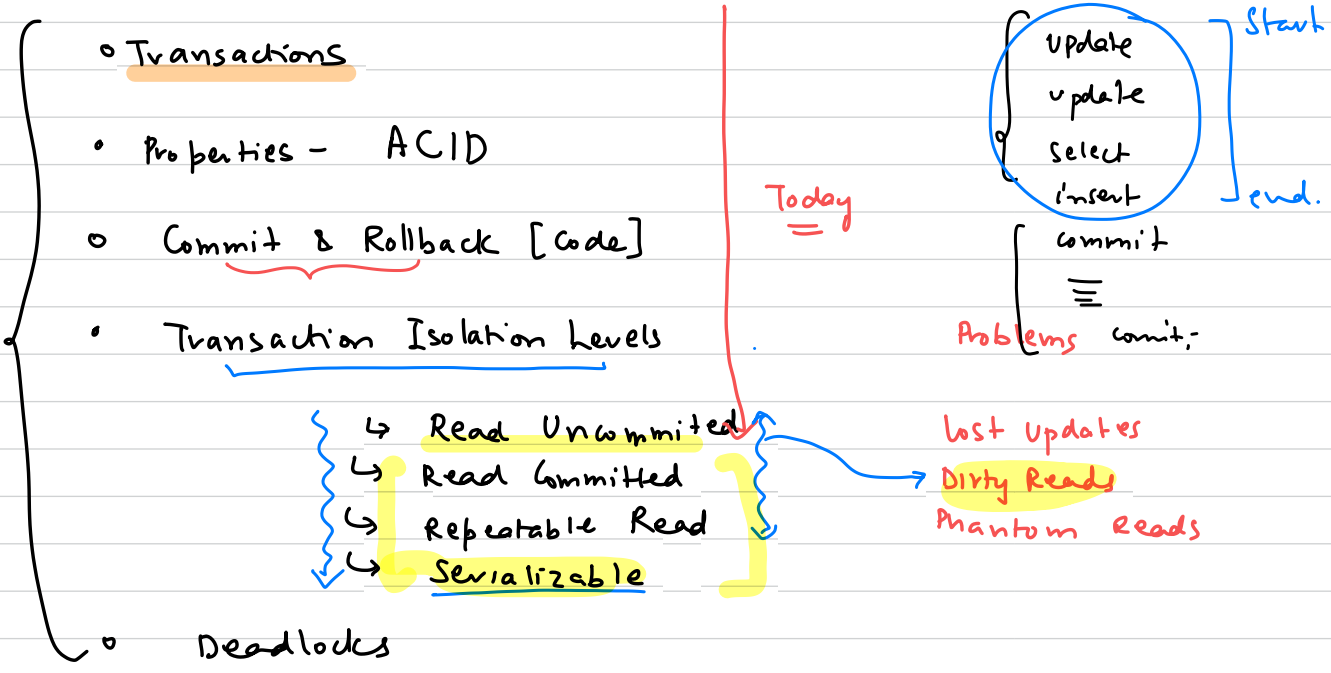
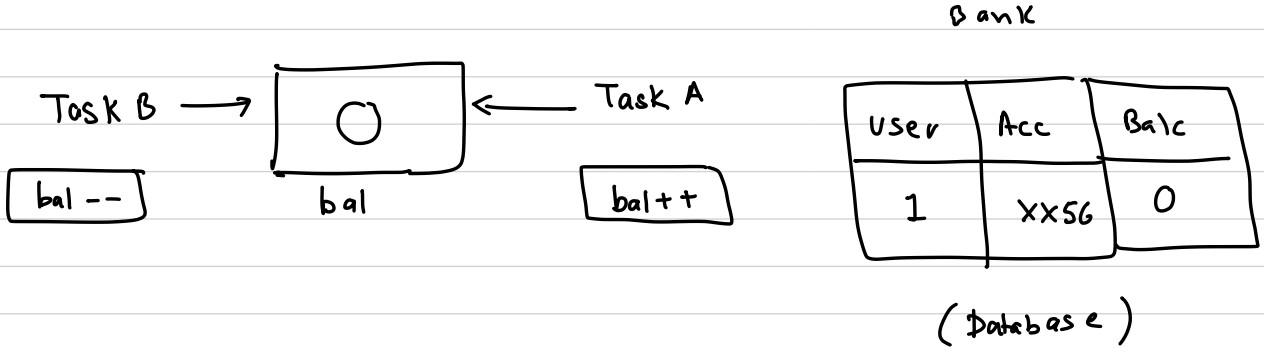
