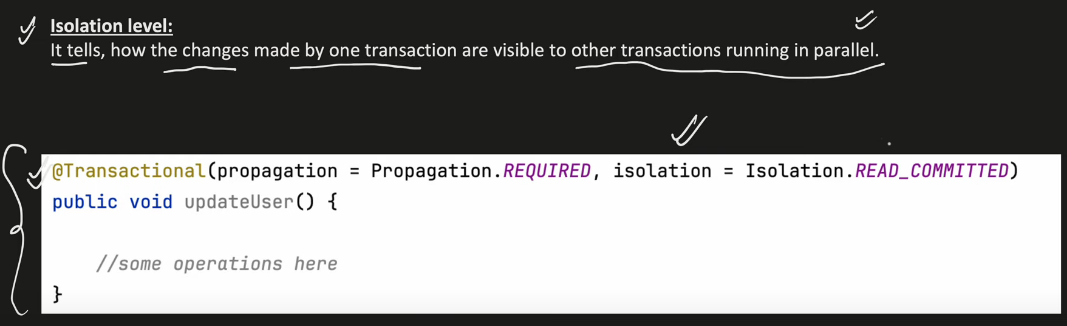
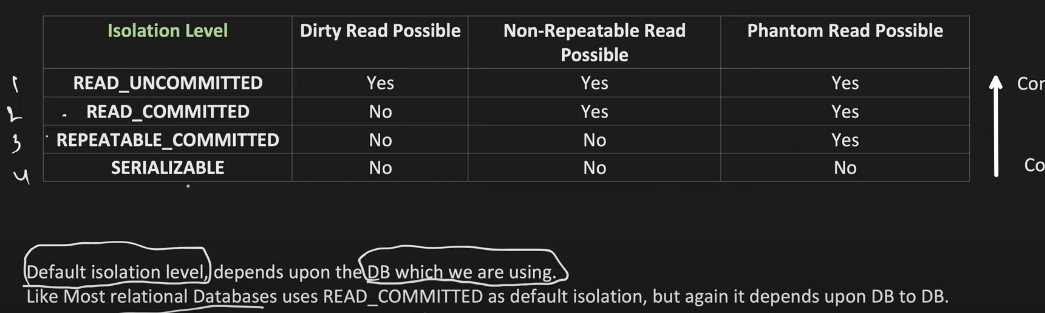
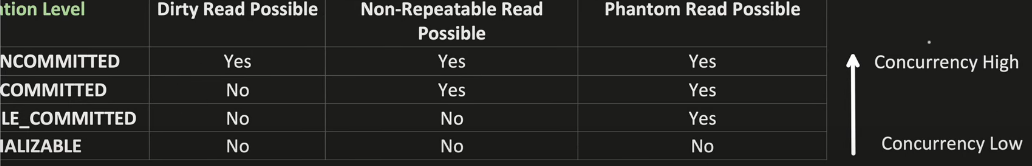
**2.transaction isolation level**





