copied at this place in the outsourcing process
- the old state of the page is overwritten

**twin-block-method** ("indirect" introducing strategy):

- for every page two copies in the background memory + bit indicating current copy
- page is copied at current copy during outsourcing

**shadow memory concept** ("indirect" introducing strategy):

- there are two copies only for modified pages

# Overview

# Three Error Categories

**local error** (in a transaction)

- only one transaction is affected
- modifications caused by this transaction have to be (quickly!) reset (local undo)

# Three Error Categories

**local error** (in a transaction)

- only one transaction is affected
- modifications caused by this transaction have to be (quickly!) reset (local undo)

**error with loss of DBMS-buffer**

- modifications caused by finished transactions have to be retained (global redo)
- modifications caused by unfinished transactions have to be reset global undo).

# Three Error Categories

**local error** (in a transaction)

- only one transaction is affected
- modifications caused by this transaction have to be (quickly!) reset (local undo)

**error with loss of DBMS-buffer**

- modifications caused by finished transactions have to be retained (global redo)
- modifications caused by unfinished transactions have to be reset global undo).

**error with background memory loss**

- data recovery via backup

# Three Error Categories

**local error** (in a transaction)

- only one transaction is affected
- modifications caused by this transaction have to be (quickly!) reset (local undo)

**error with loss of DBMS-buffer**

- modifications caused by finished transactions have to be retained (global redo)
- modifications caused by unfinished transactions have to be reset global undo).

**error with background memory loss**

- data recovery via backup

# Impact of Strategy for Outsourcing to Recovery

# Impact of Strategy for Outsourcing to Recovery

| | force | ¬force |
|---|---|---|
| ¬steal | | |
| steal | | |

# Impact of Strategy for Outsourcing to Recovery

| | force | ¬force |
|---|---|---|
| ¬steal | ■ no redo<br>■ no undo | |
| steal | | |

# Impact of Strategy for Outsourcing to Recovery

| | force | ¬force |
|---|---|---|
| ¬steal | ■ no redo<br>■ no undo | ■ redo<br>■ no undo |
| steal | | |

# Impact of Strategy for Outsourcing to Recovery

| | force | ¬force |
|---|---|---|
| ¬steal | ■ no redo<br>■ no undo | ■ redo<br>■ no undo |
| steal | ■ no redo<br>■ undo | |

# Impact of Strategy for Outsourcing to Recovery

| | force | ¬force |
|---|---|---|
| ¬steal | ■ no redo<br>■ no undo | ■ redo<br>■ no undo |
| steal | ■ no redo<br>■ undo | ■ redo<br>■ undo |

# Some Disadvantages of ¬steal + force

- forced outsourcing at transaction end is expensive
  - pages used by many transactions (*"hot spots"*) are outsourced although they remain in the buffer
  - outsourcing of pages has to be atomic ("everything or nothing")
- for ¬steal + force transactions have to block whole page

# Overview

# Taken System Configuration

*steal* pages modified by an unfinished transaction can be outsourced as well

¬*force* after transaction end modified pages do not have to be outsourced

*update-in-place* every page has exactly one corresponding place (a copy) in the background memory

small barrier granulates on dataset level (see next chapter "concurrency control"): different transactions access page simultaneously; page may contain modifications from both, finished and unfinished transactions

# Overview

# Overview

# Protocol of Modifications

**problem:**

- global undo: background memory may receive modifications of unfinished transactions (because of steal)
- global redo: modifications due to finished transactions may not be stored in background memory yet (because of ¬force)

# Protocol of Modifications

**problem:**

- global undo: background memory may receive modifications of unfinished transactions (because of steal)
- global redo: modifications due to finished transactions may not be stored in background memory yet (because of ¬force)

**solution:** storing additional information (log-file)

- information required from undo/redo
- information about start and end of a transaction

# Structure of Log-Entries

## Definition (structure of a log-entry)

```
[LSN, transactionID, pageID, redo, undo, prevLSN]
```

# Structure of Log-Entries

## Definition (structure of a log-entry)

`[LSN, transactionID, pageID, redo, undo, prevLSN]`

LSN (Log Sequence Number):

- unique identifier of log-entry
- are assigned monotonically increasing
- chronological order can be investigated

# Structure of Log-Entries

## Definition (structure of a log-entry)

`[LSN, transactionID, pageID, redo, undo, prevLSN]`

### LSN (Log Sequence Number):

- unique identifier of log-entry
- are assigned monotonically increasing
- chronological order can be investigated

### transactionID:

- identification of transaction that has lead to modification

# Structure of Log-Entries

## Definition (structure of a log-entry)

`[LSN, transactionID, pageID, redo, undo, prevLSN]`

### LSN (Log Sequence Number):

- unique identifier of log-entry
- are assigned monotonically increasing
- chronological order can be investigated

### transactionID:

- identification of transaction that has lead to modification

### pageID:

- identification of the corresponding page
- one log-entry is created per page affected

# Structure of Log-Entries

## Definition (structure of a log-entry)

`[LSN, transactionID, pageID, redo, undo, prevLSN]`

redo:

- information to track modification

# Structure of Log-Entries

## Definition (structure of a log-entry)

`[LSN, transactionID, pageID, redo, undo, prevLSN]`

redo:

- information to track modification

undo:

- information to undo modification

# Structure of Log-Entries

## Definition (structure of a log-entry)

`[LSN, transactionID, pageID, redo, undo, prevLSN]`

redo:

- information to track modification

undo:

- information to undo modification

prevLSN:

- pointer to previous log-entry of the transaction
- necessary for efficiency reasons (local undo!)

# Example of Log-Entries

## Example

| step | $T_1$ | $T_2$ | Log |
|---|---|---|---|
| 1. | **BOT** | | $[\#1, T_1, \texttt{BOT}, 0]$ |
| 2. | $r(A, a_1)$ | | |
| 3. | | **BOT** | $[\#2, T_2, \texttt{BOT}, 0]$ |
| 4. | | $r(C, c_2)$ | |
| 6. | $w(A, a_1 - 50)$ | | $[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$ |
| 8. | | $w(C, c_2 + 100)$ | $[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$ |
| 9. | $r(B, b_1)$ | | |
| 11. | $w(B, b_1 + 50)$ | | $[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$ |
| 12. | **commit** | | $[\#6, T_1, \texttt{commit}, \#5]$ |
| 13. | | $r(A, a_2)$ | |
| 15. | | $w(A, a_2 - 100)$ | $[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$ |
| 16. | | **commit** | $[\#8, T_2, \texttt{commit}, \#7]$ |

# Physical or Logical Logging

physical logging: redo- and undo-entries get state:

- `undo` receives before-image (= state before modification)
- `redo` receives after-image (= state after modification)

# Physical or Logical Logging

physical logging: redo- and undo-entries get state:

- `undo` receives before-image (= state before modification)
- `redo` receives after-image (= state after modification)

logical logging: redo- and undo-entries get operations:

- `undo` contains information how before-image is constructed from after-image
- `redo` contains information how after-image is constructed from before-image

# Physical versus. Logical Logging

## Example

| | $T_1$ | $T_2$ |
|---|---|---|
| 1 | BOT | |
| 2 | | BOT |
| 3 | read($A$, $a_1$) | |
| 4 | write($A$, $a_1 + 5$) | |
| 5 | | read($A$, $a_2$) |
| 6 | | write($A$, $a_2 + 10$) |
| 7 | abort | |

resetting only possible with logical logging

remark: violates isolation (see Concurrency Control)

# Role of LSN (Log Sequence Number)

**problem:** at recovery the DBMS has to check whether the consequences of a log-entry are already contained in a page

# Role of LSN (Log Sequence Number)

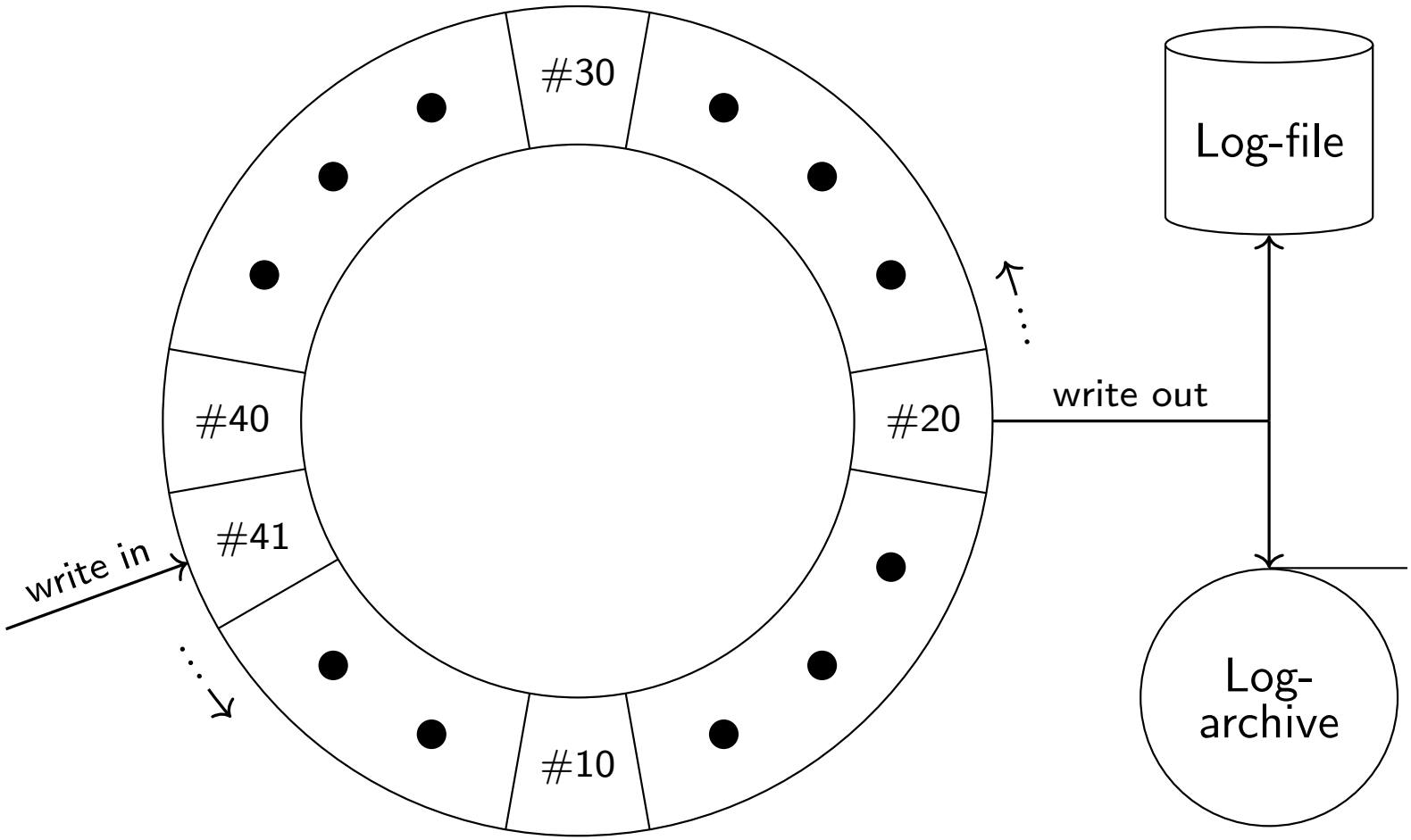
**problem:** at recovery the DBMS has to check whether the consequences of a log-entry are already contained in a page

**solution:** each page has a field for a LSN:

- construction of a log-entry: copy entry LSN to the page
- page contains LSN of the most recent log-entry corresponding to the page

# Overview

# Writing of Log-Entries



# Writing of Log-Entries

- log-buffer for log-entries
  - separated from buffer for pages
- common: ring buffer
  - continuous writing avoids peeks
- log-information is written twice:
  1. in a temporal log (background memory) for quick access
  2. in a log-repository for recovery after error with background memory loss

