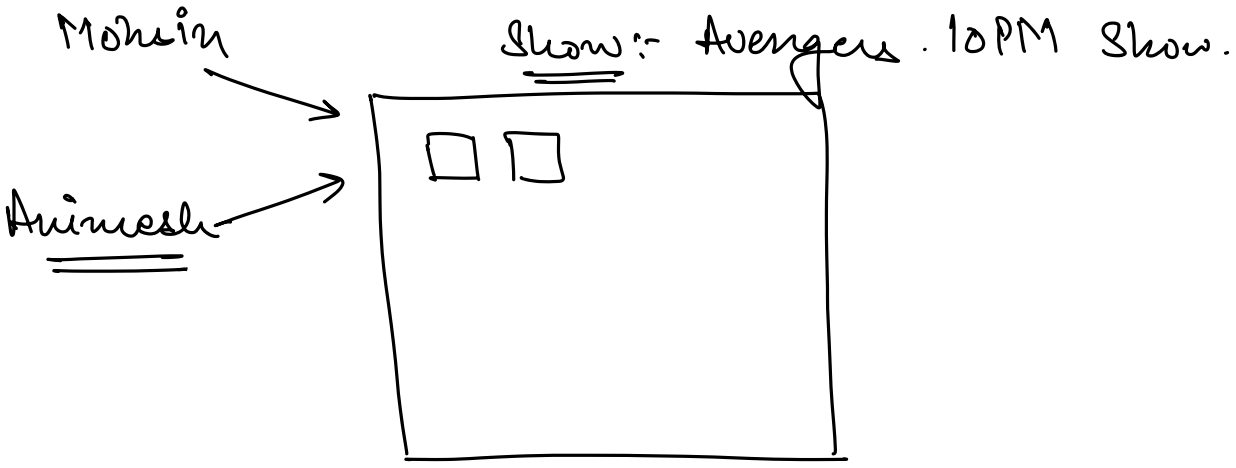
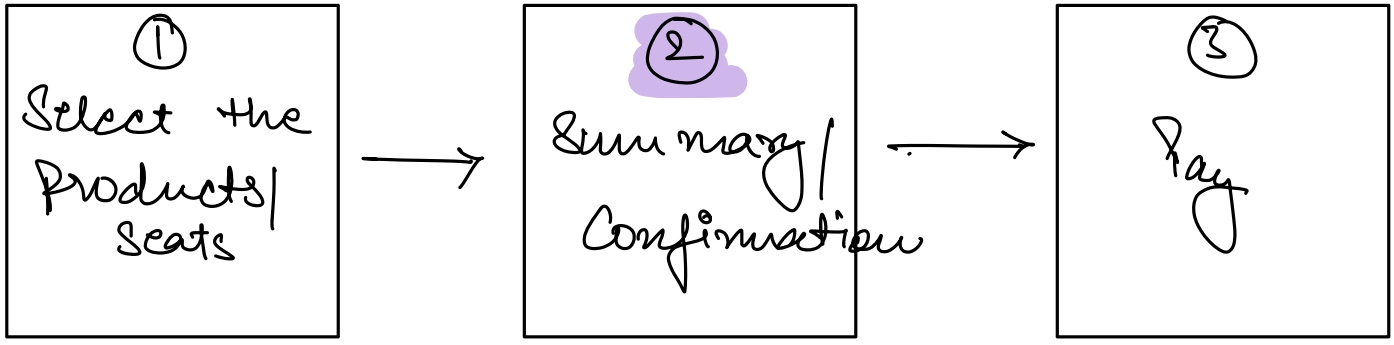
Users should be able to book seats.
(Max 10)

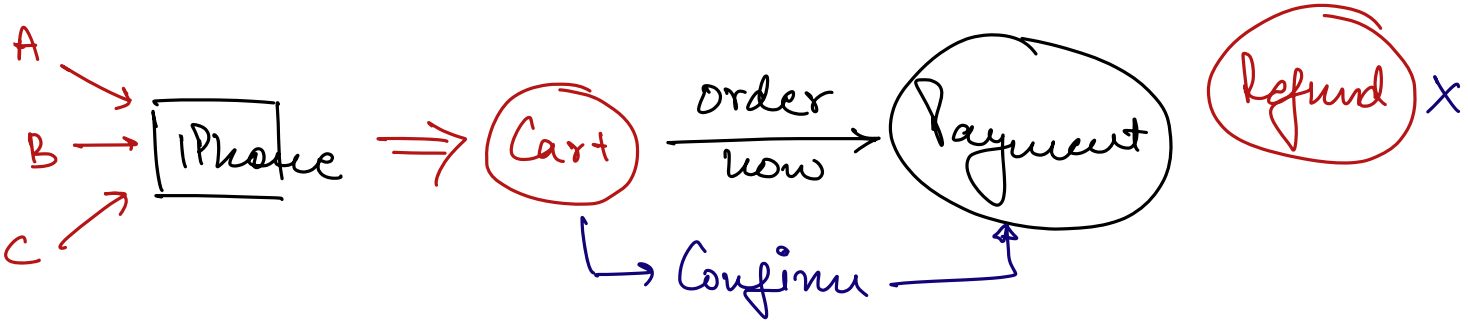
⇒ No 2 users should be allowed to book the same seats for same show.

