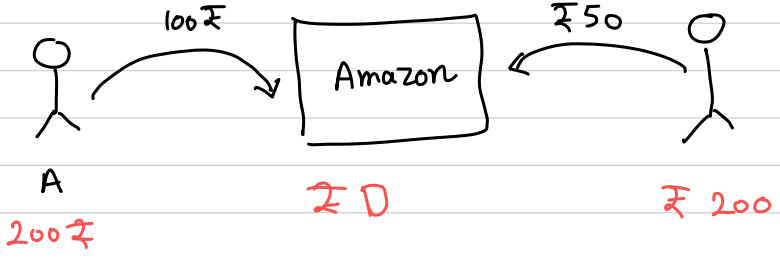
TRANSACTIONS

Agenda

- Transaction :
- ACID
- Commits & Rollbacks
- Transaction Isolation Levels
  RUC, RC, RR, Serializable
- Deadlocks

Next class

Real World Example



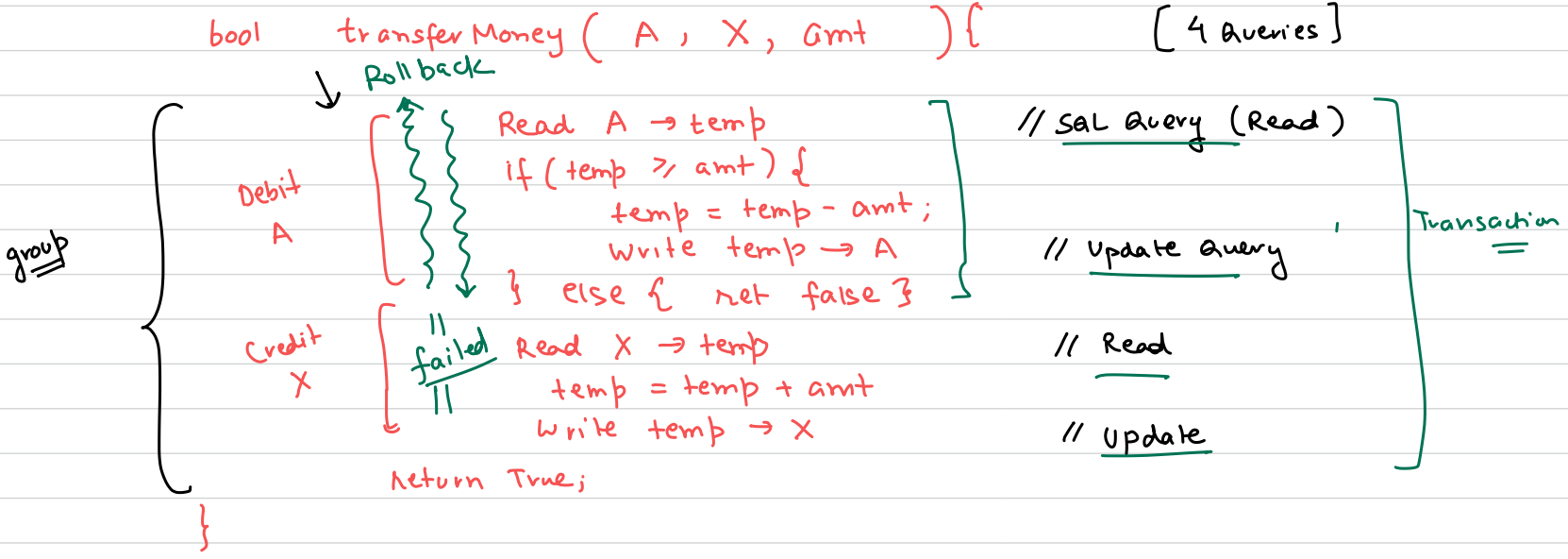
	A	Amz	B
Before	200	0	200
After	100	150	150

Expected Bank Balance.

Bank Balance

Uid	A/C	Bal.
A	=	200
B	=	200
Amz	=	0

Table (Bank Balance)

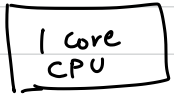


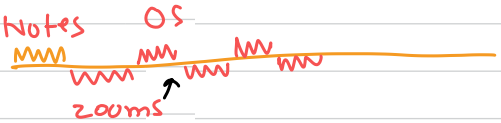
Concurrency

Concurrent Behaviour

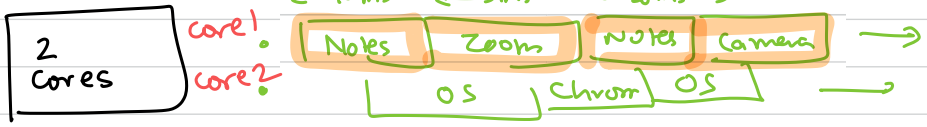
when multiples processes make progress at same time:

20 apps

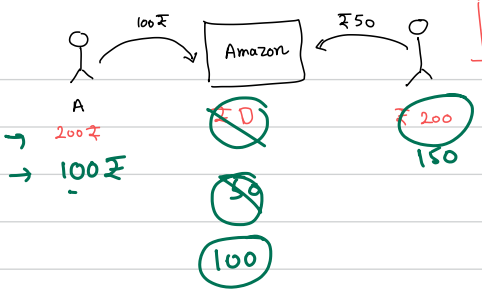




Context Switching



Each process has a priority, => get more / less CPU time
it is decided by OS scheduler.



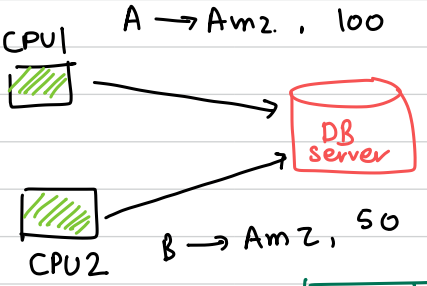
	A	Amz	B
Before	200	0	200
After	100	150	150

Expected Bank Balance.

Expected : 100, 150, 150
Actual : 100, 100, 150

Money got lost in process.

Bank Application server.



Concurrent / or truly parallel.

payments per second.

Transaction-1

Debit

1. Read A → t // 200
2. t = t - amt // 100
3. Write t → A // 100

Credit

7. Read Amz → t // 0
8. t = t + amt // 100
12. Write t → Amz // 100

as one unit

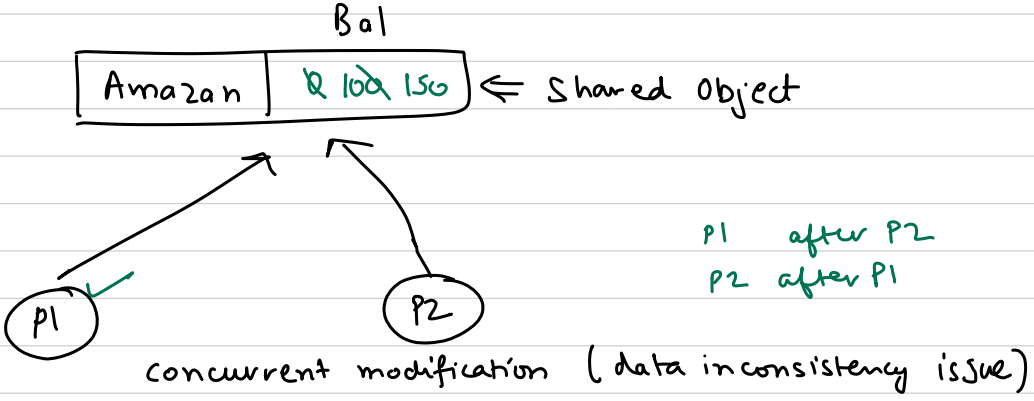
hold (waiting)

Transaction-2

4. Read B → t 200
5. t = t - amt 150
6. Write t → B 150

9. Read Amz → t 0
10. t = t + amt. 50
11. Write t → Amz

⇒ one unit



Transaction : is a set of DB operations logically grouped together to perform a task.

→ Money Transfer

→ Railway Ticket Booking
→ Movie ticket Booking

ACID Properties

Not necessary for a transaction to have all 4 properties.
depends upon use-case.

