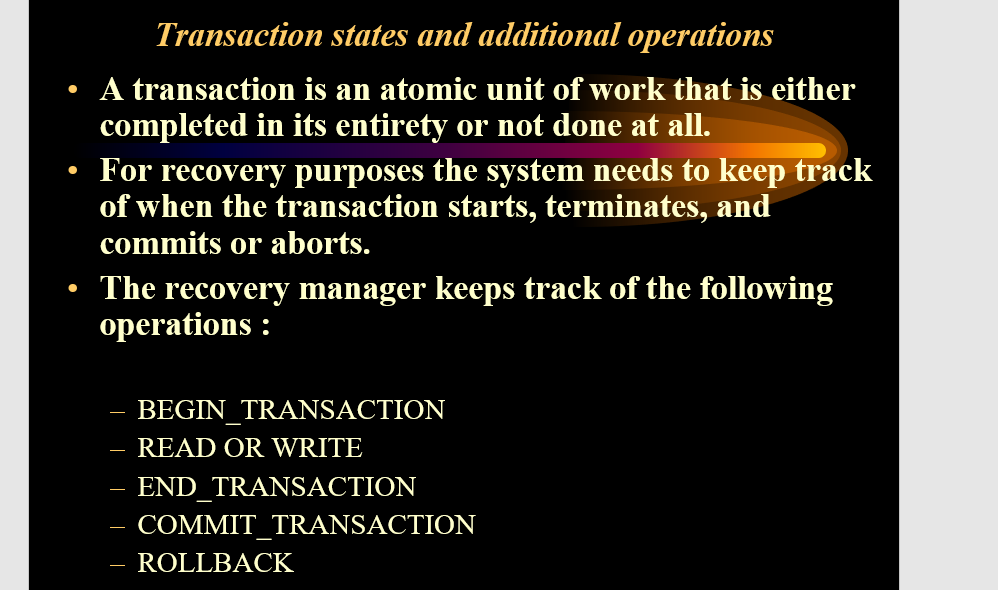
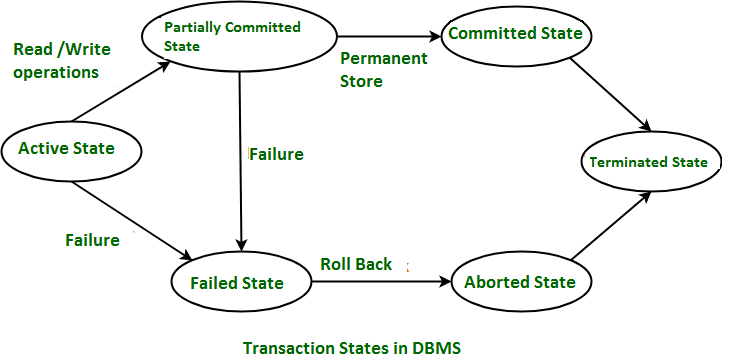
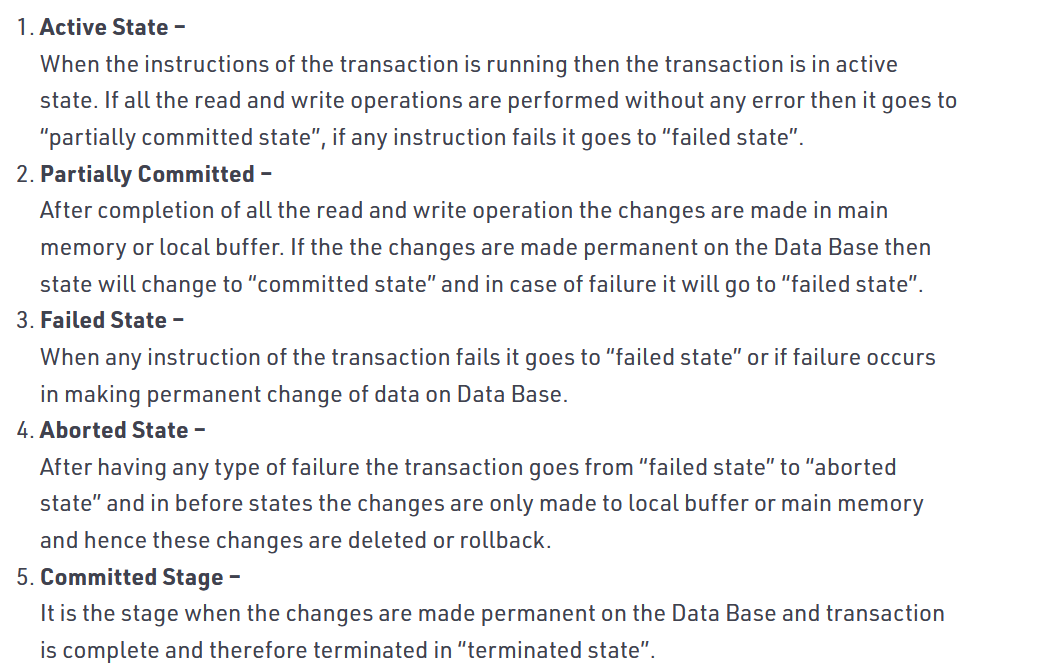
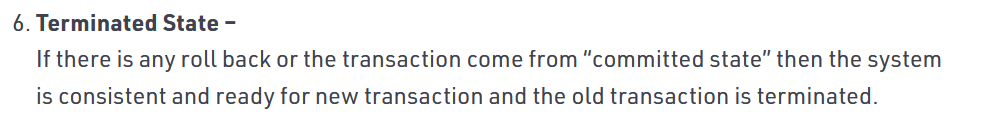
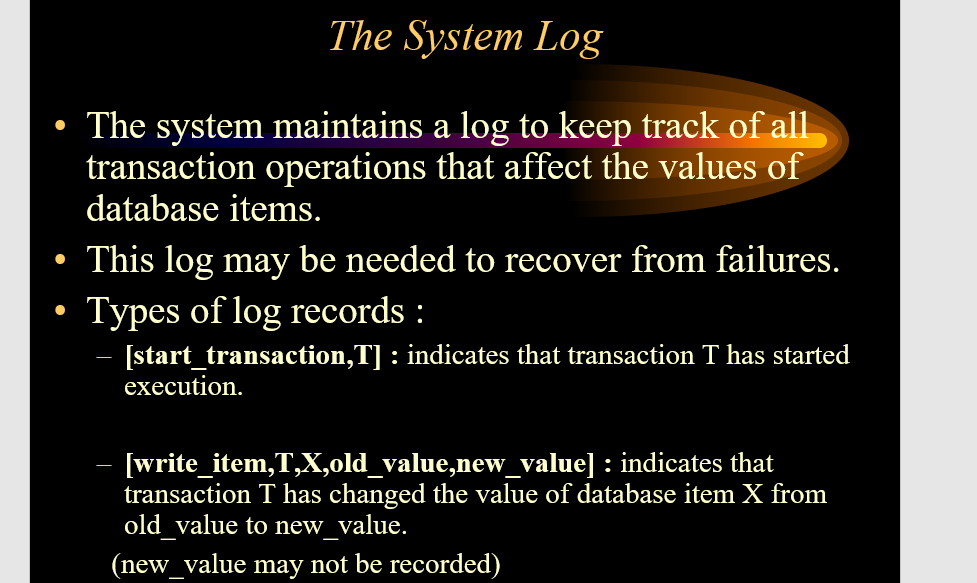
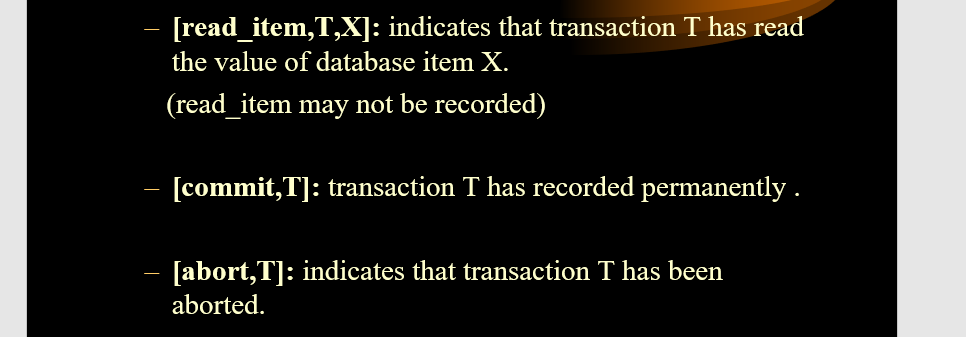
**Transaction Recovery**