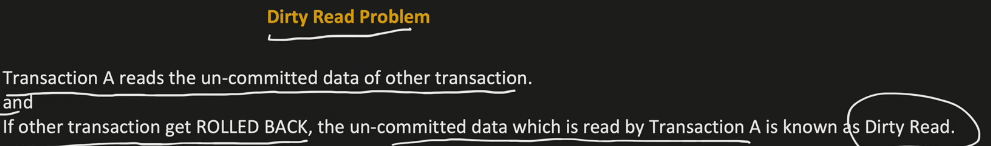


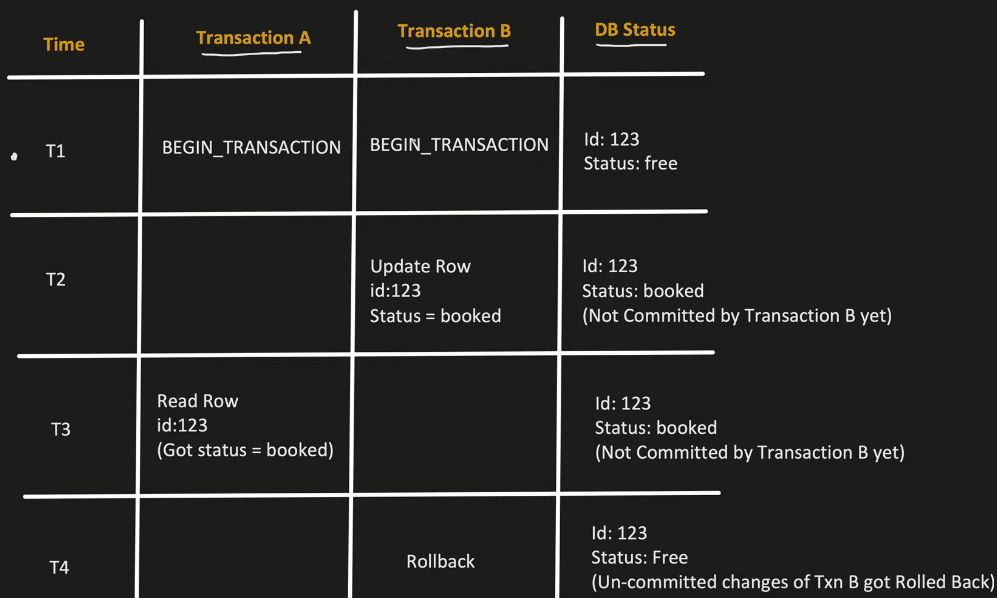
1] Read Uncommitted : You can see changes made by other transactions even if they haven’t been committed.

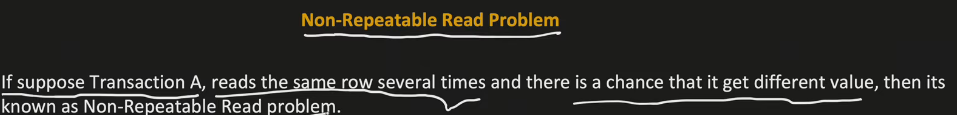
2] Read Committed: You can only see data that has been committed by other transactions. You can’t see uncommitted changes.

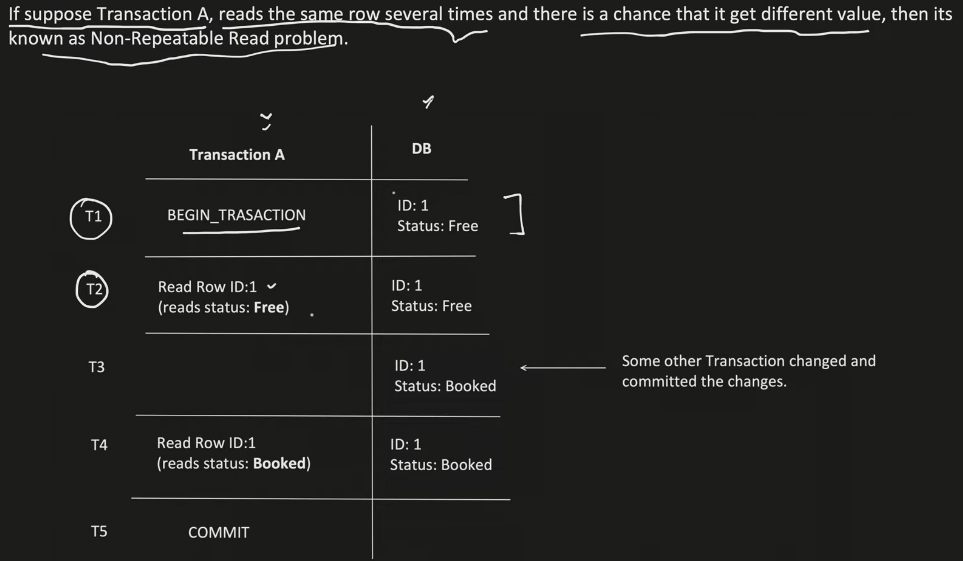
3] Repeatable Read: Once you read a row, the same copy will be used for the duration of your transaction, even if another transaction modifies it.

