Transactions - II

Update -

Extra session
↳ 16 march
Assignment problems

Transaction Isolation Levels

1) Read Uncommitted
2) Read Committed
3) Repeatable Read (MySQL Default)
4) Serializable

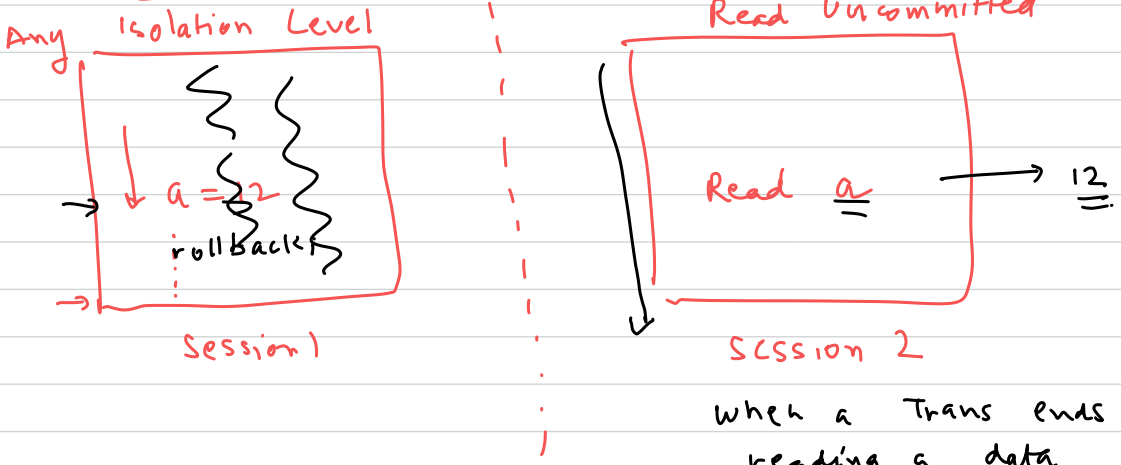
(Strict)

↓ Deadlocks

① Read Uncommitted

Last committed

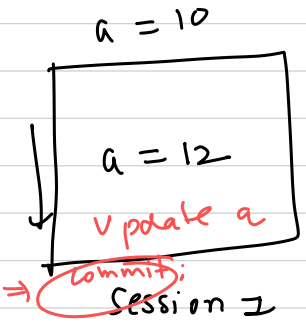
a = 10



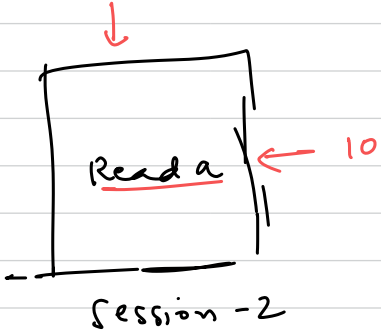
when a Trans ends up reading a data which may not be committed → DIRTY Read.

Read Commited

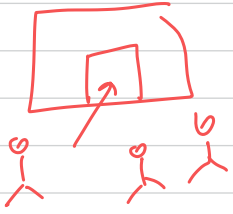
(2)

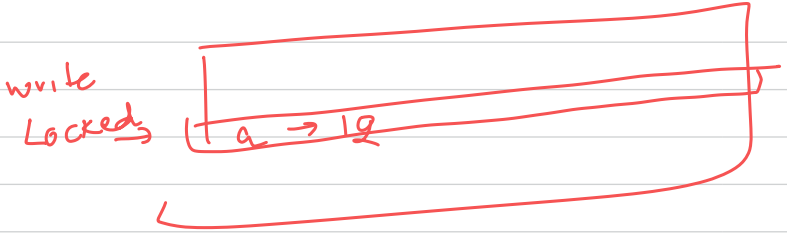


Read Committed



Last committed value of A.