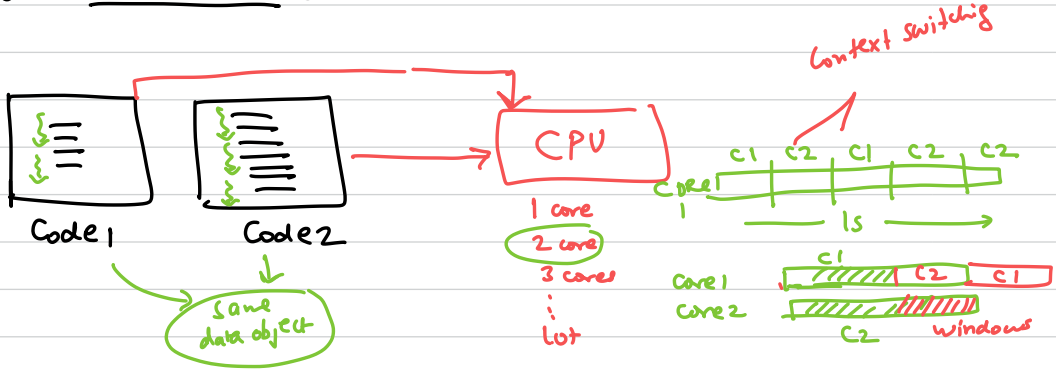
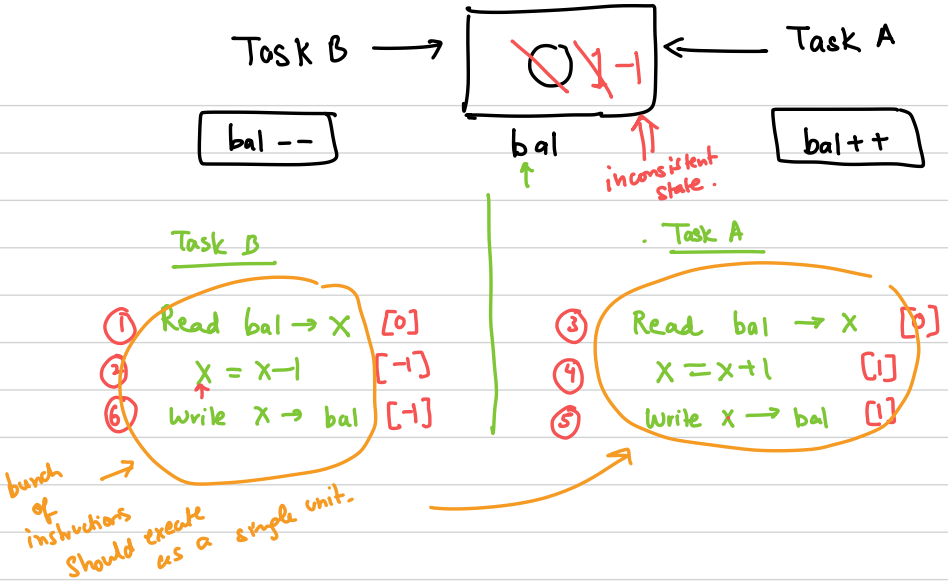
Transactions





