DBMS

K. Veerababu

Assignment - 4

2405A0513

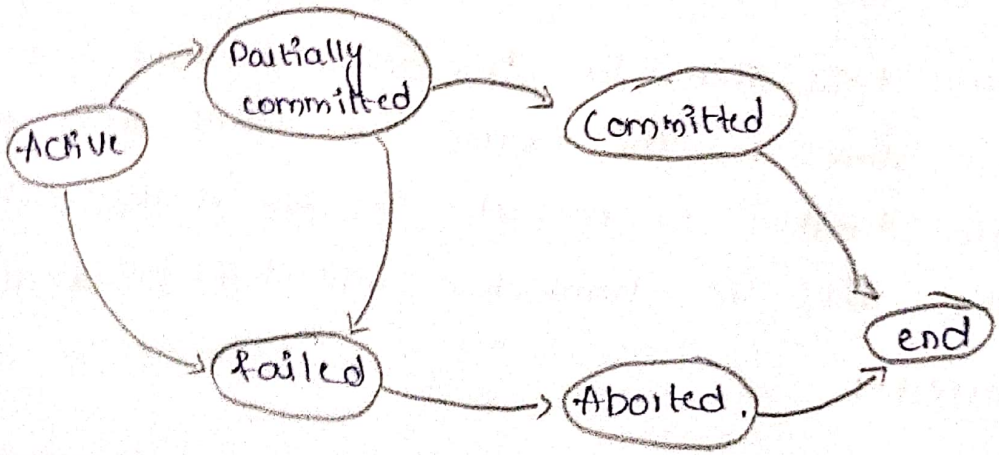
1. What is transaction in database? Explain different states of transaction its execution.

Transaction: The transaction is a set of logically related operation. It contains a group of tasks.

A transaction is an action or series of actions, it is performed by a single user to perform operations for accessing the contents of the database.

States of transaction:

In a database, the transaction can be in one of the following states.



Active state:

The active state is the first state of every transaction in this state, the transaction is being executed.

For example: Insertion or deletion or updating a record is done here. But all the records are still not saved to the database.

Partially committed:

In the partially committed state, a transaction executed its final opuation, but the data is still not solved to the database in the total mark calculation example, a final display of the total marks step is executed in this state.

committed: A transaction is said to be in a committed state if it executes all its opuations successfully. In this state all the effects are now permanently saved on the database system.

failed state:

In any of the checks made by the database recovery system fails then the transaction is said to be in the failed state. In the example of total marks calculation if the database is not able to fire a query to fetch the marks, then the transaction will fail to execute.

Aborted:

If any of the checks fail and the transaction has reached a failed state then this database recovery system will make sure that the database is in its previous consistant state.

2. Discuss serializability in detail.

The serializability of schedules is used to find non-serial schedules that allows the transaction to execute concurrently without intutering with one another.

It identifies which schedule are correct when executions of the transaction have interleaving of their operations. A non-serial schedule will be serializable if its result is equal to the result of its transactions executed serially.

Types of Serializability.

Serializability can be categorized into two main types:

1. Conflict Serializability
2. View Serializability.

1. **Conflict serializability:** two schedules are said to be conflict equivalent if all the conflicting operations in both the schedule got executed in the same order. If a schedule conflict equivalent to its serial schedule then it is called conflict serializability schedule. A schedule will be conflict serializability if it is conflict equivalent to a serial schedule.

Conflicting operations:

The two operations become conflicting if all conditions satisfy both belong to operate transactions.

2. **View serializability:** two schedules are said to be view equivalent if the order of initial read, final write and update operations is the same in both the schedules. If a schedule is view equivalent to its serial schedule then it is called view serializable if it view equivalent to a serial schedule.

A schedule will view serializable if it view equivalent to a serial schedule.

Serializability Techniques in DBMS

Serial concurrency control mechanisms enforce serializability

1. two-phase locking:

* Divides transaction into execution into:
  - → Growing phase: only acquiring locks
  - → shrinking phase: only releasing locks
* ensures conflict-serializability but can lead to deadlocks.

2. Timestamp ordering:

* Transactions receive timestamps based on their start time
* Ensures serial execution order according to timestamps.

3. Optimistic concurrency control:

* Allows transactions to execute freely and checks for conflicts before commit.
* if conflicts are found, transactions are rolled back and restarted.

Example of Serializability:

Consider two transactions:

* T_1: Read (A), A = A+10; write (A)
* T_2: Read (A), A = A×2, write (A).