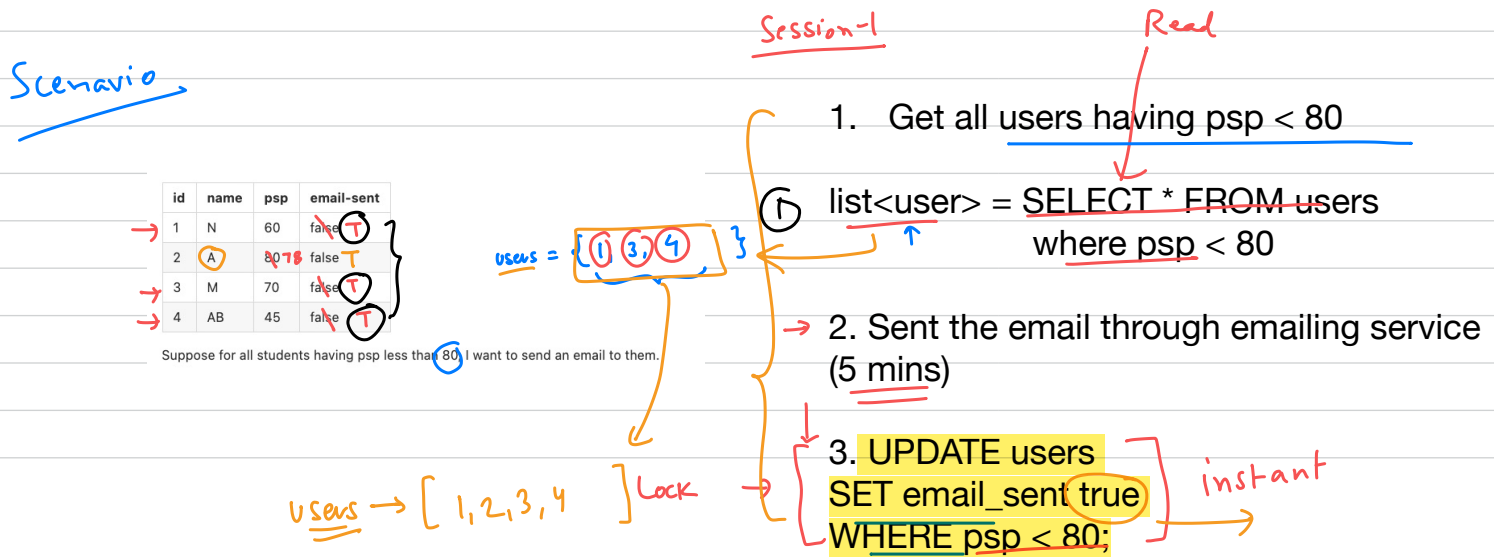


Advantages :

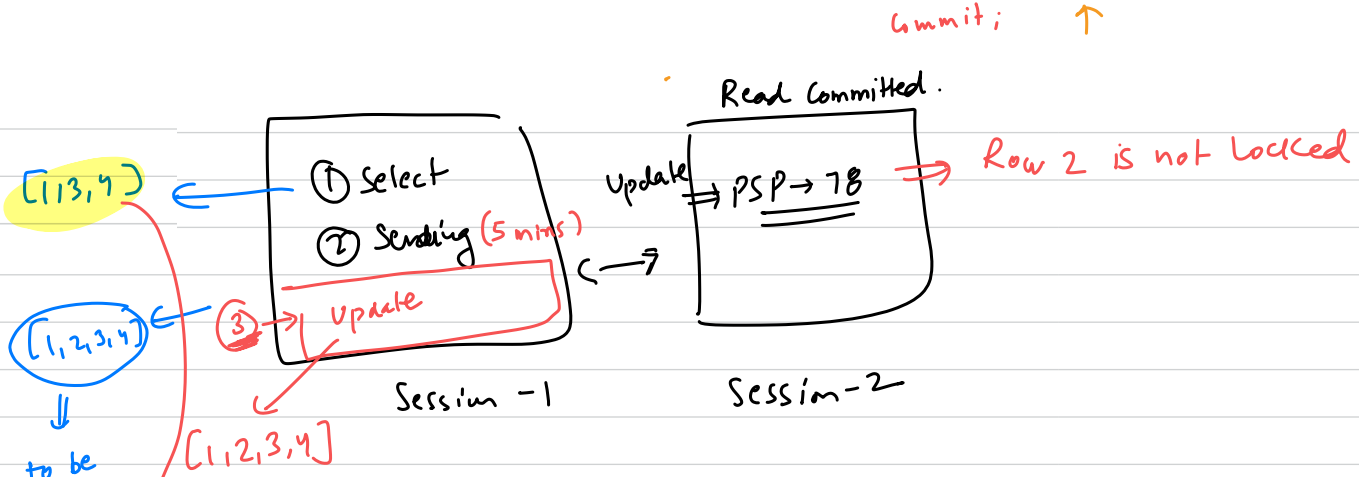
1. Read the latest committed value, dirty read will never happen.

