Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CHAI
Humanities-Centered AI

Databases

Malte Luttermann – Institute for Humanities-Centered Artificial Intelligence (CHAI)

January 22, 2026

Version from January 21, 2026

Topics for Today

- Error types
- Write-Ahead-Log (WAL)
- Recovery after failure

Error Types

What are the three types of errors a database system has to recover from?

Error Types

What are the three types of errors a database system has to recover from?

- **Transaction errors**
  - A transaction is aborted
  - All changes must be undone

Error Types

What are the three types of errors a database system has to recover from?

- **Transaction errors**
  - A transaction is aborted
  - All changes must be undone
- **System errors**
  - Database or operating system crash, power failure, etc.
  - Changes in main memory are lost
  - Ensure that no changes are lost with commit (or restore their effects)

Error Types

What are the three types of errors a database system has to recover from?

- **Transaction errors**
  - A transaction is aborted
  - All changes must be undone
- **System errors**
  - Database or operating system crash, power failure, etc.
  - Changes in main memory are lost
  - Ensure that no changes are lost with commit (or restore their effects)
- **Media errors**
  - Hard drive crashes, fire, water damage
  - Recovery from external storage media

Write-Ahead-Log (WAL)

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the log file:

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 90>
3	<START T2>
4	<WRITE T2, B, 200, 210>
5	<COMMIT T1>
6	<WRITE T2, C, 300, 280>
	<CRASH>

What is the state of the database after a system crash at this point?

Write-Ahead-Log (WAL)

- Transactions that have committed: T_1
- Transactions that have not committed: T_2
- Redo changes of T_1 and T_2, undo changes of T_2
- Final state:

$$A = 90, \quad B = 200, \quad C = 300$$

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(1) Analysis phase: Identify committed and uncommitted transactions.

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<WRITE T2, A, 150, 180>
9	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(1) Analysis phase: Identify committed and uncommitted transactions.

- Committed transactions: T_1
- Uncommitted transactions: T_2, T_3

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<WRITE T2, A, 150, 180>
9	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(2) Redo phase: Replay the operations in the log (in forward direction).

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<WRITE T2, A, 150, 180>
9	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(2) Redo phase: Replay the operations in the log (in forward direction).

Redo writes:

- T_1: `<WRITE T1, A, 100, 150>` $\rightarrow A = 150$
- T_2: `<WRITE T2, B, 200, 250>` $\rightarrow B = 250$
- T_3: `<WRITE T3, C, 300, 350>` $\rightarrow C = 350$
- T_2: `<WRITE T2, A, 150, 180>` $\rightarrow A = 180$
- T_3: `<WRITE T3, B, 250, 400>` $\rightarrow B = 400$

LSN	Log Record
1	`<START T1>`
2	`<WRITE T1, A, 100, 150>`
3	`<START T2>`
4	`<WRITE T2, B, 200, 250>`
5	`<COMMIT T1>`
6	`<START T3>`
7	`<WRITE T3, C, 300, 350>`
8	`<WRITE T2, A, 150, 180>`
9	`<WRITE T3, B, 250, 400>`
	`<CRASH>`

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(3) Undo phase: Roll back unlucky transactions (in backward direction).

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<WRITE T2, A, 150, 180>
9	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(3) Undo phase: Roll back unlucky transactions (in backward direction).

Undo operations:

- T_3: `<WRITE T3, B, 250, 400>` $\rightarrow B = 250$
- T_2: `<WRITE T2, A, 150, 180>` $\rightarrow A = 150$
- T_3: `<WRITE T3, C, 300, 350>` $\rightarrow C = 300$
- T_2: `<WRITE T2, B, 200, 250>` $\rightarrow B = 200$

LSN	Log Record
1	`<START T1>`
2	`<WRITE T1, A, 100, 150>`
3	`<START T2>`
4	`<WRITE T2, B, 200, 250>`
5	`<COMMIT T1>`
6	`<START T3>`
7	`<WRITE T3, C, 300, 350>`
8	`<WRITE T2, A, 150, 180>`
9	`<WRITE T3, B, 250, 400>`
	`<CRASH>`

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

What is the database state after recovery?

