TRANSACTIONS - 02

AGENDA

Isolation levels
- Read Committed
- Repeatable Read
- serializable
- Deadlocks in sql

start by 9:05 PM IST

Isolation level - what you as user would Read.

Read uncommitted - Reads latest data.

↳ Problem → dirty Read.

2.) Read Committed: [PSQL]

→ Always Read latest Committed values ONLY.

1. S1: start (Read UNCOMM)
2. S1: select * .....
3. S2: start (Read COMM)
4. S2: select * ....
5. S1: update value, but x commit
6. S2: select * .... - last committed value

solves my DIRTY Read issue. ✓

*) PROBLEMS:

users

id	Name	Psp	email-sent	
1	Yash	60	F	T
2	Aman	79•	F	T
3	Rahul	70	F	T
4	Puomod	45	F	T

Q.) sendemail to all students
where psp < 80;
then: update email-sent to true

solution
= S1.) Get all users with psp < 80 ✓ (3)
S2.) send EMail ✓ sent
S3.) update the flag — where <80.
=

Isol. level → Read Commited.

Psp decrease to 79 — after S2.

Problem → flag updated but EMail Not send.

is called : NON Repeatable Read.

Read uncommitted < Read committed <
Repeatable Read < serializable

Most unlent ↑ ↓ Most strict

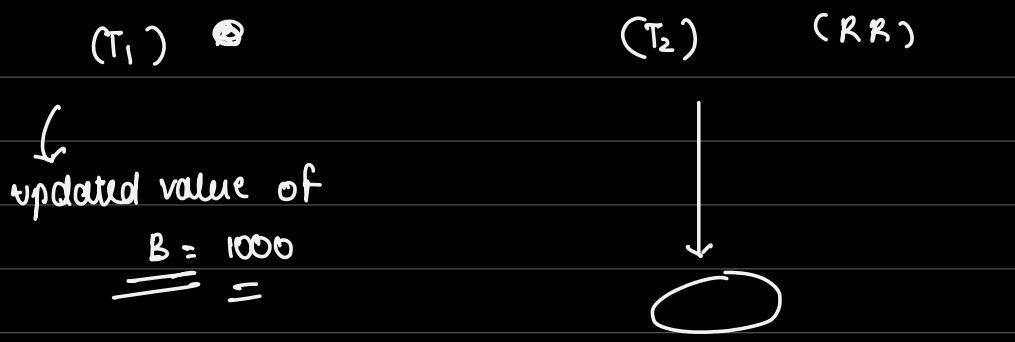
3.) Repeatable Read: (default in MYSQL)

when Reading for 1st time → Read Latest committed value.

throughout transaction keep Reading same value

•) MYSQL takes snapshot of table
→ keeps same table snapshot.

Bank: initially B = 500.



Solves: NON Repeatable Read ✓

*) PROBLEM:

① Bank Transactions still Not work

2.)

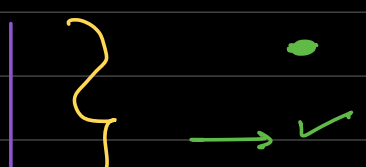
users

id	Name	psp	email_sent

(same Q as above.)

(3) send Email

s1.) select * ....
s2.) send Email
s3.) update email_sent

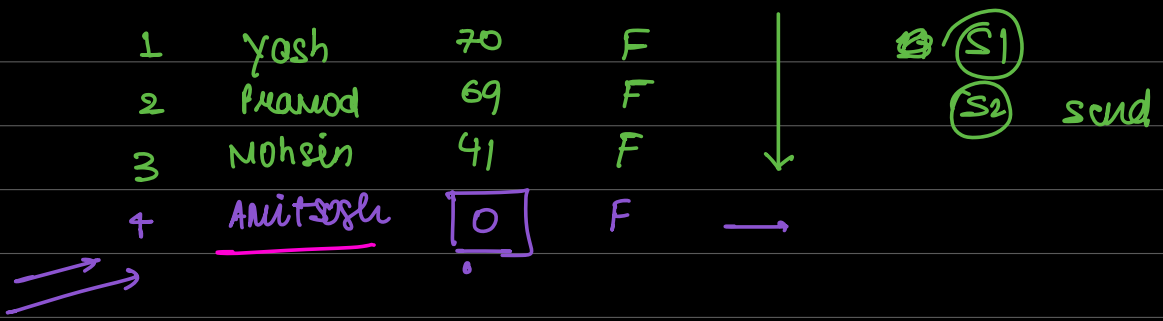


R.R. says: I will take snapshot of existing Rows.

x tell about New Rows added.

New stud Registered after s2 (New Row)

S3 ⟶ you will update.



update ⟶ for all Rows !!

↓

Problem ⟶ PHANTOM READS

Phantom Read → when a New Row is inserted in table, Even R.R. Isolation doesn't take care of it — why- because it deals with snapshot of existing data.

Read UNCOMM

↓

DIRTY Read ⟶ Read COMMitted

↓

N.R.R. ⟶ Repeatable Read

↓

Phantom Read.

serializable ⟵

10:42 PM IST

4.> SERIALIZABLE:

⟶ Anything where you want to use locks - uses serializable.

"one at a time"

if there are NO concurrency → NO ISSUE.

↳ serializable Iso. level.

lock ON JUST write Queries → shared lock (apply to all isolation levels)

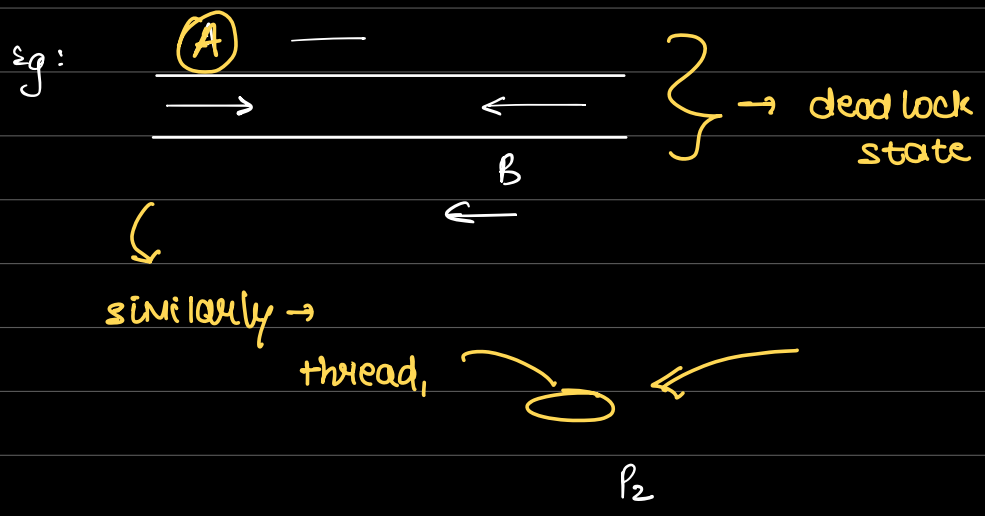
exclusive lock → only by serializable
× Read × write.

disadvantage → queries would be slow.

How do you tell MySQL → 'for update'
↓
exclusive lock.

eg: select * from film where id = 20
for update;

*) DEADLOCK:



two process working

① $P_1 \rightarrow$ 12 lock ② $P_2 \rightarrow$ 13 lock ✓

③ $P_1 \rightarrow$ 13 lock ④ $P_2 \rightarrow$ 12 lock

dead lock !!

two threads wait for resources on each other