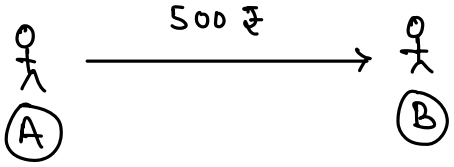
Transactions

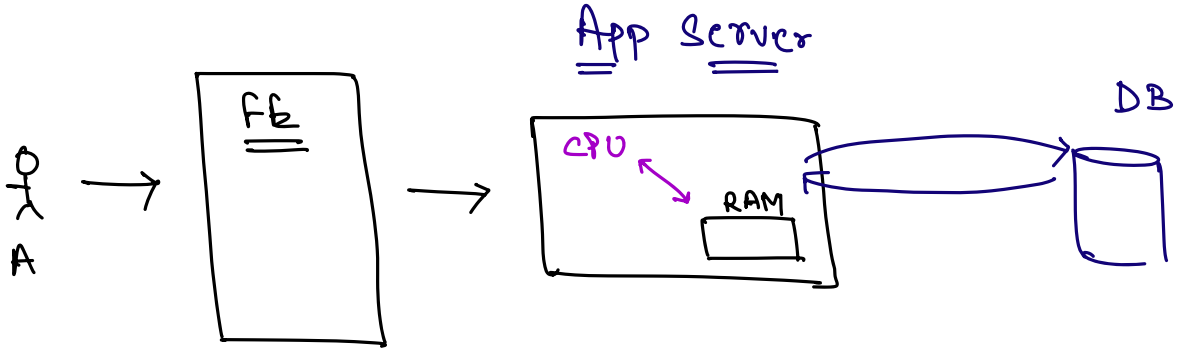
HDFC

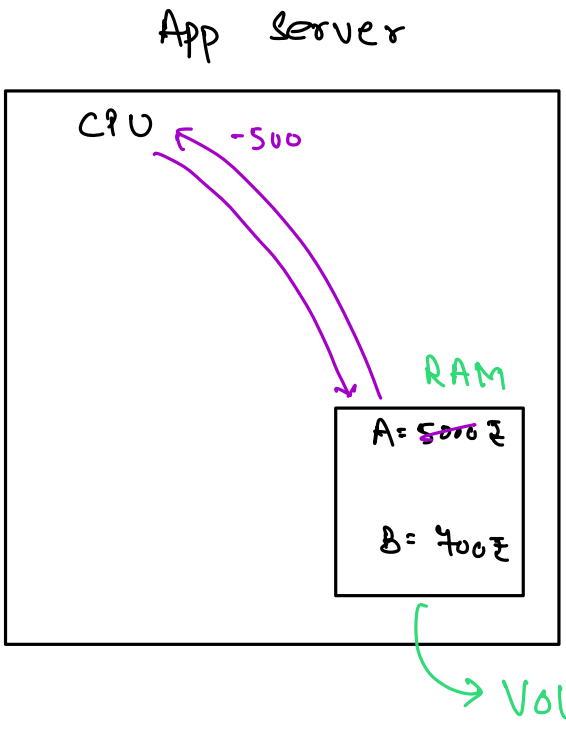


Task | Operation : Transfer 500₹ from A to B.

accounts

user	account	balance	.....
A	——	5000₹	
B	——	700 ₹	



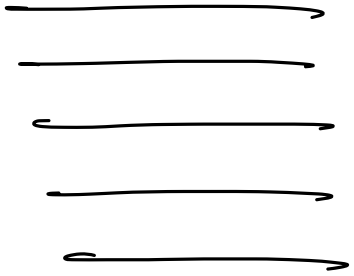


Permanent Storage.

Steps.

1) Get the balance of A.
2) Check if A's balance >= amount
3) Debit 500 from A's balance.
4) Write A's balance to DB.
5) Read B's balance & Credit 500 to B.
6) Write B's balance to DB.

func transfer_money (A, B, amount) {



}

transfer (A, B, 500) {

A = ~~5000~~ 4500

B = 700

5000 x ← Read(A)

if (x >= 500):

x = x - 500;

4500

Write(A)

700 y ← Read(B)

1200 y ← y + 500

Server crashes.