**Transaction states in DBMS**

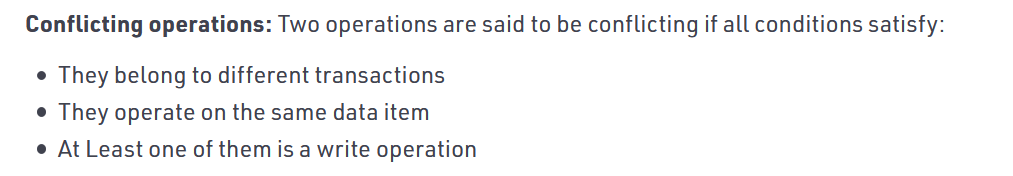
  
  


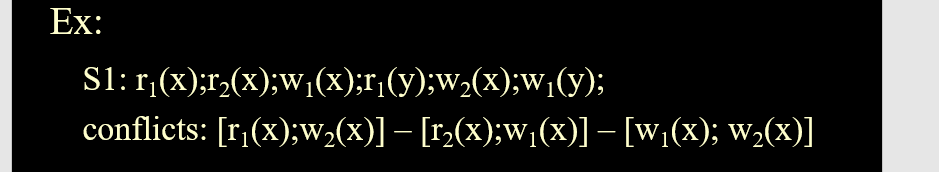
**The System Logs**

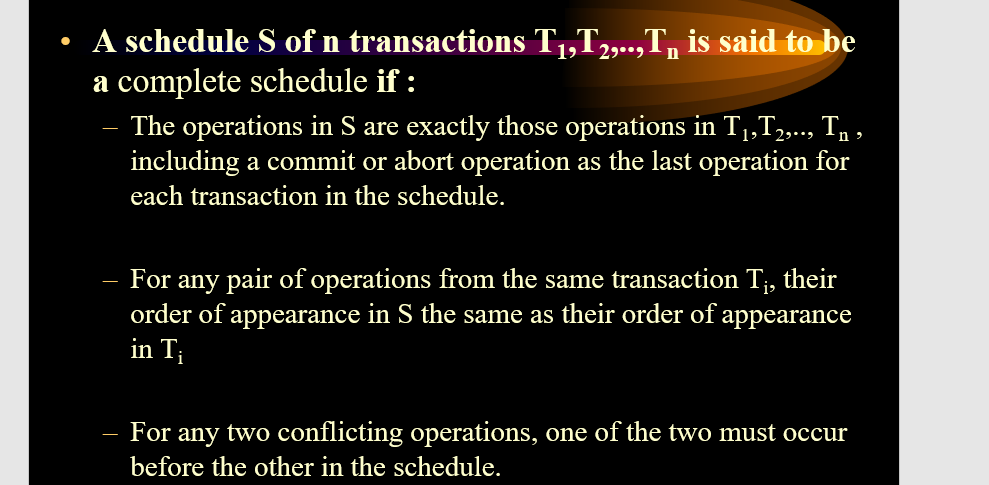
**ACID Properties**

**Schedules of Transaction**





This is a conflict since two operations occurs on the same data-item



**Types of Scheduler**

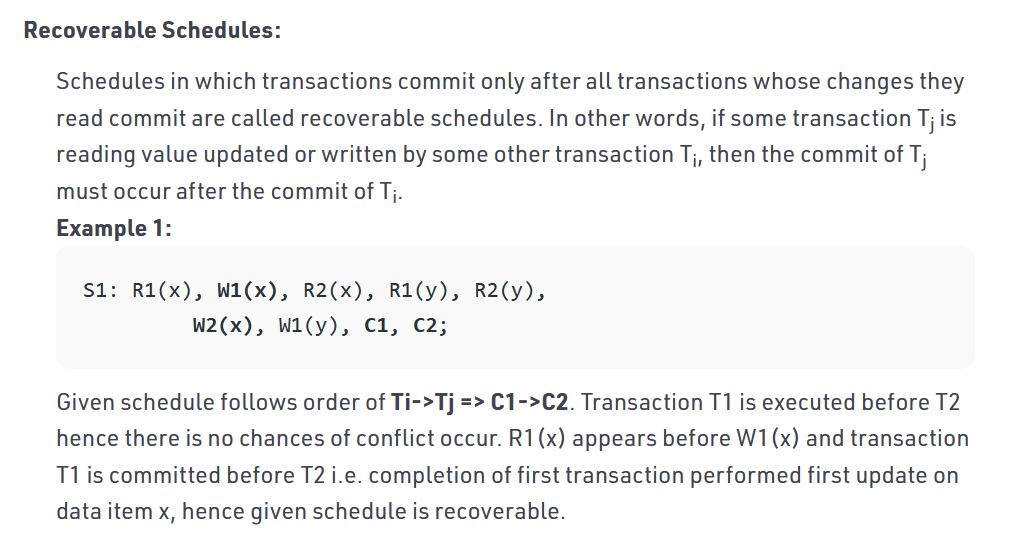
1. **Recoverability 2) Serializability**