# The WAL-Principle (Write Ahead Log)

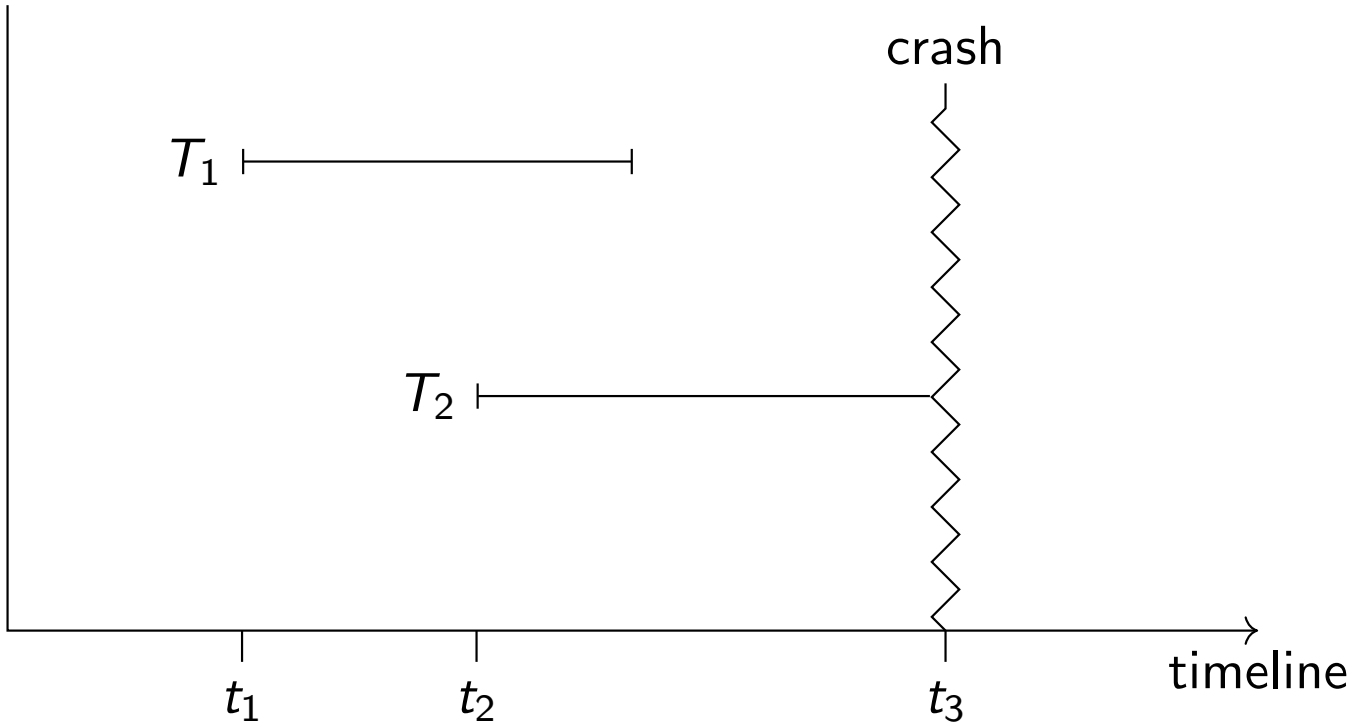
the taken configuration *steal*, ¬*force*, *update-in-place* requires the **WAL-principle** (*write ahead log*):

- before writing a transaction (successful `commit`) all log-entries of the transaction have to be written out (for redo)
- before outsourcing a modified page all corresponding log-entries have to be written out (for undo)
- the chronological order has to be preserved during writing out

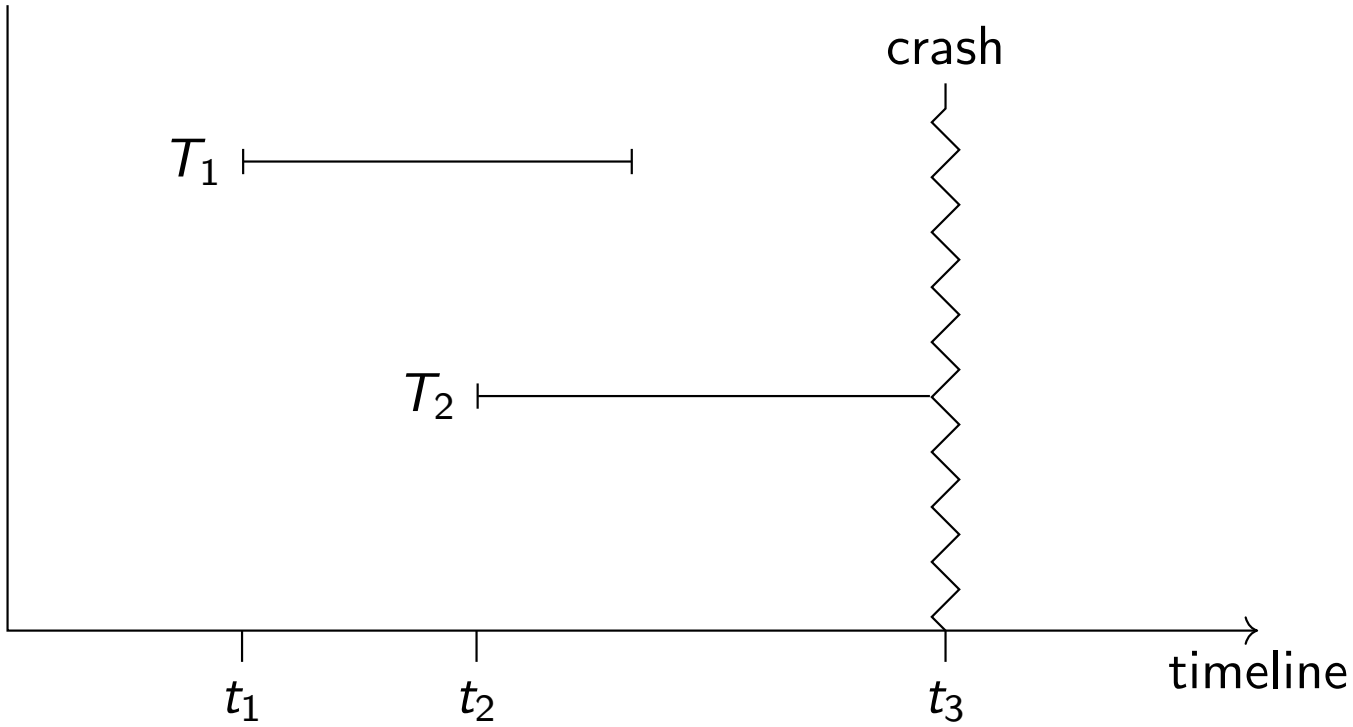
# Overview

# Overview

# We Distinguish Two Kinds of Transactions

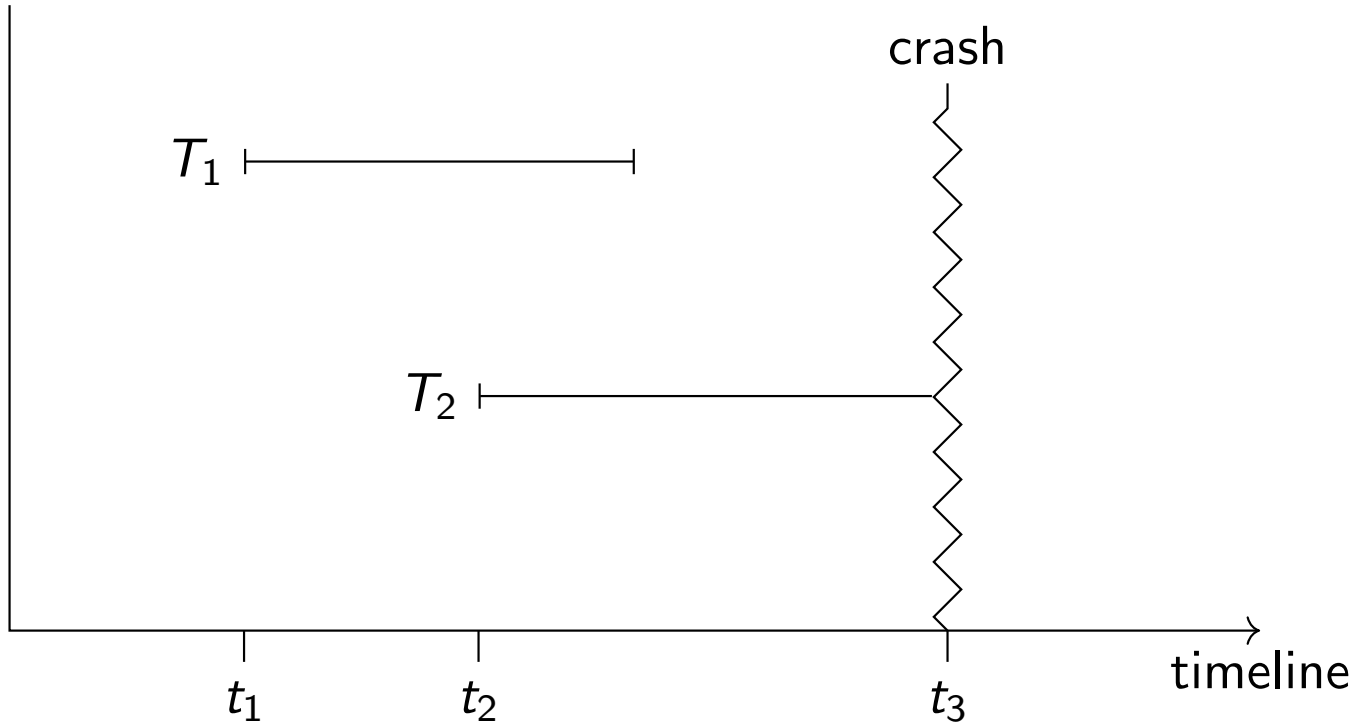


# We Distinguish Two Kinds of Transactions



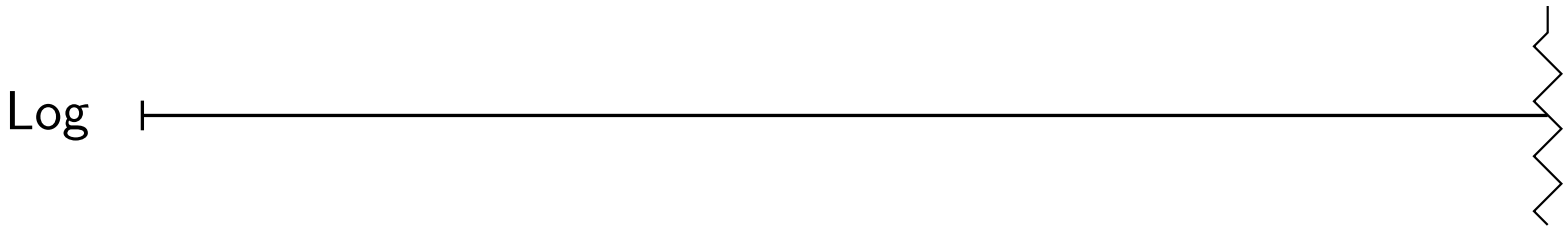

- transactions of type $T_1$ (winner):
  effect has to be completely reconstructed

# We Distinguish Two Kinds of Transactions

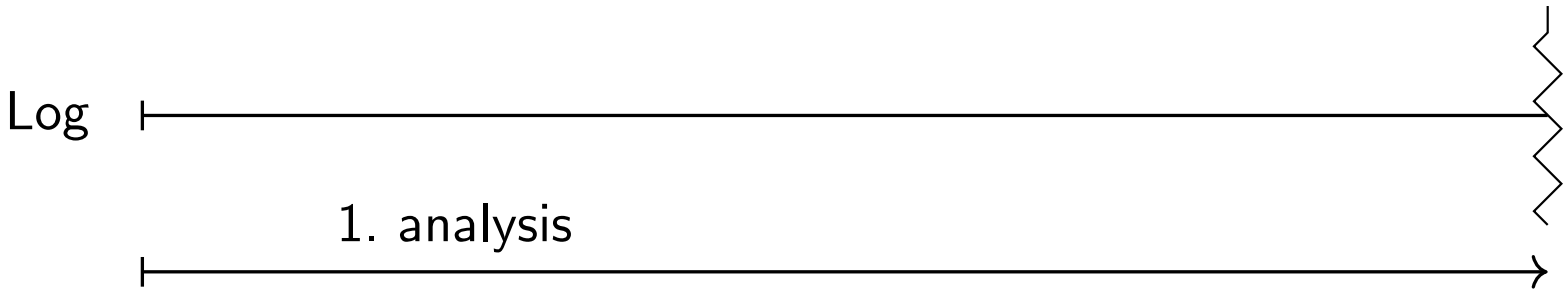



- transactions of type $T_1$ (winner):
  effect has to be completely reconstructed
- transactions of type $T_2$ (loser):
  effect has to be completely revoked