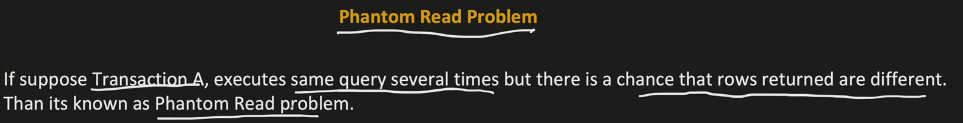
4] Serializable: Transactions are executed in a way that ensures they don’t interfere with each other, as if they ran one after another in sequence.

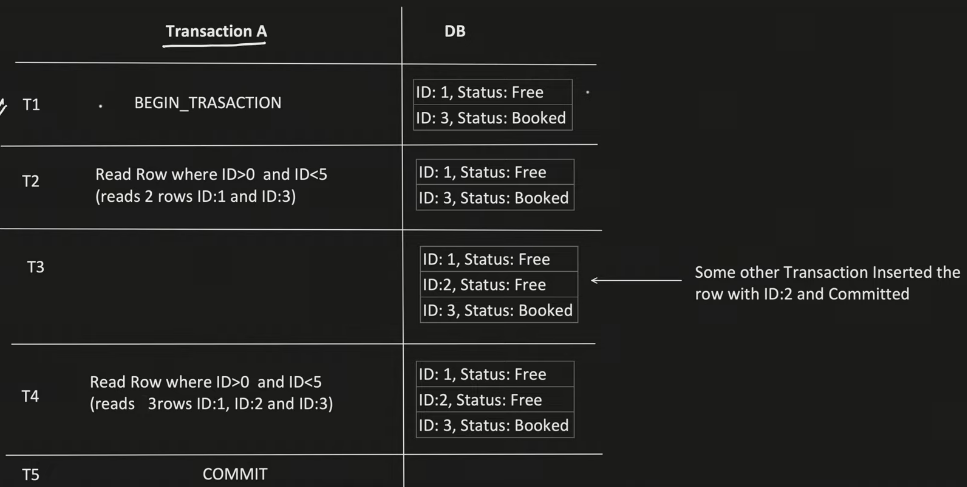


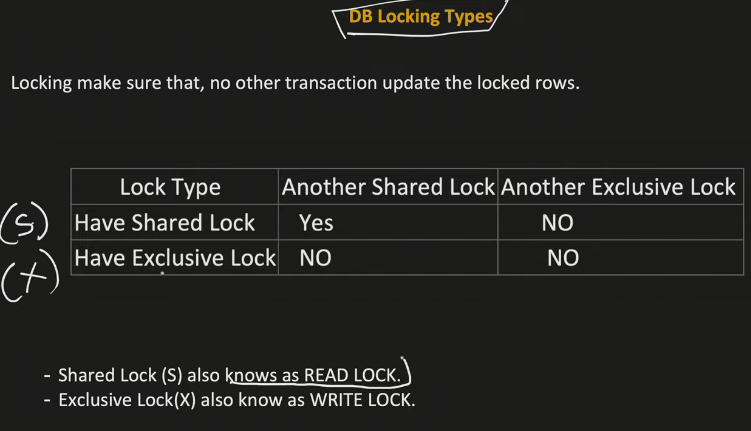


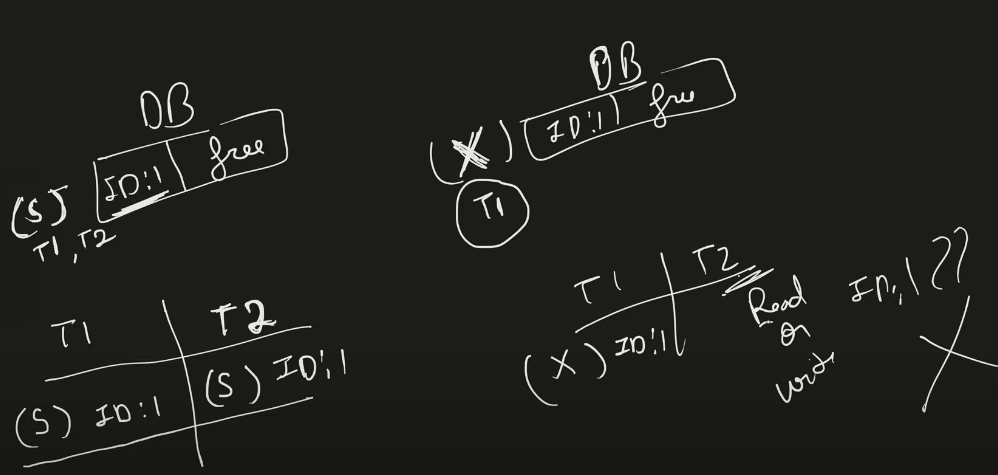






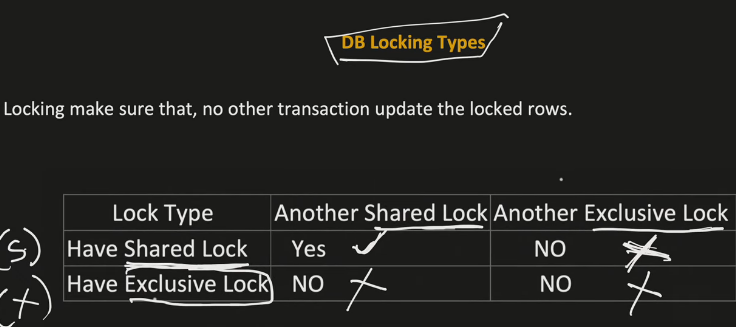