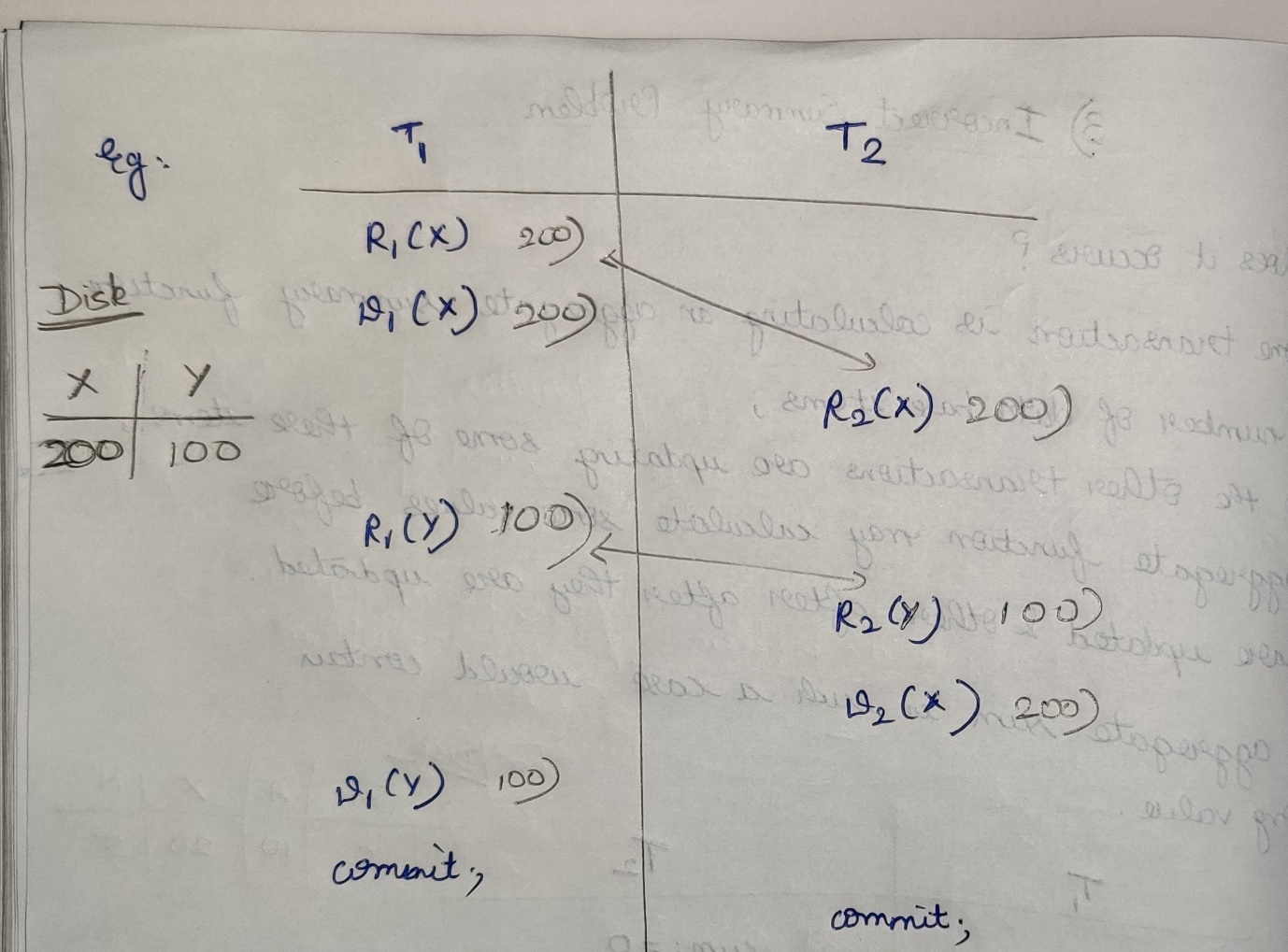
**Recoverability**

1. Recoverable Scheduler and Ir-recoverable Scheduler
2. Cascading Abort and Cascade-less Schedule
