Concurrency control

The technique used to protect data when multiple users are accessing the same data concurrently (same time) is called concurrency control.

Types of concurrency control protocol
Concurrency control protocols can be broadly divided into two categories :

1. Lock-based protocols
2. Time-stamp based protocols

Lock-based Protocols :
Database systems equipped with lock-based protocols use a mechanism by which any transaction cannot read or write data until it acquires an appropriate lock on it.

Data items can be locked in two modes :

Exclusive (X) mode: Data items can be both read as well as written. X-lock is requested using lock-X instruction.

Shared (S) mode: Data items can only be read. S-lock is requested using lock-S instruction.

Lock-compatibility matrix

	S	X
S	true	false
X	false	false

A transaction may be granted a lock on an item if the requested lock is compatible with locks already held on the item by another transaction.
Any number of transactions can hold shared locks on an item, But if any transaction holds an exclusive on the item no other transaction may hold any lock on the item.

If a lock cannot be granted, the requesting transaction is made to wait till all incompatible locks held by other transactions have been released. The lock is then granted.

Example:

```
T1: lock-S(A); // Grant-S(A,T1)
read (A);
unlock(A);
lock-S(B); // Grant-S(B,T1)
read (B);
unlock(B);
display(A+B)
```

Timestamp based Protocol

Each transaction is issued a timestamp when it enters the system. If an old transaction Ti has time-stamp TS(Ti), a new transaction Tj is assigned time-stamp TS(Tj) such that TS(Ti) <TS(Tj).

The timestamp of transaction Tiis denoted as TS(Ti).
Read time-stamp of data-item X is denoted by R-timestamp(X).
Write time-stamp of data-item X is denoted by W-timestamp(X).

Two Phase Locking Protocol

This is a protocol which ensures conflict-serializable schedules.
Phase 1: Growing Phase
In this phase transactions may obtain locks but may not release locks.

Phase 2: Shrinking Phase
In this phase transaction may release locks,but may not obtain locks.

MERITS OF 2 PHASE LOCKING PROTOCOL
The two phase locking protocol ensures conflict serializability.
Consider any transaction, the point in the schedule where the transaction has obtained it's final lock is called the lock point of the transaction.
Now, transactions can be ordered according to their lock points. This ordering is a serializability ordering for the transactions.

Two-phase locking does not ensure freedom from deadlocks.

Cascading roll-back is possible under two-phase locking. To avoid this, follow a modified protocol called strict two-phase locking. Here a transaction must hold all its exclusive locks till it commits/aborts.

Rigorous two-phase locking is even stricter: here all locks are held till commit/abort. In this protocol transactions can be serialized in the order in which they commit.