# Lecture Notes Nov 12, 2011

By-

Ali Hussain Kadar (08012304) Sahil Aggarwal (08012318)

### Locking Revisited <u>The Two-Phase Locking Protocol</u> (2PL) ensures serializability.

This protocol requires that each transaction issue lock and unlock requests in two phases:

**1. Growing phase**. A transaction may obtain locks, but may not release any lock.

**2. Shrinking phase**. A transaction may release locks, but may not obtain any new locks

Initially, a transaction is in the growing phase. The transaction acquires locks as needed. Once the transaction releases a lock, it enters the shrinking phase, and it can issue no more lock requests.

**Simple Two-phase locking**: All locks are acquired in a transaction before any lock is released



Strict Two-phase locking:



In strict 2PL first the read locks are released and then after the transaction is complete all write locks are released.

## **Rigorous Two-phase locking:**



In rigorous 2PL the transaction holds all locks until completion after which all locks are released.

One advantages of rigorous 2PL is that No cascading Rollback is needed however the resource utilization is reduced.

### Undo Redo - Two phases to recovery.

In case of a crash the DBMS resumes from the last snapshot taken before crash. It first reverts back the operations done from the point of snapshot to that of crash and then it redoes them



## <u>Serializability</u>

#### Terminology

**Schedule:** A schedule is a time ordered sequence of the actions taken by one or more transactions.

**Serial schedule**: A schedule that considers all the actions of a transaction T1, followed by all the actions of another transaction T2 and so on.

**Serializable schedule**: A schedule whose effect on the state of the Database is the same as the effect of **some** serial schedule.

Note that all serial schedules are serializable but not all serializable schedules are serial.

Consider two transactions

T1	T2
op11	op21
op12	op22
op13	op23
•	
•	

The above two transactions can be scheduled in a no of ways like

- i) T1(op11.....op1n) followed by T2(op21.....opn1)
- ii) T2(op11.....op1n) followed by T1(op21.....opn1)
- iii) op11, op21, op12, op22...... (interleaved scheduling)

However different order of operations may result into different states of the database system.

Interference between transactions can be avoided if the transactions are executed according to schedule enforcing concurrency control techniques.

**Conflict equivalence:** Two schedules are said to be conflict-equivalent if all conflicting instructions in them follow the same order.

**Conflict serializability:** A schedule is conflict serializable if it is conflict equivalent to a serial schedule.

Note : A conflict schedule is conflict serializable if two phase locking is followed.