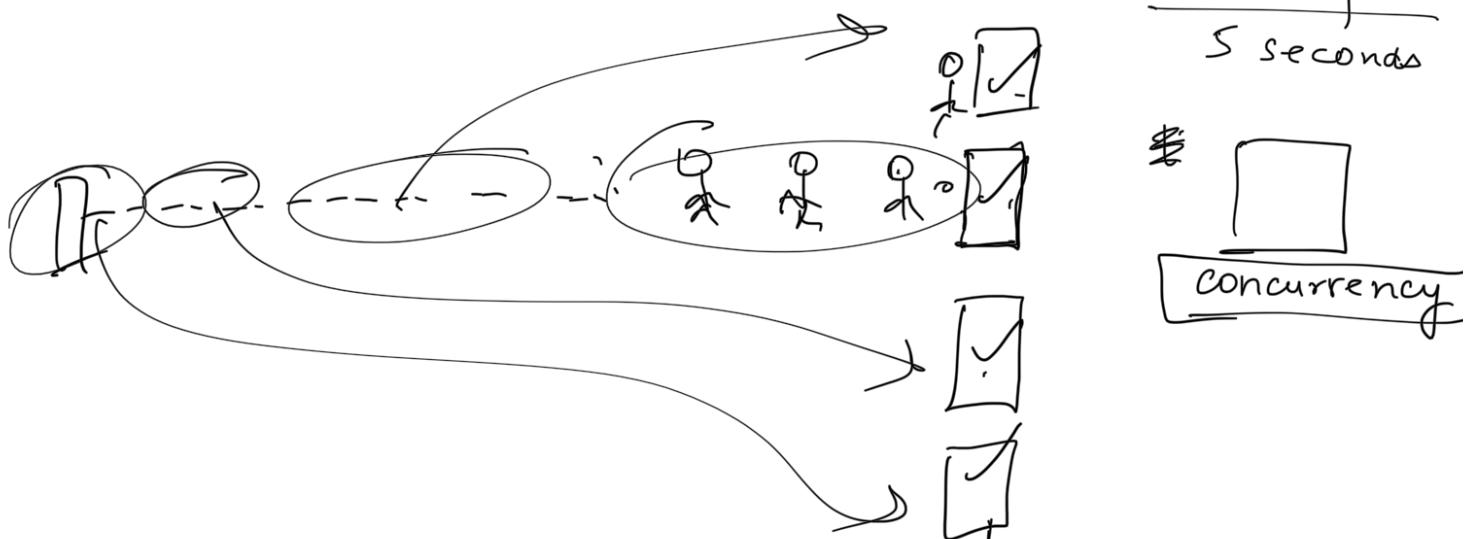
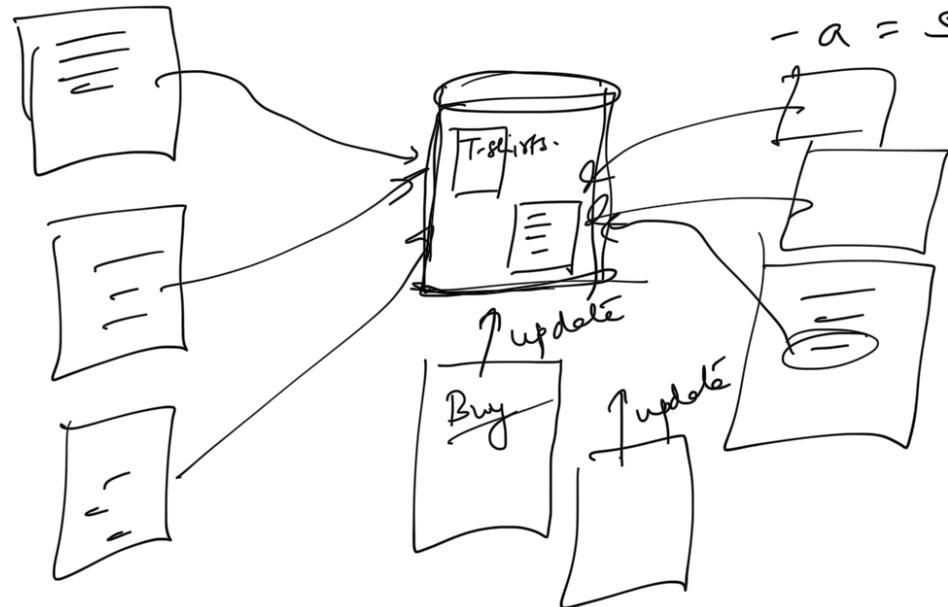
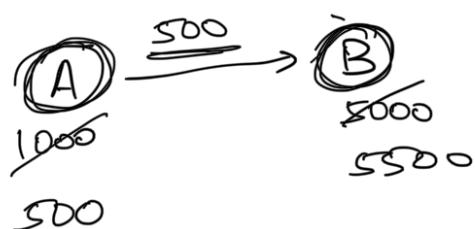