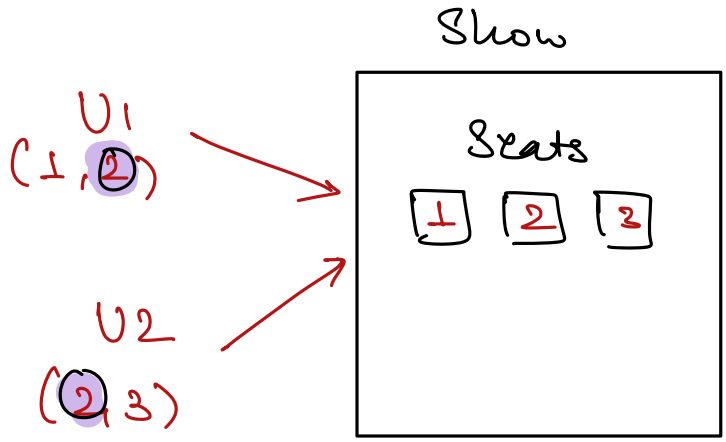
⇒ How to handle concurrency?



⇒ Every ordering website.







Show_Seat.

AVAILABLE | BOOKED | BLOCKED.

id	Show_id	Seat_id	Status.	
1			AVAILABLE	
2			AVAILABLE	
3			AVAILABLE	
⋮				

⇒ Isolation Level : Serializable.

Approach #1:

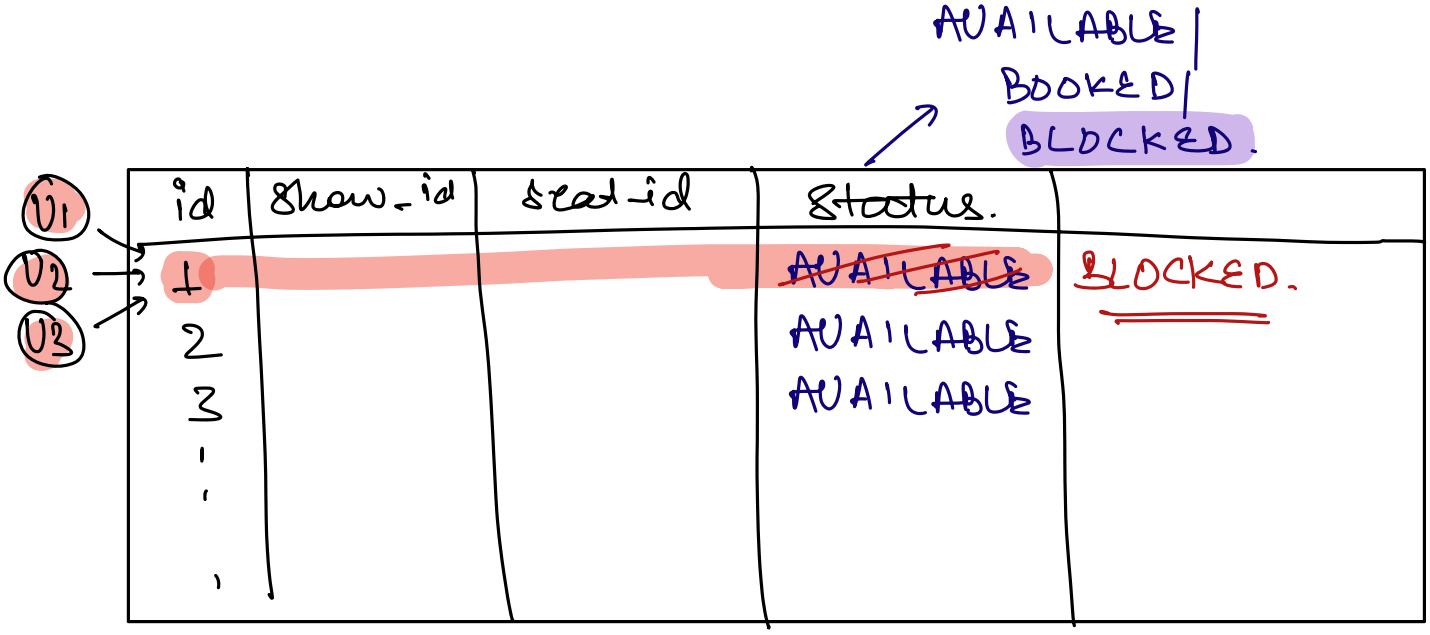
1. Take a lock on all the rows that user is trying to book.
2. Check the status if seats are Available or not.
3. Payment. Page — Pay / Go back. } 7-8 mins.
4. Release the lock.

