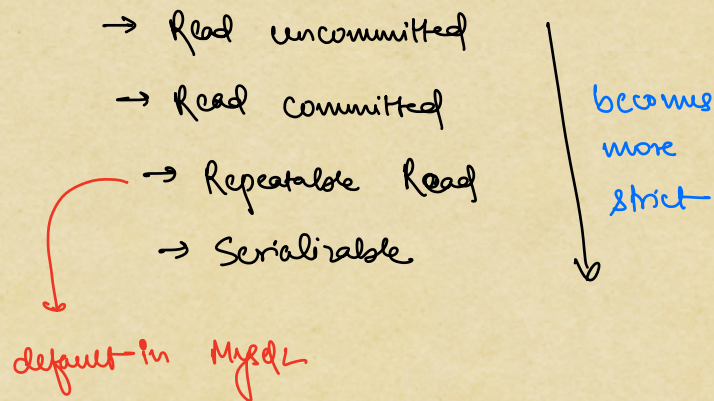


⇒ Handling failures and maintaining durability is done by 2 keywords ⇒ COMMIT and ROLLBACK

1) COMMIT : If the entire transaction goes on properly then to save the changes, we do commit.

2) ROLLBACK : If anything goes wrong or we are not able to move forward with the transaction, then to go back to the initial state and undo all changes, we do ROLLBACK.

How transactions handle concurrency :-



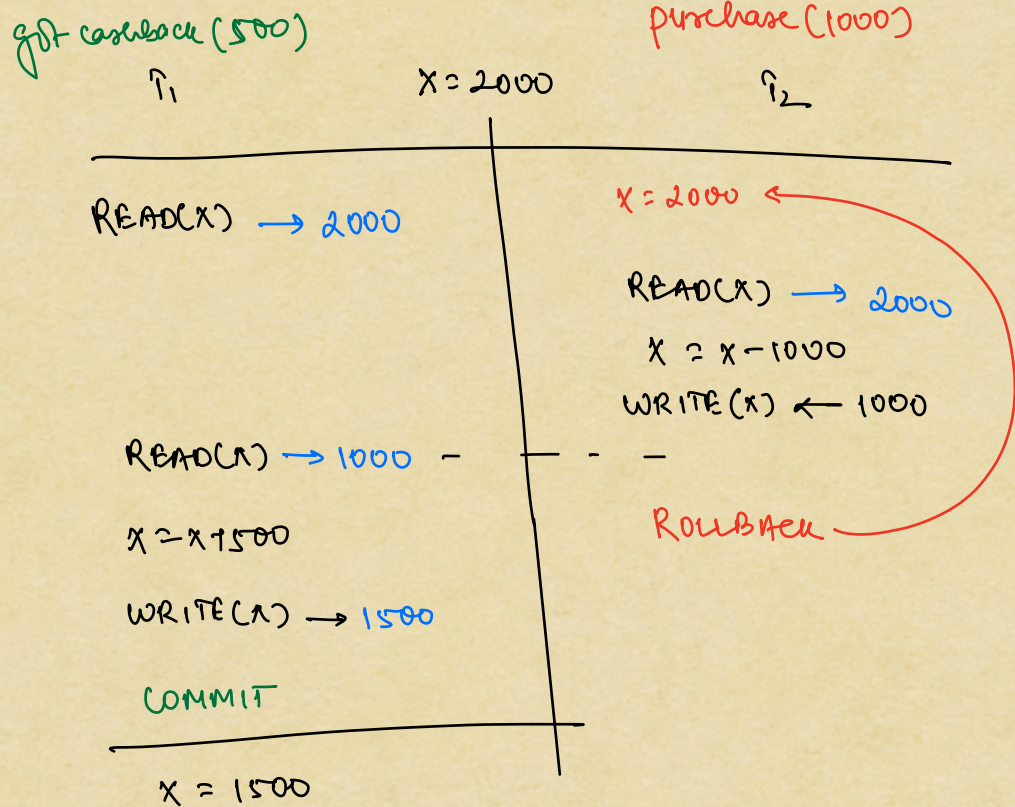
C → Create → Write

R → Read → Read

U → Update → Write

D → Delete → Write

⇒ READ UNCOMMITTED:



Whereas actual
should have been 2500

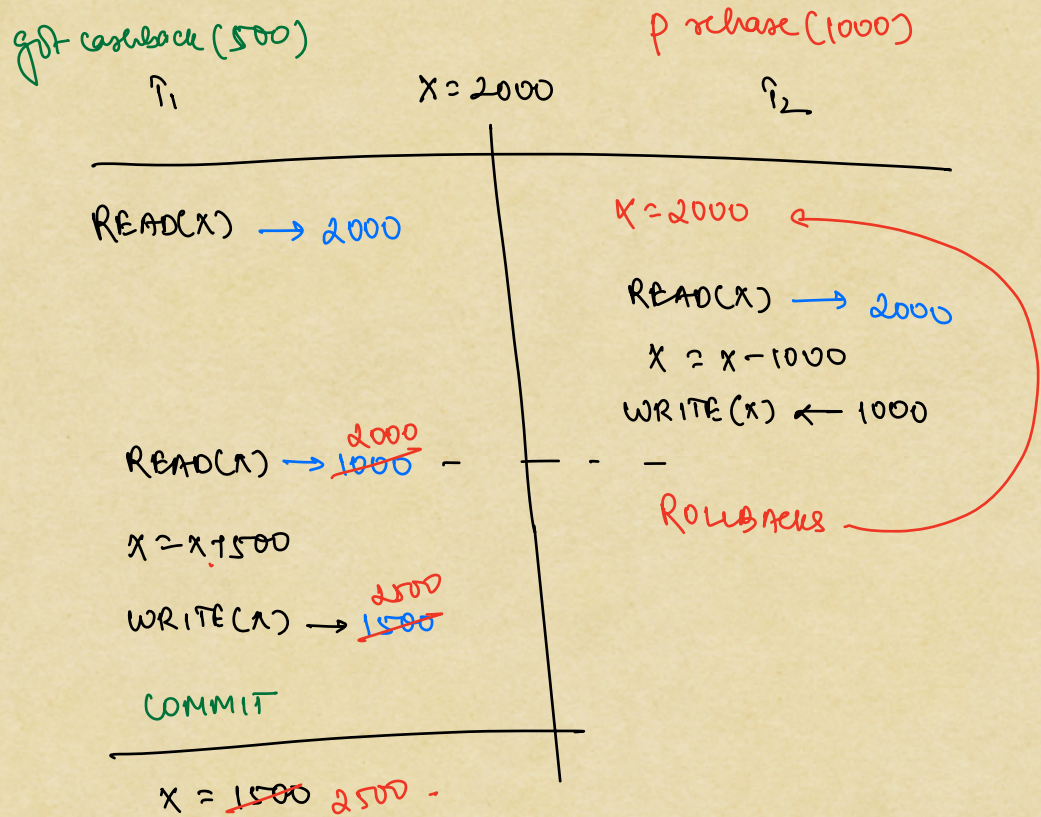
In Read Uncommitted, as the name suggests, another transaction can read uncommitted changes of another transaction.

RUC allows dirty reads.

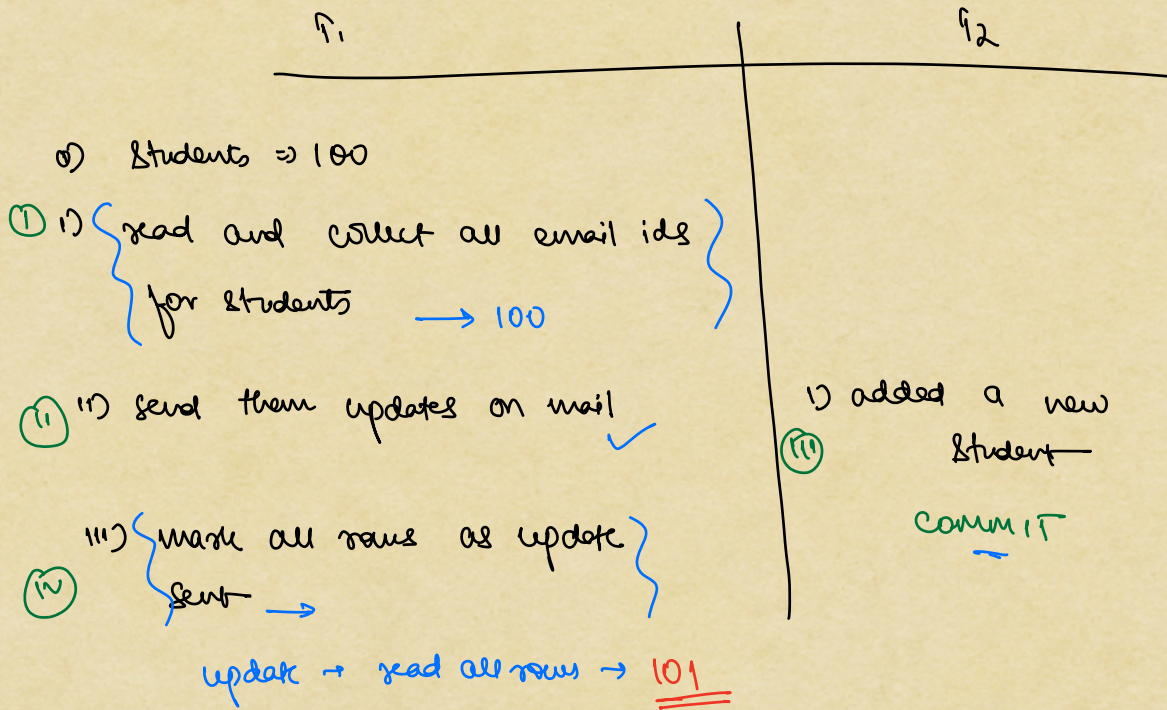
• reading uncommitted data that might/might not get committed.