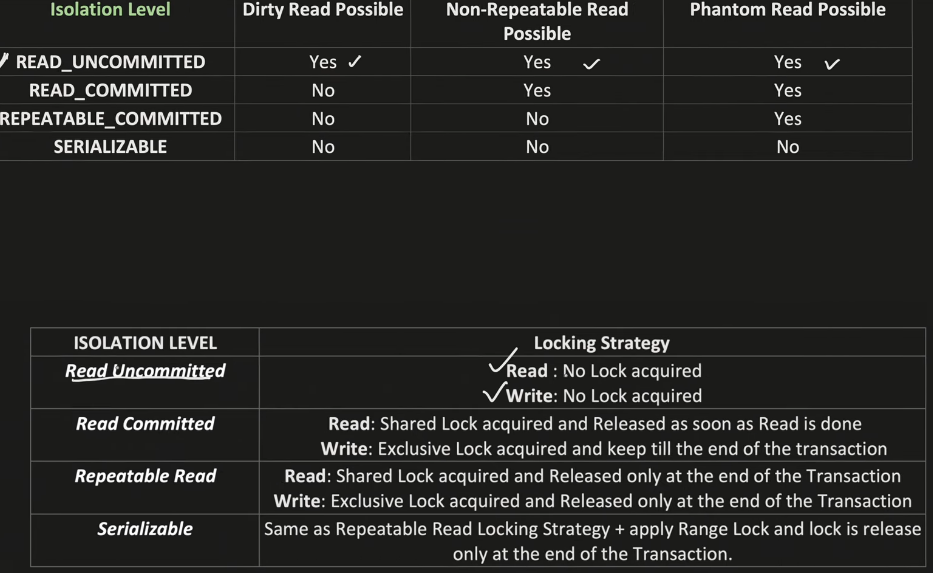
Trans t1 and t2 gets a shared lock by reading row id1. They can read with shared lock but not write.

In exclusive lock, either they can read or write. If one transaction takes exclusive lock on id1 row then no other transaction can get that lock to read or write until that transaction releases the lock on it