Concurrent Execution



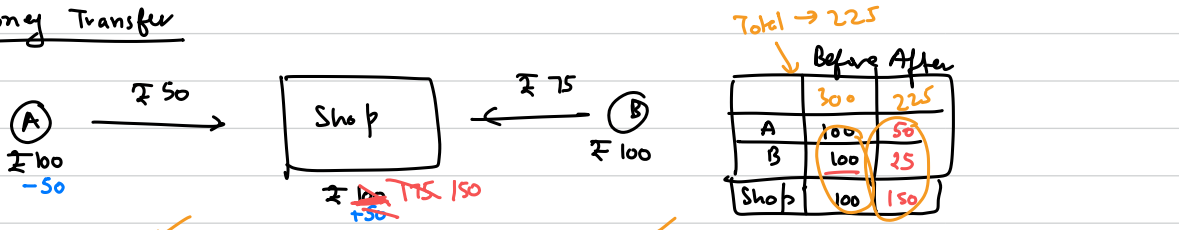


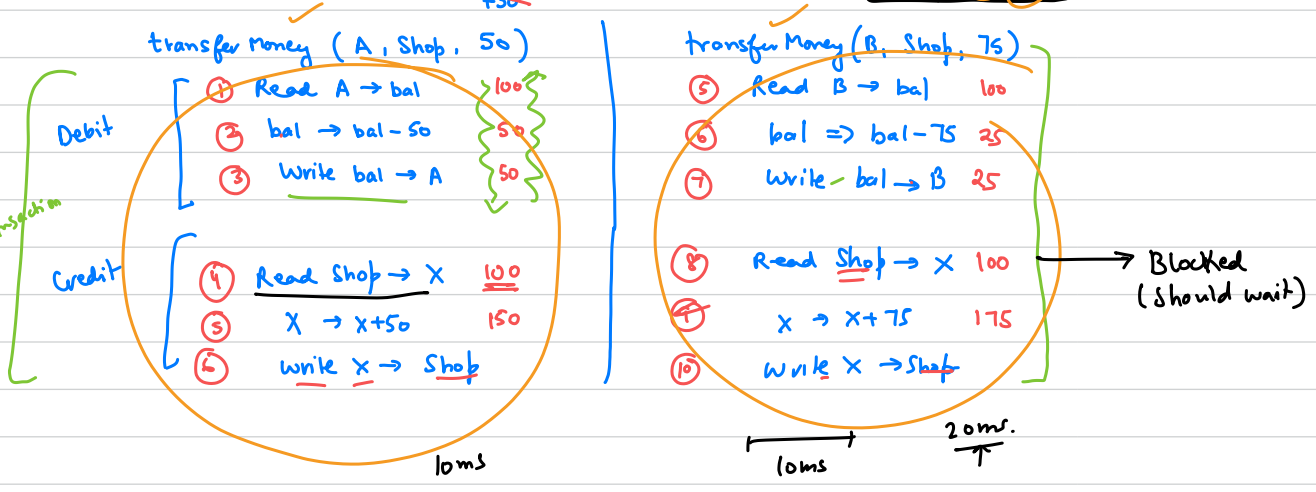
Transaction: set of database operations logically grouped together to perform task.

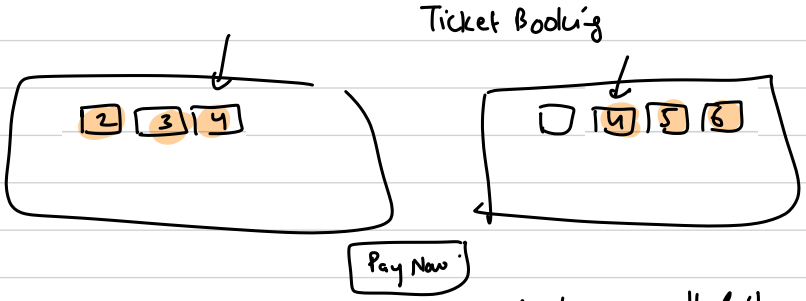
↳ solve the problem of inconsistent state

↳ Complete operation might not execute some times.

Money Transfer







one of them will fail

ACID

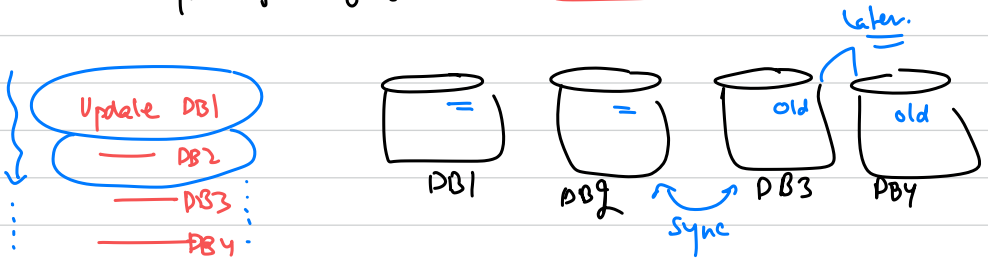
Expectations from a Transaction.

