What are transaction states?
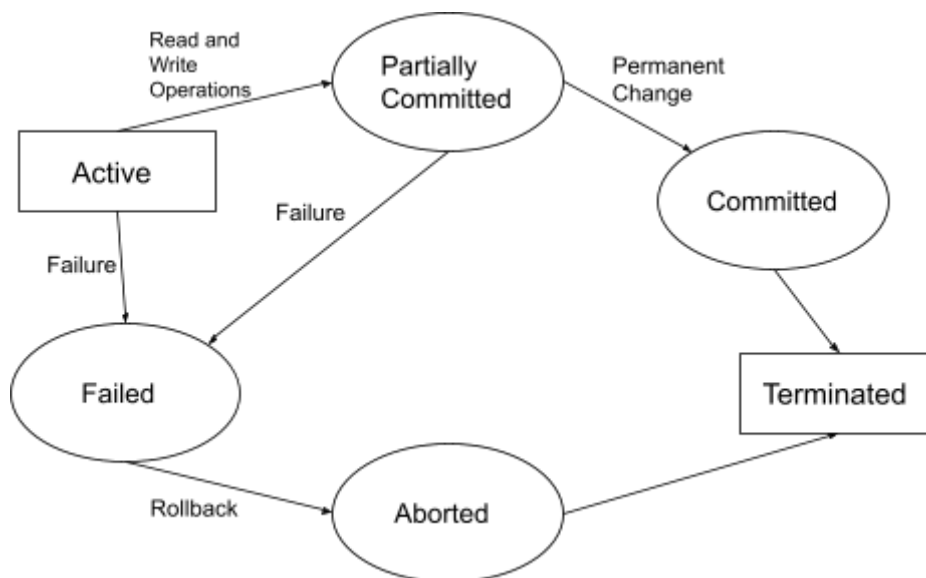
A transaction goes through numerous states throughout its life cycle.
These states are known as transaction states.

The states are:
1. Active state
2. Partially committed state
3. Committed state
4. Failed state
5. Aborted state
6. Terminated state

- **Active State:-** The very first state of the life cycle of transaction, all the read and write operations are being performed and if they execute without any error the transaction comes to 'partially committed' state, although if any error then it leads to 'failed' state.
  Note: All the changes made by the transaction now are stored in the buffer in main memory.

- **Partially Committed State:-** After the last command of the transaction is executed the changes are saved in the buffer in Main Memory. If the changes made are permanent on the Database then the state will transfer to the 'committed' state and if there's any kind of failure it will go to the 'failed' state.

- **Committed Sate:-** When the updates are made permanent on the database. Then the transaction is said to be in Committed state.
  Note: Rollback can't be done from here. At this state, a new consistent state is achieved by the database.

- **Failed State:-** When a transaction is being executed and some failure occurs due to which it becomes impossible to continue the execution. Transaction enters into a failed state. We can come to this state from an Active or Partially committed state.

- **Aborted State:-** From Failed State, when all the changes made in the buffer are reversed, and now that transaction needs to Rollback completely it enters Aborted state.

- **Terminated State:-** This is the final state of the life cycle of the transaction. It is where it ends. A transaction can come to this state only from a 'committed' or 'aborted' state.

After completing the cycle, now the database is consistent and is ready for new transactions.