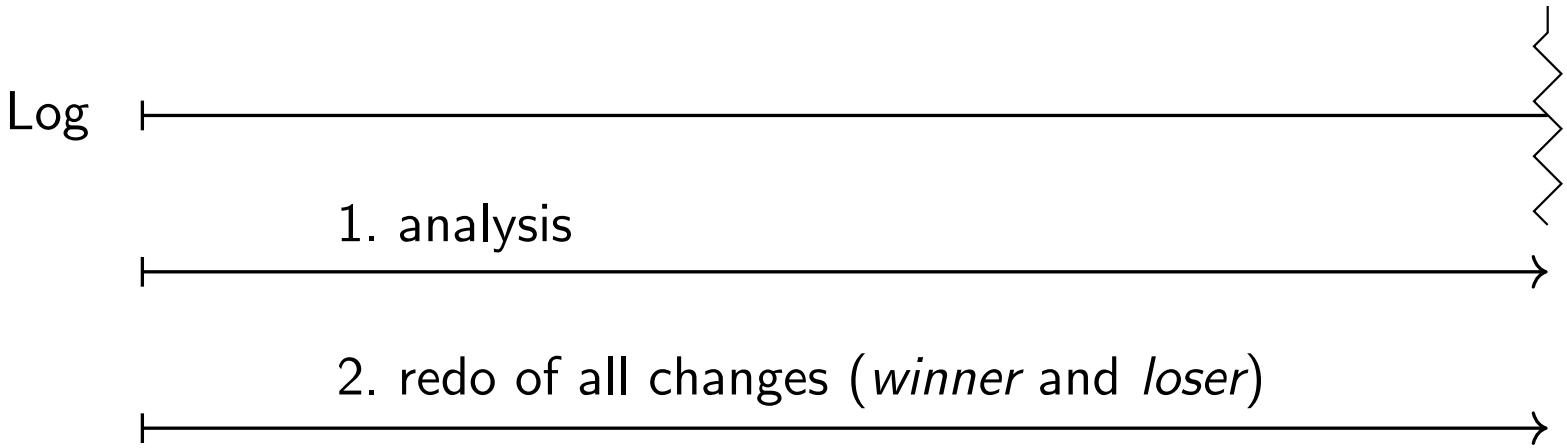
# Three Stages of Recovery



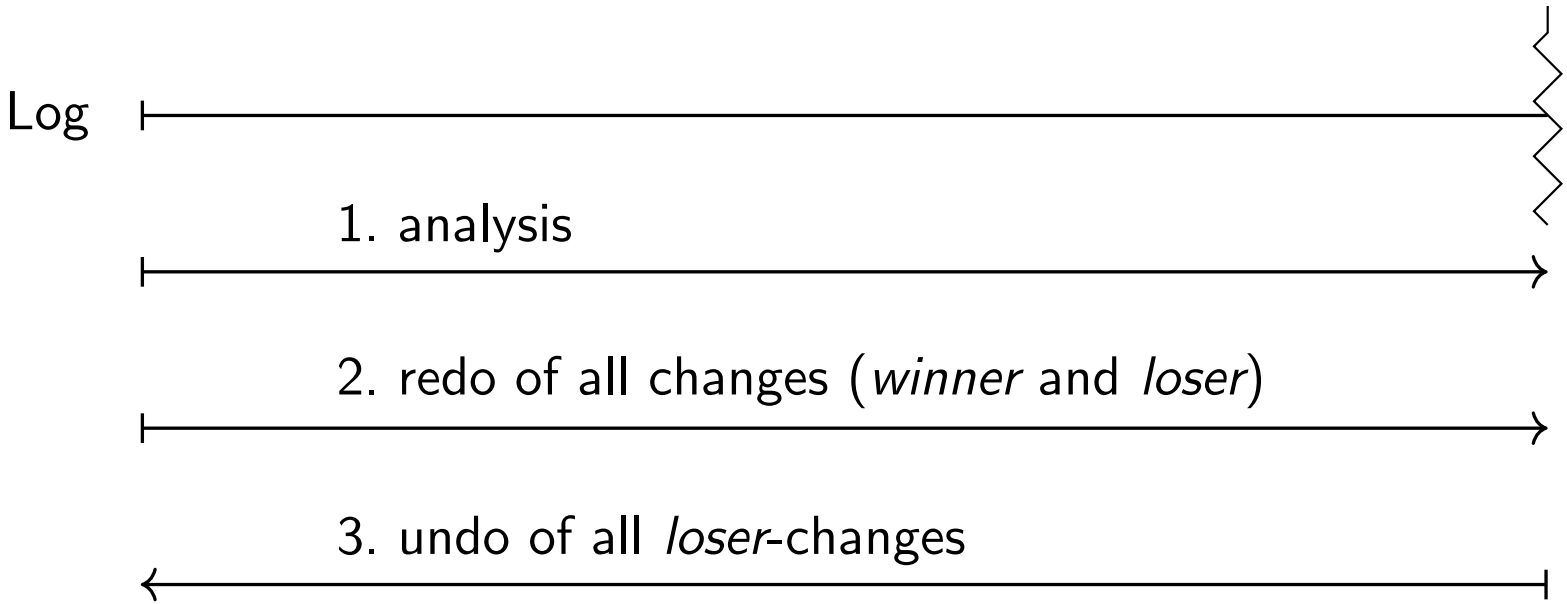
# Three Stages of Recovery



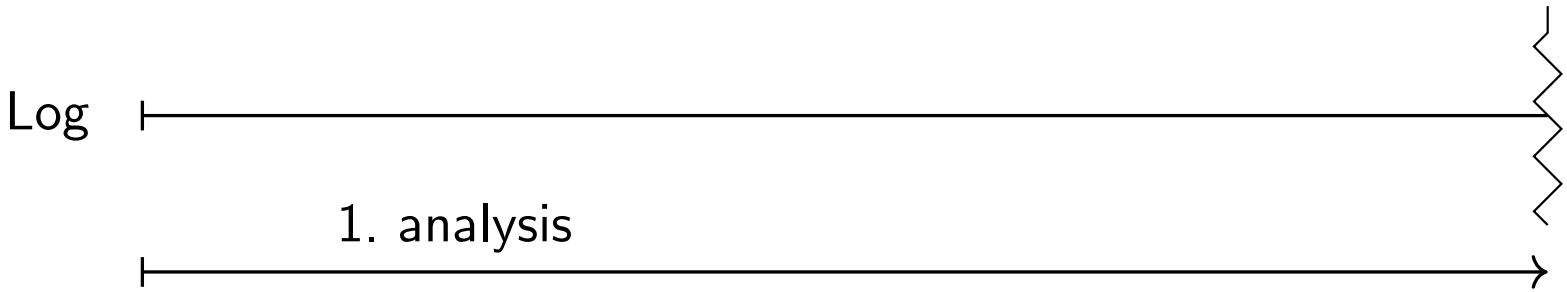
# Three Stages of Recovery



# Three Stages of Recovery

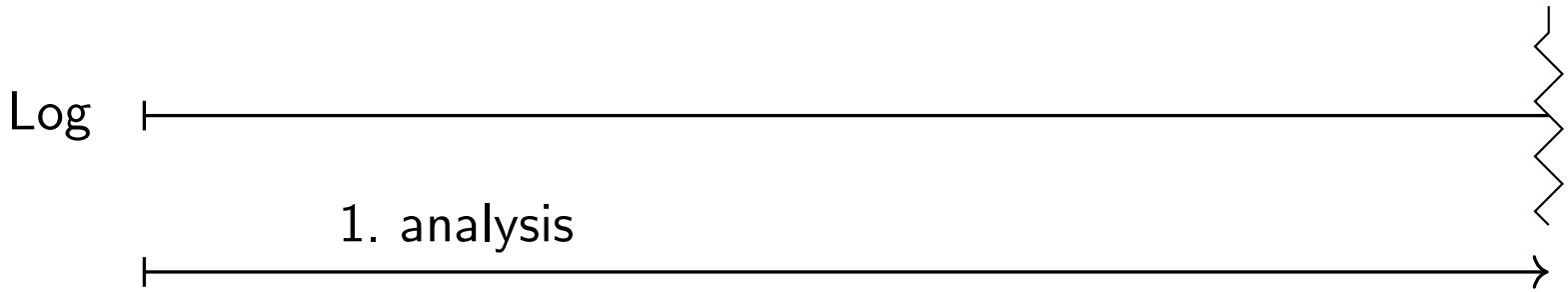


# Three Stages of Recovery: 1. Analysis



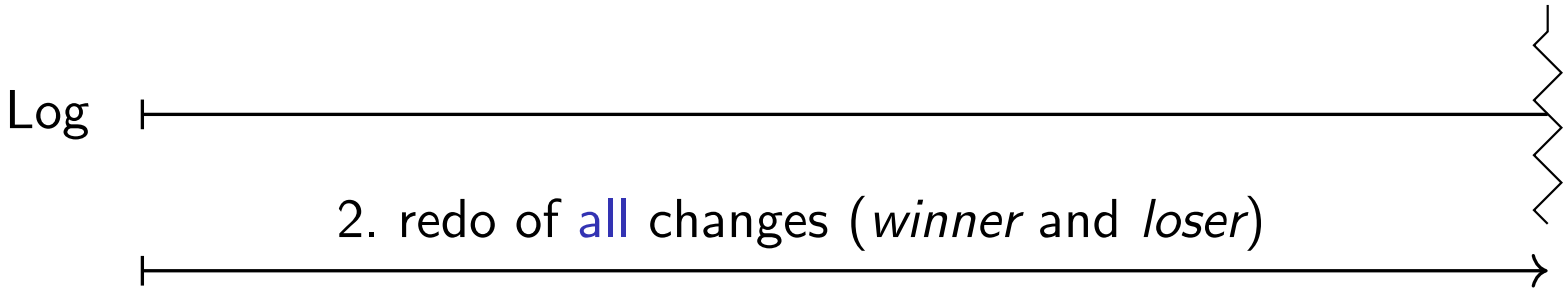

- analysis of the whole log-file

# Three Stages of Recovery: 1. Analysis



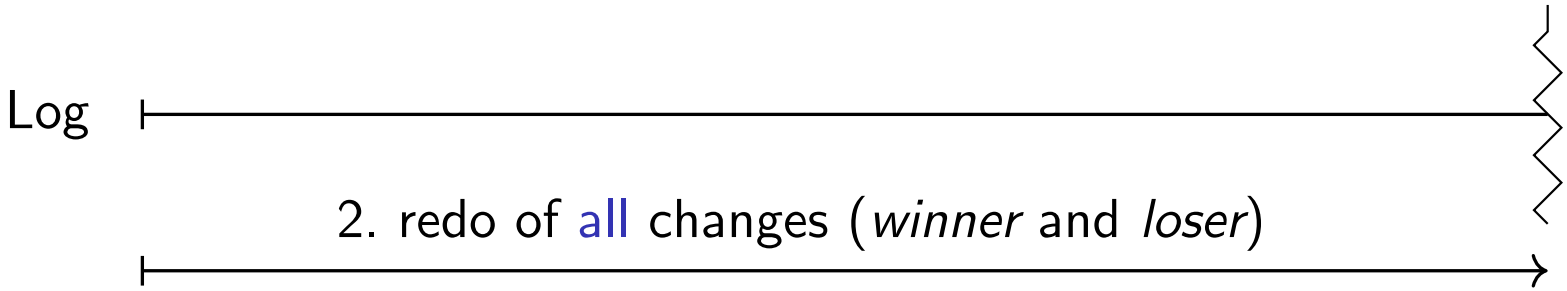

- analysis of the whole log-file
- identify loser-set
  ( = transactions without completion)
- find the highest LSN for each of those transactions

# Three Stages of Recovery: 2. Redo




database is set to state of crash

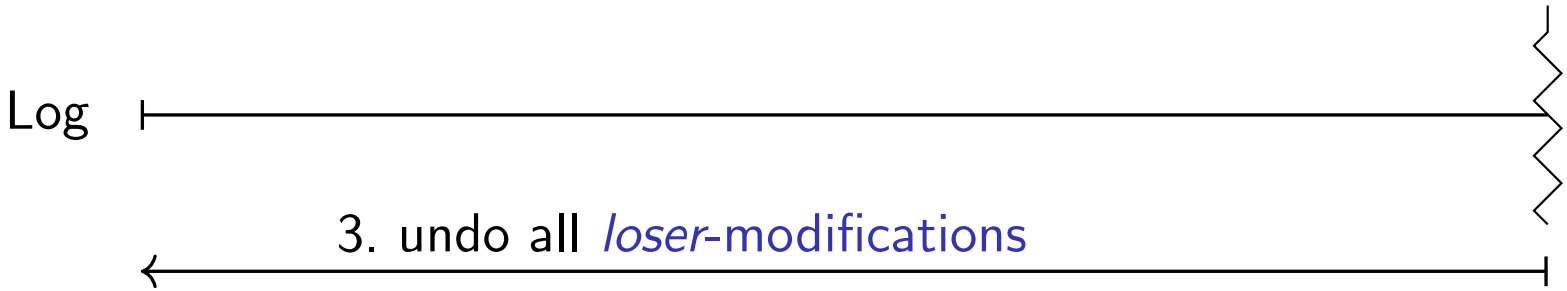
# Three Stages of Recovery: 2. Redo




database is set to state of crash

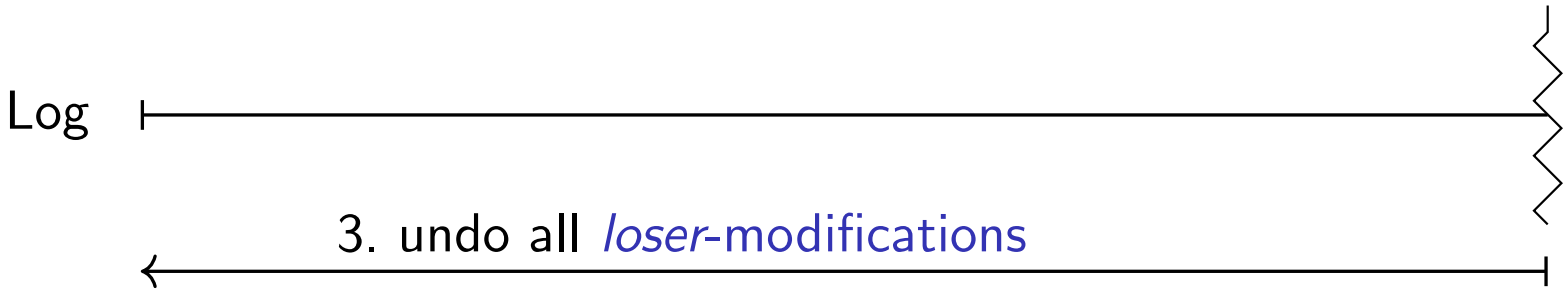
- all logged modifications are reconstructed
- in chronological order (i.e. ordered by LSN)
- page-LSN determines state of background memory:
  - LSN of log-entry $>$ page-LSN: execute redo-operation; update of page-LSN
  - otherwise: background memory contains *after-image* of operation