Disadvantage :

1. Performance will slightly degrade because of locks and concurrency controls.



id	name	psp	email-sent
1	N	60	false → T
2	A	80 → 78	false → T
3	M	70	false → T
4	AB	45	false → T



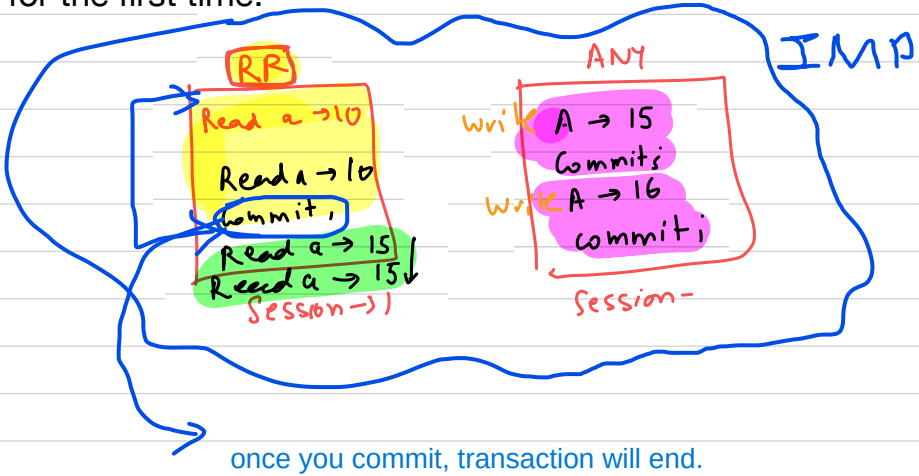
Actual emails have been sent to 3 ppl only
But in DB 4 ppl have been marked as True.
This problem is called as NON-REPEATABLE READ

Within the same session, two different queries reading the same data produce two different results.

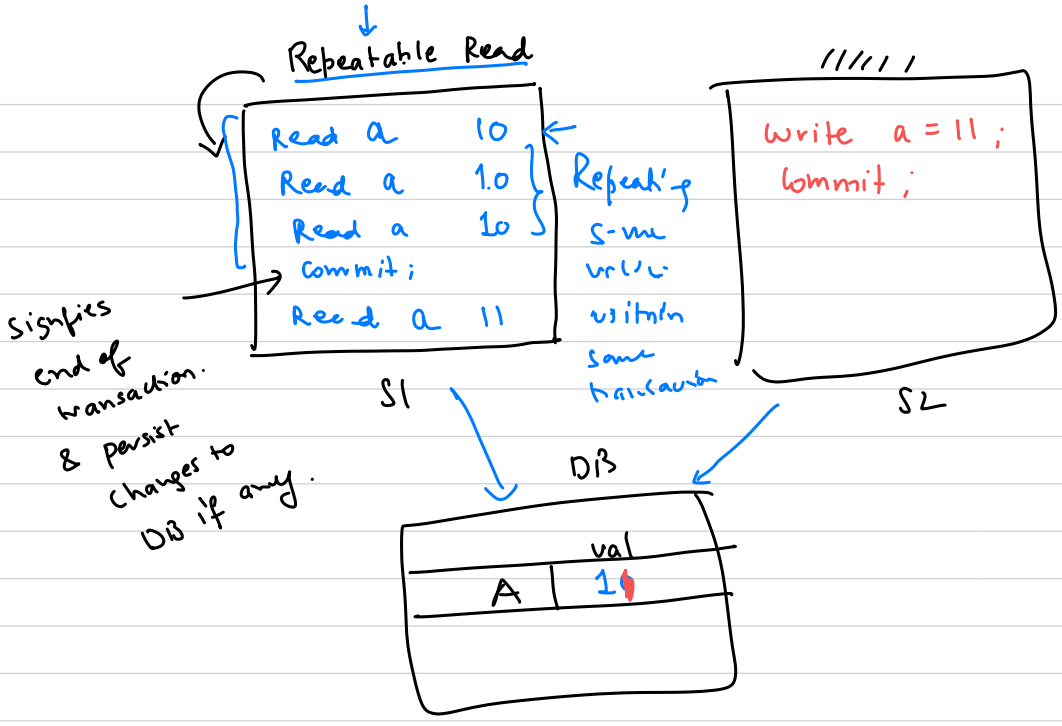
3) Repeatable Read Isolation Level (MySQL default isolation level)

