Atomicity

Consistency

Isolation

Durability

Atomicity:
- transaction is all or nothing deal
- if any part fails, whole thing gets deleted and we act like it never happened
- Uses logging mechanisms to enable rollback feature
- makes sure multiple instructions happen together or not at all

Consistency:
- transaction must follow all the rules and leave database in good state
- any data written during transaction must be valid according to constraints, triggers, etc.
- database system enforces consistency by checking for checking for constraint violations during transaction
- stops invalid data from messing up database

Isolation:

- how concurrent transactions interact with each other
- if many transactions happen at the same time, isolation makes it seem like each transaction has the database to itself

– Serializable Isolation: running each transaction one at a time. Strongest consistency, but can slow things down.

– Lower level Isolations: Allow more transaction to run simultaneously, but can lead to inconsistencies (dirty reads, phantom reads, non-repeatable reads)

- Dirty Read: transaction see data that was changed by another transaction that hasn't been committed yet
- Non-repeatable reads: transaction read same data twice and gets different results because another transaction changed the data in between
- Phantom Reads: a transaction re-runs a query and gets different results because another transaction added or deleted rows that match the search criteria

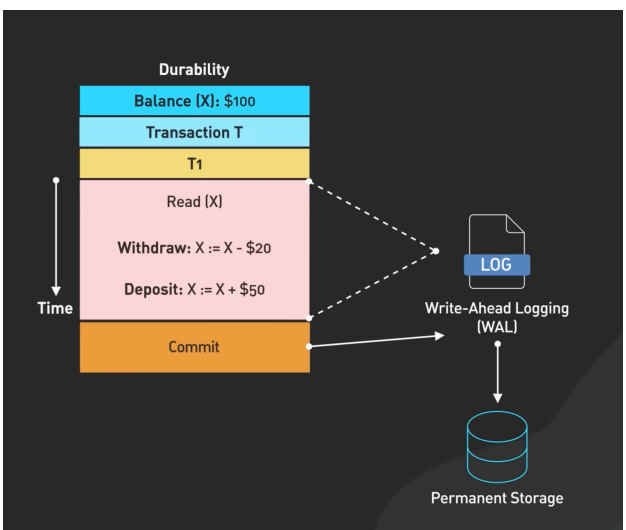
Isolation Level	Violations		
	Dirty Read	Non-repeatable Read	Phantom Read
Serializable	Don't occur	Don't occur	Don't occur
Repeatable			
Read commited	Don't occur	May occur	May occur

Durability:

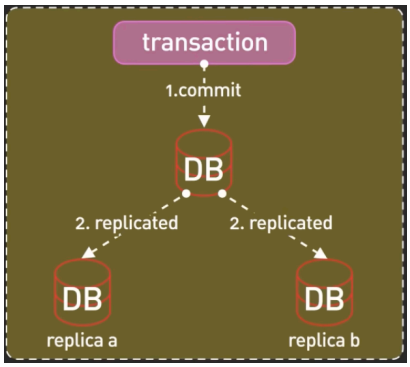
- once a transaction is committed, it is permanent, EVEN IF database CRASHES or LOSES POWER right after.
- Achieved usually through transaction logs or using write-ahead logging (WAL) to persist changes to disk/SSD before confirming commit
- ANOTHER MEANING to durability in terms of distributed databases:
  – allows for replication of data across multiple nodes

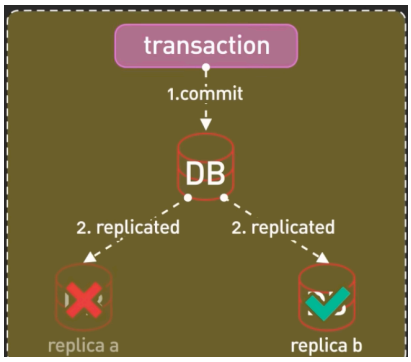
2 methods of Durability:

WAL



Replication:





if one node goes down, you'll still have the history of the committed transactions stored in b