Write(B)

}

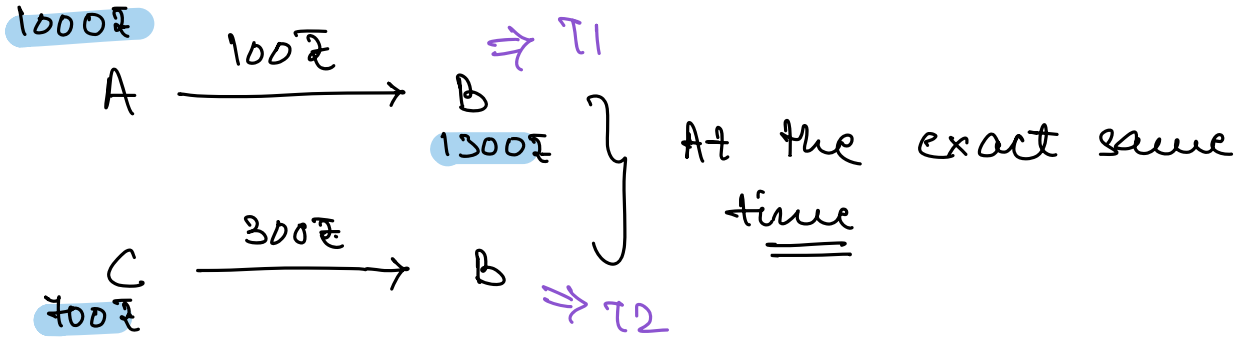
Before ⇒ 5700

After ⇒ 5200

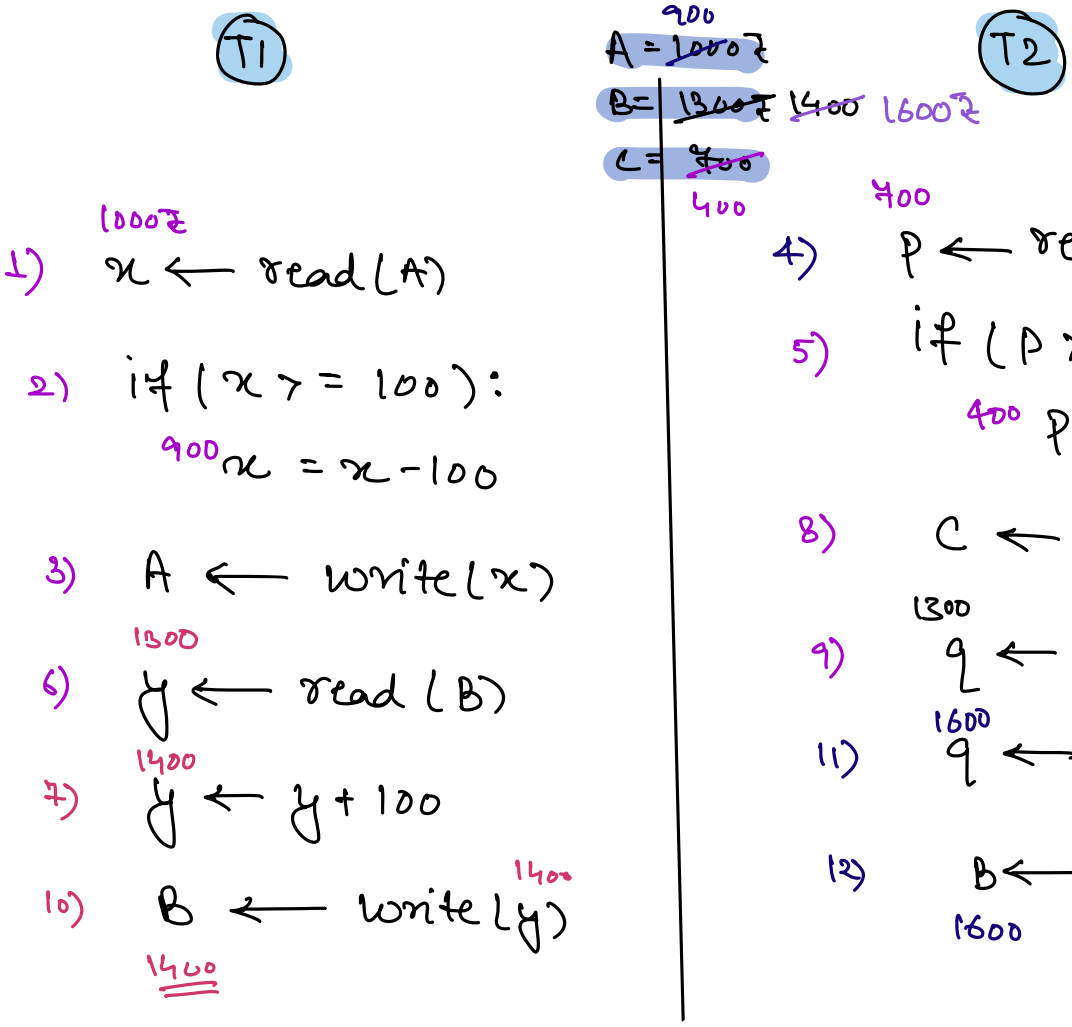
⇒ Loss of 500 ₹

Transaction

→ Set of DB operations logically grouped together to perform a task.



⇒ On account B, 2 transactions are taking place at the exact same time.



Before T1 & T2 :

A = 1000

B = 1300

C = 700

3000₹

After T1 & T2 :

A = 900₹

B = 1600₹

C = 400₹

2900₹

⇒ Loss of 100 ₹

ACID Properties.

Atomicity

Consistency

Isolation

Durability.

Atomicity

→ From outside, a transaction should look like an atomic unit.

↳ smallest unit.

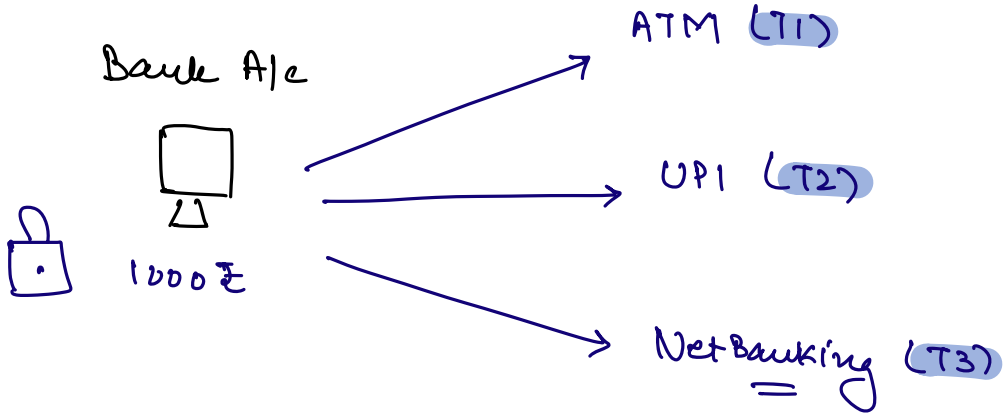
→ A transaction should either completely execute or shouldn't execute at all.

→ There should be No intermediate step.

→ If a transaction fails at any step then the steps which have already executed should be reverted / rolled back.

Consistency

→ DB should be in the correct state before & after the transaction.



Race Condition

Isolation

→ A transaction shouldn't impact another transaction running at the same.

→ Transactions should be isolated / independent from each other.

Durability.

↳ Once the transaction is done, data should be persisted to DB.

Commit :

→ It saves the data once the transaction is Completed.

→ Ensures durability.

→ MySQL gives autocommit by default.

Rollback

→ Undo the changes from previous commit.

```
select * from language;

start transaction;

update language
set name = 'Marathi'
where language_id = 5;

rollback;

commit;

show variables like 'transaction_isolation';

set transaction isolation level serializable; -- allows us to use locks.

SET SESSION TRANSACTION ISOLATION LEVEL SERIALIZABLE;

select * from language
where language_id = 3
for update;

commit;
```