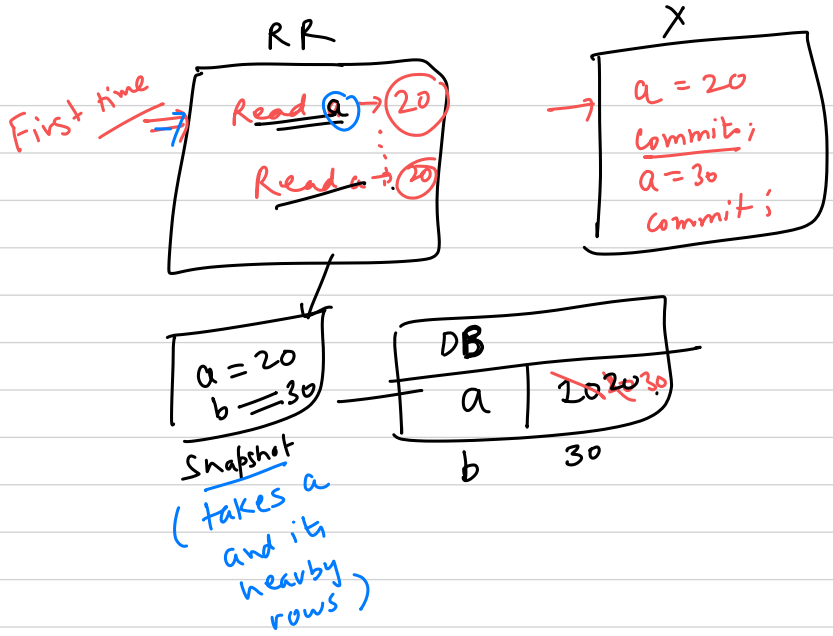
Whenever you have a read a row in this isolation level:

- For the first time, it reads the latest committed value of the current row.
- After that till the transaction completes, it keeps on reading the same value it read for the first time.

