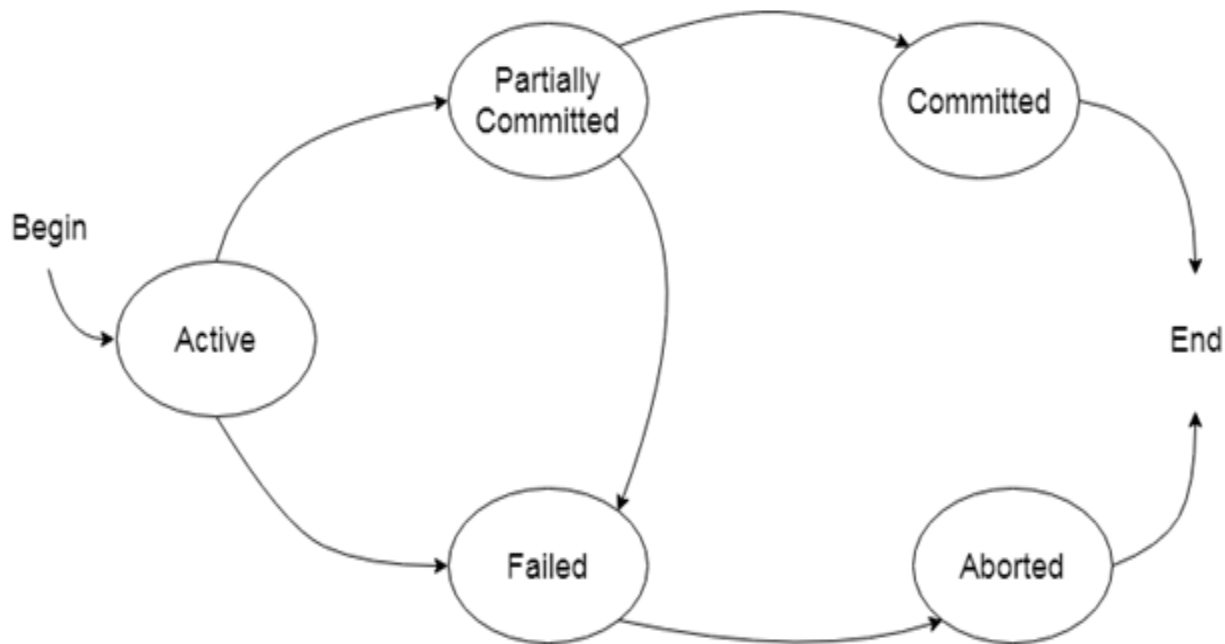


Notes for M.sc(computer science) 2<sup>nd</sup> sem  
on  
Transaction State, ACID Property(unit 4)

Rubhi Gupta  
M.Sc (Computer Science)  
SOMMAS, Jiwaji University,  
Gwalior-474011

# States of Transaction

- In a database, the transaction can be in one of the following states -



## Active state

- ▶ The active state is the first state of every transaction. In this state, the transaction is being executed.
- ▶ For example: Insertion or deletion or updating a record is done here. But all the records are still not saved to the database.

## Partially committed

- ▶ In the partially committed state, a transaction executes its final operation, but the data is still not saved to the database.
- ▶ In the total mark calculation example, a final display of the total marks step is executed in this state.

# Committed

- ▶ A transaction is said to be in a committed state if it executes all its operations successfully. In this state, all the effects are now permanently saved on the database system.

# Failed state

- ▶ If any of the checks made by the database recovery system fails, then the transaction is said to be in the failed state.
- ▶ In the example of total mark calculation, if the database is not able to fire a query to fetch the marks, then the transaction will fail to execute.

# Aborted

- ▶ If any of the checks fail and the transaction has reached a failed state then the database recovery system will make sure that the database is in its previous consistent state. If not then it will abort or roll back the transaction to bring the database into a consistent state.
- ▶ If the transaction fails in the middle of the transaction then before executing the transaction, all the executed transactions are rolled back to its consistent state.
- ▶ After aborting the transaction, the database recovery module will select one of the two operations:
  - ▶ Re-start the transaction
  - ▶ Kill the transaction

# ACID Property

- ▶ A transaction is a very small unit of a program and it may contain several low level tasks. A transaction in a database system must maintain Atomicity, Consistency, Isolation, and Durability – commonly known as ACID properties – in order to ensure accuracy, completeness, and data integrity.
- ▶ **Atomicity** – This property states that a transaction must be treated as an atomic unit, that is, either all of its operations are executed or none. There must be no state in a database where a transaction is left partially completed. States should be defined either before the execution of the transaction or after the execution/abortion/failure of the transaction.

# ACID Property

- ▶ **Consistency** – The database must remain in a consistent state after any transaction. No transaction should have any adverse effect on the data residing in the database. If the database was in a consistent state before the execution of a transaction, it must remain consistent after the execution of the transaction as well.
- ▶ **Durability** – The database should be durable enough to hold all its latest updates even if the system fails or restarts. If a transaction updates a chunk of data in a database and commits, then the database will hold the modified data. If a transaction commits but the system fails before the data could be written on to the disk, then that data will be updated once the system springs back into action.

# ACID Property

- ▶ **Isolation** - In a database system where more than one transaction are being executed simultaneously and in parallel, the property of isolation states that all the transactions will be carried out and executed as if it is the only transaction in the system. No transaction will affect the existence of any other transaction.