# Three Stages of Recovery: 3. Undo




removes effects of unfinished transactions

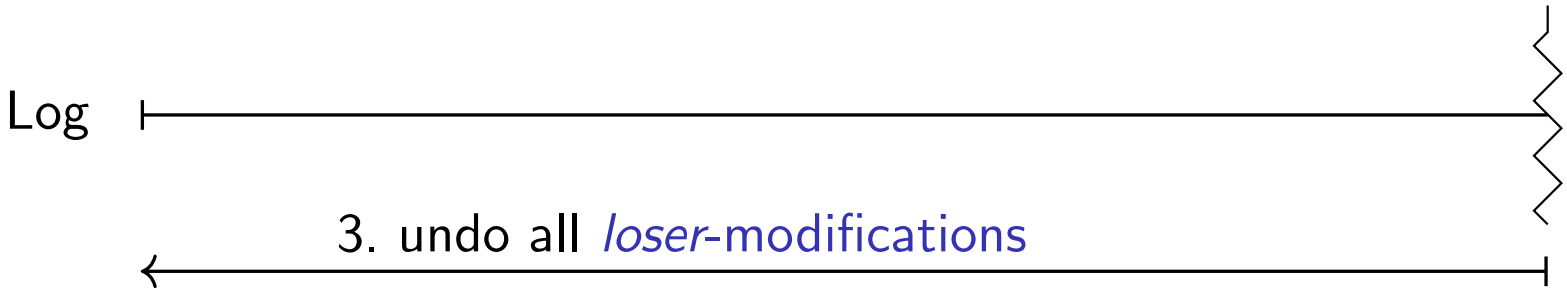
# Three Stages of Recovery: 3. Undo




removes effects of unfinished transactions

- for every log-entry of a loser-transaction: use undo-information for recovery of *before-image*
- in reverse chronological order (by LSN)

# Three Stages of Recovery: 3. Undo




removes effects of unfinished transactions

- for every log-entry of a loser-transaction: use undo-information for recovery of *before-image*
- in reverse chronological order (by LSN)
- construct special log-entry for every executed undo

# Fault Tolerance (Idempotence) of Recovery

**problem:** during recovery an error with loss of buffer may occur

# Fault Tolerance (Idempotence) of Recovery

**problem:** during recovery an error with loss of buffer may occur

**requirements:**

$$undo(undo(\cdots(undo(a))\cdots)) = undo(a)$$
$$redo(redo(\cdots(redo(a))\cdots)) = redo(a)$$

# Fault Tolerance (Idempotence) of Recovery

**problem:** during recovery an error with loss of buffer may occur

**requirements:**

$$undo(undo(\cdots(undo(a))\cdots)) = undo(a)$$
$$redo(redo(\cdots(redo(a))\cdots)) = redo(a)$$

**realization:**

- redo-stage: via LSN (in log-entry resp. page-LSN)

# Fault Tolerance (Idempotence) of Recovery

**problem:** during recovery an error with loss of buffer may occur

**requirements:**

$$undo(undo(\cdots(undo(a))\cdots)) = undo(a)$$
$$redo(redo(\cdots(redo(a))\cdots)) = redo(a)$$

**realization:**

- redo-stage: via LSN (in log-entry resp. page-LSN)
- undo-stage: via CLRs (compensation log record)

# Compensation Log Record (CLR)

## Definition (structure of compensation log entry)

⟨`LSN`, `transaktionsID`, `pageID`, `redo`, `prevLSN`, `UndoNxtLSN`⟩

no undo-information:

- revoking of undo-steps not necessary:
  either successfully outsourced or *redo*

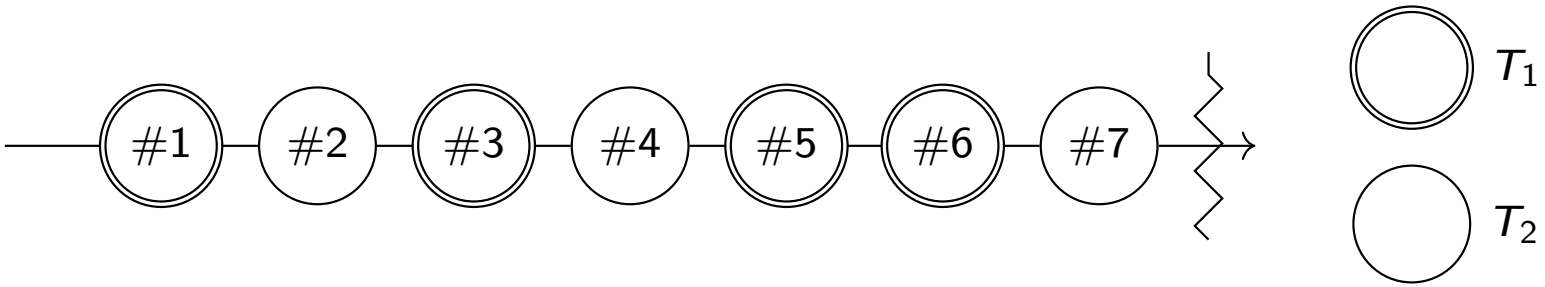
prevLSN (similar to log-entries):

- Pointer to previous log- or *CLR-entry* of the transaction

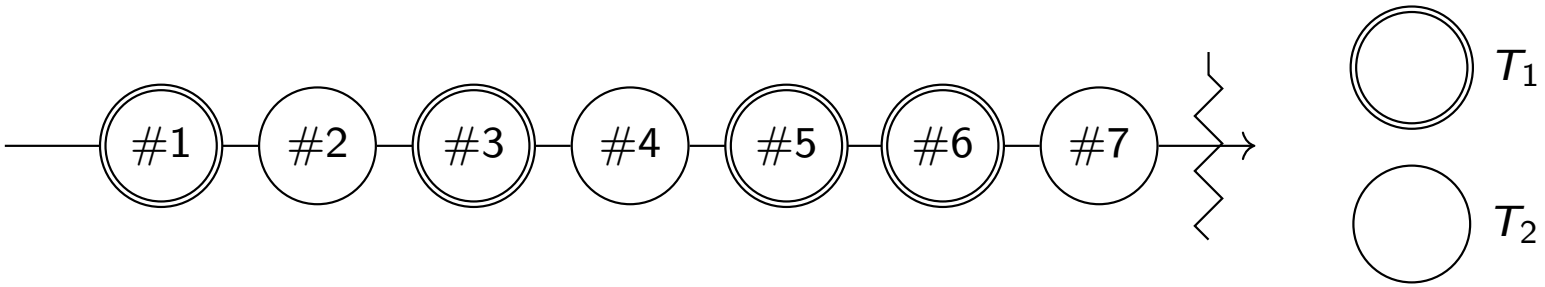
UndoNxtLSN:

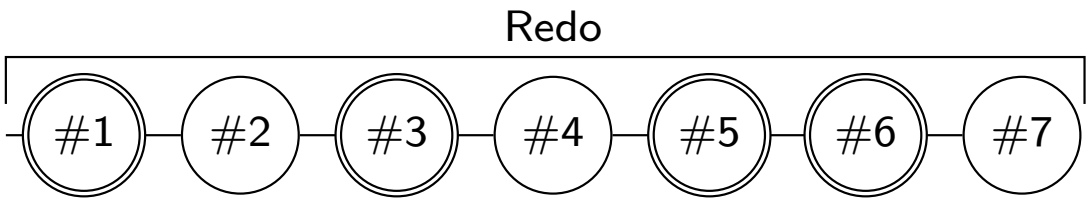
- pointer to modification of transaction that has to be revoked next
- `prevLSN` of revoked log-entry

# Example: Compensationentries in Log

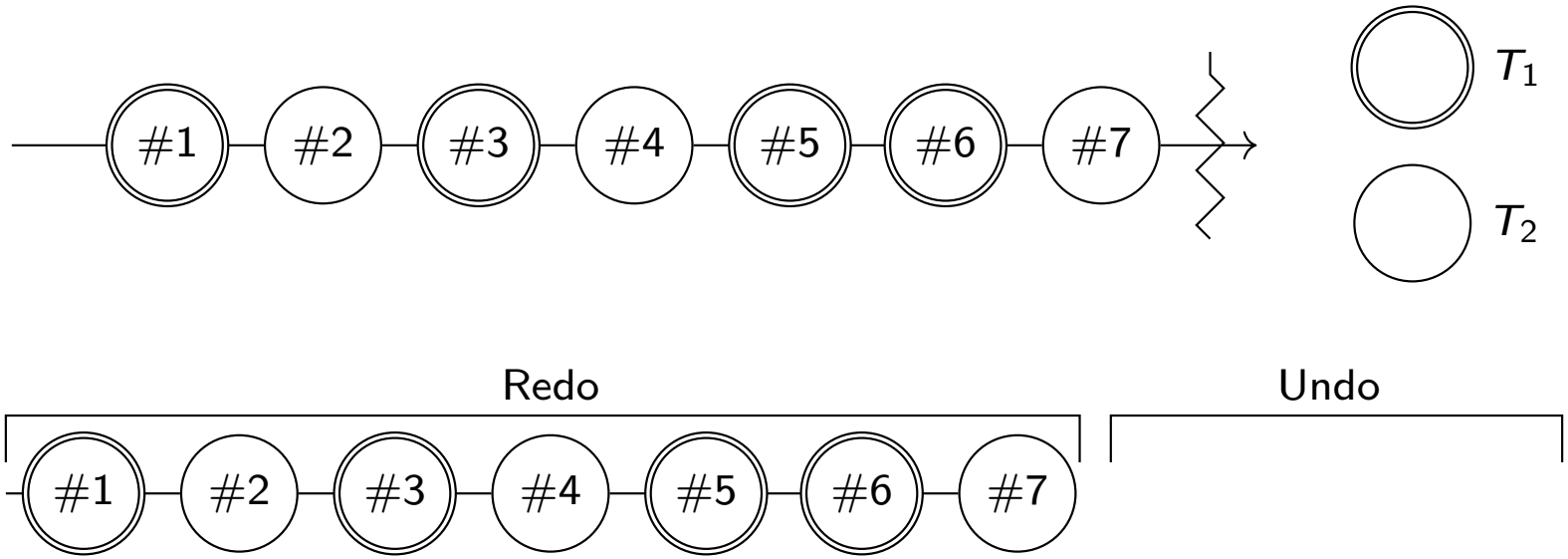


# Example: Compensationentries in Log



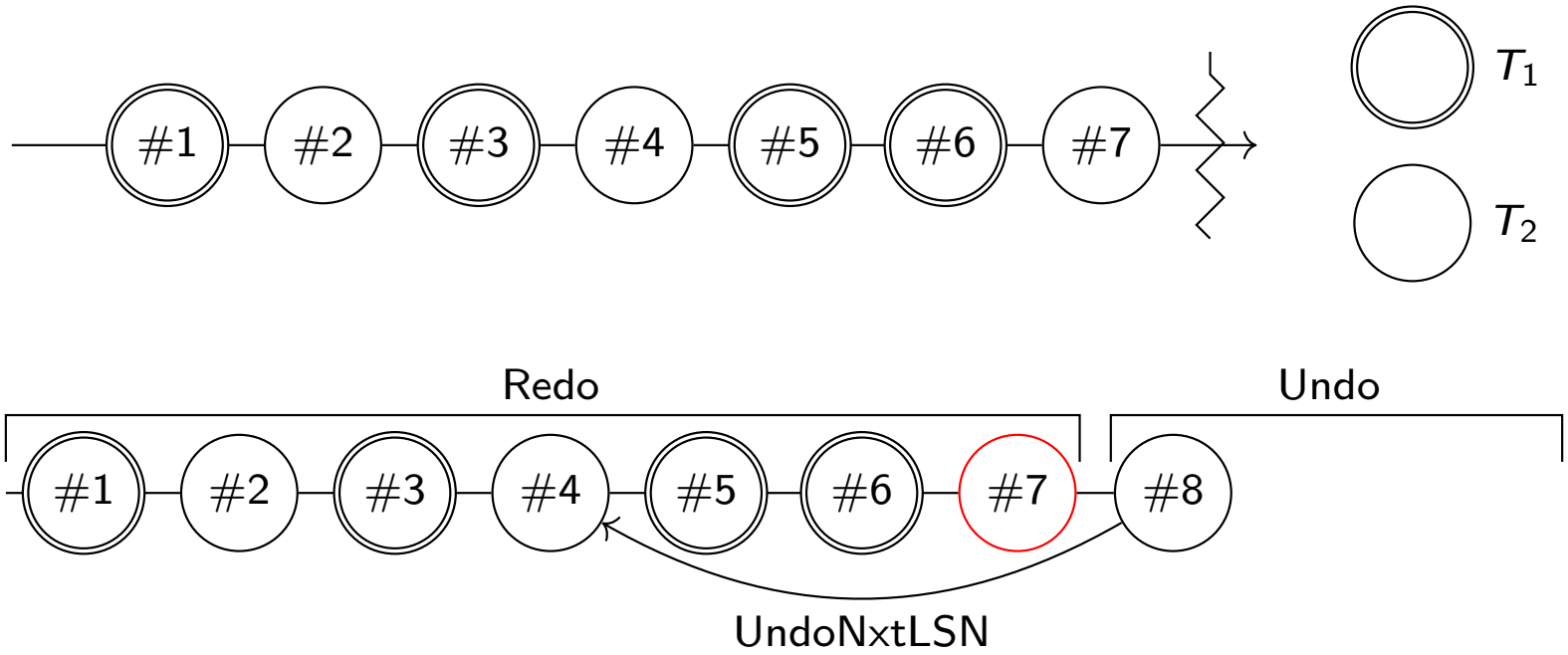



# Example: Compensationentries in Log




CLRs for revoked changes and for BOT:

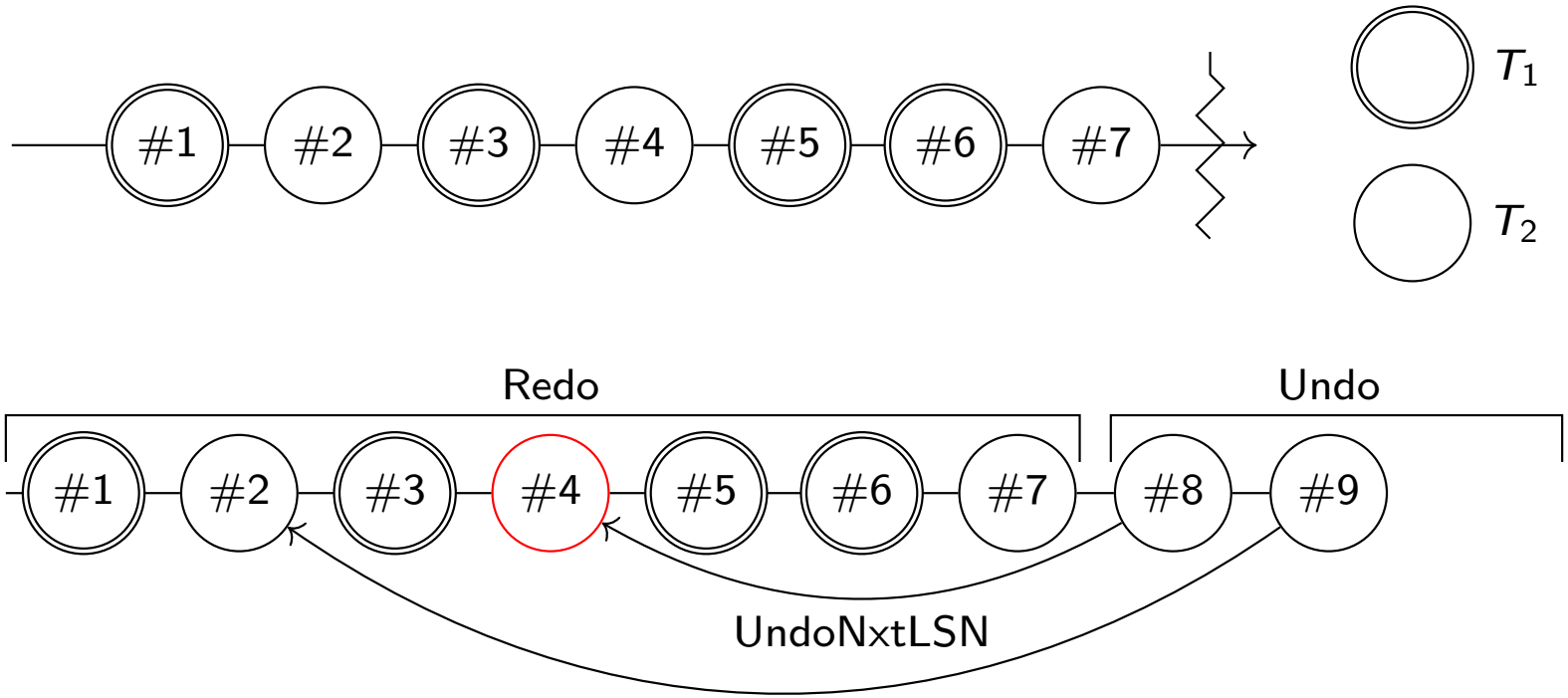
# Example: Compensationentries in Log




CLRs for revoked changes and for BOT:

- #8 is CLR for #7

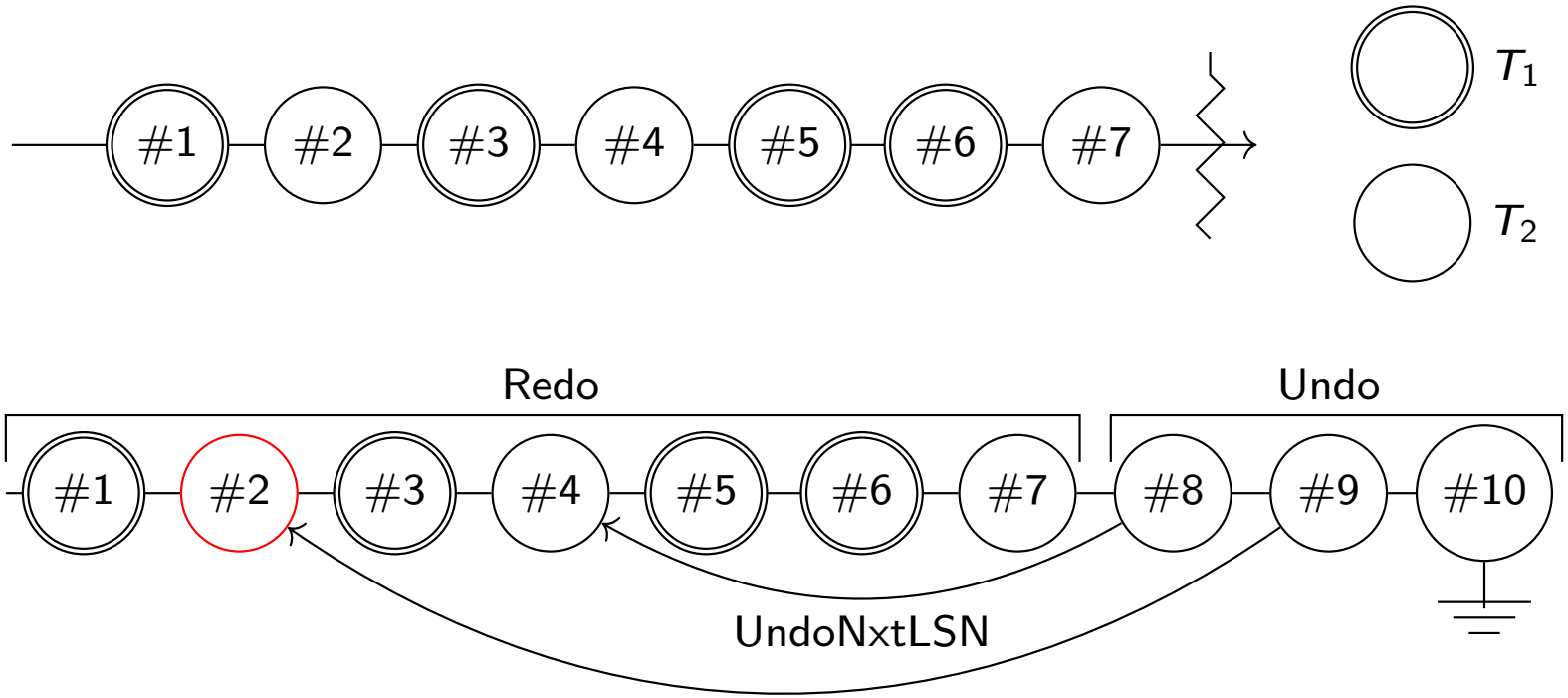
# Example: Compensationentries in Log




CLRs for revoked changes and for BOT:

- #8 is CLR for #7
- #9 is CLR for #4

# Example: Compensationentries in Log




CLRs for revoked changes and for BOT:

- #8 is CLR for #7
- #9 is CLR for #4
- #10 is CLR for #2 (= BOT)

# Example Log-Entries and Recovery

## Example

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$
$$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$
$$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$$
$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$

# Example Log-Entries and Recovery

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$
$$[\#2, T_2, \texttt{BOT}, 0]$$
$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$
$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$
$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$
$$[\#6, T_1, \texttt{commit}, \#5]$$
$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$
$$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$$
$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$
$$\langle \#10, T_2, \text{-}, \text{-}, \#9, 0 \rangle$$

# Recovery with CLR

**2. redo:**

- edit whole log and CLRs

**3. undo:**

- manage list of all losers + highest current LSN per loser
- for the loser with maximal current LSN:
  - "normal" log-entry:
    (1) undo; (2) construct CLR;
    (3) set current LSN to `PrevLSN`
  - CLR:
    (1) set current LSN to `UndoNxtLSN`

# Example: Logging and Recovery

**Schedule**

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$
$r_1(A, a_1)$
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

**Log Entries**

**Pages**

| $P_A$ | LSN: #0 |
|---|---|
| $A = 75$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #0 |
|---|---|
| $C = 10$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|

# Example: Logging and Recovery

**Schedule**

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$
$r_1(A, a_1)$
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

**Log Entries**

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$

**Pages**

| $P_A$ | LSN: #0 |
|---|---|
| $A = 75$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #0 |
|---|---|
| $C = 10$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

**Schedule**

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

**Log Entries**

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$

**Pages**

| $P_A$ | LSN: #0 |
|---|---|
| $A = 75$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #0 |
|---|---|
| $C = 10$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

**Schedule**

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

**Log Entries**

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$

**Pages**

| $P_A$ | LSN: #0 |
|---|---|
| $A = 75$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #0 |
|---|---|
| $C = 10$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

**Schedule**

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

**Log Entries**

$[\#1,\ T_3,\ \texttt{BOT},\ \#0]$
$[\#2,\ T_1,\ \texttt{BOT},\ \#0]$
$[\#3,\ T_2,\ \texttt{BOT},\ \#0]$
$[\#4,\ T_1,\ P_A,\ \text{A-=50},\ \text{A+=50},\ \#2]$

**Pages**

| $P_A$ | LSN: #4 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #0 |
|---|---|
| $C = 10$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

| Schedule | Log Entries |
|---|---|
| $\mathtt{BOT}_3$ | [#1, $T_3$, BOT, #0] |
| $\mathtt{BOT}_1$ | [#2, $T_1$, BOT, #0] |
| $\mathtt{BOT}_2$ | [#3, $T_2$, BOT, #0] |
| $r_3(C, c_3)$ [10] | [#4, $T_1$, $P_A$, A-=50, A+=50, #2] |
| $r_1(A, a_1)$ [75] | |
| $w_1(A, a_1 - 50)$ | |
| $r_2(B, b_2)$ [120] | |
| $w_2(A, b_2 - 95)$ | |
| $w_3(C, c_3 + 25)$ | |
| $w_2(B, b_2 + 50)$ | |
| $r_1(C, c_1)$ | |
| $w_1(C, c_1 + 25)$ | |
| ... | |

**Pages**

| $P_A$ | LSN: #4 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #0 |
|---|---|
| $C = 10$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

| Schedule | Log Entries |
|---|---|
| $\mathtt{BOT}_3$ | [#1, $T_3$, BOT, #0] |
| $\mathtt{BOT}_1$ | [#2, $T_1$, BOT, #0] |
| $\mathtt{BOT}_2$ | [#3, $T_2$, BOT, #0] |
| $r_3(C, c_3)$ [10] | [#4, $T_1$, $P_A$, A-=50, A+=50, #2] |
| $r_1(A, a_1)$ [75] | [#5, $T_2$, $P_A$, A+=0, A-=0, #3] |
| $w_1(A, a_1 - 50)$ | |
| $r_2(B, b_2)$ [120] | |
| $w_2(A, b_2 - 95)$ | |
| $w_3(C, c_3 + 25)$ | |
| $w_2(B, b_2 + 50)$ | |
| $r_1(C, c_1)$ | |
| $w_1(C, c_1 + 25)$ | |
| ... | |

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #0 |
|---|---|
| $C = 10$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 5 |
| 3 | 1 |

# Example: Logging and Recovery

| Schedule | Log Entries |
|---|---|
| $\mathtt{BOT}_3$ | [#1, $T_3$, BOT, #0] |
| $\mathtt{BOT}_1$ | [#2, $T_1$, BOT, #0] |
| $\mathtt{BOT}_2$ | [#3, $T_2$, BOT, #0] |
| $r_3(C, c_3)$ [10] | [#4, $T_1$, $P_A$, A-=50, A+=50, #2] |
| $r_1(A, a_1)$ [75] | [#5, $T_2$, $P_A$, A+=0, A-=0, #3] |
| $w_1(A, a_1 - 50)$ | [#6, $T_3$, $P_C$, C+=25, C-=25, #1] |
| $r_2(B, b_2)$ [120] | |
| $w_2(A, b_2 - 95)$ | |
| $w_3(C, c_3 + 25)$ | |
| $w_2(B, b_2 + 50)$ | |
| $r_1(C, c_1)$ | |
| $w_1(C, c_1 + 25)$ | |
| ... | |

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #6 |
|---|---|
| $C = 35$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 5 |
| 3 | 6 |

# Example: Logging and Recovery

**Schedule**

$\texttt{BOT}_3$
$\texttt{BOT}_1$
$\texttt{BOT}_2$
$r_3(C, c_3)$ [10]
$r_1(A, a_1)$ [75]
$w_1(A, a_1 - 50)$
$r_2(B, b_2)$ [120]
$w_2(A, b_2 - 95)$
$w_3(C, c_3 + 25)$
$w_2(B, b_2 + 50)$
$r_1(C, c_1)$
$w_1(C, c_1 + 25)$
...