Non - serialitable schedule.

step	Transaction	opuation.
1	T_1	Read (A)
2	T_2	Read (A)
3	T_1	write (A) (A = A+10)
4	T_2	write (A) (A = A*2)

* This leads to an incorrect result due to lost update.

Serialitable schedule.

step	Transaction	opuation
1	T_1	Read (A)
2	T_1	write (A) (A=A+10)
3	T_2	Read (A)
4	T_2	write (A) (A=A*2)

* Here, the outcome is consistent and correct.

3. Explain about lock based protocal with example.

Lock - Based protocol :

In this type of protocol, any transaction cannot read or write data until it acquires an appripriate lock on it there are two types of lock.

1. Shared lock
2. Exclusive locks.

1. Shared lock : It is also known as a Read-only lock. In a shared lock, the data item can only read by the

transactions because when the transaction holds a lock then it can't update the data on the data item.

2. Exclusive lock:

In this exclusive lock, the data item can be both read as well as write by the transaction.

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

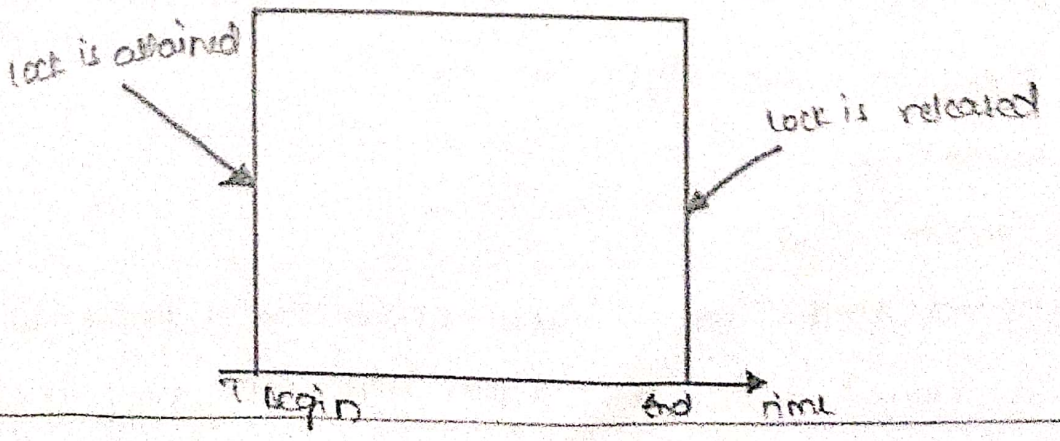
Lock Protocols:

1. Simplistic lock protocol:

It is the simplest way of locking the data while transactions simplistic lock-based protocols allow all the transactions to get simplistic lock on the data before insert or delete or update on it. It will unlock the data item after completing the transaction.

2. Pre-claiming lock protocol:

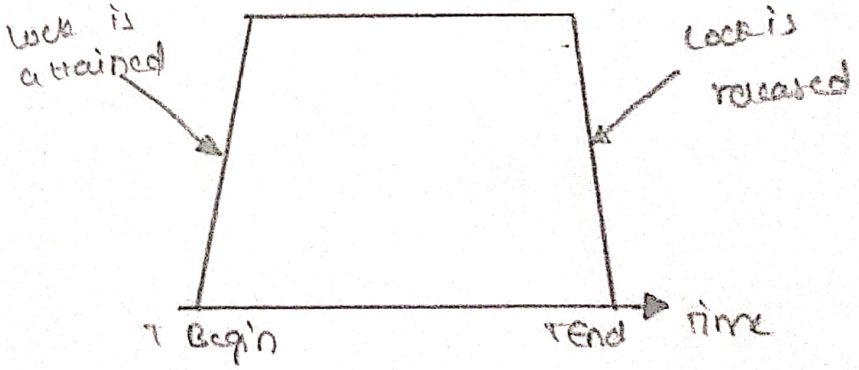
Pre-claiming lock protocols evaluate the transaction to list all the data items on which they need locks before. Initializing on execution of the transaction, it requests DBMS for all the lock on all those data items.



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 only new locks it only releases the acquired locks.



Example :

	T_1	T_2
0	lock-S(A)	
1		lock-S(A)
2	lock-X	
3	—	—
4	unlock(A)	
5		lock-X(C)
6	unlock(B)	
7		unlock(A)
8		unlock(C)
9	—	

4. Strict Two-phase locking:

The first phase of Strid-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 a lock after using it strict-2PL waits until the whole transaction to commit and then it releases all the locks at a time

Strict-2PL protocol does not have shrinking phase of lock release.