3. Strict schedule

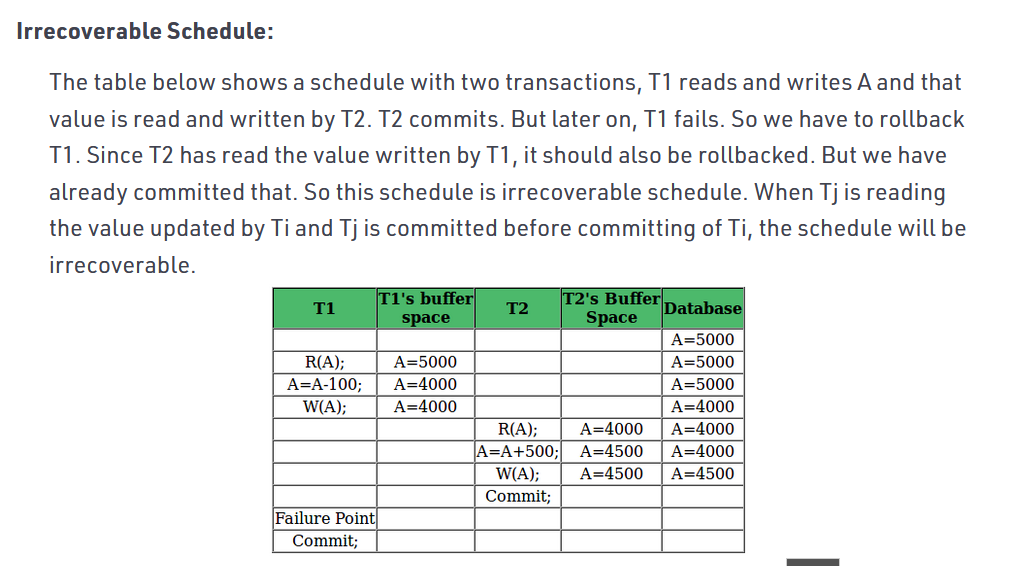
**Recoverability**

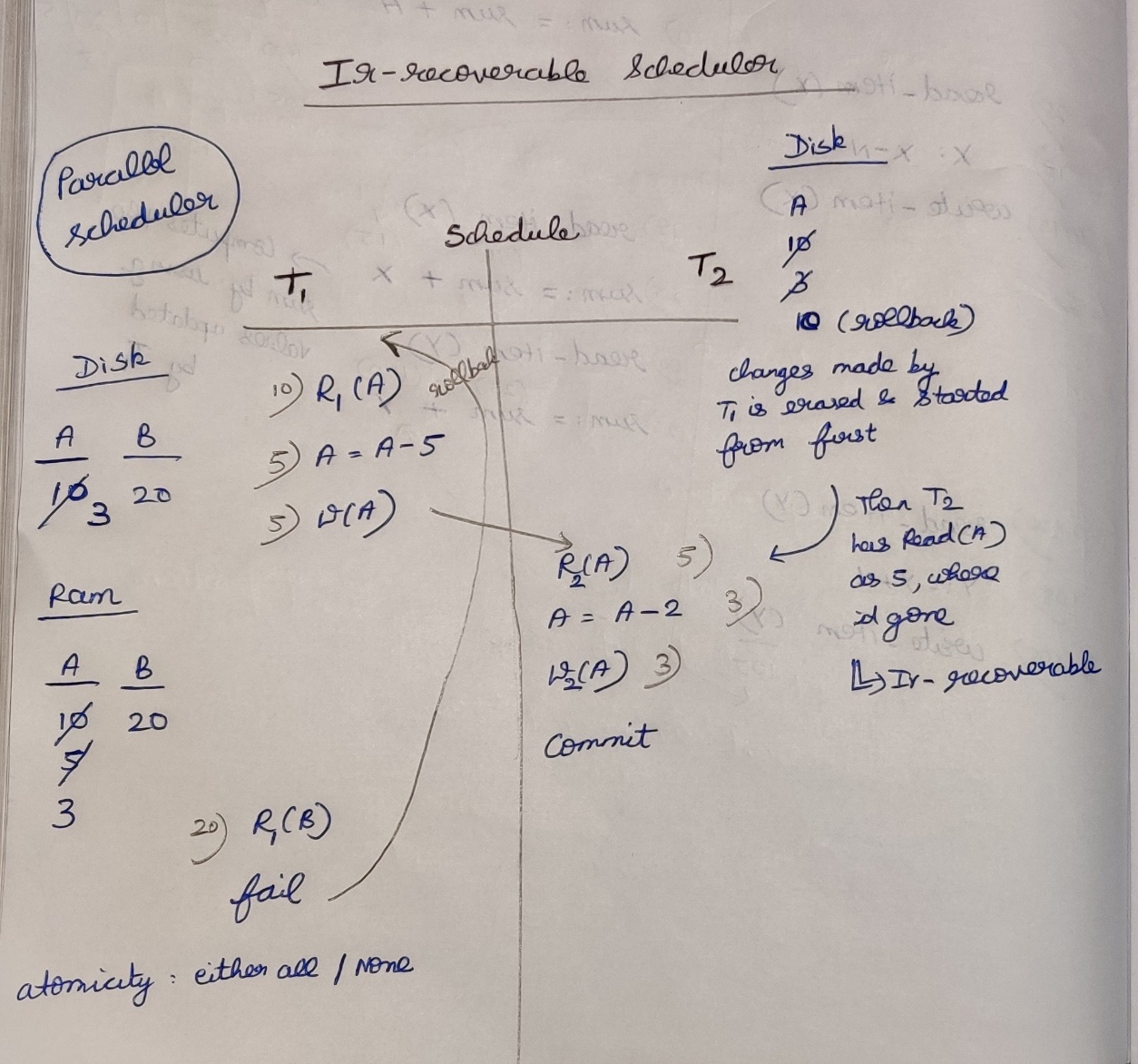
**Recoverable Schedules**