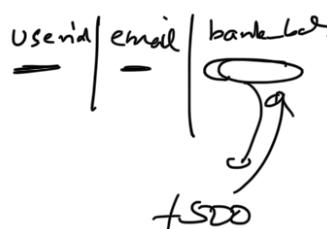
TRANSACTIONS



Bank



Users



① Check current balance of A (bal_A)

\rightarrow t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4 \rightarrow t_5 \rightarrow t_6

$\rightarrow \leftarrow$ Is $\text{balA} >= 500$:

③ $\text{balA} = \text{balA} - 500$, update

④ SELECT balB -

⑤ $\text{balB} = \text{balB} + 500$, update DB.

$A \rightarrow 1000$, $B \rightarrow 5000$, $C \rightarrow 2000 = [8000]$



① $\text{balA} = \text{current balance of A}$

$$\text{balA} = 1000$$

② Is $\text{balA} >= 700$:

① $\text{balA} = 700$

② $\text{balB} = 5000$

$$700 \rightarrow \boxed{\text{balB} = 5700}$$



① $\text{balA} = \text{current balance of A}$

$$\text{balA} \leftarrow 100 =$$

② If $\text{balA} >= 800$:

① $\text{balA} - 800 = 200$

② $\text{balC} = 2000$

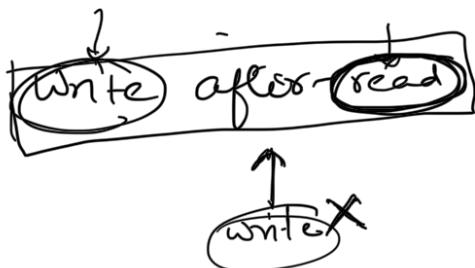
$$\boxed{\text{balC} \rightarrow 2800}$$

B	A	C	$\Rightarrow [8700]$
5700	200	2800	

Transaction:

1 or more SQL statements

- ↳ Read CRUD
- ↳ update/insert



$[f(\text{source_user}, \text{dest_user}, \text{amount})]$

bal A :- SELECT bal from users where
email = source-user

if bal A >= amount :



A diagram illustrating a process flow. On the left, there is a square tank. Two pipes, labeled T_1 and T_2 , originate from the bottom of this tank. Pipe T_1 leads to a cylindrical tank containing a liquid with horizontal striations. Pipe T_2 leads to another cylindrical tank, which appears to be empty or contains a very light liquid. The labels $f(A, B, 700)$ and $f(A, C, 800)$ are written near their respective tanks.

