LOCKING PROTOCOLS

When multiple transactions occur at the same time, they might try to access the same data. Locking protocols help control how these transactions access data, ensuring that the database stays consistent and correct.

Locking protocols in a Database Management System (DBMS) manage access to data items to maintain consistency and isolation in a multi-user environment. They control how and when locks are applied to avoid conflicts such as lost updates, dirty reads, and uncommitted data.

Types of Locks

1. Shared Locks (S-Locks)

A shared lock allows multiple transactions to read a data item at the same time but prevents any from writing to it. This lock is also called a read-only lock and is requested using the lock-S instruction.

Purpose: Allows read-only operations to proceed without interference and ensures no modifications occur while data is being read.

Characteristics:

Multiple transactions can hold shared locks on the same data item simultaneously.

Prevents data modifications while shared locks are held.

Example: Transaction T1 and T2 both want to read the balance of account A. Both can hold a shared lock on account A at the same time.

2. Exclusive Locks (X-Locks)

An exclusive lock allows a transaction to both read and write a data item. It ensures that no other transaction can read or write the data item until the exclusive lock is released.

Purpose: Ensures complete control over a data item for modifications and prevents other transactions from accessing the data item.

Characteristics:

Only one transaction can hold an exclusive lock on a data item at any time.

Blocks both read and write access for other transactions.

Example: Transaction T1 wants to update the balance of account A. It acquires an exclusive lock on account A, preventing any other transaction from accessing account A until T1 releases the lock.