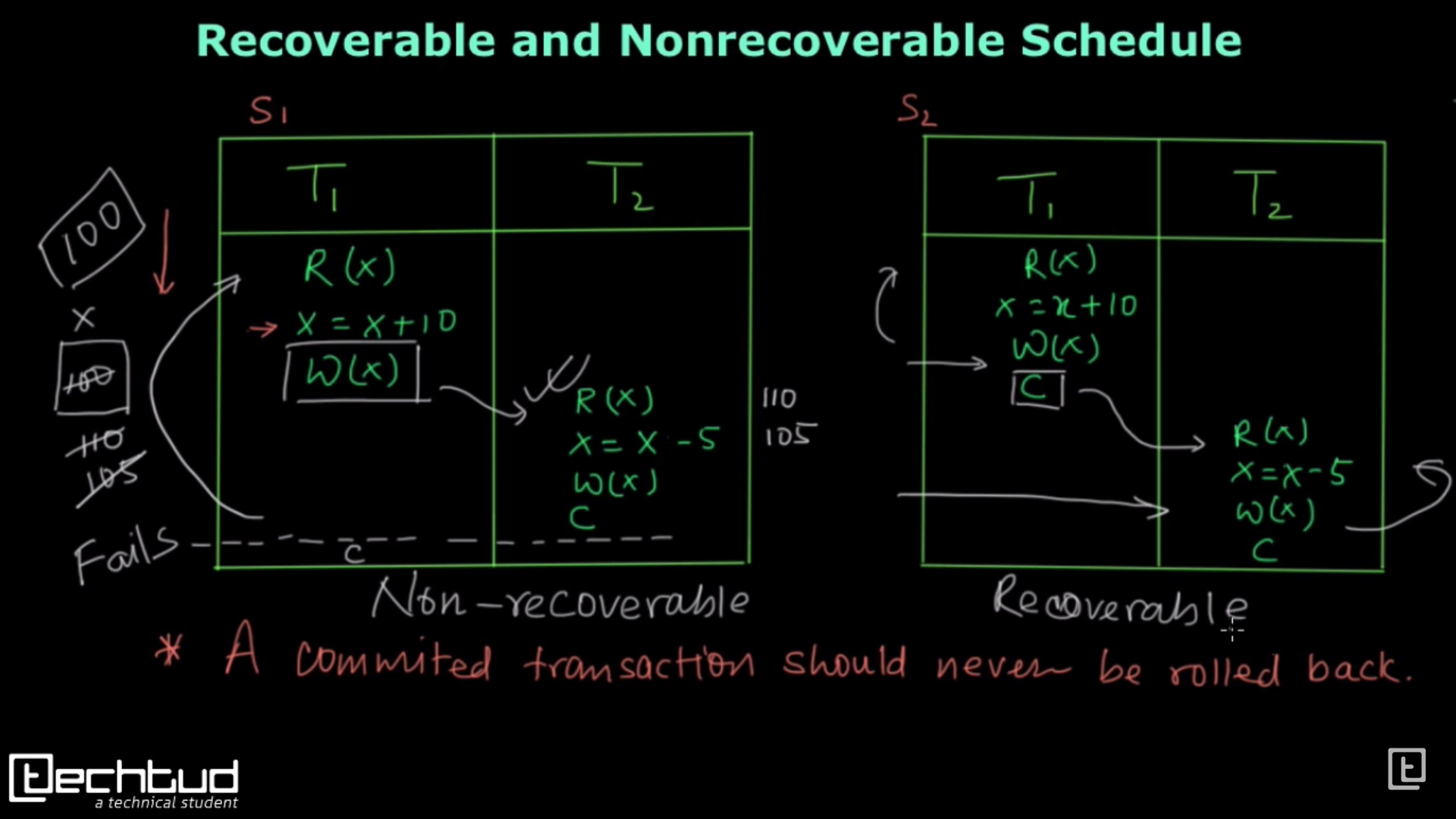




**Ir-recoverable Schedules**

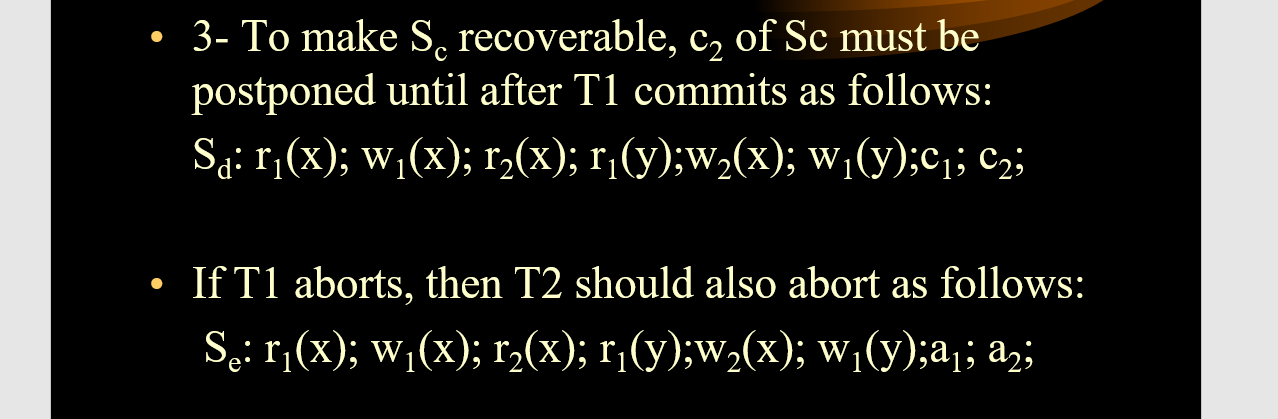
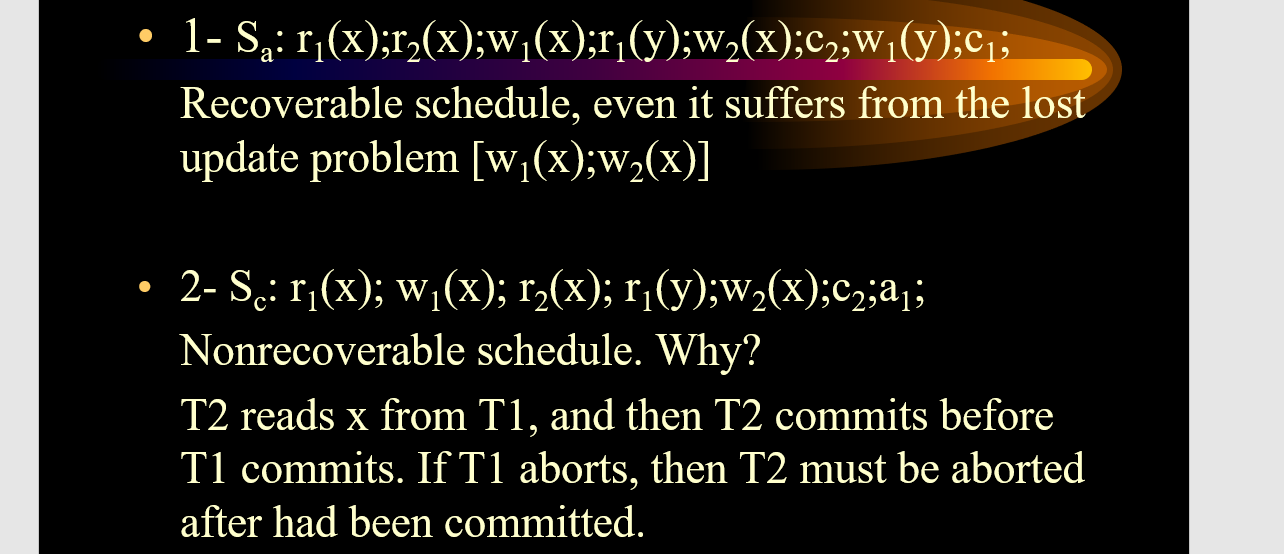