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<WRITE T2, A, 150, 180>
9	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

What is the database state after recovery?

$$A = 150, \quad B = 200, \quad C = 300$$

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<WRITE T2, A, 150, 180>
9	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(1) Analysis phase: Identify committed and uncommitted transactions.

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<CHECKPOINT>
9	<WRITE T2, A, 150, 180>
10	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(1) Analysis phase: Identify committed and uncommitted transactions.

- Committed transactions: None
- Uncommitted transactions: T_2 (minLSN = 4), T_3 (minLSN = 7)
- Dirty page table: A (LSN 9), B (LSN 4), C (LSN 7)

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<CHECKPOINT>
9	<WRITE T2, A, 150, 180>
10	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(2) Redo phase: Replay the operations in the log (in forward direction).

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<CHECKPOINT>
9	<WRITE T2, A, 150, 180>
10	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(2) Redo phase: Replay the operations in the log (in forward direction).

Redo writes:

- T_2: `<WRITE T2, B, 200, 250>` $\rightarrow B = 250$
- T_3: `<WRITE T3, C, 300, 350>` $\rightarrow C = 350$
- T_2: `<WRITE T2, A, 150, 180>` $\rightarrow A = 180$
- T_3: `<WRITE T3, B, 250, 400>` $\rightarrow B = 400$

LSN	Log Record
1	`<START T1>`
2	`<WRITE T1, A, 100, 150>`
3	`<START T2>`
4	`<WRITE T2, B, 200, 250>`
5	`<COMMIT T1>`
6	`<START T3>`
7	`<WRITE T3, C, 300, 350>`
8	`<CHECKPOINT>`
9	`<WRITE T2, A, 150, 180>`
10	`<WRITE T3, B, 250, 400>`
	`<CRASH>`

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(3) Undo phase: Roll back unlucky transactions (in backward direction).

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<CHECKPOINT>
9	<WRITE T2, A, 150, 180>
10	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

(3) Undo phase: Roll back unlucky transactions (in backward direction).

Undo operations:

- T_3: `<WRITE T3, B, 250, 400>` $\rightarrow B = 250$
- T_2: `<WRITE T2, A, 150, 180>` $\rightarrow A = 150$
- T_3: `<WRITE T3, C, 300, 350>` $\rightarrow C = 300$
- T_2: `<WRITE T2, B, 200, 250>` $\rightarrow B = 200$

LSN	Log Record
1	`<START T1>`
2	`<WRITE T1, A, 100, 150>`
3	`<START T2>`
4	`<WRITE T2, B, 200, 250>`
5	`<COMMIT T1>`
6	`<START T3>`
7	`<WRITE T3, C, 300, 350>`
8	`<CHECKPOINT>`
9	`<WRITE T2, A, 150, 180>`
10	`<WRITE T3, B, 250, 400>`
	`<CRASH>`

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

What is the database state after recovery?

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<CHECKPOINT>
9	<WRITE T2, A, 150, 180>
10	<WRITE T3, B, 250, 400>
	<CRASH>

Recovery after Failure

Consider a database with initial state $A = 100$, $B = 200$, and $C = 300$ as well as the given log.

What is the database state after recovery?

$$A = 150, \quad B = 200, \quad C = 300$$

LSN	Log Record
1	<START T1>
2	<WRITE T1, A, 100, 150>
3	<START T2>
4	<WRITE T2, B, 200, 250>
5	<COMMIT T1>
6	<START T3>
7	<WRITE T3, C, 300, 350>
8	<CHECKPOINT>
9	<WRITE T2, A, 150, 180>
10	<WRITE T3, B, 250, 400>
	<CRASH>