A transaction is a unit of program execution that accesses and possibly updates one or more data items in the database.

For example,

Money transfer from an account to another account.
Transaction to transfer Rs 1000 from account A to account B:

1. Read(A)
2. A := A – 500
3. Write(A)
4. Read(B)
5. B := B + 500
6. Write(B)

Two main issues to deal in transaction:

1. Failures of various kinds, such as hardware failures and system crashes
2. Concurrent execution of multiple transactions

A transction must follow ACID properties.

Acid properties:

1. Atomicity
2. Consistency
3. Isolation
4. Durability

1. Atomicity

Either all operations of the transaction are properly reflected in the database or none are.

2. Consistency

Execution of a transaction in isolation preserves the consistency of the database.

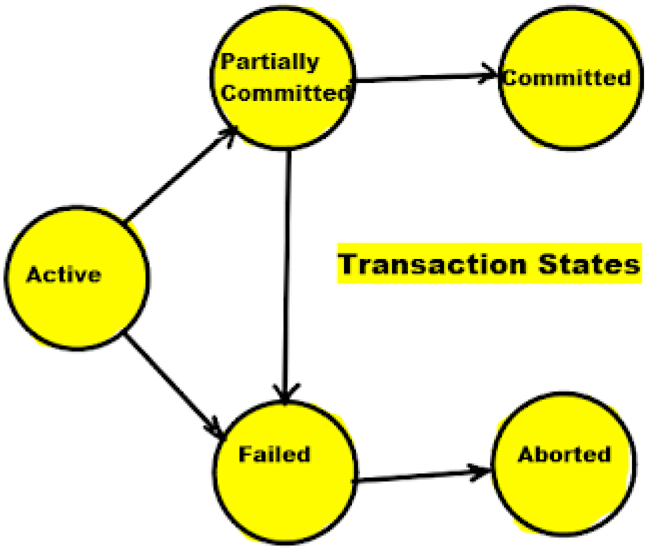
3. Isolation

Although multiple transactions may execute concurrently, each transaction must be unaware of other concurrently executing transactions.

4. Durability

After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

Transaction states:



1. Active

Active, the initial state; the transaction stays in this state while it is executing

2. Partially commited

Partially committed, after the final statement has been executed.

3. Failed

Failed, after the discovery that normal execution can no longer proceed.

4. Aborted

Aborted, after the transaction has been rolled back and the database restored to its state prior to the start of the transaction. Two options after it has been aborted:

- restart the transaction – only if no internal logical error
- kill the transaction

5. Committed

Committed, after successful completion.