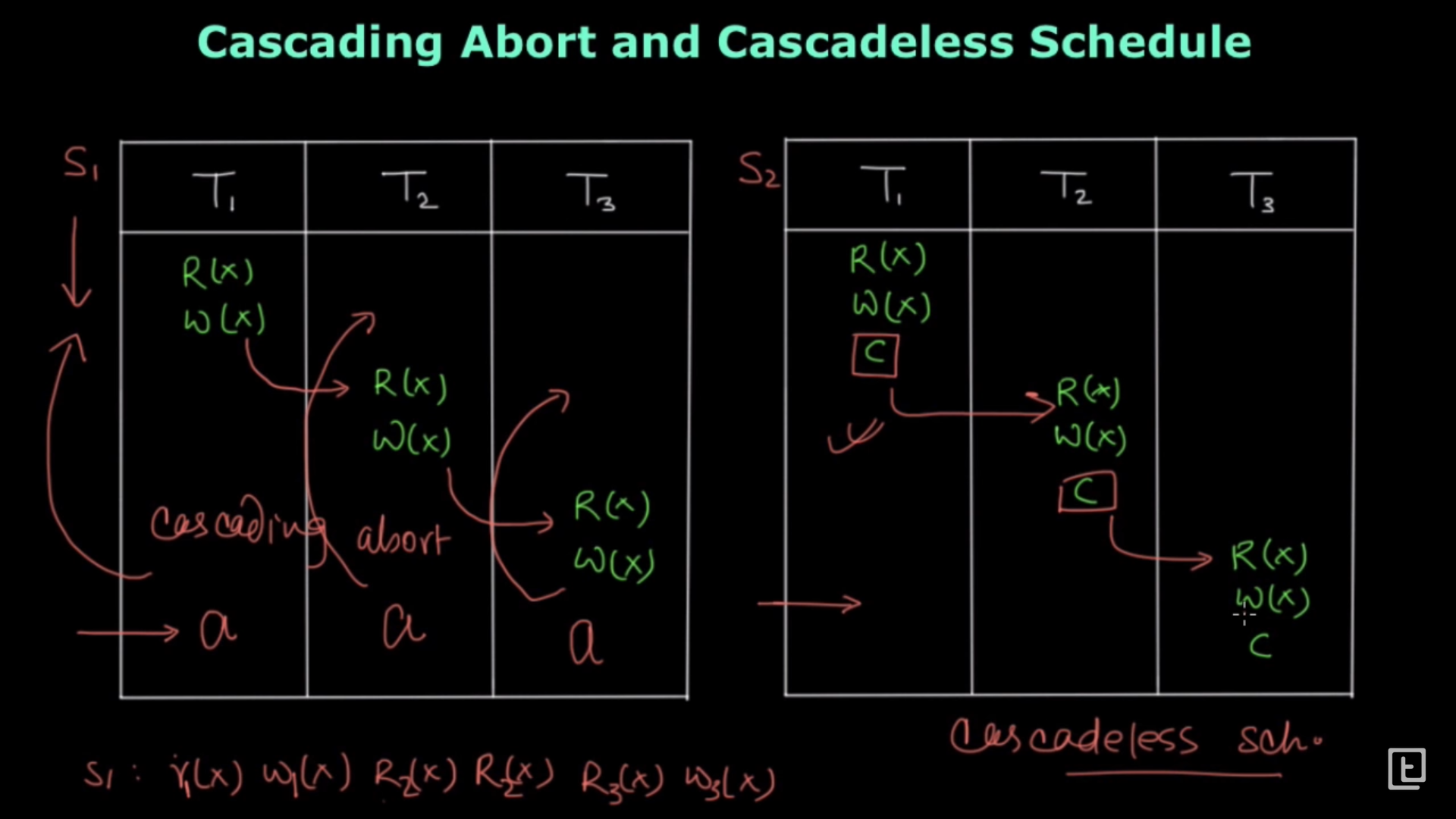


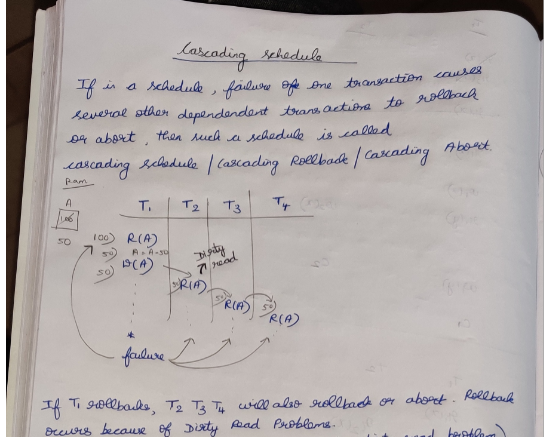
**Non-Recoverable:**  
If any failure happens in T1, then it will rollback so X’s value 🡪 100, but in T2 Read(X) 🡪 110 which lead to data-inconsistent.

**Recoverable:**If any failure happens in T1, then it will roll-back until that T2 will not start its operation.  
If any failure happens in T2, then it will roll-back to T2 itself so there is no effect in T1.