CPU 1 → [Update users set bal = 200
WHERE email = xyz]

```

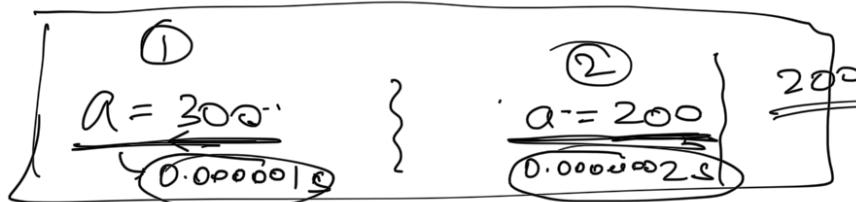
graph LR
    A((CPU 1)) -- T0 --> B["T1  
1, 2, 3, 4, 5"]
    B --> C((CPU 2))
    C -- T2 --> D["T2  
1, 2, 3, 4, 5"]
    D --> E["T3  
1, 2, 3, 4, 5"]
    E --> F["T_f"]
  
```

The diagram illustrates a merging process. It starts with two ovals labeled "CPU 1" and "CPU 2". An arrow labeled T_0 points from "CPU 1" to a horizontal line above which are the numbers $1, 2, 3, 4, 5$. From this line, an arrow points to another oval labeled "CPU 2". An arrow labeled T_2 points from "CPU 2" to a horizontal line above which are the numbers $1, 2, 3, 4, 5$. An arrow labeled T_3 points from the second horizontal line to a final oval at the bottom labeled T_f .

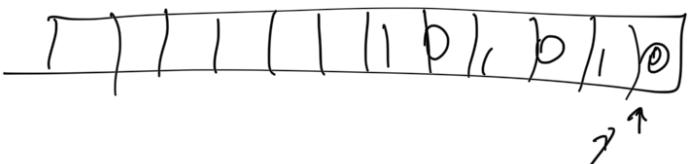
\sqsubset_1 all steps.
 \sqsubset_2 all steps.