**Log Entries**

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #7 |
|---|---|
| $B = 170$ | |

| $P_C$ | LSN: #6 |
|---|---|
| $C = 35$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 7 |
| 3 | 6 |

# Example: Logging and Recovery

| Schedule | Log Entries |
|---|---|
| $\mathtt{BOT}_3$ | [#1, $T_3$, BOT, #0] |
| $\mathtt{BOT}_1$ | [#2, $T_1$, BOT, #0] |
| $\mathtt{BOT}_2$ | [#3, $T_2$, BOT, #0] |
| $r_3(C, c_3)$ [10] | [#4, $T_1$, $P_A$, A-=50, A+=50, #2] |
| $r_1(A, a_1)$ [75] | [#5, $T_2$, $P_A$, A+=0, A-=0, #3] |
| $w_1(A, a_1 - 50)$ | [#6, $T_3$, $P_C$, C+=25, C-=25, #1] |
| $r_2(B, b_2)$ [120] | [#7, $T_2$, $P_B$, B+=50, B-=50, #5] |
| $w_2(A, b_2 - 95)$ | |
| $w_3(C, c_3 + 25)$ | |
| $w_2(B, b_2 + 50)$ | |
| $r_1(C, c_1)$ [35] | |
| $w_1(C, c_1 + 25)$ | |
| ... | |

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #7 |
|---|---|
| $B = 170$ | |

| $P_C$ | LSN: #6 |
|---|---|
| $C = 35$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 7 |
| 3 | 6 |

# Example: Logging and Recovery

| Schedule | Log Entries |
|---|---|
| $\texttt{BOT}_3$ | [#1, $T_3$, BOT, #0] |
| $\texttt{BOT}_1$ | [#2, $T_1$, BOT, #0] |
| $\texttt{BOT}_2$ | [#3, $T_2$, BOT, #0] |
| $r_3(C, c_3)$ [10] | [#4, $T_1$, $P_A$, A-=50, A+=50, #2] |
| $r_1(A, a_1)$ [75] | [#5, $T_2$, $P_A$, A+=0, A-=0, #3] |
| $w_1(A, a_1 - 50)$ | [#6, $T_3$, $P_C$, C+=25, C-=25, #1] |
| $r_2(B, b_2)$ [120] | [#7, $T_2$, $P_B$, B+=50, B-=50, #5] |
| $w_2(A, b_2 - 95)$ | [#8, $T_1$, $P_C$, C+=25, C-=25, #4] |
| $w_3(C, c_3 + 25)$ | |
| $w_2(B, b_2 + 50)$ | |
| $r_1(C, c_1)$ [35] | |
| $w_1(C, c_1 + 25)$ | |
| ... | |

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #7 |
|---|---|
| $B = 170$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 7 |
| 3 | 6 |

# Example: Logging and Recovery

| Schedule | Log Entries |
|---|---|
| ... | [#1, $T_3$, BOT, #0] |
| $r_2(C, c_2)$ [60] | [#2, $T_1$, BOT, #0] |
| $r_3(B, b_3)$ [170] | [#3, $T_2$, BOT, #0] |
| $w_3(B, b_3 - 75)$ | [#4, $T_1$, $P_A$, A-=50, A+=50, #2] |
| $w_2(C, c_2 + 25)$ | [#5, $T_2$, $P_A$, A+=0, A-=0, #3] |
| commit$_3$ | [#6, $T_3$, $P_C$, C+=25, C-=25, #1] |
| abort$_2$ | [#7, $T_2$, $P_B$, B+=50, B-=50, #5] |
| | [#8, $T_1$, $P_C$, C+=25, C-=25, #4] |

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #7 |
|---|---|
| $B = 170$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 7 |
| 3 | 6 |

# Example: Logging and Recovery

| Schedule | Log Entries |
|---|---|
| ... | [#1, $T_3$, BOT, #0] |
| $r_2(C, c_2)$ [60] | [#2, $T_1$, BOT, #0] |
| $r_3(B, b_3)$ [170] | [#3, $T_2$, BOT, #0] |
| $w_3(B, b_3 - 75)$ | [#4, $T_1$, $P_A$, A-=50, A+=50, #2] |
| $w_2(C, c_2 + 25)$ | [#5, $T_2$, $P_A$, A+=0, A-=0, #3] |
| commit$_3$ | [#6, $T_3$, $P_C$, C+=25, C-=25, #1] |
| abort$_2$ | [#7, $T_2$, $P_B$, B+=50, B-=50, #5] |
| | [#8, $T_1$, $P_C$, C+=25, C-=25, #4] |
| | [#9, $T_3$, $P_B$, B-=75, B+=75, #6] |

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 7 |
| 3 | 9 |

# Example: Logging and Recovery

| Schedule | Log Entries |
|---|---|
| ... | [#1, $T_3$, BOT, #0] |
| $r_2(C, c_2)$ [60] | [#2, $T_1$, BOT, #0] |
| $r_3(B, b_3)$ [170] | [#3, $T_2$, BOT, #0] |
| $w_3(B, b_3 - 75)$ | [#4, $T_1$, $P_A$, A-=50, A+=50, #2] |
| $w_2(C, c_2 + 25)$ | [#5, $T_2$, $P_A$, A+=0, A-=0, #3] |
| commit$_3$ | [#6, $T_3$, $P_C$, C+=25, C-=25, #1] |
| abort$_2$ | [#7, $T_2$, $P_B$, B+=50, B-=50, #5] |
| | [#8, $T_1$, $P_C$, C+=25, C-=25, #4] |
| | [#9, $T_3$, $P_B$, B-=75, B+=75, #6] |
| | [#10, $T_2$, $P_C$, C+=25, C-=25, #7] |

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #10 |
|---|---|
| $C = 85$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 10 |
| 3 | 9 |

# Example: Logging and Recovery

**Schedule**

$\ldots$
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

**Log Entries**

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #10 |
|---|---|
| $C = 85$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 10 |

# Example: Logging and Recovery

**Schedule**

$\ldots$
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

**Log Entries**

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #10 |
|---|---|
| $C = 85$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 10 |

# Example: Logging and Recovery

**Schedule**

$\ldots$
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

**Log Entries**

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #10 |
|---|---|
| $C = 85$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 10 |

# Example: Logging and Recovery

**Schedule**

$\ldots$
$r_2(C, c_2)$ [60]
$r_3(B, b_3)$ [170]
$w_3(B, b_3 - 75)$
$w_2(C, c_2 + 25)$
$\texttt{commit}_3$
$\texttt{abort}_2$

**Log Entries**

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
$\langle$#12, $T_2$, $P_C$, C-=25, #10, #7$\rangle$

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #12 |
|---|---|
| $C = 60$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 7 |

# Example: Logging and Recovery

| Schedule | Log Entries |
|---|---|
| ... | [#1, $T_3$, BOT, #0] |
| $r_2(C, c_2)$ [60] | [#2, $T_1$, BOT, #0] |
| $r_3(B, b_3)$ [170] | [#3, $T_2$, BOT, #0] |
| $w_3(B, b_3 - 75)$ | [#4, $T_1$, $P_A$, A-=50, A+=50, #2] |
| $w_2(C, c_2 + 25)$ | [#5, $T_2$, $P_A$, A+=0, A-=0, #3] |
| commit$_3$ | [#6, $T_3$, $P_C$, C+=25, C-=25, #1] |
| abort$_2$ | [#7, $T_2$, $P_B$, B+=50, B-=50, #5] |
| | [#8, $T_1$, $P_C$, C+=25, C-=25, #4] |
| | [#9, $T_3$, $P_B$, B-=75, B+=75, #6] |
| | [#10, $T_2$, $P_C$, C+=25, C-=25, #7] |
| | [#11, $T_3$, COMMIT, #9] |
| | ⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩ |
| | crash |

**Pages**

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #12 |
|---|---|
| $C = 60$ | |

**active trans.**

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 7 |

# Example: Logging and Recovery

**Log-Entries**

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

**Pages**

| $P_A$ LSN: #4 |
|---|
| $A = 25$ |

| $P_B$ LSN: #0 |
|---|
| $B = 120$ |

| $P_C$ LSN: #8 |
|---|
| $C = 60$ |

# Example: Logging and Recovery

## 1. analysis stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #4 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *Status* |
|---|---|
| 1 | loser |
| 2 | loser |
| 3 | winner |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #4 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #4 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #4 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 2 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #4 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #4 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #4 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 3 |
| 3 | 1 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3] ✗
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ LSN: #5 |
|---|
| $A = 25$ |

| $P_B$ LSN: #0 |
|---|
| $B = 120$ |

| $P_C$ LSN: #8 |
|---|
| $C = 60$ |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 5 |
| 3 | 1 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3] ✗
[#6, $T_3$, $P_C$, C+=25, C-=25, #1] ✓
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #0 |
|---|---|
| $B = 120$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 5 |
| 3 | 6 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3] ✗
[#6, $T_3$, $P_C$, C+=25, C-=25, #1] ✓
[#7, $T_2$, $P_B$, B+=50, B-=50, #5] ✗
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #7 |
|---|---|
| $B = 170$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 4 |
| 2 | 7 |
| 3 | 6 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3] ✗
[#6, $T_3$, $P_C$, C+=25, C-=25, #1] ✓
[#7, $T_2$, $P_B$, B+=50, B-=50, #5] ✗
[#8, $T_1$, $P_C$, C+=25, C-=25, #4] ✓
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ LSN: #5 |
|---|
| $A = 25$ |

| $P_B$ LSN: #7 |
|---|
| $B = 170$ |

| $P_C$ LSN: #8 |
|---|
| $C = 60$ |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 7 |
| 3 | 6 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3] ✗
[#6, $T_3$, $P_C$, C+=25, C-=25, #1] ✓
[#7, $T_2$, $P_B$, B+=50, B-=50, #5] ✗
[#8, $T_1$, $P_C$, C+=25, C-=25, #4] ✓
[#9, $T_3$, $P_B$, B-=75, B+=75, #6] ✗
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #8 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 7 |
| 3 | 9 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3] ✗
[#6, $T_3$, $P_C$, C+=25, C-=25, #1] ✓
[#7, $T_2$, $P_B$, B+=50, B-=50, #5] ✗
[#8, $T_1$, $P_C$, C+=25, C-=25, #4] ✓
[#9, $T_3$, $P_B$, B-=75, B+=75, #6] ✗
[#10, $T_2$, $P_C$, C+=25, C-=25, #7] ✗
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #10 |
|---|---|
| $C = 85$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 10 |
| 3 | 9 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3] ✗
[#6, $T_3$, $P_C$, C+=25, C-=25, #1] ✓
[#7, $T_2$, $P_B$, B+=50, B-=50, #5] ✗
[#8, $T_1$, $P_C$, C+=25, C-=25, #4] ✓
[#9, $T_3$, $P_B$, B-=75, B+=75, #6] ✗
[#10, $T_2$, $P_C$, C+=25, C-=25, #7] ✗
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #10 |
|---|---|
| $C = 85$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 10 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3] ✗
[#6, $T_3$, $P_C$, C+=25, C-=25, #1] ✓
[#7, $T_2$, $P_B$, B+=50, B-=50, #5] ✗
[#8, $T_1$, $P_C$, C+=25, C-=25, #4] ✓
[#9, $T_3$, $P_B$, B-=75, B+=75, #6] ✗
[#10, $T_2$, $P_C$, C+=25, C-=25, #7] ✗
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩ ✗

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #12 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 12 |

# Example: Logging and Recovery

## 2. redo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2] ✓
[#5, $T_2$, $P_A$, A+=0, A-=0, #3] ✗
[#6, $T_3$, $P_C$, C+=25, C-=25, #1] ✓
[#7, $T_2$, $P_B$, B+=50, B-=50, #5] ✗
[#8, $T_1$, $P_C$, C+=25, C-=25, #4] ✓
[#9, $T_3$, $P_B$, B-=75, B+=75, #6] ✗
[#10, $T_2$, $P_C$, C+=25, C-=25, #7] ✗
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩ ✗

