Distributed Database Unit-2
for final year students of CSE and IT

Developed and Presented By:

Dileep Kumar Yadav

Assistant Professor

CSE Dept.

V.B.S P.U,Jaunpur

Mb. No.8726943272

Email-dileep1482@gmail.com

Lock

- Lock is a variable associated with data item that describes the status of the item with respect to possible operation is accessing a data item, no other transaction can modify that data item.
- All lock requests are made to the concurrency control manager, transactions proceed only once the lock request is granted.

Lock Based Protocols

- It manage the order between the conflicting pairs among transactions at the time of execution. There are various modes in which a data item may be locked..
- Binary lock
- Shared lock
- Exclusive lock

Cont…

- **Binary Lock :**A binary lock on a data item can either locked or unlocked states.
- **Shared lock(s):** A shared lock is also called 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.

Cont…

- **Exclusive lock(x):**A exclusive a data item can be read as well as write.
- This is exclusive and can not be held concurrently on the same data item.
- For example:

T1	T2
Lock –s(Q)	Lock-s(Q)
Lock-x(Q)	

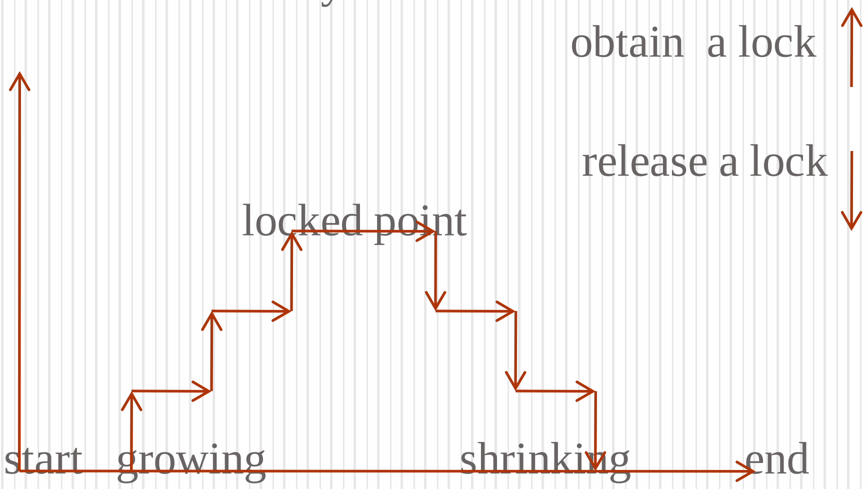
Cont…

	shared	exclusive
Shared	**T**	**F**
Exclusive	F	F

Compatibility table

Two Phase Locking Protocol

- Two phase locking protocol ensure serializability.
- It also insure conflict serializability.

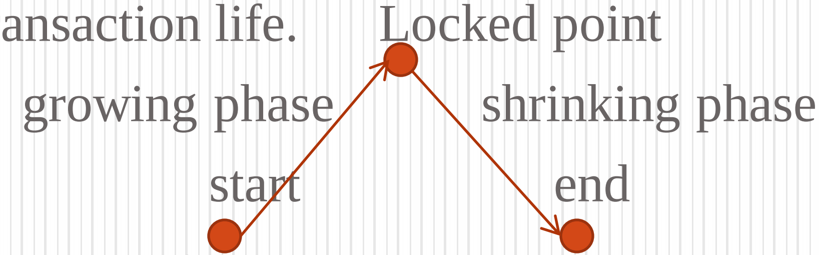


Cont…

- This protocol requires that each transaction in a schedule will be two phase i.e. growing phase and shrinking phase.
- In growing phase transaction can only obtain locks but can not release any lock.
- In shrinking phase transaction can only release lock but can not obtain any lock.
- Transaction can perform read/write operation both in growing and shrinking phase.

Cont…

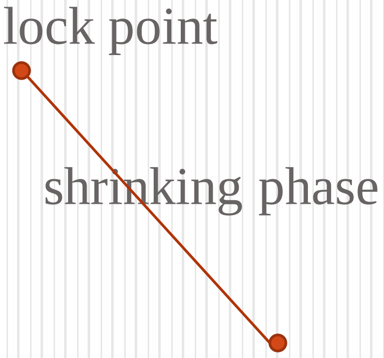
- In database and transaction processing two phase locking protocol is a concurrency control method that guarantees serializability.
- The protocol utilizes locks applied by a transaction to data which may block other transaction from accessing the same data during the transaction life.