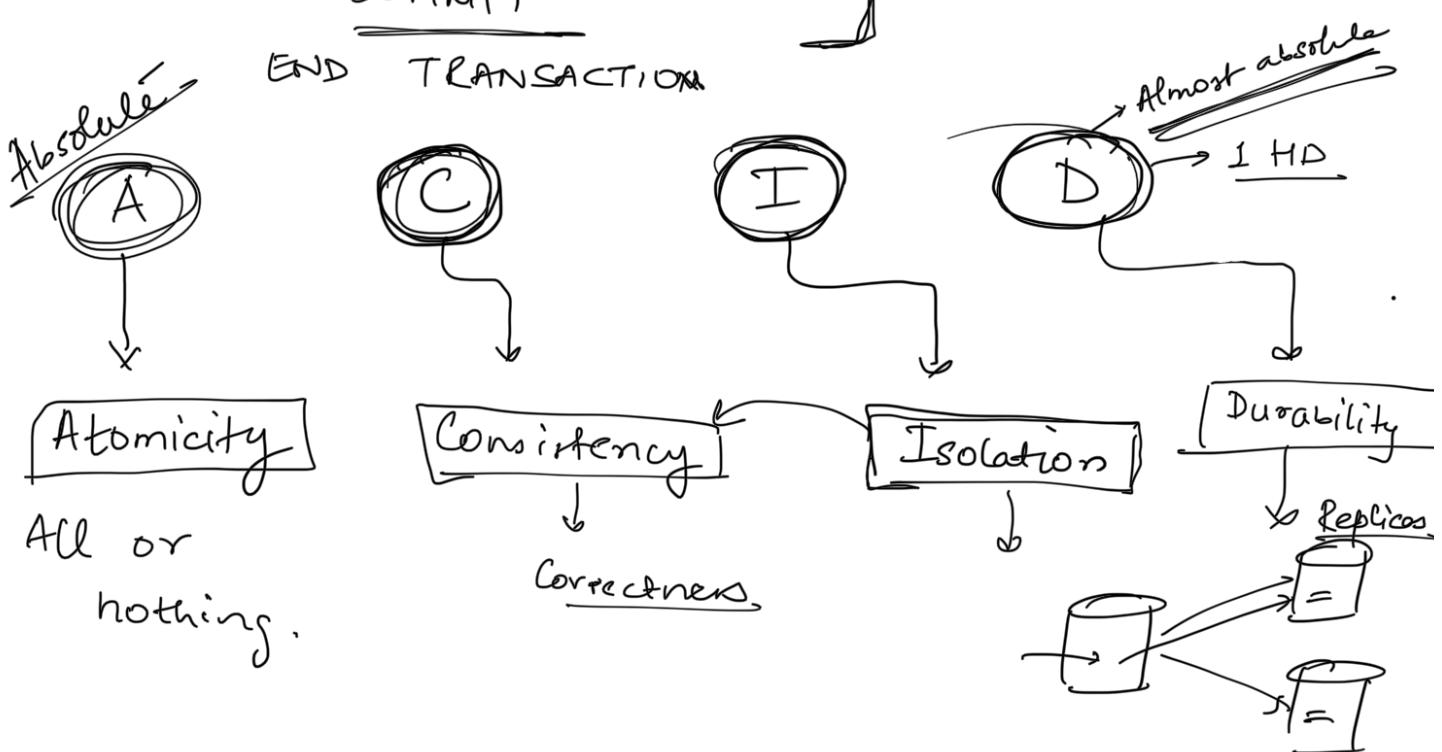
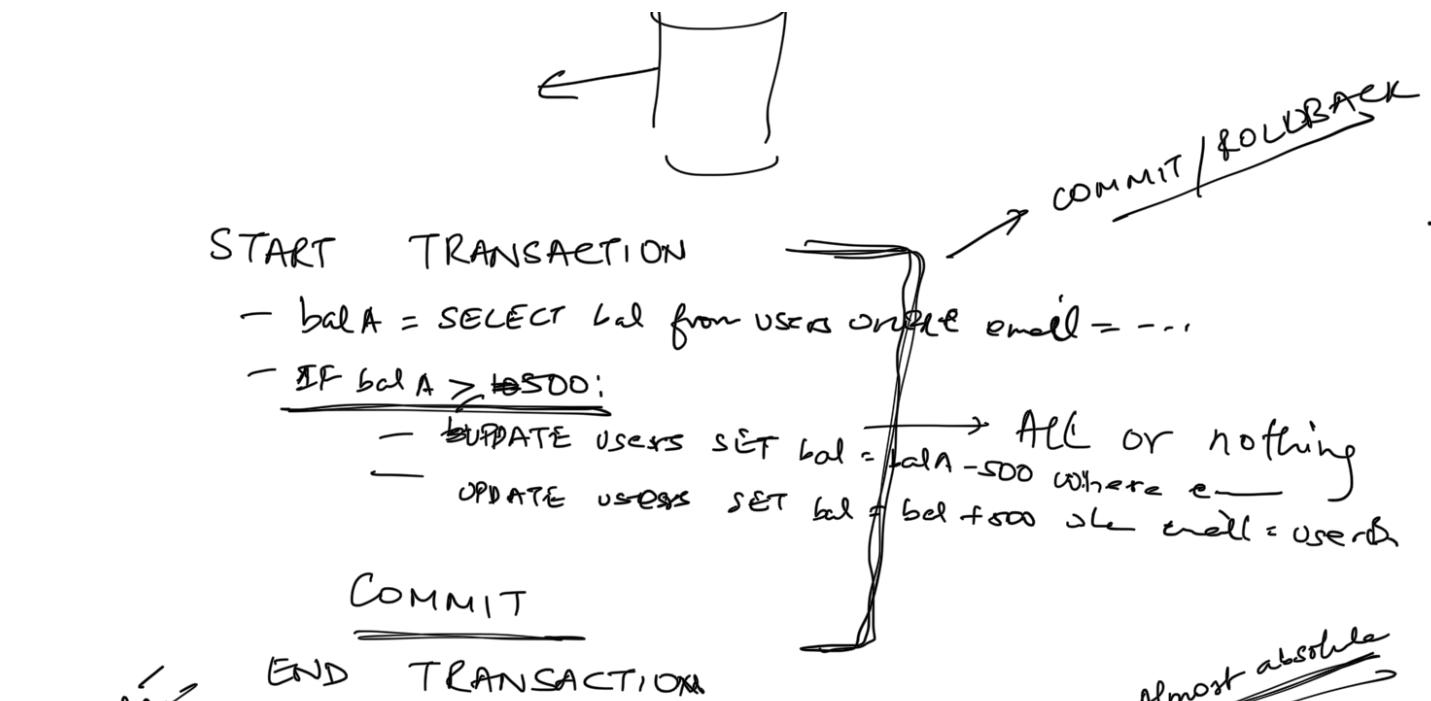
CPU $\xrightarrow{x_1}$ x_1 $\xrightarrow{x_2}$ $x_3, x_4, x_5, x_6, x_7, \dots, x_{10}$

