$$\begin{tikzcd} A \arrow[d] & C \arrow[dd] & I \arrow[d] & D \arrow[d] \\ \text{Atomicity} & & \text{Isolation} & \text{Durability} \\ & \text{Consistency} & & \end{tikzcd}$$

<u>Isolation</u>

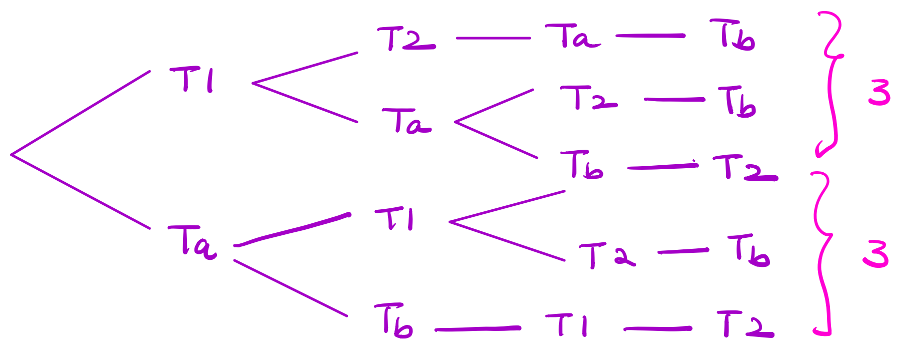
* database guarantees that transactions on <u>same</u> machine are ordered by time.

  user: $\xrightarrow{Ta\ ;\ Tb}$

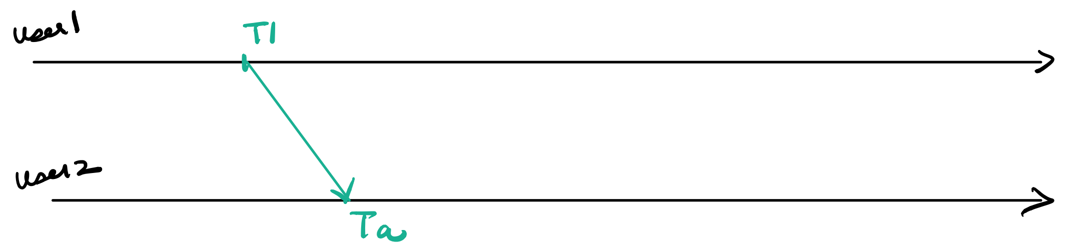
* Txs on different machines may be interleaved, but output must be equivalent to some serial order of all Txs.

<u>Example</u>: user1 : T1 ; T2
user 2: Ta ; Tb

List all possible serial orders:



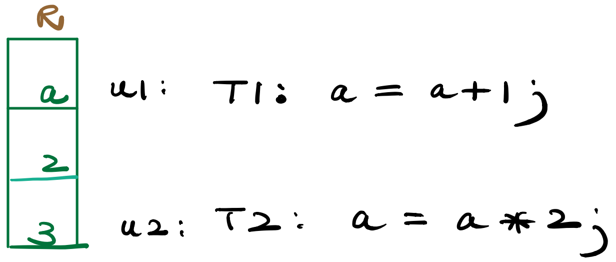
6 possible serial orders.



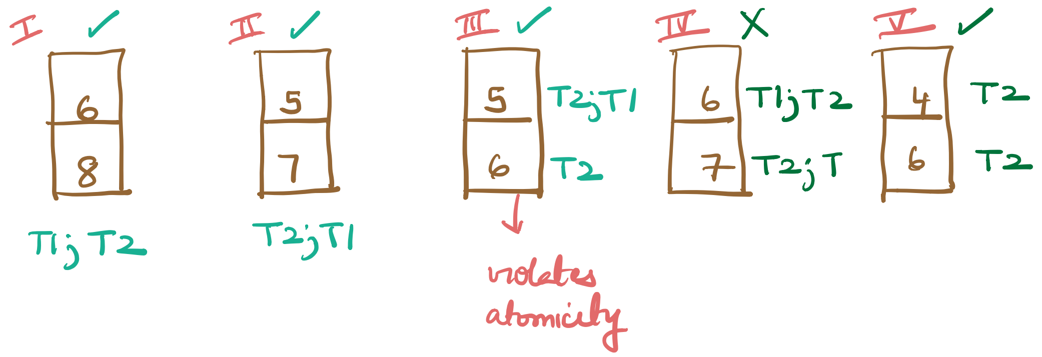
* want Ta to execute after T1; different computers

* must be done at application level.

Question :



Which output(s) do NOT violate isolation?



MySQL TX

I * TXs begin automatically on first SQL statement

* by default, every query is a separate TX.

AUTO COMMIT is executed at the end of every query.

↓ written safely to disk

SET AUTOCOMMIT = 1; (default)

II SET AUTOCOMMIT = 0;

T1 { Q1: update dept set dname = 'CS' where dname = 'EE';
Q2: select * from dept; → shows the update

ROLLBACK;

T2 { Q3: select * from dept; → shows the original dept 'EE'

COMMIT; OR SET AUTOCOMMIT = 1 } ending T1 with a commit

III START TRANSACTION

Q1

Q2

⋮

Qn

COMMIT or ROLLBACK

Isolation is achieved with locks.

locks $\Rightarrow$ serializability $\Rightarrow$ Performance $\downarrow$
(less concurrency)

Allow concurrency of Reads (when possible)

ISOLATION Levels : Reads

SET TRANSACTION ISOLATION LEVEL <level>

START TRANSACTION

⋮

COMMIT or ROLLBACK

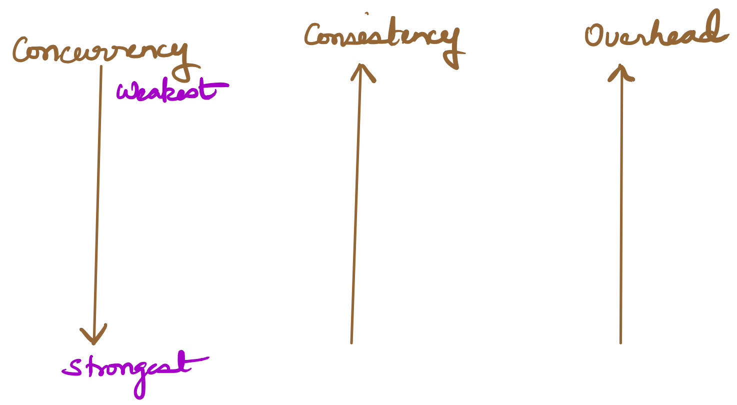
<level>

Serializable

Repeatable Read

Read Committed

Read Uncommitted



* isolation levels are for reads (not writes)
* each TX has its own isolation level.
* Ui's isolation level only affects Ui.
* isolation controls the possible sequence of outcomes, not the possible sequence of executions.