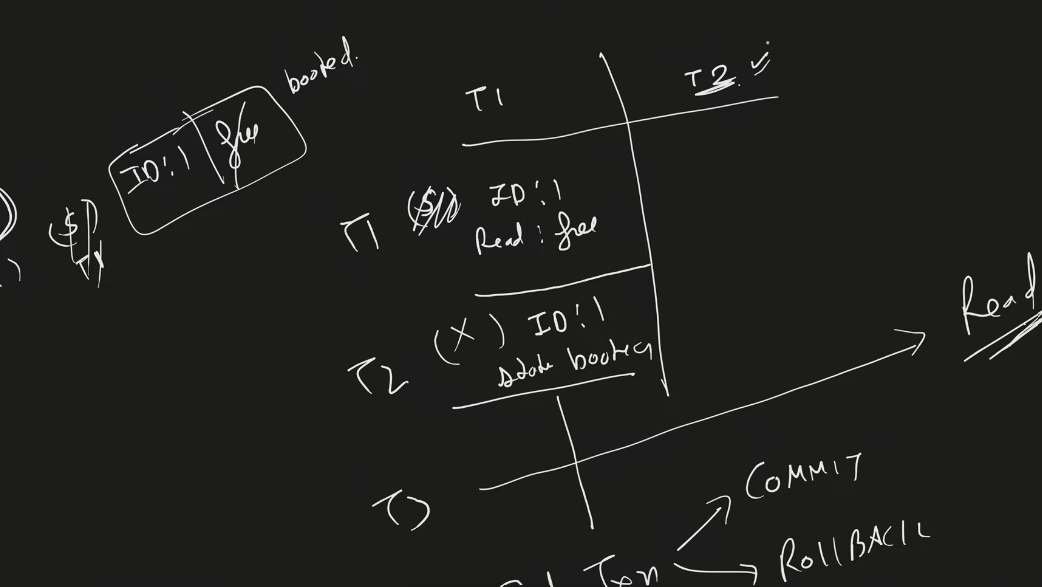
When somebody has taken shared lock, then other transaction cannot take exclusive lock on it but shared lock can be accessed.

When somebody has taken exclusive lock, then other transaction cannot take both shared and exclusive lock



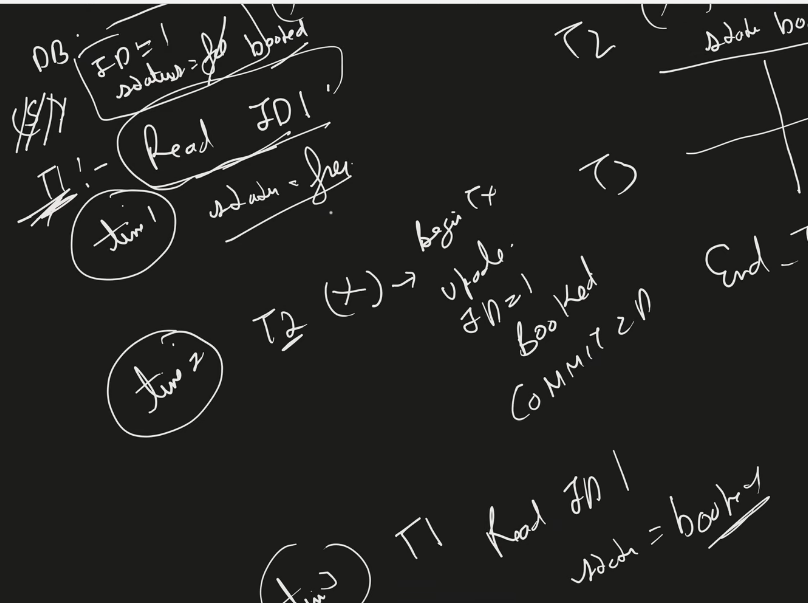
Read uncommited – Read Application, Concurrency High

