Concurrency control protocols

The concurrency control protocols ensure the atomicity, consistency, isolation, durability and serializability of the concurrent execution of the database transactions. Therefore, these protocols are categorized as:

1. Lock Based concurrency control protocol
2. Time stamp concurrency control protocol
3. validation Based concurrency control protocol.

1. Lock Based concurrency control protocol

In this type of protocol, any transaction cannot read or

write data until it acquires an appropriate lock on it. There are two types of lock.

① Shared Lock:-

→ It is also known as a Read only lock. In a shared lock, the data item can only read by the transaction.

→ It can be shared between the transactions.

② Exclusive lock:- In the exclusive lock,

→ the data item can be both reads as well as written by the transaction.

→ This lock is exclusive, and in this lock, multiple transactions do not modify the same data simultaneously.

There are four types of lock protocol available!

① Simplistic lock protocol :-

It is the simplest way of locking the data while transaction.

Simplistic lock-based protocols allow all the transactions to get the lock on the data before insert or delete or update on it. It will unlock the data item after completing the transaction.

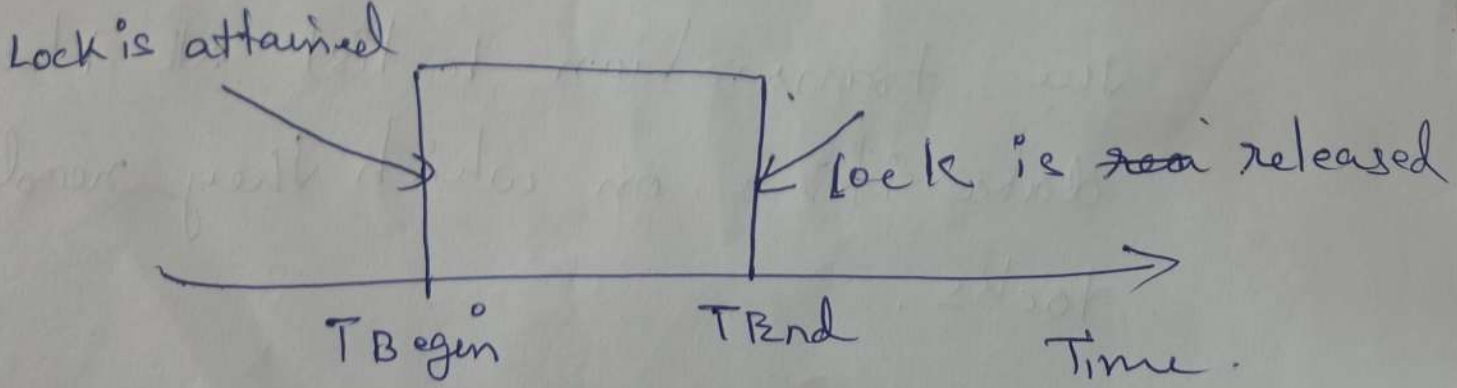
② Preclaiming Lock protocol :-

→ preclaiming lock protocols evaluate the transaction to list all the data items on which they need locks.

→ Before initiating an execution of the transaction, it requests DBMS for all the lock on all those data items.

→ If all the locks are granted then this protocol allows the transaction to begin. when the transaction is completed then it releases all the lock.

→ If all the locks are not granted then this protocol allows the transaction to rolls back and waits until all the locks are granted.



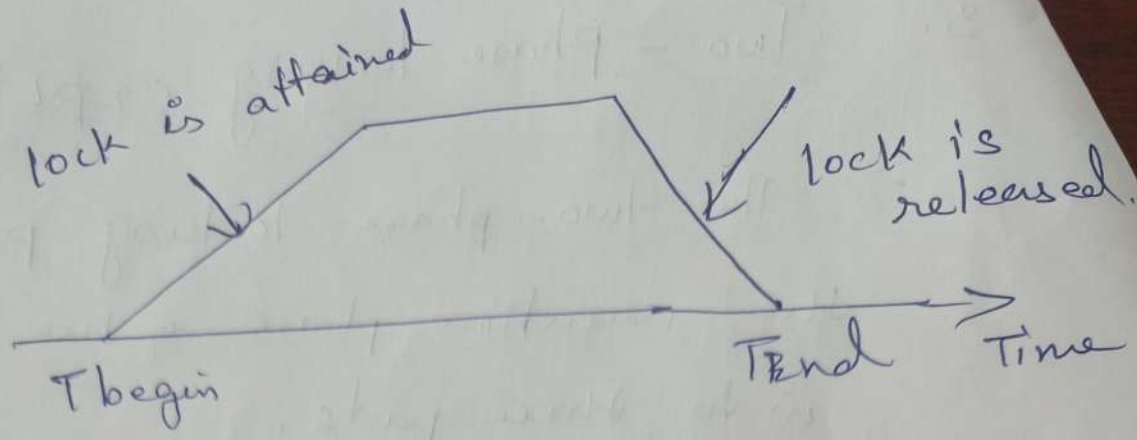
3. Two - phase locking (2PL)

→ The two-phase locking protocol divides the execution phase of the transaction into three parts.

→ In the first part, when the execution of the transaction starts, it seeks permission for the lock it requires.

→ In the second part, the transaction acquires all the locks. The third phase is started as soon as the transaction releases its first lock.

→ In the third phase, the transaction cannot demand any new locks. It only releases the acquired locks.



