[Transaction Management]

SQL query

Transaction -> collection of operation as single logical unit of work

E.g -> Transfer of money from one account to another

1. Read account 1 tuple
2. bal amount -> Subtract
3. update account 2 tuple

↓

2 SQL query -> should be 1 transaction

* Transaction objective is to ensure ACID

* In real world DB, multiple user are accessing DBMS concurrently.

* If there is some shared data b/w two transaction → it should be controlled

⇓

Similar to Process Synchronisation in OS where critical data is share by multiple process / threads.

* Transaction Manger of DBMS handles all this, allows application developers to focus on implementing transaction

[ACID]

- **Atomicity**. Either all operations of the transaction are reflected properly in the database, or none are.
- **Consistency**. Execution of a transaction in isolation (that is, with no other transaction executing concurrently) preserves the consistency of the database.
- **Isolation**. Even though multiple transactions may execute concurrently, the system guarantees that, for every pair of transactions T_i and T_j, it appears to T_i that either T_j finished execution before T_i started or T_j started execution after T_i finished. Thus, each transaction is unaware of other transactions executing concurrently in the system.
- **Durability**. After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

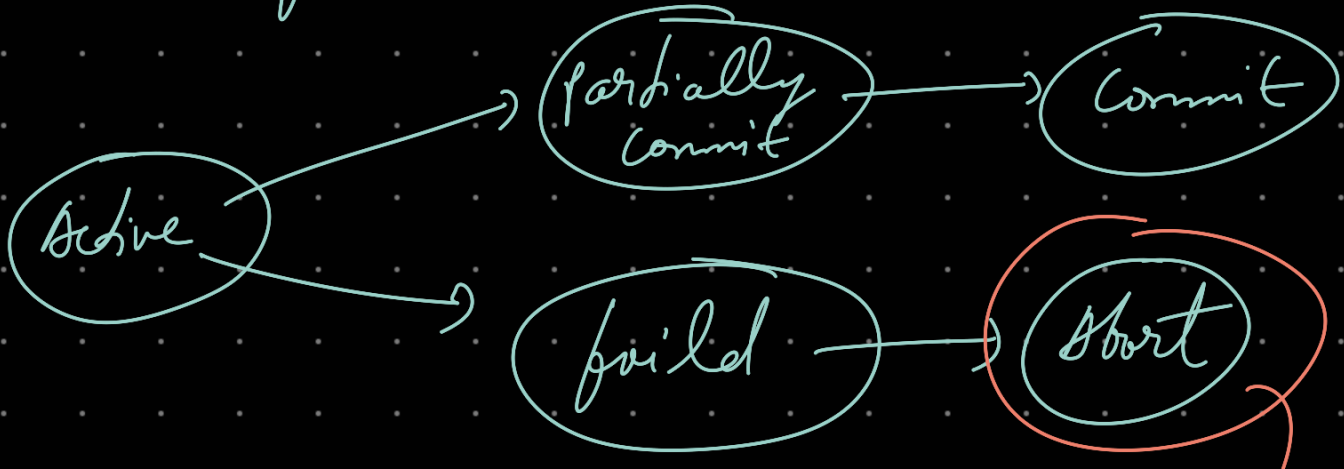
Atomicity -> Transaction manager

Consistency -> Application programmer

Isolation -> Concurrency manager

Durability -> Rollback manager

[State of Transaction]



Rollback

The motivation for using concurrent execution in a database is essentially the same as the motivation for using **multiprogramming** in an operating system.

When several transactions run concurrently, the isolation property may be violated, resulting in database consistency being destroyed despite the correctness of each individual transaction. In this section, we present the concept of sched-

[Problems with Concurrent Transaction]

* Lost update (Write-Write conflict)
* Dirty read (Write-Read conflict)
* Unrepeatable read (Read-write)
* Phantom tuple
* Incorrect summary