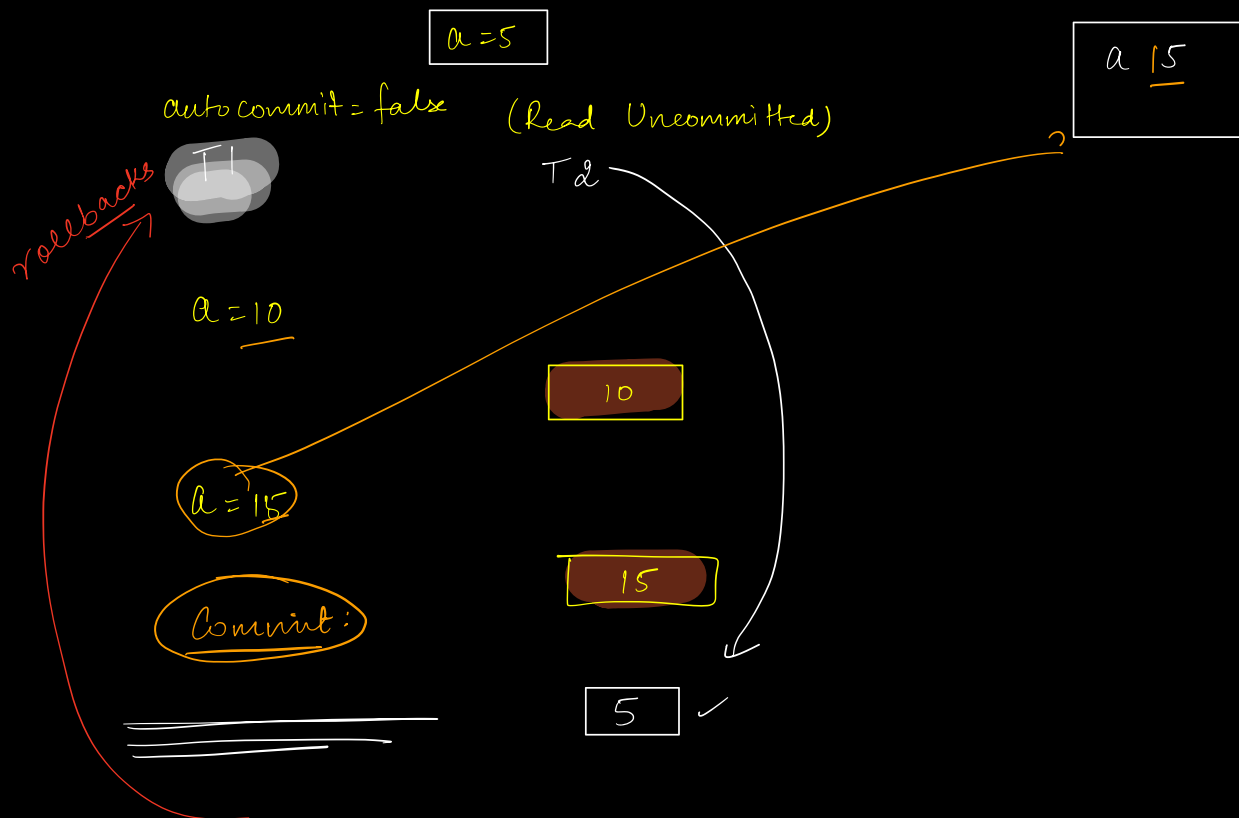


Transaction Isolation Levels :-

Read Uncommitted :- Allows to read uncommitted data from other transaction.



Dirty Read :- Reading some data that is not confirmed yet.

Read Committed :- Your transaction will only read latest committed data.

Someone

Read Uncommitted

updating a row

locks that
row for any
write

Anyone

Read ✓

Write X



id	Name	psp	email_Sent
1	A	60 ✓	false ✓
2	B	80	false ✓
3	C	81 70 ✓	false ✓
4	D	45 ✓	false ✓

1
3
4

Send an email to all st whose psp < 80 & update email_sent to true for them.

1)

Select * from Students
where psp < 80

psp

✓ 1 60 ✓
✓ 3 70 ✓
✓ 4 45 ✓

2) Send email to all these students (work) 5 min

3)

update Students
Set email_Sent = true
where psp < 80 x

}

1	60
3	81
4	45

 →

The problem is known as **Non Repeatable Read**
different

which will be solved by

Repeatable Read :-

Within a transaction if i read the same
row again, it must have same value.

Only guaranteed for row that you have read

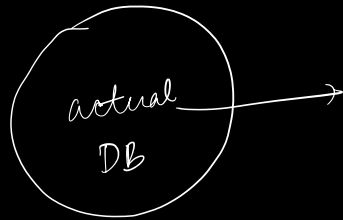
Read Uncommitted :- Dirty Read

Read Committed :- Non Repeatable Read

Repeatable Read :- Phantom Read

Default for MySQL

Serializable ✓



for first time,

whatever rows you read,

creates a snapshot for them

it continues to use

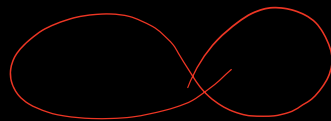
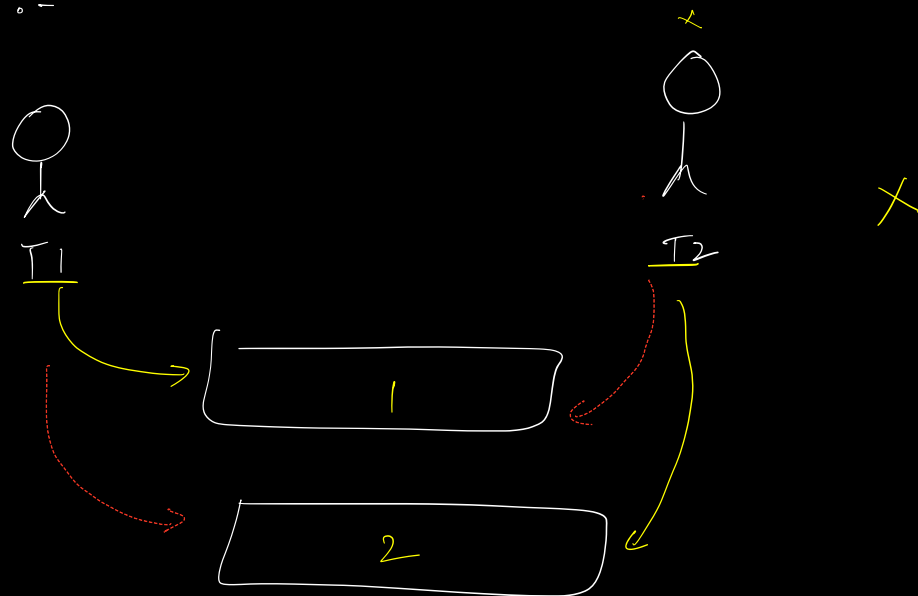
the snapshot for that

transaction.

Phantom Read - A row magically comes
that you had not seen before.

Phantom = Ghost = Sudden occurrence

Deadlock :-



Deadlock

T_1
locks A
~~locks B~~

T_2
locks B
~~locks A~~

T_1 will not get a lock until T_2 is done

& vice versa.

SOL detects deadlock, it automatically rollsback one of the transactions.

Overall :- Read Uncommitted

↳ Dirty Read
Read Committed

↳ Non Repeated Read
Repeatable Read
Phantom Read

Schema Design

2 classes