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #12 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | *LSN* |
|---|---|
| 1 | 8 |
| 2 | 12 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #12 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 8 | 8 |
| 2 | 12 | 12 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #12 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 8 | 8 |
| 2 | 12 | 12 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #12 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 8 | 8 |
| 2 | 7 | 12 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #12 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 8 | 8 |
| 2 | 7 | 12 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #12 |
|---|---|
| $C = 60$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 8 | 8 |
| 2 | 7 | 12 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩
→ ⟨#13, $T_1$, $P_C$, C-=25, #8, #4⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 4 | 13 |
| 2 | 7 | 12 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #9 |
|---|---|
| $B = 95$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 4 | 13 |
| 2 | 7 | 12 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩
→ ⟨#14, $T_2$, $P_B$, B-=50, #12, #5⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #14 |
|---|---|
| $B = 45$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 4 | 13 |
| 2 | 5 | 14 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #5 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #14 |
|---|---|
| $B = 45$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 4 | 13 |
| 2 | 5 | 14 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩
→ ⟨#15, $T_2$, $P_A$, A-=0, #14, #3⟩

### Pages

| $P_A$ | LSN: #15 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #14 |
|---|---|
| $B = 45$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 4 | 13 |
| 2 | 3 | 15 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #15 |
|---|---|
| $A = 25$ | |

| $P_B$ | LSN: #14 |
|---|---|
| $B = 45$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 4 | 13 |
| 2 | 3 | 15 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩
→ ⟨#16, $T_1$, $P_A$, A+=50, #13, #2⟩

### Pages

| $P_A$ | LSN: #16 |
|---|---|
| $A = 75$ | |

| $P_B$ | LSN: #14 |
|---|---|
| $B = 45$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 2 | 16 |
| 2 | 3 | 15 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #16 |
|---|---|
| $A = 75$ | |

| $P_B$ | LSN: #14 |
|---|---|
| $B = 45$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 2 | 16 |
| 2 | 3 | 15 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩
→ ⟨#17, $T_2$, BOT, #15 ⟩

### Pages

| $P_A$ | LSN: #16 |
|---|---|
| $A = 75$ | |

| $P_B$ | LSN: #14 |
|---|---|
| $B = 45$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 2 | 16 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩

### Pages

| $P_A$ | LSN: #16 |
|---|---|
| $A = 75$ | |

| $P_B$ | LSN: #14 |
|---|---|
| $B = 45$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| *Tr.* | NxtU | LSN |
|---|---|---|
| 1 | 2 | 16 |

# Example: Logging and Recovery

## 3. undo-stage

### Log-Entries

[#1, $T_3$, BOT, #0]
[#2, $T_1$, BOT, #0]
[#3, $T_2$, BOT, #0]
[#4, $T_1$, $P_A$, A-=50, A+=50, #2]
[#5, $T_2$, $P_A$, A+=0, A-=0, #3]
[#6, $T_3$, $P_C$, C+=25, C-=25, #1]
[#7, $T_2$, $P_B$, B+=50, B-=50, #5]
[#8, $T_1$, $P_C$, C+=25, C-=25, #4]
[#9, $T_3$, $P_B$, B-=75, B+=75, #6]
[#10, $T_2$, $P_C$, C+=25, C-=25, #7]
[#11, $T_3$, COMMIT, #9]
⟨#12, $T_2$, $P_C$, C-=25, #10, #7⟩
→ ⟨#18, $T_1$, BOT, #16 ⟩

### Pages

| $P_A$ | LSN: #16 |
|---|---|
| $A = 75$ | |

| $P_B$ | LSN: #14 |
|---|---|
| $B = 45$ | |

| $P_C$ | LSN: #13 |
|---|---|
| $C = 35$ | |

### transactions

| Tr. | NxtU | LSN |
|---|---|---|
| | | |

# Recovery with CLR – Example

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$$

# Recovery with CLR – Example

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$$

**redo-phase:**

- log-entries #1 – #8
- consider page-LSN

# Recovery with CLR – Example

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$$

**redo-phase:**

- log-entries #1 – #8
- consider page-LSN

**undo-phase:**

- follow UndoNxtLSN

# Recovery with CLR – Example

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$$

**redo-phase:**

- log-entries #1 – #8
- consider page-LSN

**undo-phase:**

- follow UndoNxtLSN

# Recovery with CLR – Example

## Example

$[\#1, T_1, \texttt{BOT}, 0]$

$[\#2, T_2, \texttt{BOT}, 0]$

$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$

$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$

$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$

$[\#6, T_1, \texttt{commit}, \#5]$

$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$

$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$

**redo-phase:**

- log-entries #1 – #8
- consider page-LSN

**undo-phase:**

- follow UndoNxtLSN

# Recovery with CLR – Example

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$$

$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$

$$\langle \#10, T_2, \text{-}, \text{-}, \#9, 0 \rangle$$

# Recovery with CLR – Example

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A{+}{=}100, \#7, \#4 \rangle$$

$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$

$$\langle \#10, T_2, \text{-}, \text{-}, \#9, 0 \rangle$$

**redo-phase:**

- log entries #1 – #10
- note the page-LSN

# Recovery with CLR – Example

## Example

$$[\#1, T_1, \texttt{BOT}, 0]$$

$$[\#2, T_2, \texttt{BOT}, 0]$$

$$[\#3, T_1, P_A, A\text{-}{=}50, A\text{+}{=}50, \#1]$$

$$[\#4, T_2, P_C, C\text{+}{=}100, C\text{-}{=}100, \#2]$$

$$[\#5, T_1, P_B, B\text{+}{=}50, B\text{-}{=}50, \#3]$$

$$[\#6, T_1, \texttt{commit}, \#5]$$

$$[\#7, T_2, P_A, A\text{-}{=}100, A\text{+}{=}100, \#4]$$

$$\langle \#8, T_2, P_A, A\text{+}{=}100, \#7, \#4 \rangle$$

$$\langle \#9, T_2, P_C, C\text{-}{=}100, \#8, \#2 \rangle$$

$$\langle \#10, T_2, \text{-}, \text{-}, \#9, 0 \rangle$$

**redo-phase:**

- log entries #1 – #10
- note the page-LSN

**undo-phase:**

- follow UndoNxtLSN
- UndoNxtLSN = 0
  $\Rightarrow$ $T_2$ is done

# ARIES

- described procedure forms the kernel of the ARIES procedure
- further refinement mainly for reduction of access to the background memory
- very flexible recovery method

# ARIES

- described procedure forms the kernel of the ARIES procedure
- further refinement mainly for reduction of access to the background memory
- very flexible recovery method

- core components of ARIES:
  - WAL
  - recovery of complete history in redo-step
  - construction of CLRs in undo-step

# Overview

# Local Reset of Transaction

1. investigate the last log-entry of the transaction
2. local undo: modifications of the transaction are revoked
   - log passed against chronological order
   - CLRs are constructed

# Local Reset of Transaction

1. investigate the last log-entry of the transaction
2. local undo: modifications of the transaction are revoked
   - log passed against chronological order
   - CLRs are constructed

**efficiency:**

- usually for every transaction a pointer to the last log-entry is maintained
- backward chaining via `PrevLSN` allows for quick log traversing
- most log-entries of active transactions with reasonable size are still in the buffer

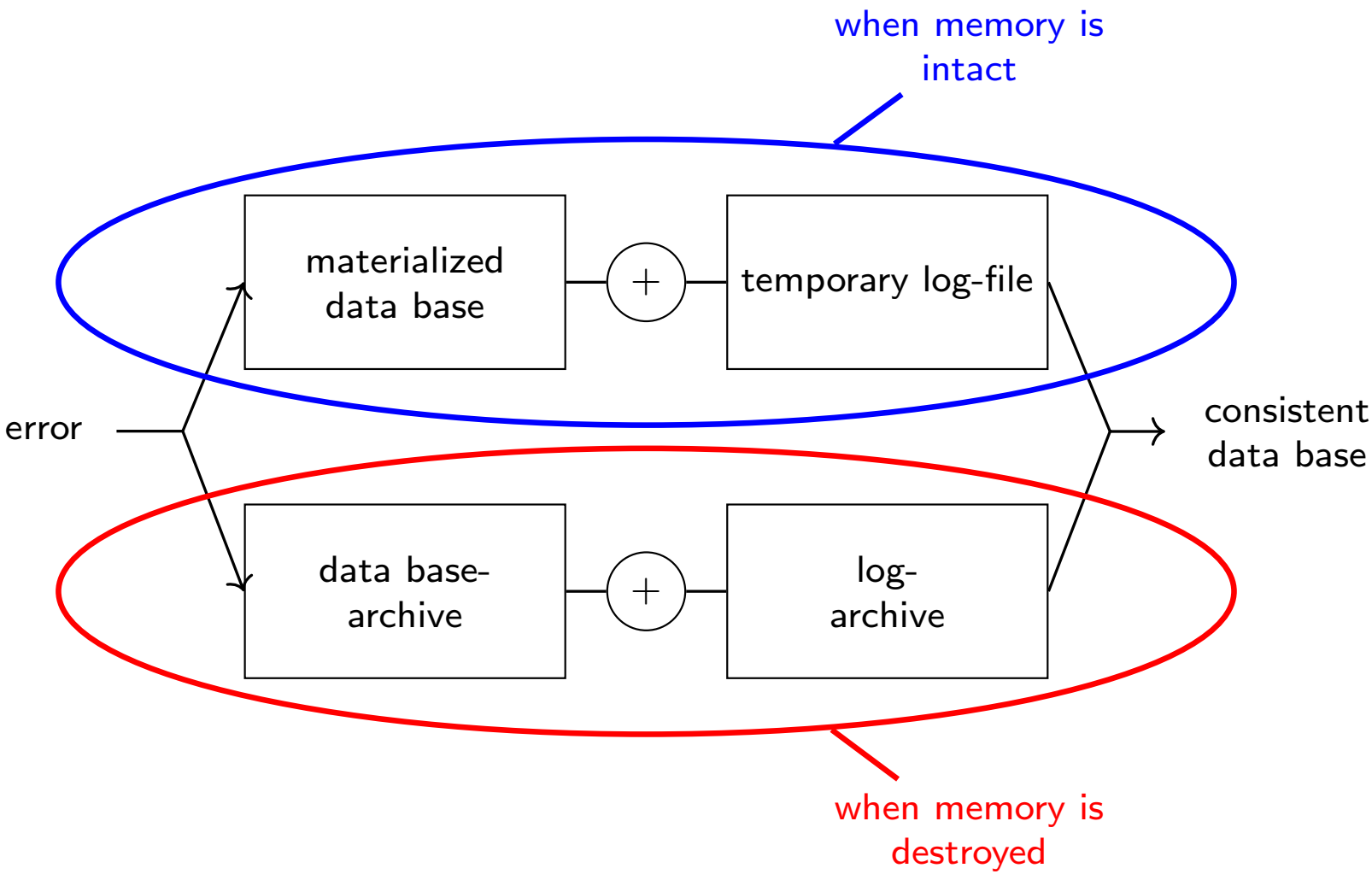
# Partial Reset of a Transaction

1. reset modifications and construct CLRs starting from the last log-entry (as for a local undo).
2. difference: do not go back to `BOT`, but reset the transaction up to the intended point

save points possible

# Overview

# Media-Recovery



# Overview

# Why Checkpoints

**problem** with recovery method:

- we have to traverse the whole log
- log gets bigger and bigger after some time

# Why Checkpoints

**problem** with recovery method:

- we have to traverse the whole log
- log gets bigger and bigger after some time

**solution:**

- enforce (occasional) writing of modified pages = checkpoint
- this way the log is needed only after a certain LSN
- the required minimal LSN is based on time and the kind of the checkpoint

# Why Checkpoints

**problem** with recovery method:

- we have to traverse the whole log
- log gets bigger and bigger after some time

**solution:**

- enforce (occasional) writing of modified pages = checkpoint
- this way the log is needed only after a certain LSN
- the required minimal LSN is based on time and the kind of the checkpoint

attention! checkpoint $\neq$ save point!