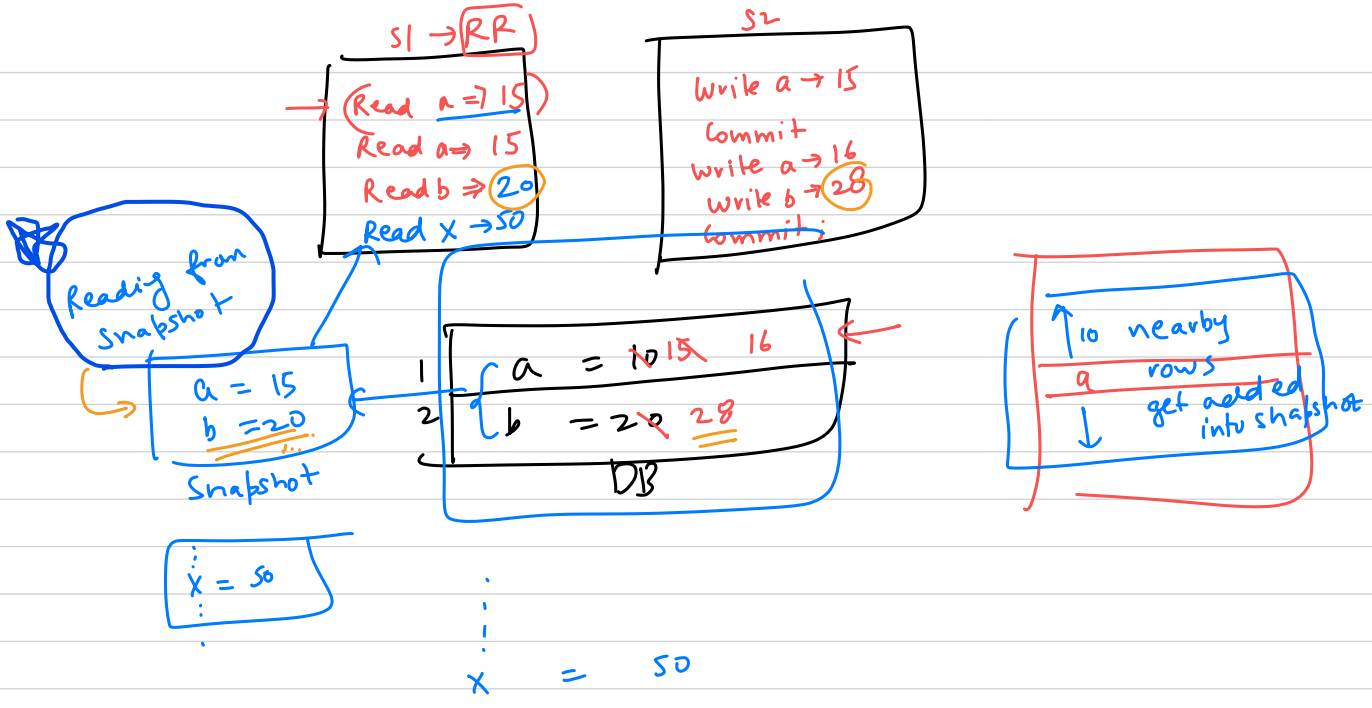


once you commit, transaction will end.


Repeatable Read
Read a 10
Read a 10
Read a 10
Commit;
Read a 11
Signifies end of transaction & persist changes to DB if any.
Repeating same read within same transaction
S1
Write a = 11;
Commit;
S2
DB
val
A
10


RR
X
First time
Read a
20
Read a
20
a = 20
commit;
a = 30
commit;
a = 20
b = 30
Snapshot
(takes a and its nearby rows)
DB
a
20
20
30
b
30

10·25


S1 → RR
Read a → 15
Read a → 15
Read b → 20
Read X → 50
S2
Write a → 15
Commit
write a → 16
write b → 28
commit;
Reading from Snapshot
a = 15
b = 20
Snapshot
1
2
a = 10 15 16
b = 20 28
DB
10 nearby
a rows
get added into snapshot
X = 50
X = 50

Phantom Read (Problem in RR)

