Concurrency Control

By :
Dr. Rinkle Rani
Associate Professor, CSED
TIET, Patiala

Concurrency Control

˝ **Concurrency Control** in Database Management System is <mark>a procedure of managing simultaneous operations without conflicting with each other</mark>.

˝ It ensures that Database transactions are performed concurrently and accurately to produce correct results <mark>without violating data integrity</mark> of the respective Database.

˝ Concurrent access is quite easy if all users are just reading data. There is no way they can interfere with one another. Though for any practical Database, it would have a mix of READ and WRITE operations and hence the concurrency is a challenge.

DBMS Concurrency Control is used to address such conflicts, which mostly occur with a multi-user system.

Therefore, Concurrency Control is the ==most important== element for proper functioning of a Database Management System where ==two or more database transactions are executed simultaneously,== which require access to the same data.

Why use Concurrency method?

Reasons for using Concurrency control method is DBMS:

˝ To ==apply Isolation== through mutual exclusion between conflicting transactions
˝ To ==resolve read-write and write-write== conflict issues
˝ To ==preserve database consistency== through constantly preserving execution obstructions
˝ The system needs to ==control the interaction== among the concurrent transactions. This control is achieved using concurrent-control schemes.
˝ Concurrency control helps to ==ensure serializability==

Concurrency Control Protocols

Different concurrency control protocols offer different benefits between the amount of concurrency they allow and the amount of overhead that they impose. Following are the Concurrency Control techniques in DBMS:

1. Lock-Based Protocols

2. Two Phase Locking Protocol

3. Timestamp-Based Protocols

4. Validation-Based Protocols

Lock-based Protocols

˝ A lock is a data variable which is associated with a data item. This lock signifies that operations that can be performed on the data item.

˝ Lock Based Protocols in DBMS is a mechanism in which a transaction cannot Read or Write the data until it acquires an appropriate lock.

˝ Lock based protocols help to eliminate the concurrency problem in DBMS for simultaneous transactions by locking or isolating a particular transaction to a single user.

Locks are of two kinds

    a) Binary Locks

    b) Shared/exclusive

Binary Locks   A Binary lock on a data item can either ==locked or unlocked states==.

Shared/exclusive
Shared Lock (S):
A shared lock is also called a ==Read-only lock==. With the shared lock, the data item can be shared between transactions. This is because you will never have permission to update data on the data item.

For example, consider a case where two transactions are ==reading the account balance of a person==. The database will let them read by placing a shared lock. However, if another transaction wants to update that account's balance, shared lock prevent it until the reading process is over.

Exclusive Lock (X):

With the Exclusive Lock, a data item can be read as well as written. This is exclusive and can't be held concurrently on the same data item. X-lock is requested using lock-x instruction. Transactions may unlock the data item after finishing the 'write' operation.

For example, when a transaction needs to update the account balance of a person. You can allows this transaction by placing X lock on it. Therefore, when the second transaction wants to read or write, exclusive lock prevent this operation.

Simplistic Lock Protocol

Simplistic lock-based protocols allow transactions to obtain a lock on every object before a 'write' operation is performed. Transactions may unlock the data item after completing the 'write' operation.

Pre-claiming Lock Protocol

Pre-claiming protocols evaluate their operations and create a list of data items on which they need locks. Before initiating an execution, the transaction requests the system for all the locks it needs beforehand. If all the locks are granted, the transaction executes and releases all the locks when all its operations are over. If all the locks are not granted, the transaction rolls back and waits until all the locks are granted.

Problems with Pre-claiming Lock Protocol

Starvation

Starvation is the situation when a transaction needs to wait for an indefinite period to acquire a lock.