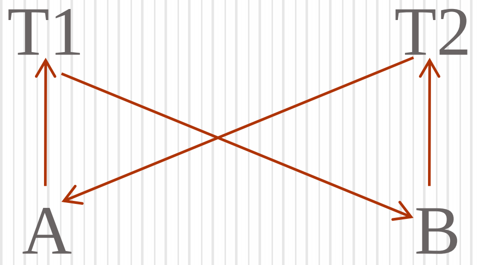
Strict Two Phase Locking

- Strict two phase locking waits until the whole transaction to commit and then it releases all the locks at a time.

T1	T2
Lock s(A)	
Read(A) gp	
	Lock x(B)
	Read(B) gp
	Write (B)
Lock s(B)	
	Lock x(A)
Further not	growing and go to dead lock



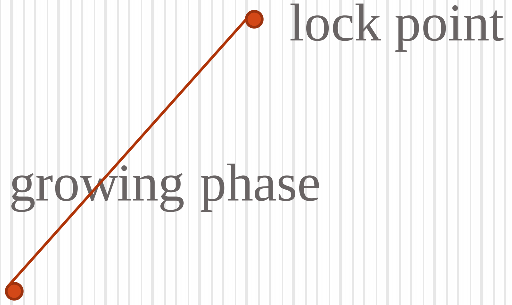
Dead Lock



dead lock condition

Regorous two phase locking

- It holds all locks until the transaction has already successfully committed or aborted.



Cont…

- It is an important over 2pl protocol whose we try to ensure recoverability and cascadlessness.
-

T1	T2
Lock x(A)	
R(A)	
W(A)	DR
Unlock (A)	
	Lock s(A)
	R(A)

T1-unlock(A)

T2-lock(A)

- If transaction commit successful then its already go to unlock condition

Time Stamp Based Protocol

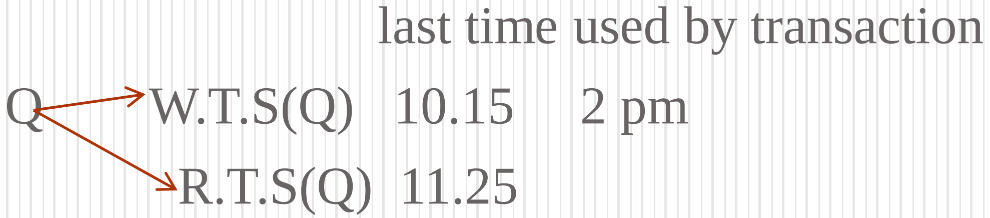
- The basic idea of time stamp is to decode the order between the transaction before that enters into the system so that in case of conflict during execution we can resolve the conflict using order.
- The reason we call time stamp not stamp because for stamping we use of system clock(as it will always be unique can never repeat itself).
- 1 T1 2 T2 3 T3

  10.15AM 11.25 AM

Time Stamp with Transaction

- With each transaction Ti we associated a timestamp by Ts(Ti), it is the value of the system clock when a transaction enters into the system so if a new transaction Tj enters after Ti..
- Then Ts(Ti)<Ts(Tj) always unique will remain fixed through the execution also determine serializability order if Ts(Ti)<Ts(Tj).
- Then system ensure that in the resultant conflict serial schedule Ti will execute 1st before Tj.