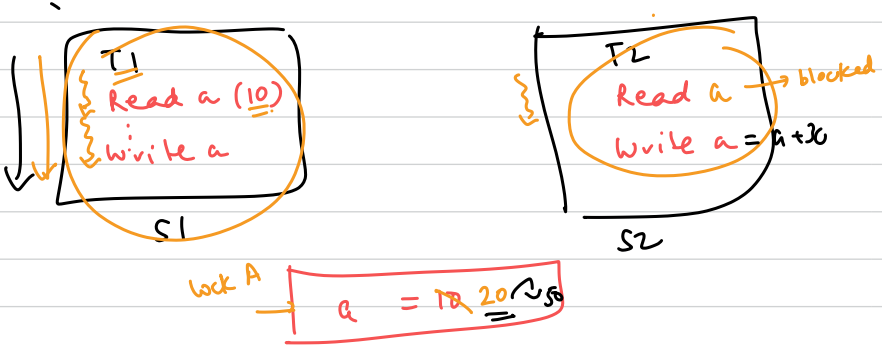
↓

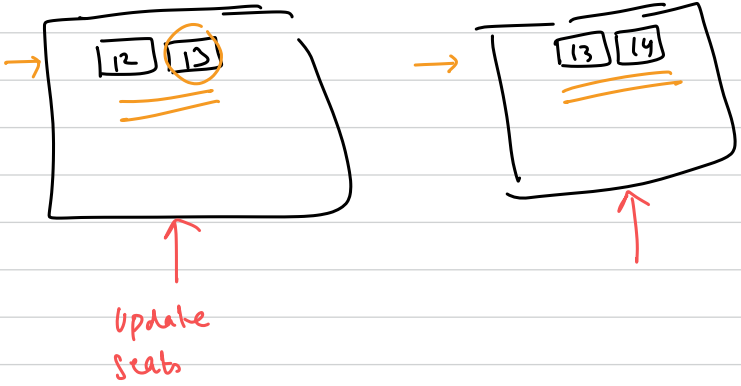
Rows appear magically if insertions were done & committed from other session. (S1)

S2 → will see the newly inserted rows if you try to update all rows

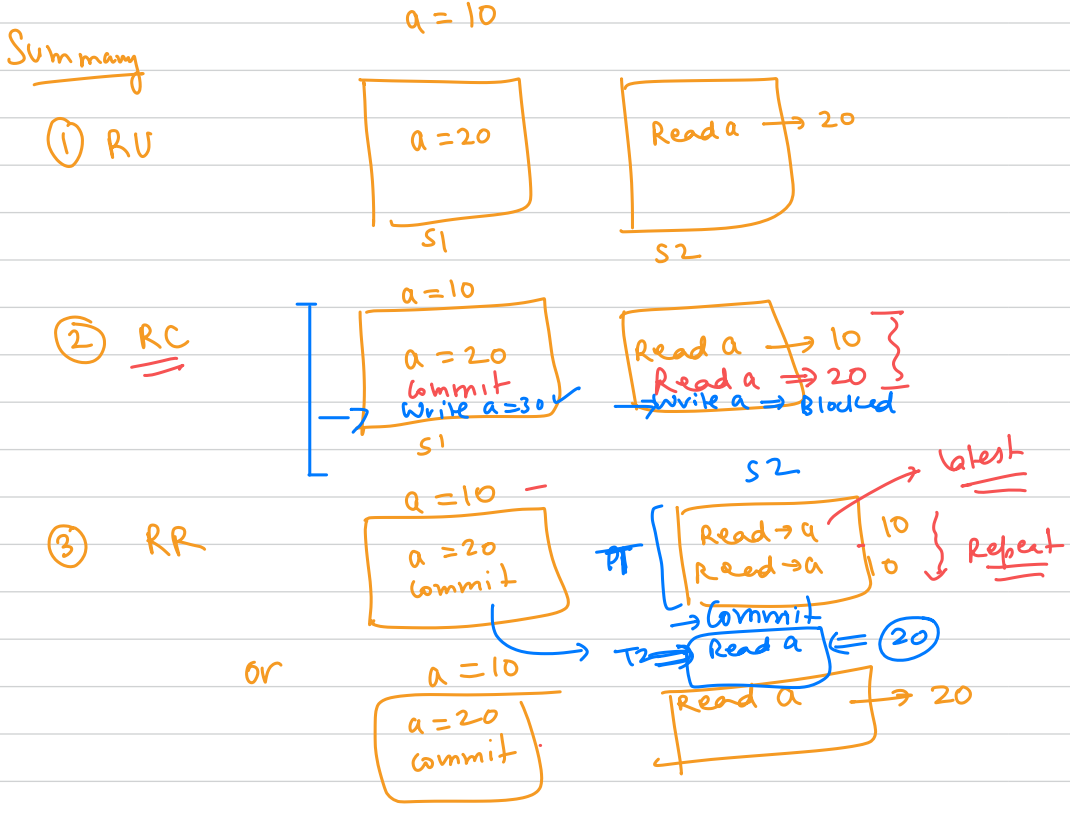
(4) Serializable

↳ sequential execution, one after the another




12
13
13
14
Update
seats

Summary


a = 10
① RU
a = 20
S1
Read a → 20
S2
② RC
a = 10
a = 20
commit
Write a = 30
S1
Read a → 10
Read a → 20
Write a → Blocked
③ RR
a = 10
a = 20
commit
S2
Read → a
Read → a
10
10
Repeat
latest
PT
commit
T2
Read a
20
or
a = 10
a = 20
commit
Read a → 20

4 Serializable