Deadlock

Deadlock refers to a specific situation where two or more processes are waiting for each other to release a resource or more than two processes are waiting for the resource in a circular chain.
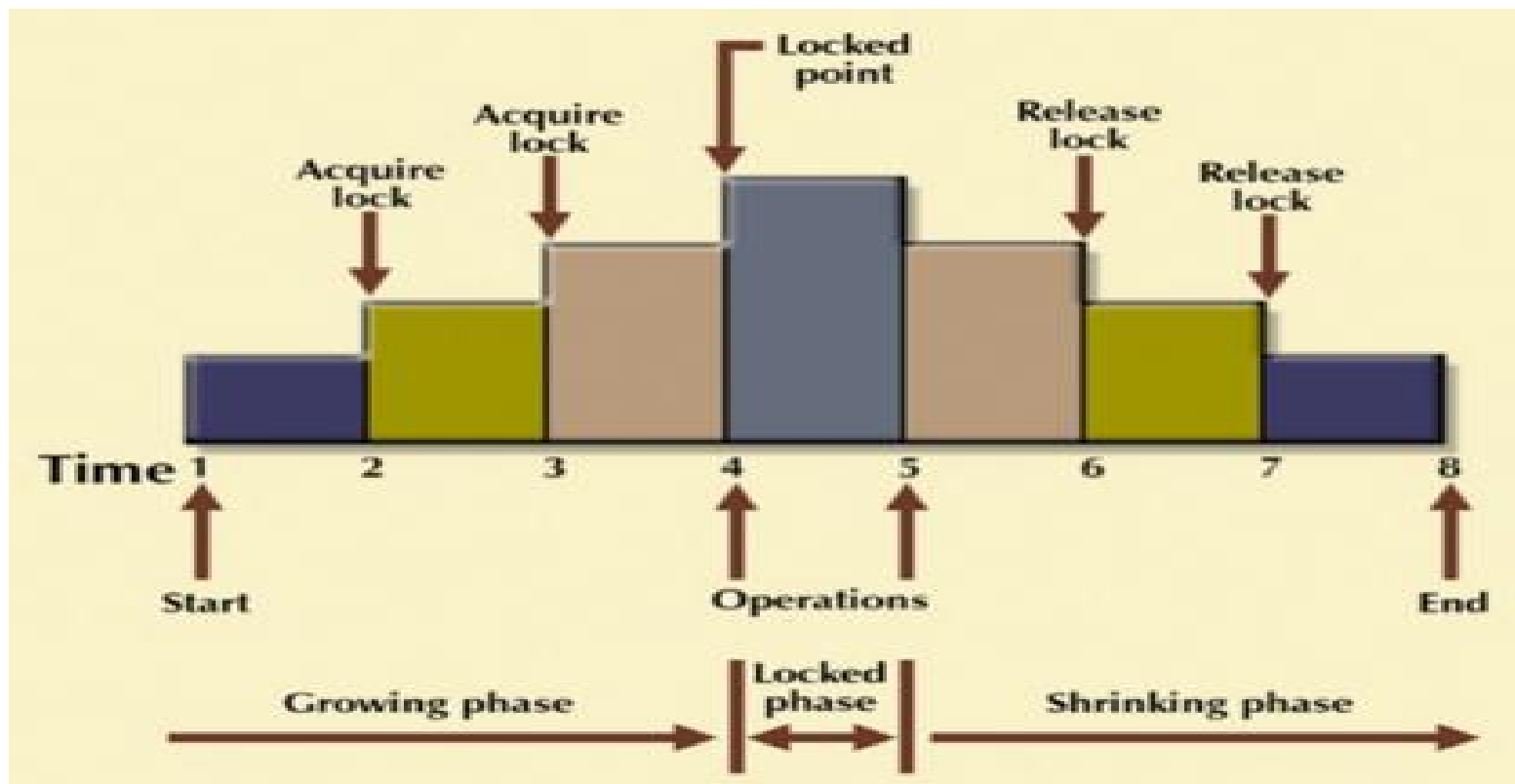
Two Phase Locking Protocol

˝ Two Phase Locking Protocol also known as 2PL protocol is a method of concurrency control in DBMS that ensures serializability by applying a lock to the transaction data which blocks other transactions to access the same data simultaneously.

˝ This locking protocol divides the execution phase of a transaction into three different parts.

- o In the first phase, when the transaction begins to execute, it requires permission for the locks it needs.
- o The second part is where the transaction obtains all the locks. When a transaction releases its first lock, the third phase starts.
- o In this third phase, the transaction cannot demand any new locks. Instead, it only releases the acquired locks.

A transaction is said to follow Two Phase Locking protocol if Locking and Unlocking can be done in two phases.

Growing Phase: New locks on data items may be acquired but none can be released.

Shrinking Phase: Existing locks may be released but no new locks can be acquired.

Strict Two-Phase Locking Method

˝ Strict-Two phase locking system is almost similar to 2PL. The only difference is that Strict-2PL never releases a lock after using it.

˝ It holds all the locks until the commit point and releases all the locks at one go when the process is over.

Timestamp-based Protocols

˝ The most commonly used concurrency protocol is the timestamp based protocol. This protocol uses either system time or logical counter as a timestamp.

˝ Lock-based protocols manage the order between the conflicting pairs among transactions at the time of execution, whereas timestamp-based protocols start working as soon as a transaction is created.

˝ Every transaction has a timestamp associated with it, and the ordering is determined by the age of the transaction. A transaction created at 0002 clock time would be older than all other transactions that come after it. For example, any transaction 'y' entering the system at 0004 is two seconds younger and the priority would be given to the older one.

Advantages:

a) Schedules are serializable just like 2PL protocols

b) No waiting for the transaction, which eliminates the possibility of deadlocks.

Validation Based Protocol

Validation based Protocol in DBMS also known as <mark>Optimistic Concurrency Control Technique is a method to avoid concurrency in transactions.</mark> In this protocol, the local copies of the transaction data are updated rather than the data itself, which results in less interference while execution of the transaction.

The Validation based Protocol is performed in the following three phases:

a) Read Phase

b) Validation Phase

c) Write Phase

Read Phase

In the Read Phase, the data values from the database can be read by a transaction but the write operation or updates are only applied to the local data copies, not the actual database.

Validation Phase

In Validation Phase, the data is checked to ensure that there is no violation of serializability while applying the transaction updates to the database.

Write Phase

In the Write Phase, the updates are applied to the database if the validation is successful, else; the updates are not applied, and the transaction is rolled back.