⇒ lock was there for the duration entire transaction.

Approach #2

⇒ Introduce a new Show seat Status as BLOCKED, and use this instead of DB lock.

⇒ We'll not keep the transaction running till the end of payment.



⇒ Only one user should be allowed to acquire a lock.

1. Check the seat status. If the status is AVAILABLE

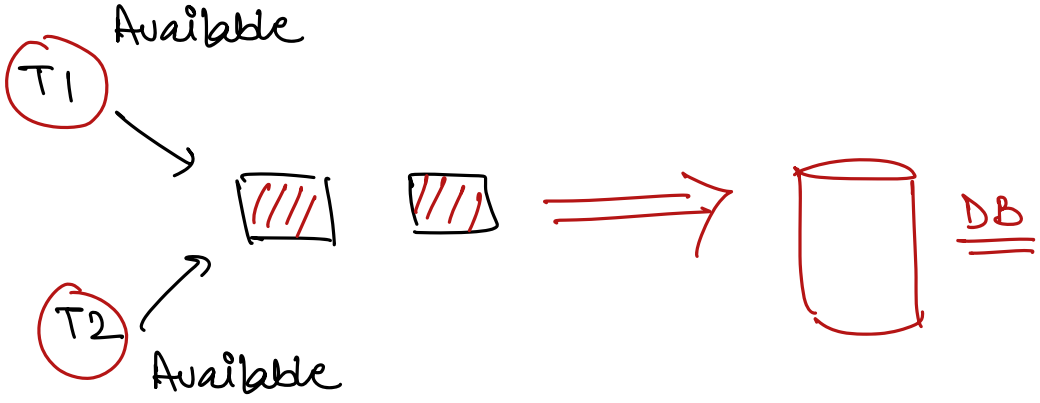
2. Take a lock over the seat id's.
   - Only 1 transaction will be able to acquire the lock.

3. DOUBLE CHECK LOCKING.
   Check the status again if it is AVAILABLE

4. Change the seat status to be BLOCKED.

5. Release the lock

6. Payment -
   - Pay ⇒ BLOCKED → BOOKED.
   - Go back. ⇒ BLOCKED → AVAILABLE

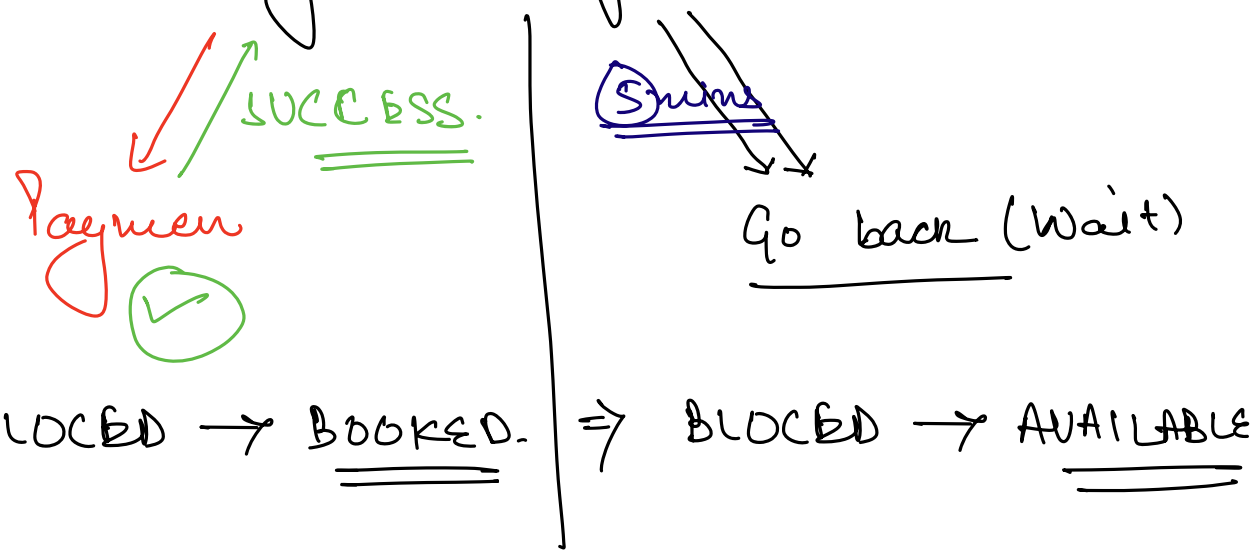


⇒ T1 acquires the lock.
⇒ Again checks the status.
⇒ Changes the status to BLOCKED.

T2 ⇒ Something is Not right

⇒ T1 releases the lock.

⇒ Go the Payment page. (T1)



⇒ BLOCED → BOOKED.

⇒ BLOCED → AVAILABLE.