There are two phases of 2PL

growing phase :- In the growing phase, a new lock on the data item may be acquired by the transaction, but none can be released.

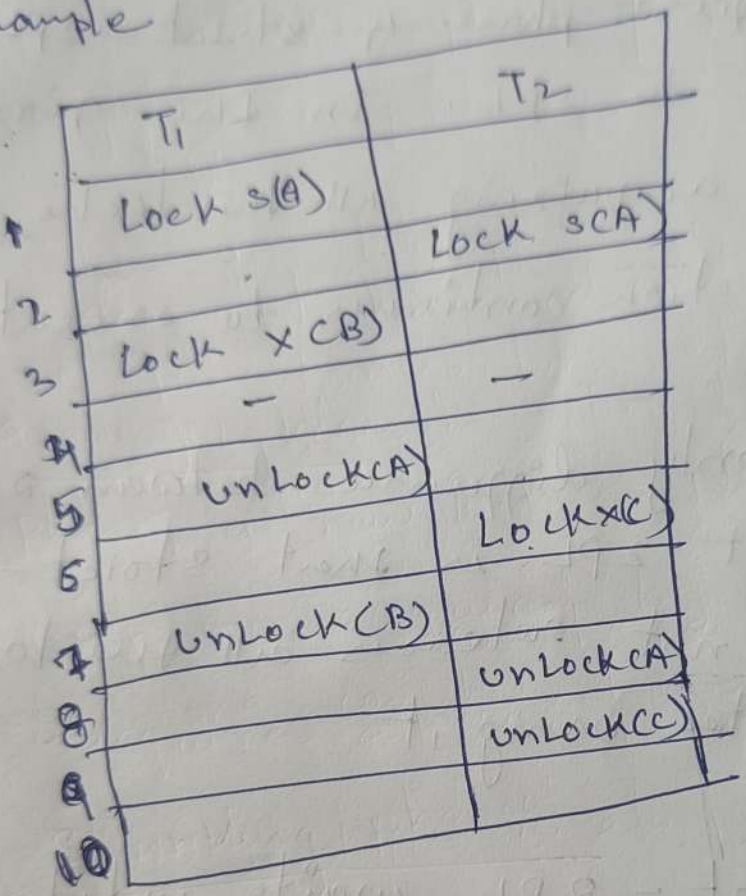
shrinking phase :- In the shrinking phase, existing lock held by the transaction may be ~~reaker~~ released, but no new locks can be acquired.

Lock conversion S -> sharing lock
X -> exclusive lock

(1) upgrading a lock (from S(a)) to X(a) is allowed in growing phase

(2) Downgrading of lock (from x(a) to s(a)) must be done in shrinking phase.

example



Transaction T1:

Growing phase: from step 1-3

shrinking phase: from step 5-7

Lock point: at 3

Transaction T2:

Growing phase: from step 2-6

shrinking phase: from step 8-9

Lock point at 6

Validation

(4) Strict Two-phase locking (strict-2PL)

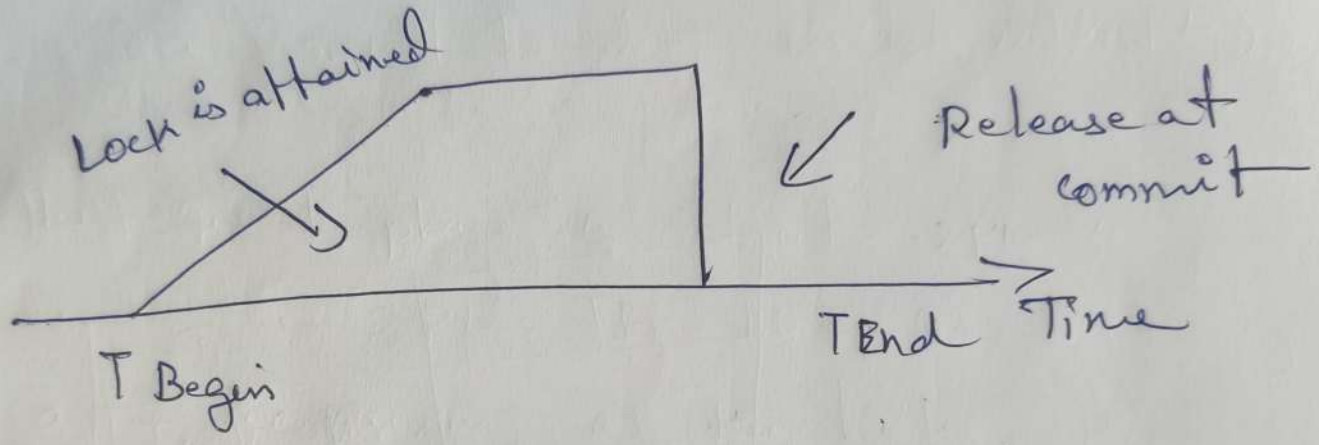
→ The first phase of strict-2PL is similar to 2PL. In the first phase, after acquiring all the locks, the transaction continues to execute normally.

→ The only difference between 2PL and strict 2PL is that strict-2PL does not release all the locks after using it

→ strict-2PL waits until the whole transaction to commit, and it releases all the locks at a time

→ strict 2PL protocol does not have shrinking phase of lock release.

(3) Validation Based protocol



It does not have cascading abort as 2PL does.