Time Stamp with Data Item



last time used by transaction

Q → W.T.S(Q) 10.15 2 pm

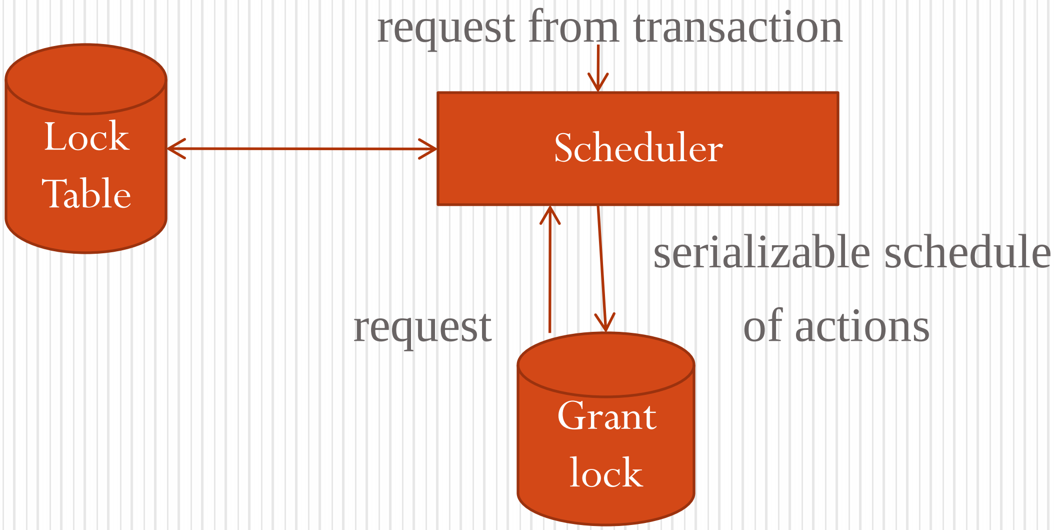
Q → R.T.S(Q) 11.25

- **W time stamp(Q):** It is the largest timestamp of any transaction that executed write(Q) successfully.
- **R time stamp(Q):** It is the largest time stamp of any transaction that executed read(Q) successfully.

Enforcing Serializability by Locks

- Locks
- Two phase locking
- Locking scheduler

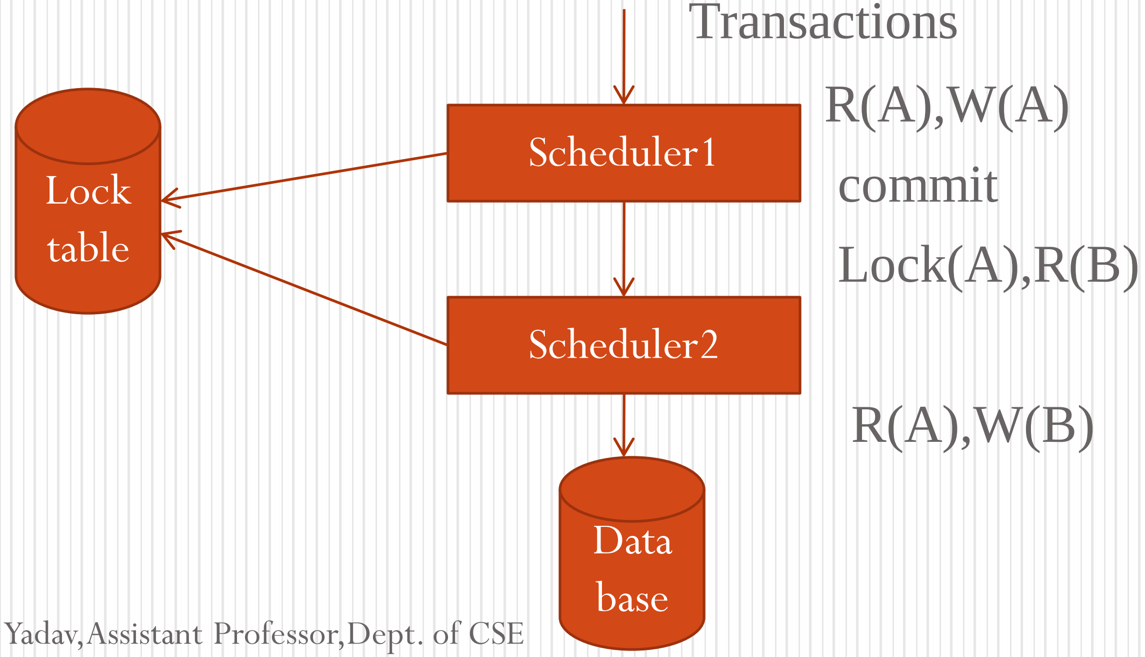
Locking Scheduler



Cont…

- Lock table stores the information about current locks on the elements.
- Grant lock request only if it is in a legal schedule.
- No two transaction can acquire the lock on same element without the prior one releasing it.

Architecture of Locking Scheduler



Transactions

Scheduler1

R(A),W(A)

commit

Lock(A),R(B)

Lock table

Scheduler2

R(A),W(B)

Data base

Cont…

- Transactions can not request for lock it is the scheduler that take care of it.
- Scheduler inserts the lock and release the lock when it is notified by the transaction manager.
- Scheduler take cares of the actions which are requested by the transactions.
- Scheduler1 handles all the request generated by the transaction and insert lock actions in front of all database access operations.

Cont…

- The database actions are transmitted to part II.
- The scheduler2 is responsible for performing the database access actions passed by scheduler1.
- When a transaction commits or aborts scheduler1 is notified by the transaction manager and release all locks held by T.
- If any transaction is waiting for any of the locks, scheduler1 notifies scheduler2.

Cont…

- When scheduler2 is notified that a lock on same database element x is available it determines the next transaction or transactions that can now be given a lock on x.
- The lock table is a relation that associates database elements with locking information about that element.

Reference

- Database management system by Abraham Silberschatz and Henry F. Korth, Tata McGraw-hill publications.