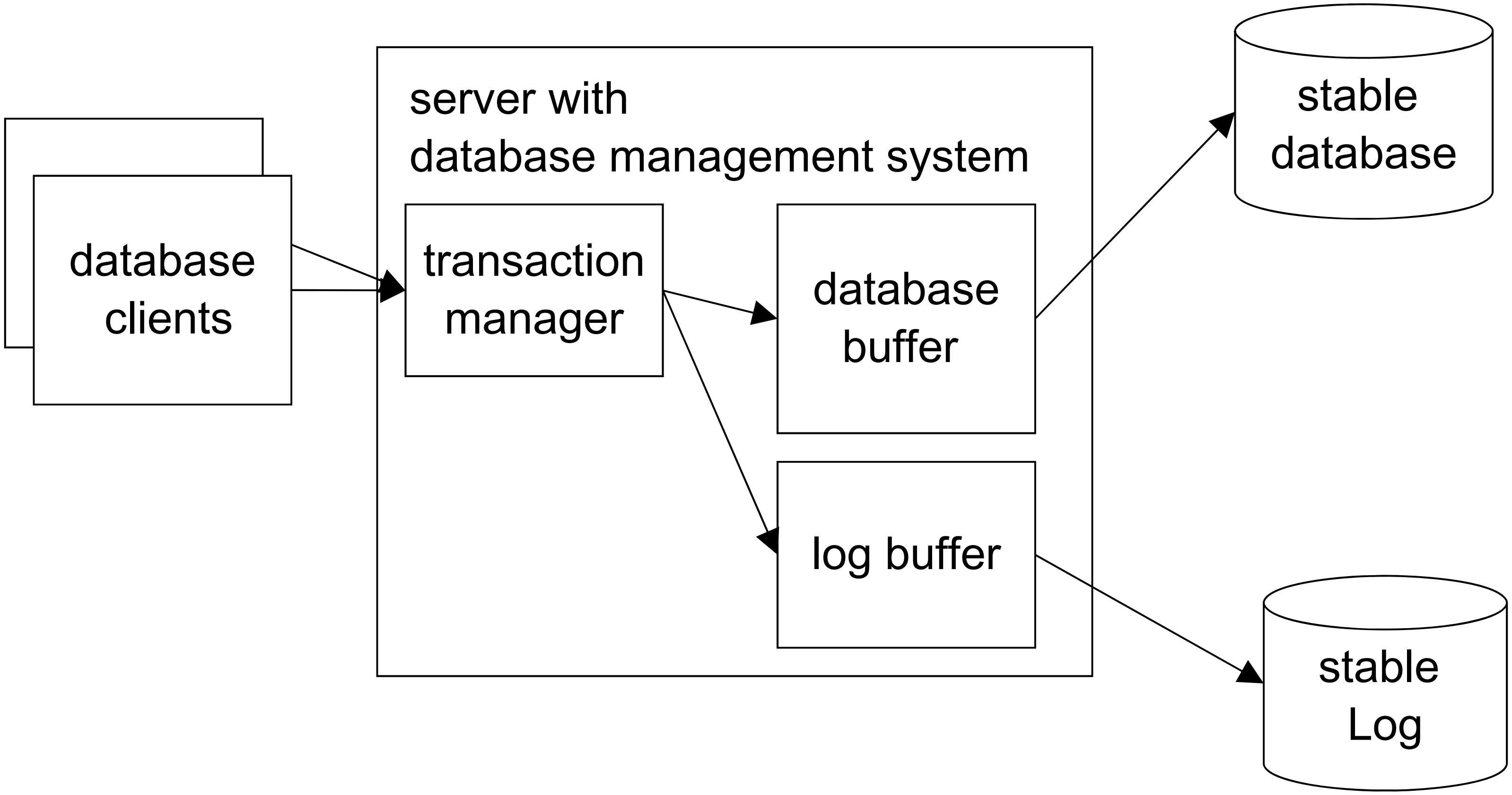
ACID atomicity

- Database logs
- Rollback

ACID atomicity requires rollback

- For ACID atomicity, the database must be able to undo write operations of uncommitted transactions.
- We consider here strategies that work with a *log.*
- The log is a list of records for write operations in the order of execution
  - optionally with some datastructures for easier access.
  - The log is related to the schedule, without reads.
- The log will be also important for recovery and ensuring ACID durability and will contain more information than we immediately need for rollback in the context of ACID atomicity.

system architecture



the database log

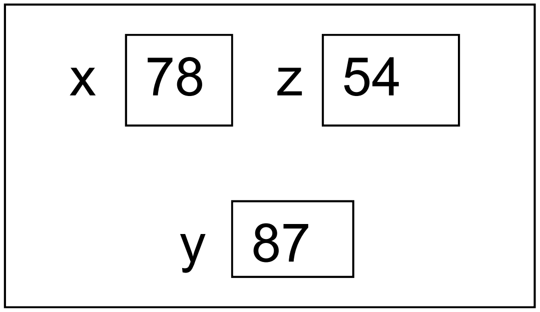
- is a central feature of typical database managers.
- The log contains a list of
  - the write operations performed.
  - The transaction demarcations: (BOT), abort, commit.
- We consider undo/redo logs: for each write operation, a log entry is written that contains:
  - a log sequence number, or LSN (denoted as “nr:”)
  - The transaction ID of the executing transaction (“ta:”).
  - An identifier of the object affected (“obj:”)
  - the value before the write operation (before-image, “b:”).
  - the value after the write operation (after-image, “a:”).

