Database Transactions and ACID Properties

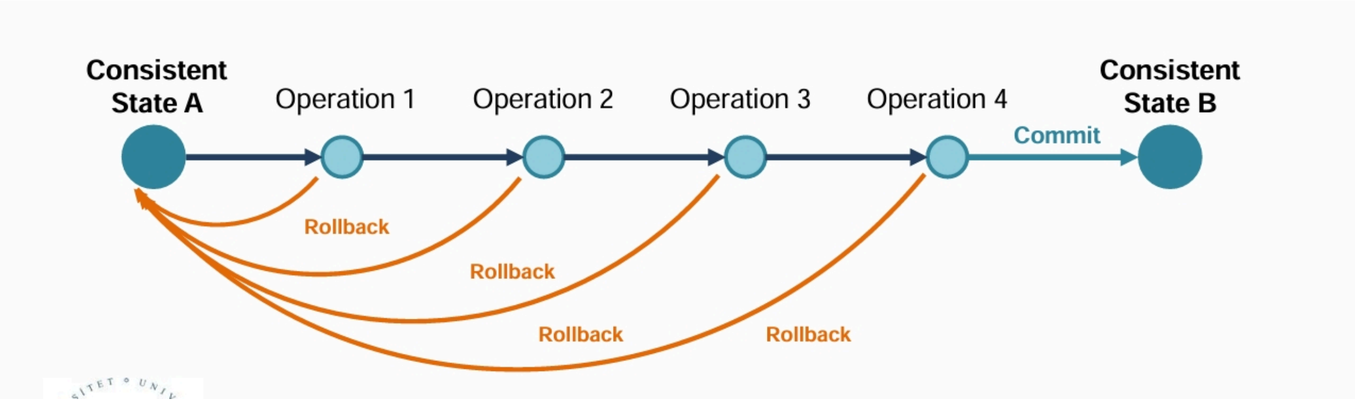
> Tags: Database Transactions, Database, ACID Properties, [[]]
> Date: 2025-12-07

Transactions

A **Transaction** is a sequence of operations performed as a single logical unit of work.

A **Transaction** generally represents any change in a database.

A **database transaction** by definition, must be **ACID**.

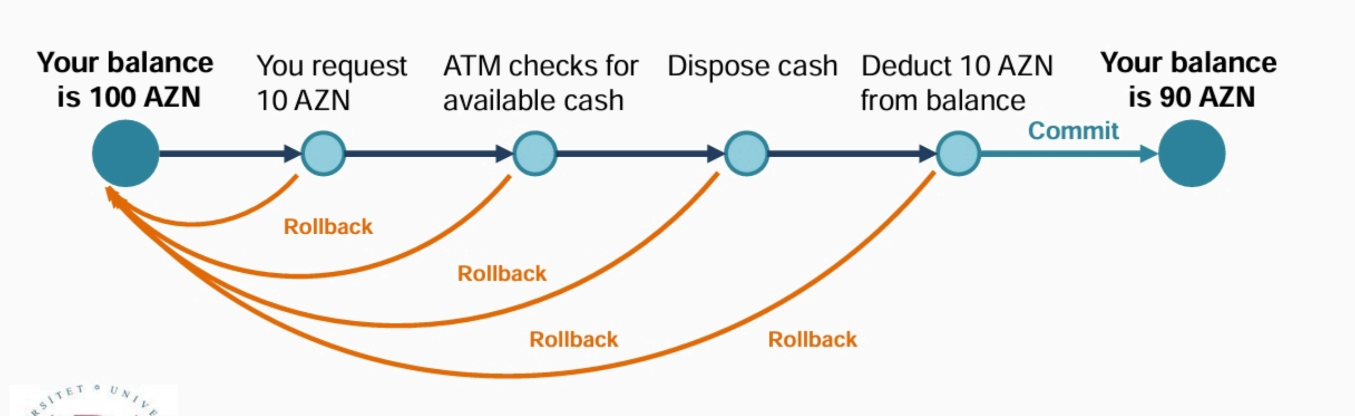


- Changing from **State A** to **State B** may require the execution of some operations. If all operations are successful, we **commit** (accept/approve) all operations and move to State B. If there's a fail in any of operations, we reject all previously made operations and **roll-back** - move to the previous State A.

Example: Transactions

The example below shows the simplified transaction flow for the ATM cash disposal.

- **State A** - is out current balance in our credit card
- **State B** - the amount that supposed to be in our credit card after withdrawal



ACID

Atomicity

A transaction must be an atomic unit of work; either all of its data modifications are performed, or none of them are performed.

- As in the previous example - either we commit all changes or roll back, no partial state is accepted

Consistency

When completed, a transaction must leave all data in a **consistent state**. In a relational database, all rules must be applied to the transaction's modifications to maintain all data integrity. All internal data structures must be correct at the end of the transaction.

- The data saved can't violate any of the database's integrity. Interrupted changes are rolled back to ensure the database placed in a state prior to the change.
- Storing student database in 2 files (tables) and when one student changes his department, it changes in one file - other file remains the older data
- Customer has 2 records in a database with different ID card numbers.

Isolation

Modifications made by concurrent (parallel, happening at the same time) transactions must be isolated from the modifications made by any other concurrent transactions (a transaction in process and not yet committed must remain isolated from any other transaction)

- *Example*: you have a 100 AZN in your credit card. Even if you open 2 payment page in online shopping portal (one with 70 AZN and another 50 AZN payment) and press "Pay" at the same time, your operations will be sequenced and one of payments will be rejected.

Durability

After a transaction has completed, its **effects are permanently in place** in their system. The modifications persist even in the event of a system failure.

- Once the transaction is committed any failure or system restart returns the data in a correct state.