Read commited: avoid dirty reads

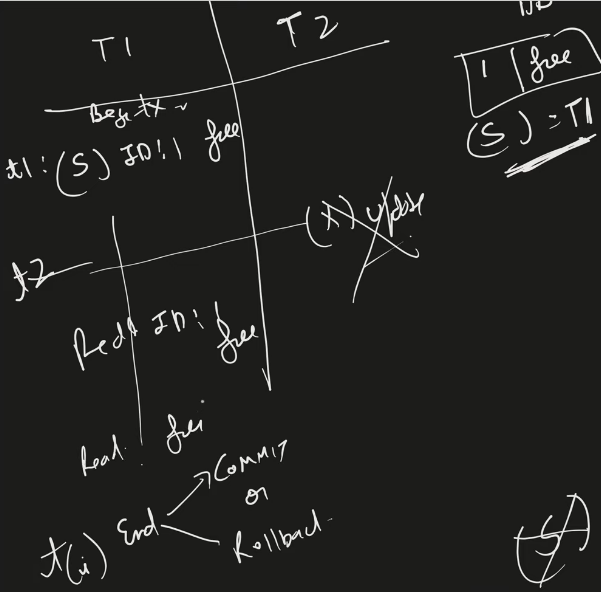


T3 cannot read uncommited data

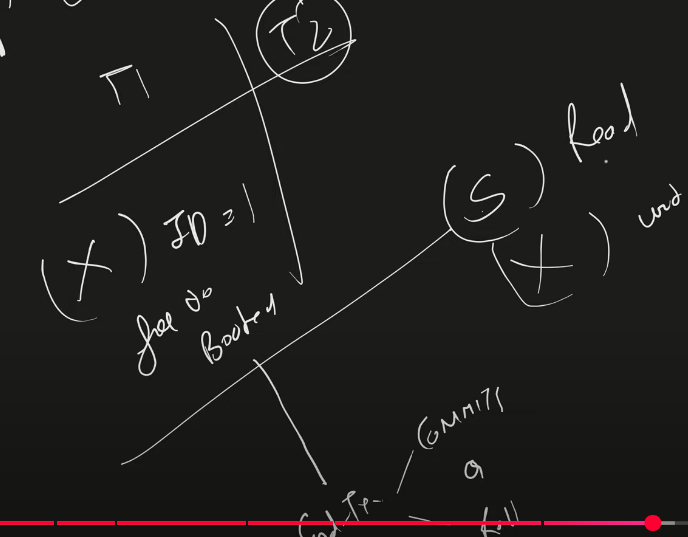
Repeatable read problem still exist

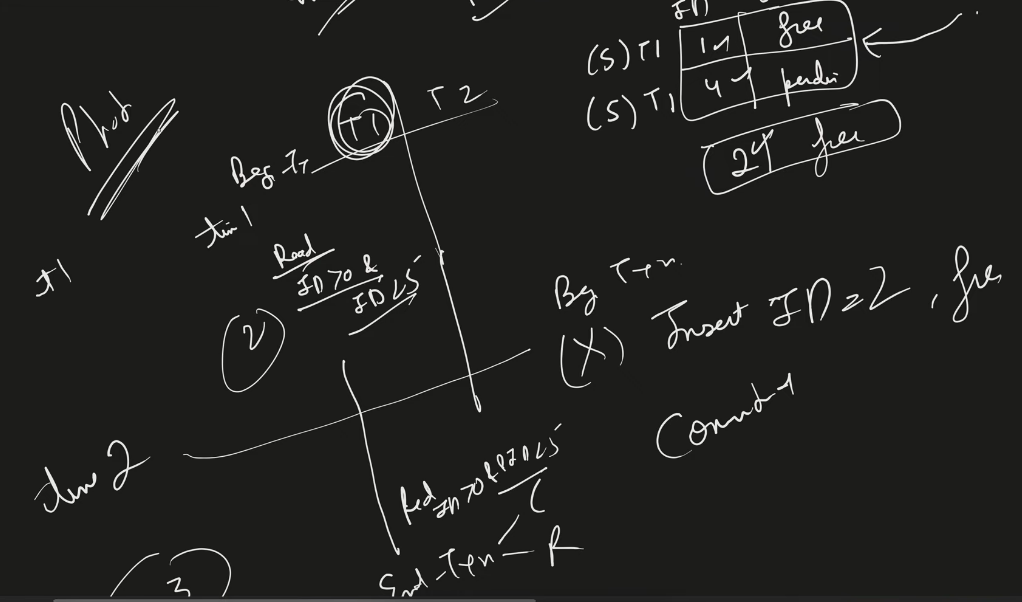


Repeatable Read isolation: no other transactions between this transaction can change it.



The other transaction cannot even take shared or exclusive lock until it releases. It cannot even read it





Serializable – range lock