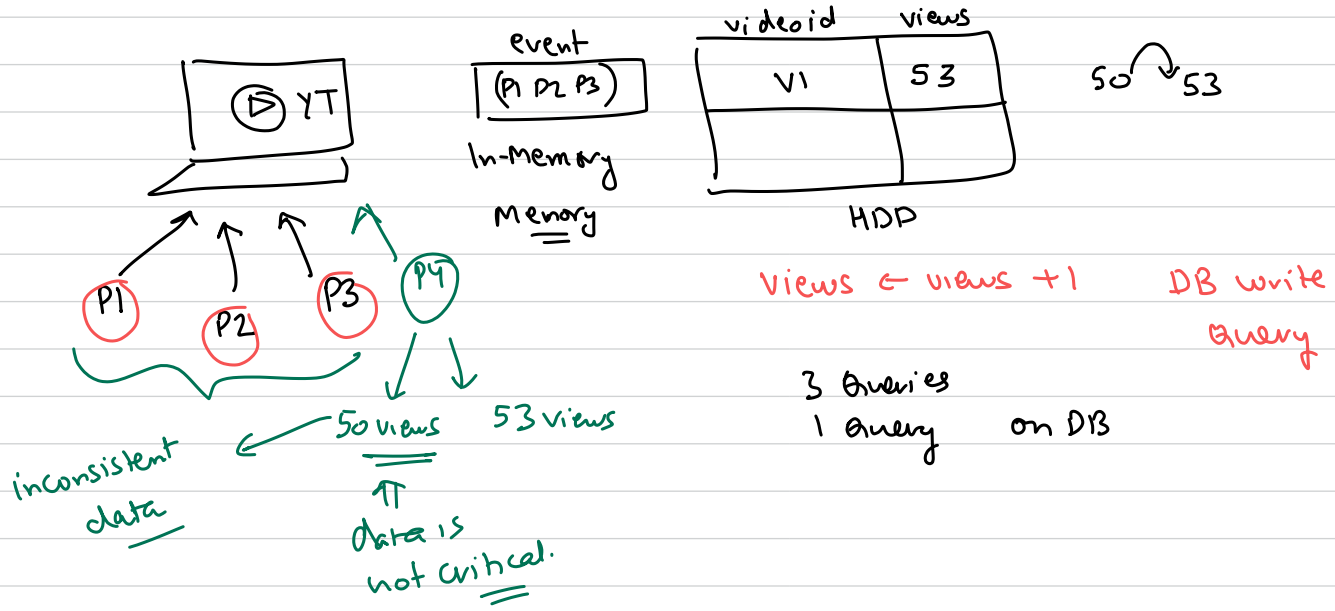
1) Atomicity
2) Consistent
3) Isolation
4) Durability

① **Atomicity** : Opn/trans should appear as atomic/one entity to an end user.

UPI Payment → single outcome
↳ successful (all the opns)
↳ fail (nothing has happened)

② Consistency — logically correct

Banking ⇒ Consistency is expected.



③ Isolation

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

"Lock" → mechanism to provide "Isolation"

↓ 4 levels of Transaction Isolation

④ Durable

↳ long lasting

↳ once change is made, it should persist in DB forever.

- Data → HDD / SD card + Backup
- Cloud storage (Multiple Backups)

↓

across different Regions of world.

10:20

- Commit & Rollback → undo the changes since last commit
  - Saving changes to DB (Durable)

SET [autocommit = 0]

- TRANSACTION ISOLATION Levels.

committed: M1
uncommitted: M1-dino

What "state" of data you will end up reading.

4 Transaction "Isolation" Levels defined for every session.

Lenient

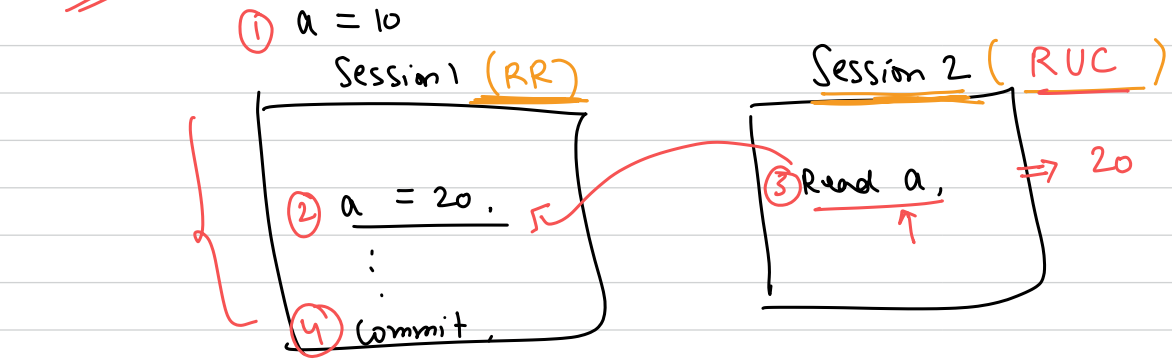
1) Read Uncommitted (RU) ⇒ dirty read problem

2) Read Committed (RC)

3) Repeatable Read (RR) [Default in MySQL]

4) Serializable

Strict



RUC

good : efficient in execution, no locks or concurrency control are reqd.

bad : dirty Reads, inconsistent data.

Init : 100, ~~50~~ 0, 100 ⇒ 200

finaly : ~~50~~ 100, 100, 50 ⇒ 250