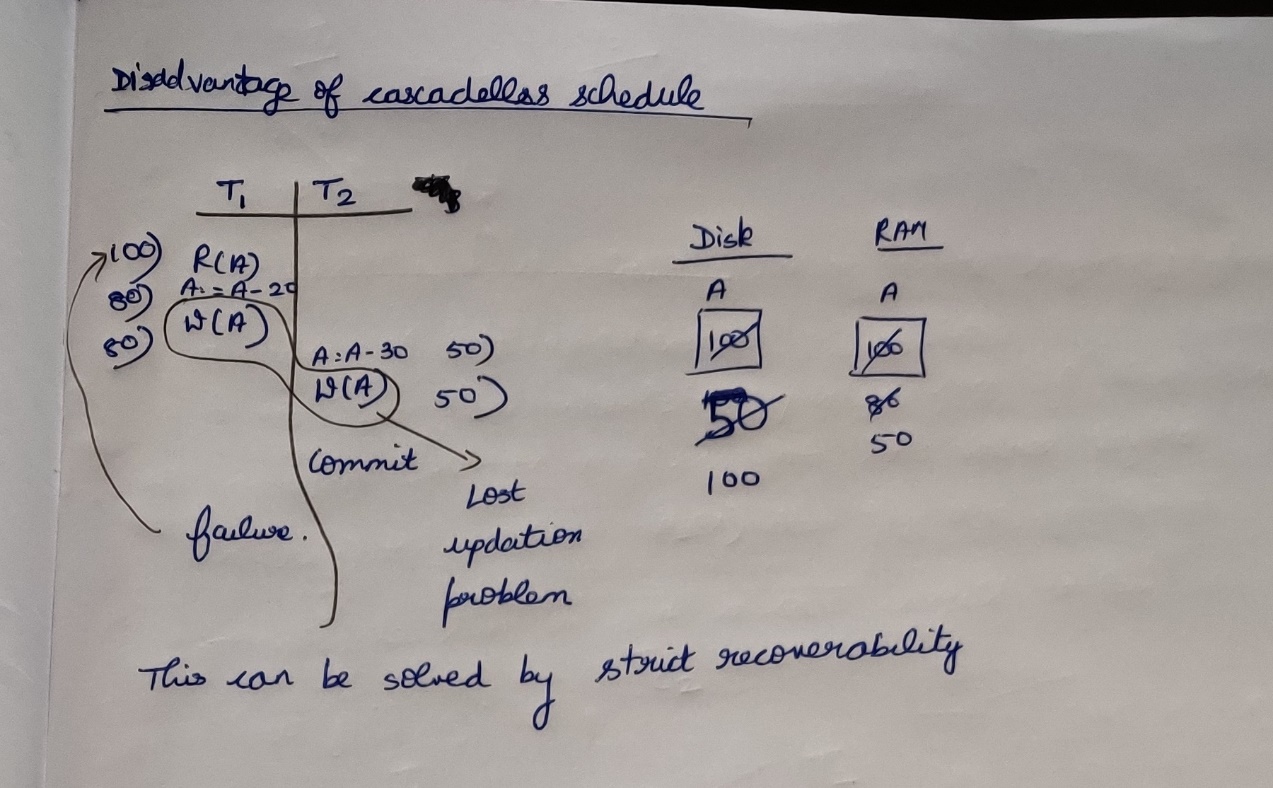
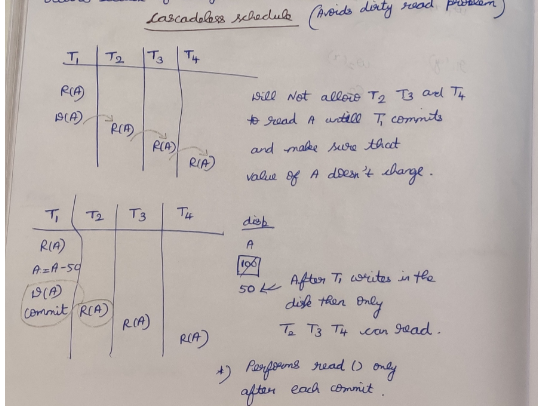


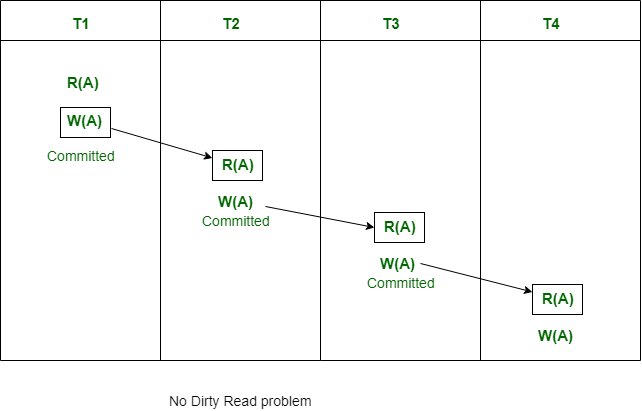
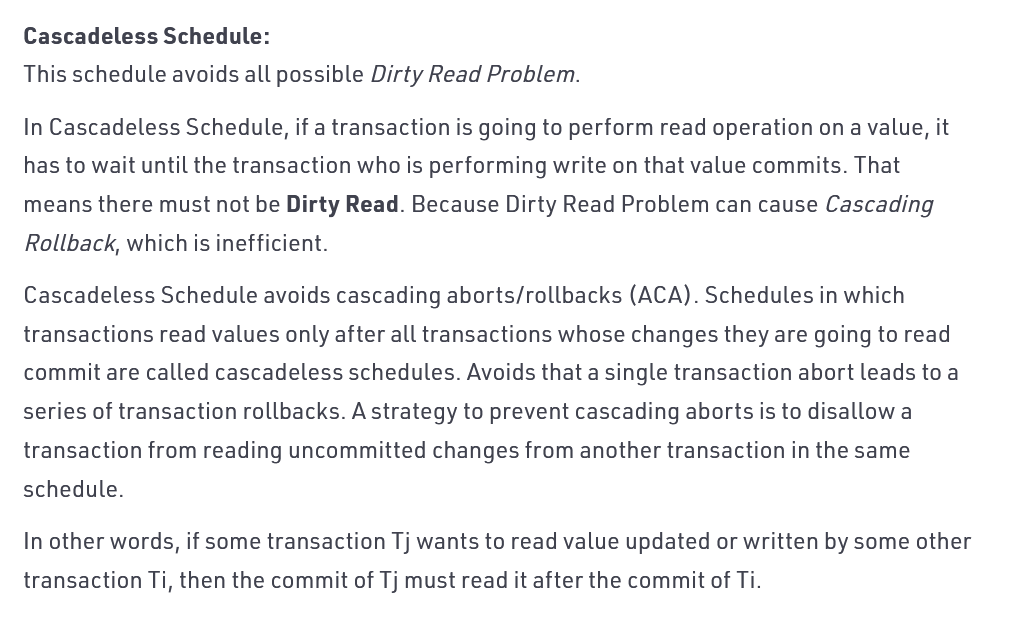
**Cascading Abort**