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

(1) Atomicity → To the outside the transn should appear as a single ~~entity~~ operation

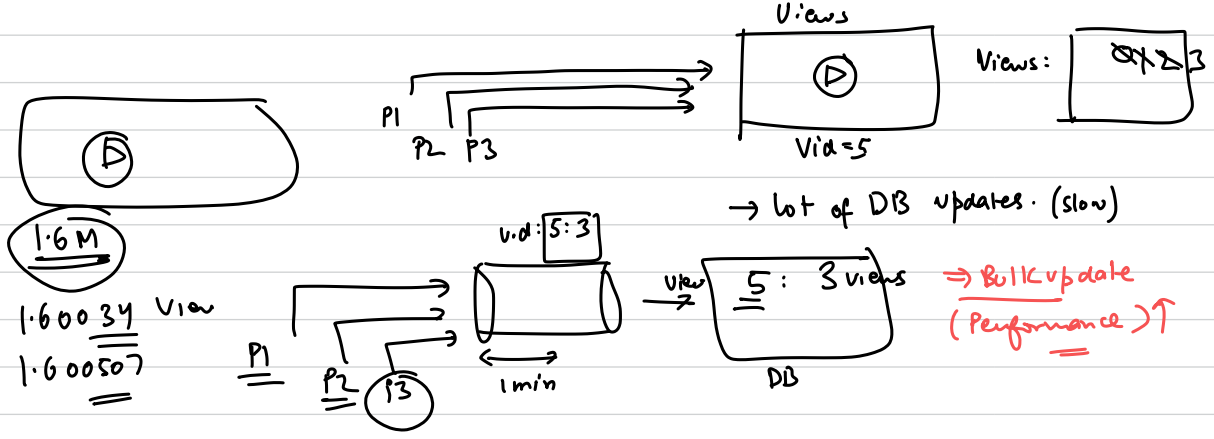
→ All instructions are executed or nothing has happened.

Update your google accnt profile pic



(2) Consistency

→ logical correctness of data..



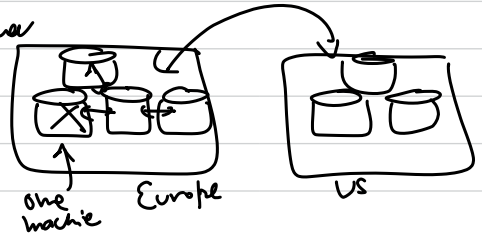
(3) Isolation

Means one transaction shouldn't affect another Transaction running at same time on the same DB in a wrong way.