CPU2 x2



$\rightarrow \text{int } @ = \underline{100};$





$A \xrightarrow{500} B$

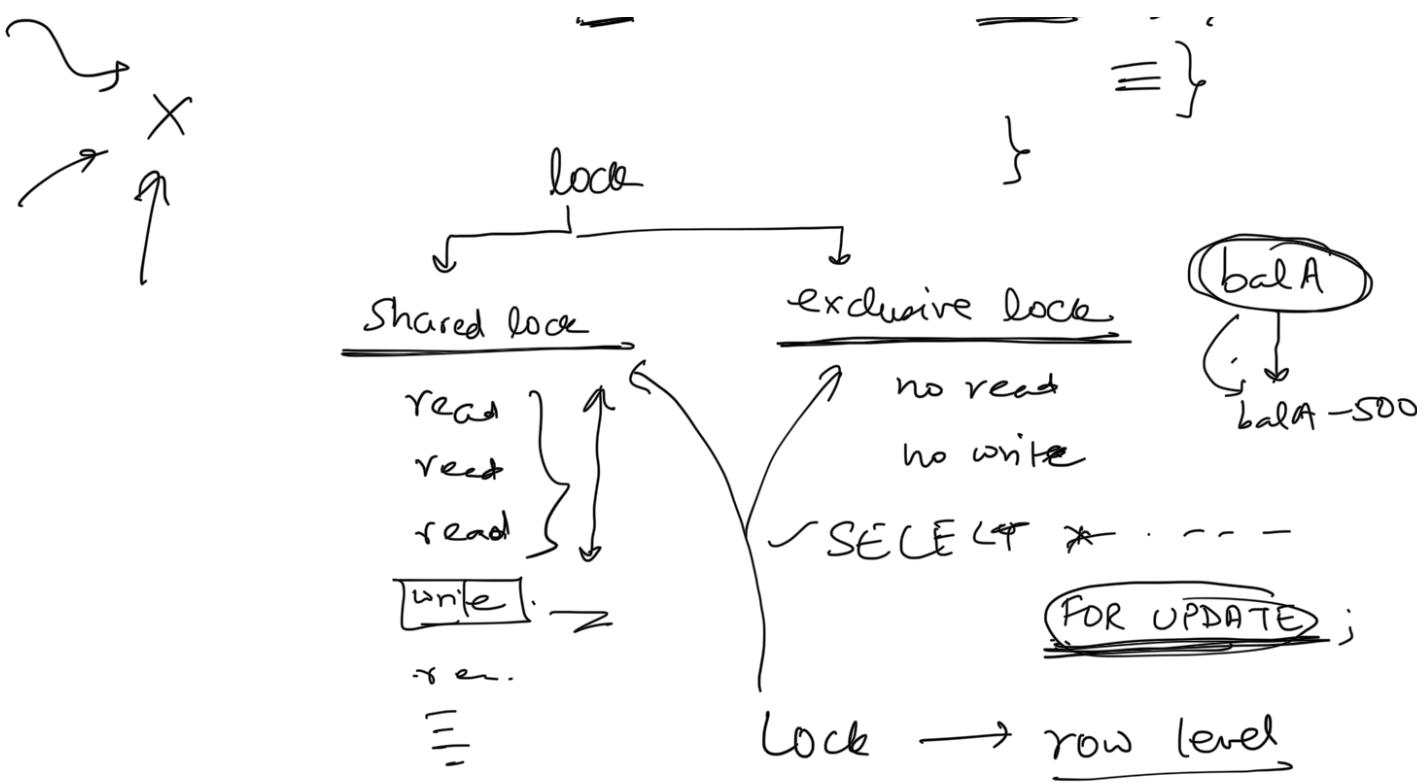
$A \xrightarrow{600} C$

LOCK

balA

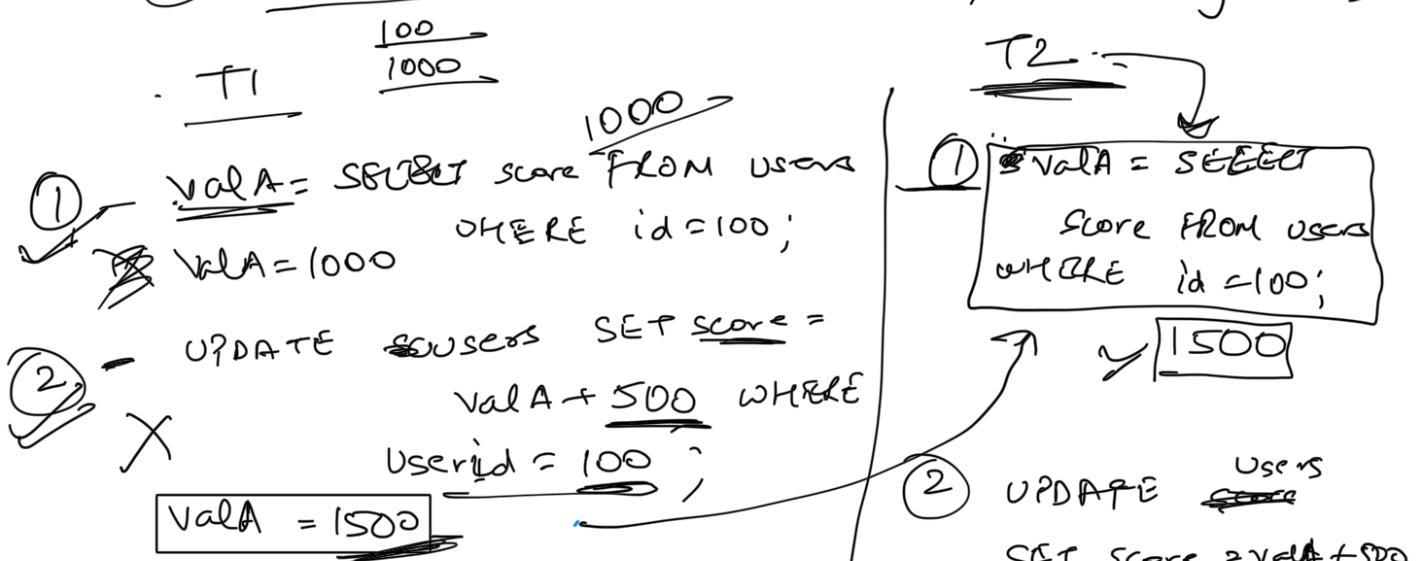
int $\lceil x \rceil = 10;$

lock(x) {



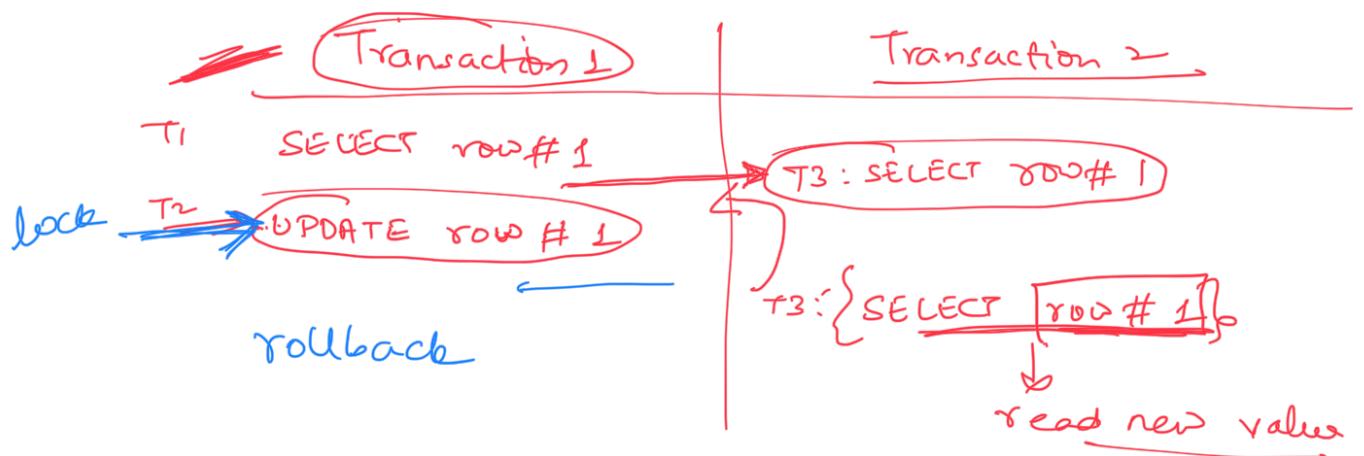
4 levels of isolation

① Read uncommitted (Facts, consistency issues)



ROLLBACK

② Read committed. (Postgres DB)



Reads → completely fine

<u>Write</u>	→ exclusive lock
--------------	------------------

repeatable reads

Transaction 1
① SELECT email FROM users WHERE psp < 50%

② UPDATE email SET last-sent = NOW() WHERE email IN (SELECT email FROM users WHERE psp < 50%)

T1
Email
① SELECT psp < 50%
Repeated Reads

T2
① Update psp for some users
Commit → released lock

② update where psp < 50%

③ Repeatable Reads. (Transaction)