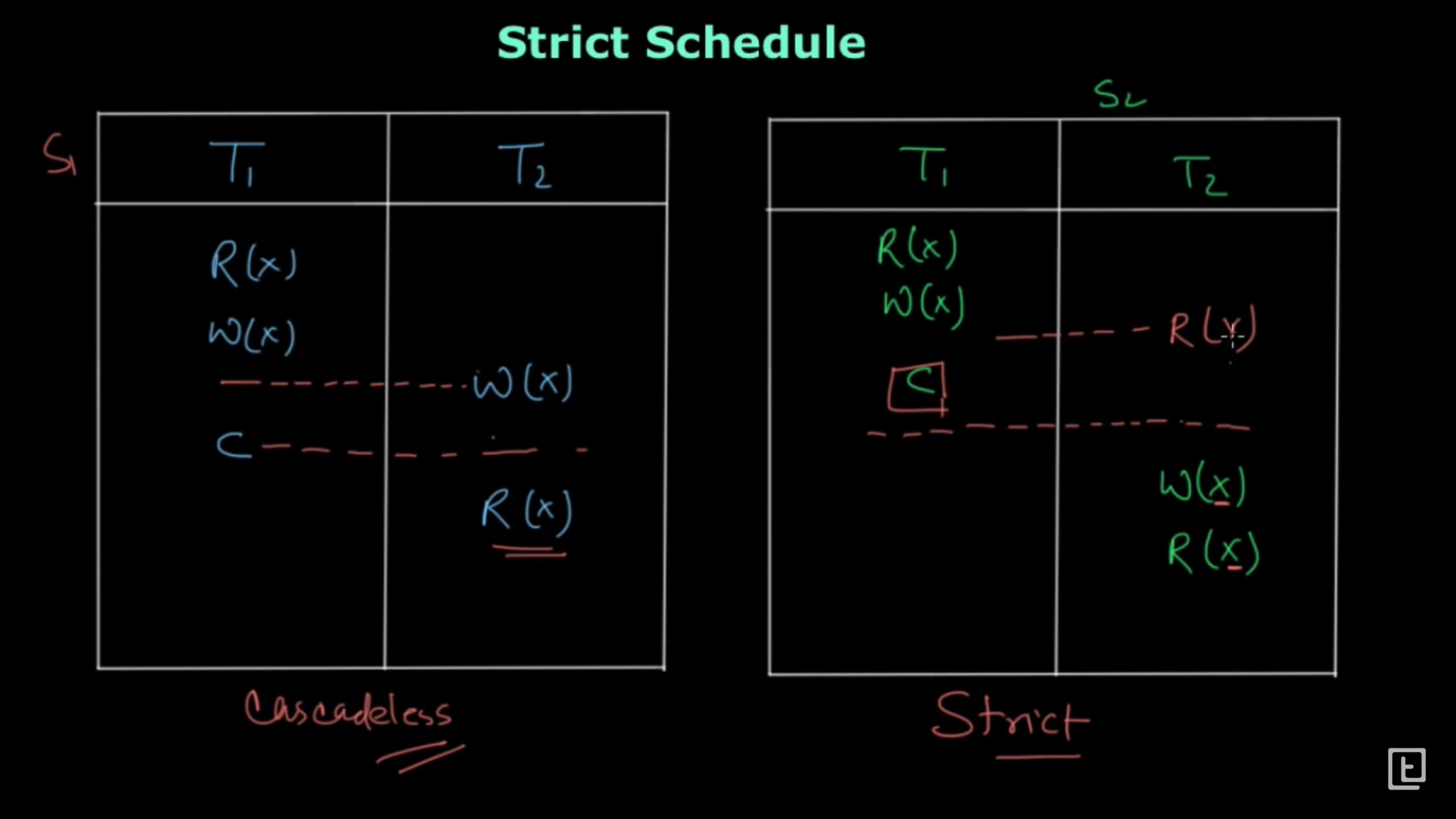


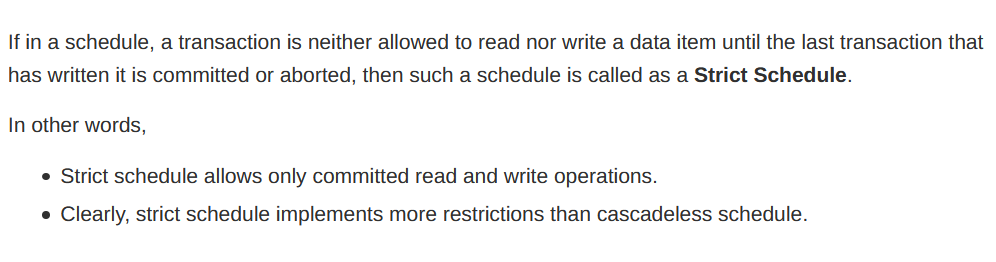
**Cascade less Schedules *(Read is allowed only after committing)***



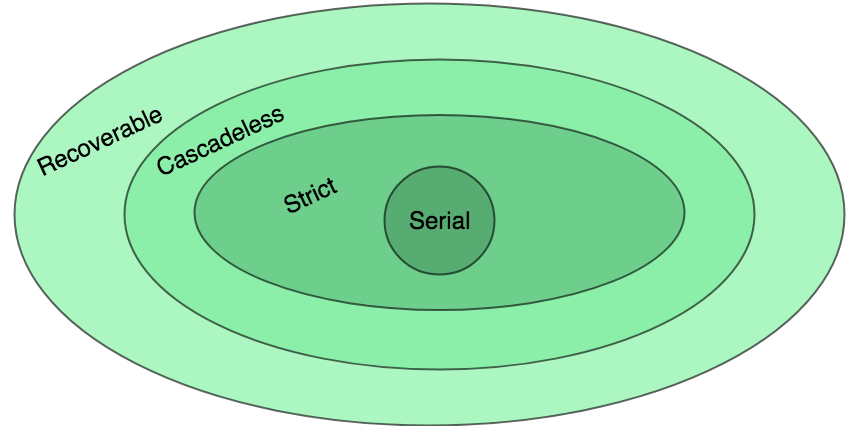


**Strict Schedule**





Single data-item (X) 🡪 strict schedule can also become serial scheduler  
Multiple data-item (X,Y,…) 🡪 strict schedule is not serial scheduler



**Serializability**