example log and database

- The log of the database at a certain point in time could look the following way (new operations are appended at lower end)
- We dont’ write BOT records in our sample logs.
- Commit/abort records contain only LSN and TA.

```
......
[nr: 110, ta: 22, obj: x, b: 2, a: 53]
[nr: 111, ta: 22, commit]
[nr: 112, ta: 23, obj: x, b: 53, a: 46]
[nr: 113, ta: 24, obj: y, b: 34, a: 87]
[nr: 114, ta: 23, obj: z, b: 23, a: 54]
[nr: 115, ta: 23, obj: x, b: 46, a: 78]
```

Log Sequence Number (LSN)



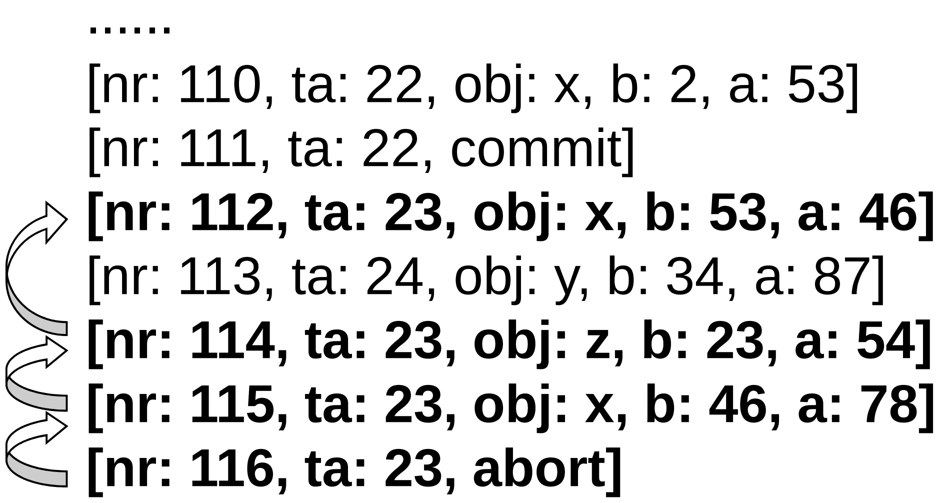
The state of the database at this point in time

rollback

- A transaction that is aborted has to be undone in a *rollback*. Rollback is the undoing of a transaction during normal operation (also known as *transaction recovery,* but we do not use this term).
- This rollback is based on the log.
  - An abort record is entered into the log.
- For all write operations performed so far, the transaction still holds the write lock:
  - The before-image is restored.
  - Afterwards, all write locks are released.

example: rollback of TA 23

- In case of a rollback, the operations are undone and the before values are restored.



......

[nr: 110, ta: 22, obj: x, b: 2, a: 53]

[nr: 111, ta: 22, commit]

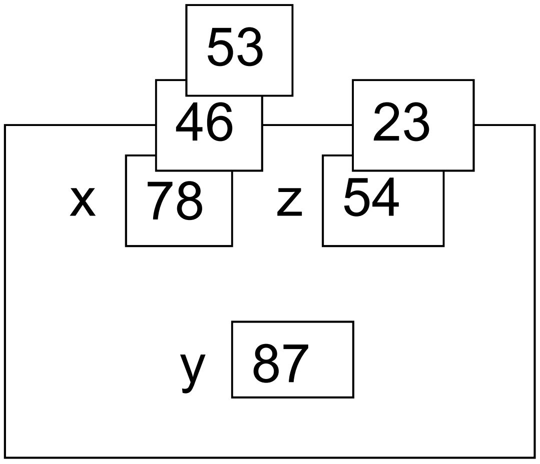
[nr: 112, ta: 23, obj: x, b: 53, a: 46]

[nr: 113, ta: 24, obj: y, b: 34, a: 87]

[nr: 114, ta: 23, obj: z, b: 23, a: 54]

[nr: 115, ta: 23, obj: x, b: 46, a: 78]

[nr: 116, ta: 23, abort]



rollback vs. compensating transaction

- If a transaction is aborted, the database does the rollback.
- Afterwards the situation is as if the transaction was never started.
- This is different from a compensating transaction.
- A compensating transaction is a transaction that, if committed, exactly undoes the effect of an earlier committed transaction. Example could be the cancellation of a purchase; a typical operation for bookkeeping.
- Both transactions remain part of the record.
- A compensating transaction can fail, because isolation has ended and other operations may have been made.