RUC, since it doesn't take any lock, it's highly optimised.
 we can use it for places where dirty reads will not be
 an issue.

⇒ READ COMMITTED: As the name suggests we only
 read committed data.



⇒ REPEATABLE READ:-



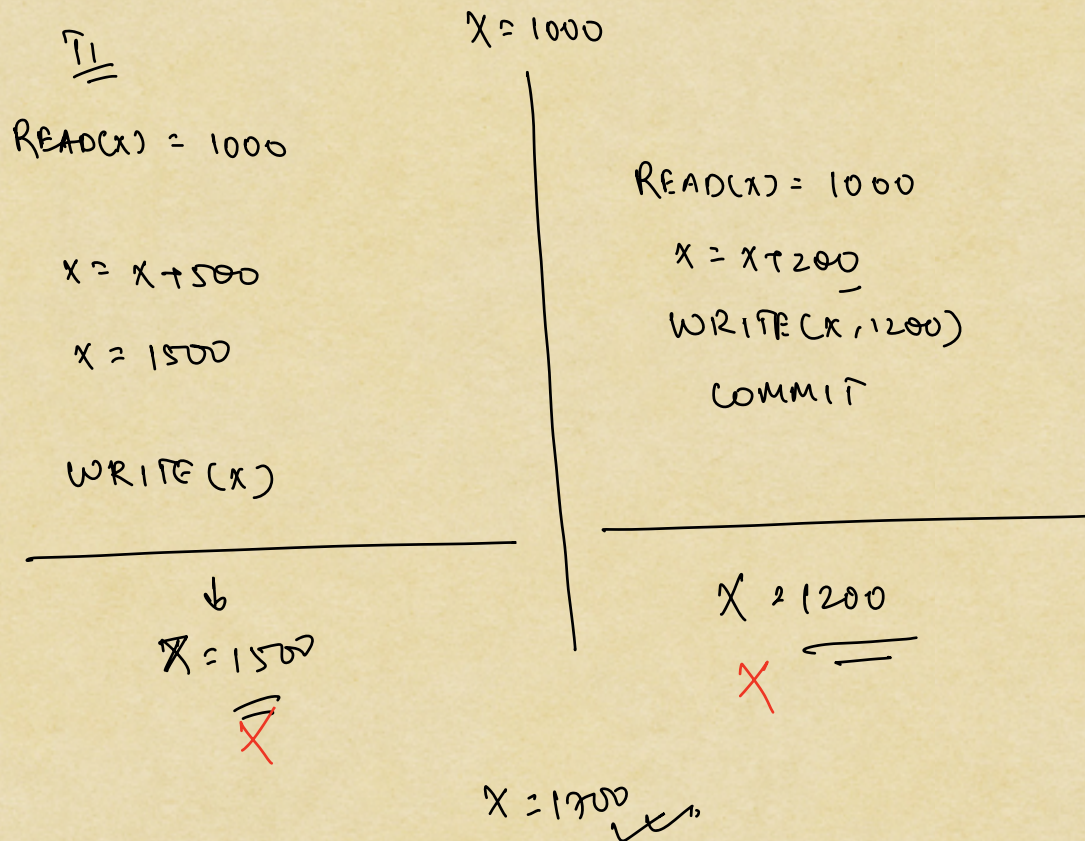
We want to read the same data

that we read in step ①, also in step ③

To read the same data within a transaction, we

will use REPEATABLE READ

↓
default isolation
level in mysql



4) SERIALIZABLE:

- * uses lock behind the scenes
- * if one transaction has a lock over the data, no other transactions will be allowed to even read that row.
- * each transaction will have to wait until another trans. gets over.
- * As locks are involved, system will be slower.

* How Serializable works :

- * Lock the row and won't allow any reads if I am updating the row.
- * it will read (just read) the data before the transaction.
- * it will never read for update