Txn
S1

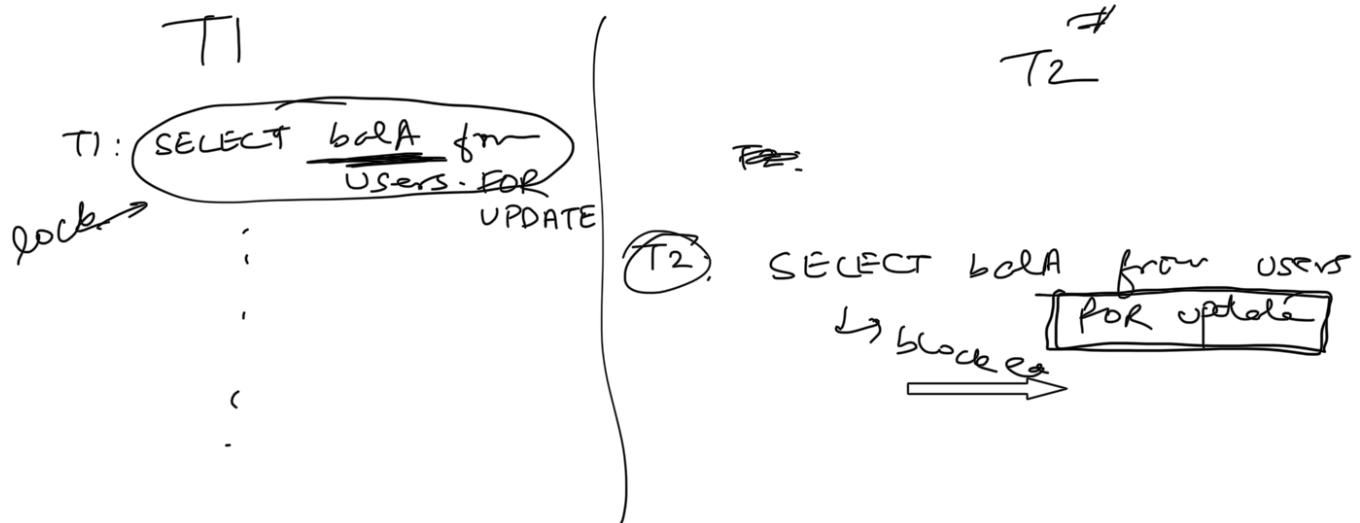
Banks
✓④

Serializable Read → Banks

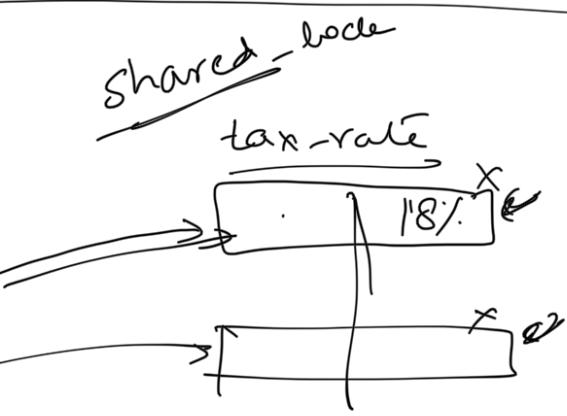
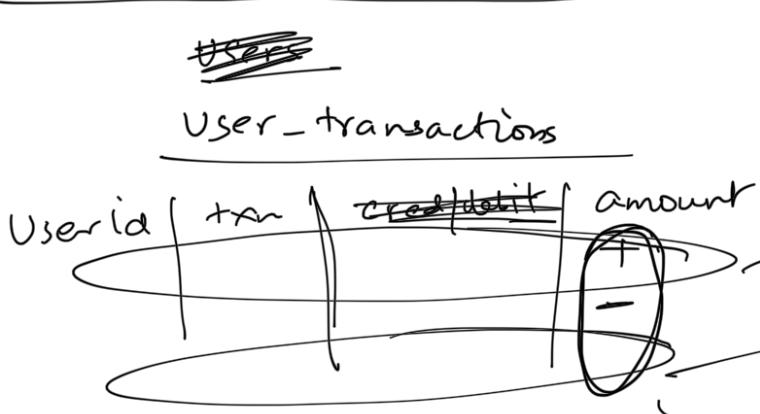
write → exclusive lock # rows that are affected

read → shared lock # row

read with intention of changing rows later
→ exclusive lock



SERIALIZABLE READ FOR UPDATE

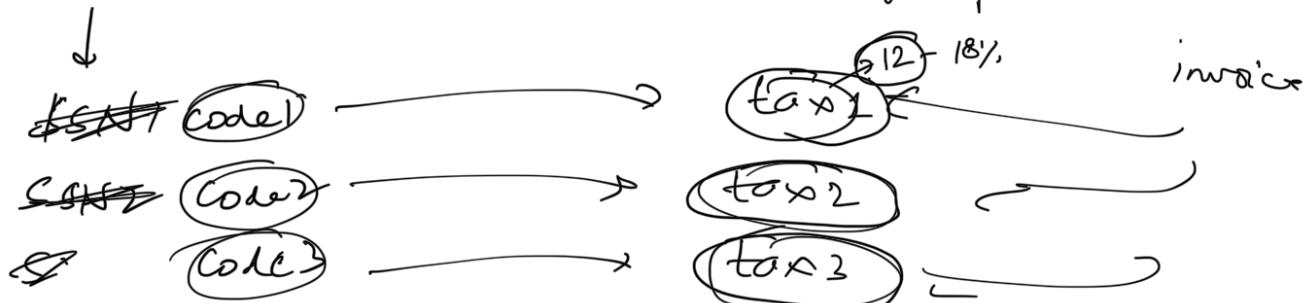


SERIALIZABLE

Read # row → Shared lock

invoic

↳ write # row → exclusive
↳ Read for update # row ↑



SERIALIZABLE
READ

Transaction

Seat 23 A

exclusive lock

(FOR UPDATE)

SERIALIZABLE
READ

Seats

IA

IB

IC

⋮
⋮