# Kinds of Checkpoints

**transaction-consistent checkpoints**

- "best" quality for recovery, but expensive

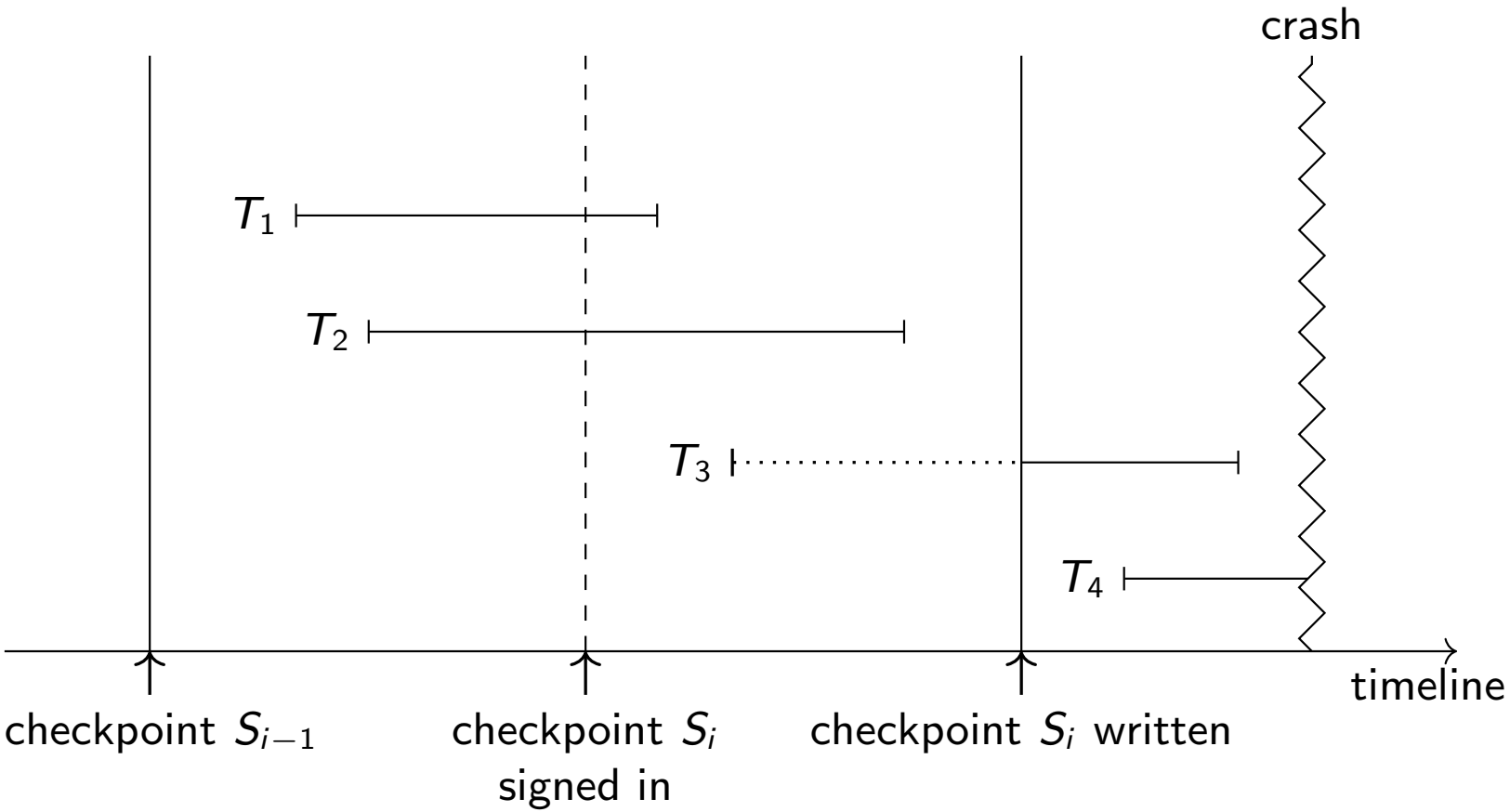
**action-consistent checkpoints**

- *undo* requires "older" log-entries, leads to peaks in load

**fuzzy checkpoints**

- *undo* and *redo* require "older" log-entries, continuous writing

# Transaction-Consistent Checkpoints



# Transaction-Consistent Checkpoints

**idea:**

- background memory should receive all modifications of transactions completed at time $S_i$
  $\Rightarrow$ no redo beyond $S_i$ needed
- at time $S_i$ no active transactions allowed
  $\Rightarrow$ no undo beyond $S_i$ needed

# Transaction-Consistent Checkpoints

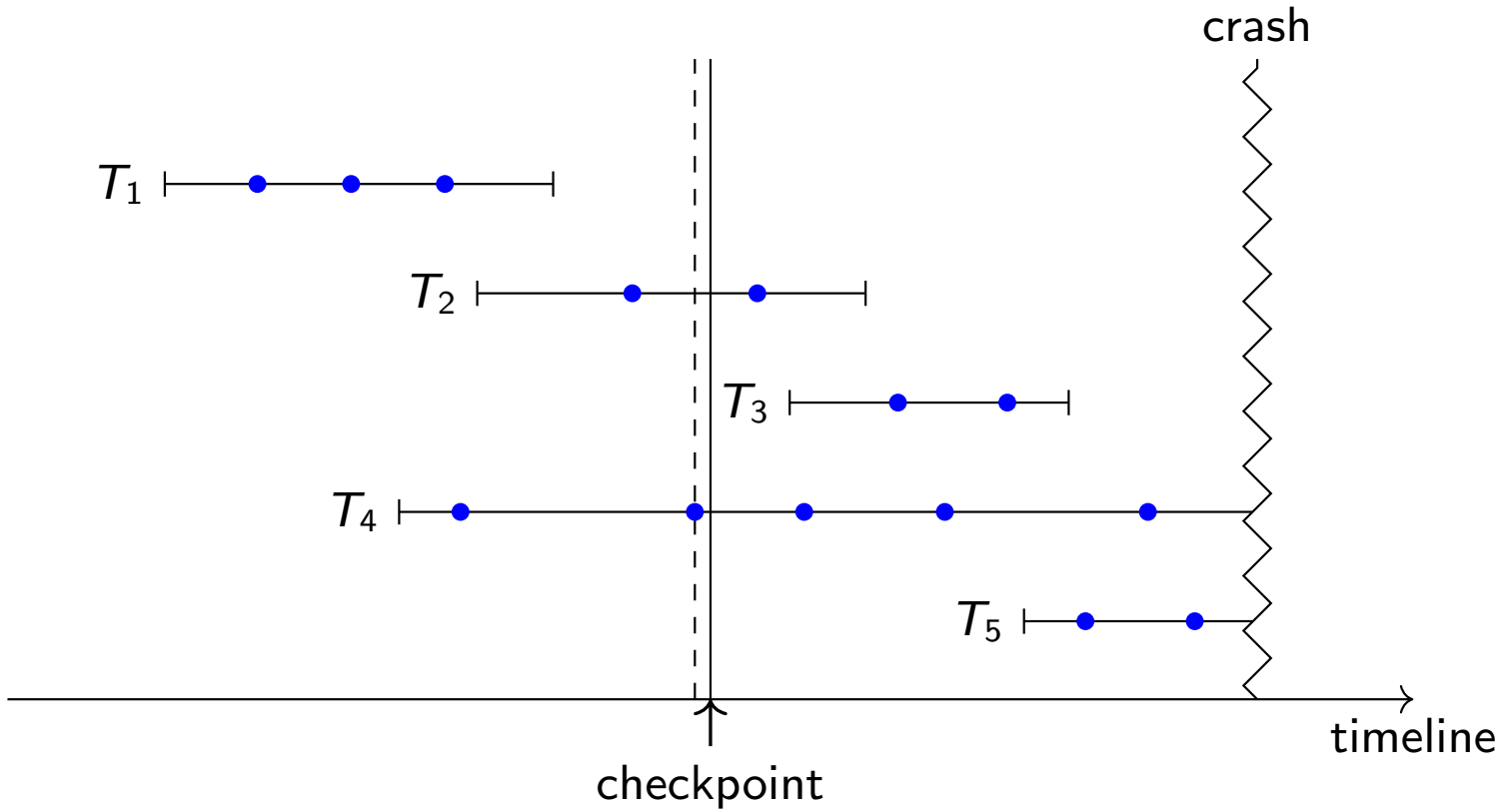
**idea:**

- background memory should receive all modifications of transactions completed at time $S_i$
  $\Rightarrow$ no redo beyond $S_i$ needed
- at time $S_i$ no active transactions allowed
  $\Rightarrow$ no undo beyond $S_i$ needed

**problem:**

- no new transactions are allowed to start between signing in and writing of checkpoints
- leads to usually not acceptable latency

# Action-Consistent Checkpoints



# Action-Consistent Checkpoints

**idea:**

- all active modifying operations should be terminated
- then writing all modified pages
- only start of next modifying operation is delayed, but not the start of new transactions

# Action-Consistent Checkpoints

**idea:**

- all active modifying operations should be terminated
- then writing all modified pages
- only start of next modifying operation is delayed, but not the start of new transactions

**recovery:**

- analysis-stage starting from $S_i$
- no redo beyond $S_i$ necessary
- in general an undo beyond $S_i$ is necessary; up to *MinLSN* (= smallest LSN of the transaction active at checkpoint)

# Fuzzy Checkpoints

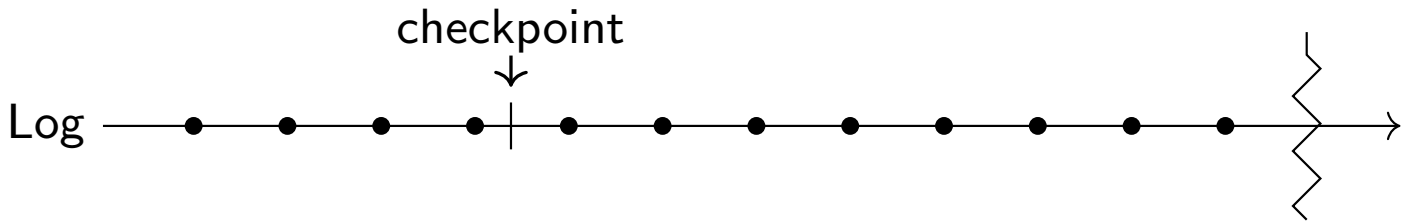
**idea:**

- pages should be written continuously
  no abrupt writing of many pages because of checkpoint
- at checkpoint only the identification of all modified pages (= dirty pages) is written
- additionally *MinDirtyPageLSN* (= minimal LSN whose modifications have not been written yet) is maintained and written at checkpoint

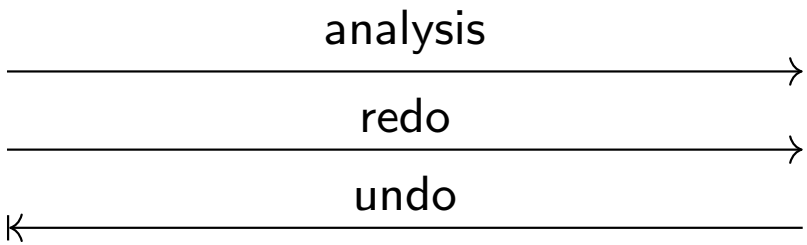
**problem:**

- „hot-spots“ (constantly needed pages) are not written for a long time, writing is enforced when a page occurred among the „dirty pages“ several times

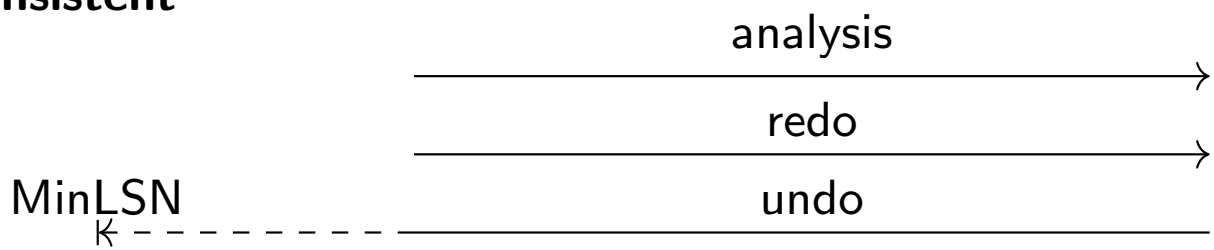
# Three Checkpoint Qualities – Summary



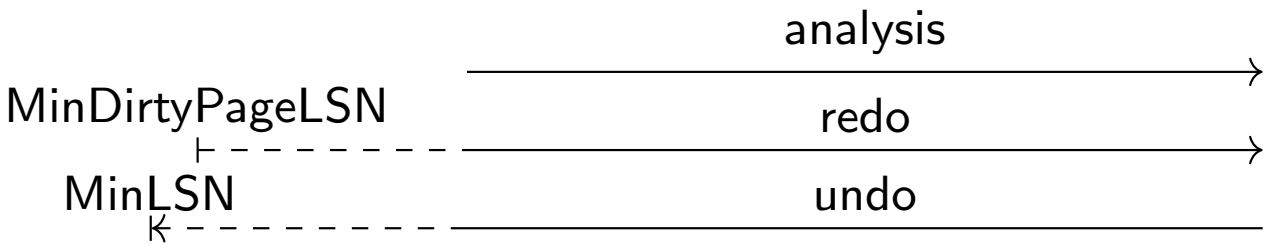

(a) **transaction consistent**




(b) **action consistent**




(c) **fuzzy**



# Learning Objectives

- What are pages?
- Which strategies for the replacement/outsourcing of buffer pages are there? Advantages? Disadvantages?
- Which introducing strategies are there and how do the work?
- Which kinds of error categories are there?
- What kind of information is stored in a log-entry?
  - Difference between logical and physical logging.
  - What is the LSN? Why do we need it?
- What is the WAL-principle?
- How does recovery work after an error with loss of buffer?
  - Stages of recovery and how do they work?
  - How is fault tolerance for recovery implemented?
  - What are CLRs?
- Why do we need checkpoint and which kinds of checkpoint are there?