↳ Transaction Isolations Levels

(4) Durability

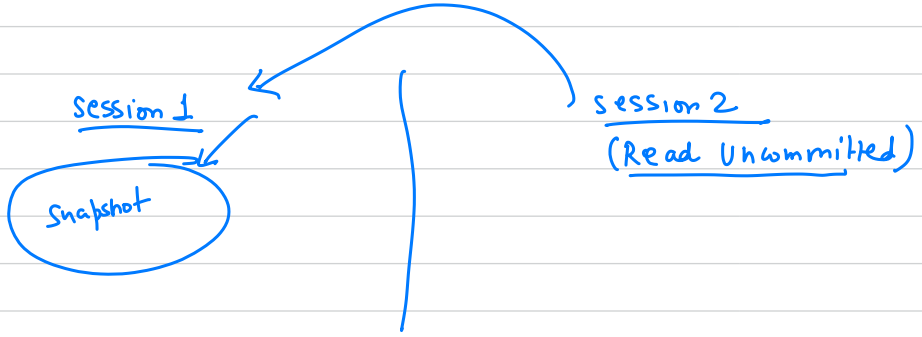
↳ should persist in DB forever



Relaxed

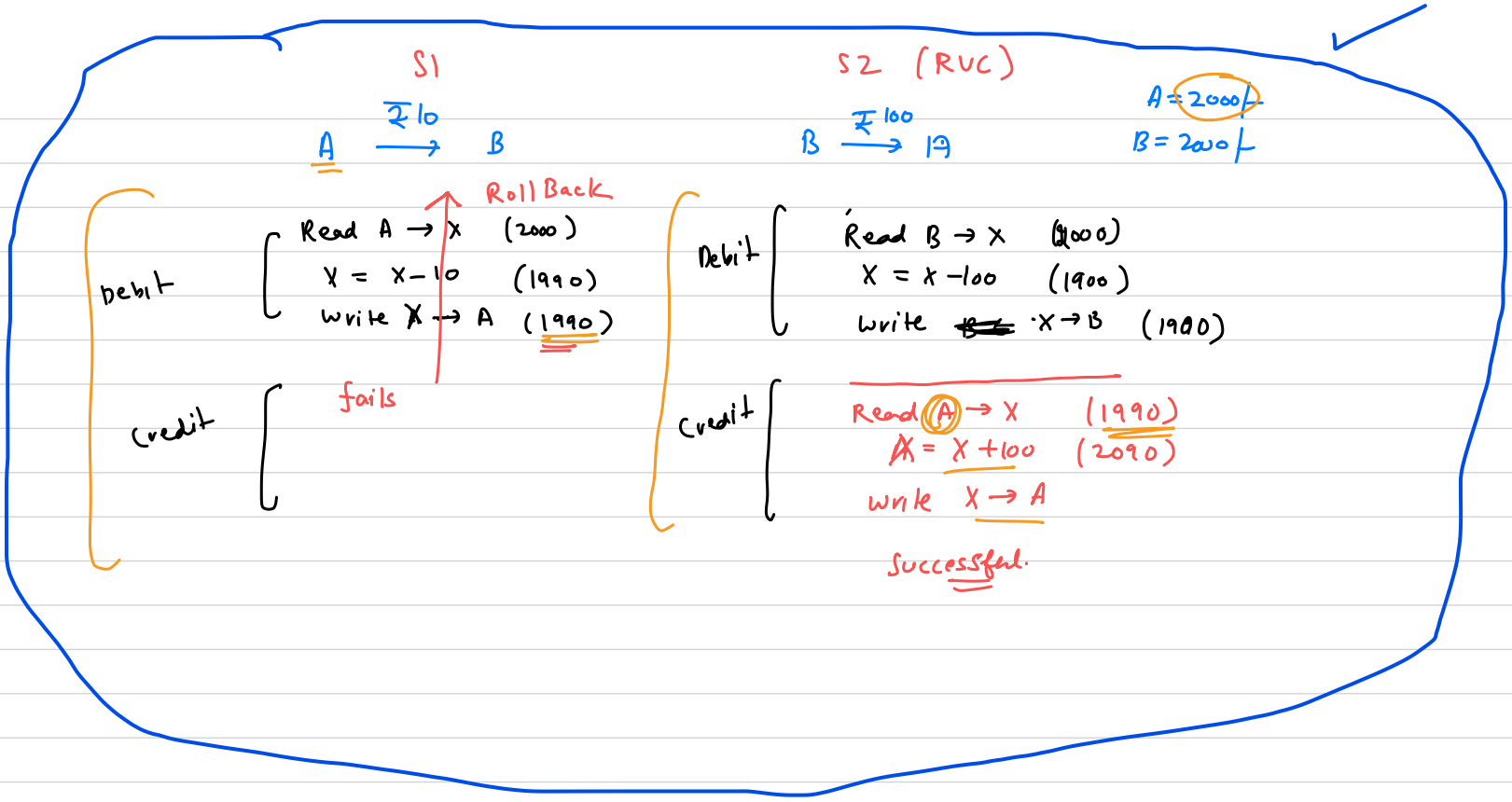
1) ⇒ Read Uncommitted (Today)
2) ⇒ Read Committed ----------> PostgreSQL default
3) ⇒ Repeatable Read → MySQL Default
4) ⇒ Serializable

Strict



Fast
but lead to inconsitency

Dirty Read is a problem in Read Uncommitted.


S1
₹10
A
B
Roll Back
debit
Read A → X (2000)
X = X - 10 (1990)
write X → A (1990)
credit
fails
S2 (RUC)
₹100
B
19
A = 2000
B = 2000
Debit
Read B → X (2000)
X = X - 100 (1900)
write X → B (1900)
Credit
Read A → X (1990)
X = X + 100 (2090)
write X → A
Successful.