- Deadlock recovery.

a. Most common method for recovery is roll back
b. Selection of victim transaction (starvation)

- Transaction having minimum cost will rollback.

Min cost → No of data item used and required

→ Time required for execution

No of transaction involed in rollback

- Deadlock Prevention

Two main approches for deadlock prevention.

a. Accuring all locks before execution.
b. Performing roll back instead of waiting for a lock.

Wait die   Wound wait

1. Wait die:

Condition: $Ts(ti) < Ts(tj)$

Ti is requesting for lock from tj. Allow to wait otherwise rollback.

Q. comparision between wait die and wound die.

2. Wound die:

$Ts(ti) > Ts(tj)$

$Ts(ti)$ will wait for roll back.

PAGE NO.
DATE:

Deadlock exist:

Eg(2)

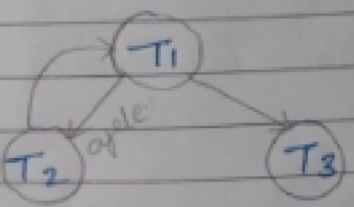
Time	Transaction	Operation.
t_1	T_1	Lock (A, x)
t_2	T_2	Lock (B, x)
t_3	T_3	Lock (A, S)
t_4	T_4	Lock (B, S)
t_5	T_5	Lock (B, S)
t_6	T_6	Lock (D, x)
t_7	T_7	Lock (D, S)
t_8	T_8	Lock (C, x)

in the graph then dead lock exist.

Time	Transaction	Operation.
t_1	T_1	Lock (A, x)
t_2	T_2	lock (B, S)
t_3	T_3	Lock (A, S)
t_4	T_4	Lock (C, x)
t_5	T_5	lock (D, x)
t_6	T_6	lock (D, S)
t_7	T_7	lock (C, S)

T_1	T_2	T_3
W(A)		
	R(B)	
		R(A)
W(I)		
	W(D)	
R(n)		
	R(c)	

Graph.



- Dead Lock:

Each transaction is waiting for another transaction to execute and none of the transaction can execute this situation is called as dead lock.

There are two methods deal with dead lock.

a. Dead Lock deduction and recovery.
b. Dead Lock prevention.

a. Dead Lock deduction and recovery.

It uses wait-for-graph technique

Basic steps:

i Keep information of currently allocated data items and also the outstanding request.
ii Find wheather dead lock exist or not
iii (Use graph)
Recover from dead lock if it exist.

Deadlock can be detected using directed group called as wait-for-graph

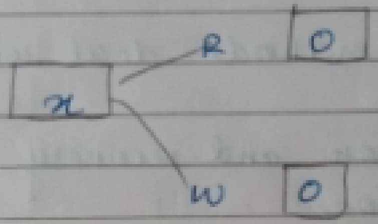
Q. Define wait-for-graph.

It is denoted as G(V,E), vertices cover all transaction. Edges is mapping between t(i) and t(j)

If there is edge from t(i) to t(j) it means t(i) is waiting for t(j). If there is cycle

Eg.

T_1	T_2
R(w)	
	w(x)
R(w)	



if (0>1)
read (x)
R (ts (x) = 1)

- Thomas write rule

If RTs(x) > Ts(T_1), if condition true abort and rollback.

If wTs(x) > Ts(T_1), If condition true ignore write and continue execution

HW 1] Difference between Time stamp ordinary protocol and lock based od ordinary protocol.

2. Explain optimistic conquerency control

It is denoted by Ts (ti) If there is two transaction ti and t_2 then Ts (ti) < Ts (t2) To implement this two its ts values are associated with each data item.

a· Read Time stamp
b· write Time stamp.

a· Write Time stamp – It denotes largest time stamp of any transaction that execute write successfully.

b. Read Time stamp - It denotes largest time stamp of any Transaction that execute read successfully.

whenever there is read or write operation completed successfully this time stamp are updated.

- Rules for reading and writing :

a· If a transaction issues read request for (x) is greater than time stamp (ti) it will roll back else read (x)
ie; (w Ts (x) > Ts (ti))

b· If a transaction issues write request for (x) i·e; (w Ts (x) > Ts (T1) or RTs (x) > Ts (T1)

PAGE NO.
DATE:

2. Strict 2PL :

In this method transaction can apply any lock on any data item but for exclusive lock once the transaction request for exclusive lock it will not release that lock ~~with~~ untill transaction comitt

- Advantage : No risk of dead lock.
- Disadvantage: There may be cascading roll back.

3. Rigorous 2PL :

In this method transaction will apply lock on different data item and will not release any of the lock until transaction committed.

- Advantage : This is best method, most of the database system uses this method
  No risk of dead lock and cascadding rollback.

5. The point in the schedule where transaction has obtained its final lock is called as lock point.

6. On the basis of lock point we can find order of serializibitity

Advantage:

It always ensure serializibility

Disadvantage:

- Cascading rollback may occur.
- It does not ensure freedom from deadlock.

- ~~Ensuring~~ Variations in 2PL.

Strict 2PL　　Rigorous 2PL　　Conservative 2PL.

1. Conservatine 2PL:

- In This method Transaction locks all the data items or variables before starting its execution
- Early prediction of how many data items are required
- This method is deadlock free
- Practical implemention of this method is very difficult

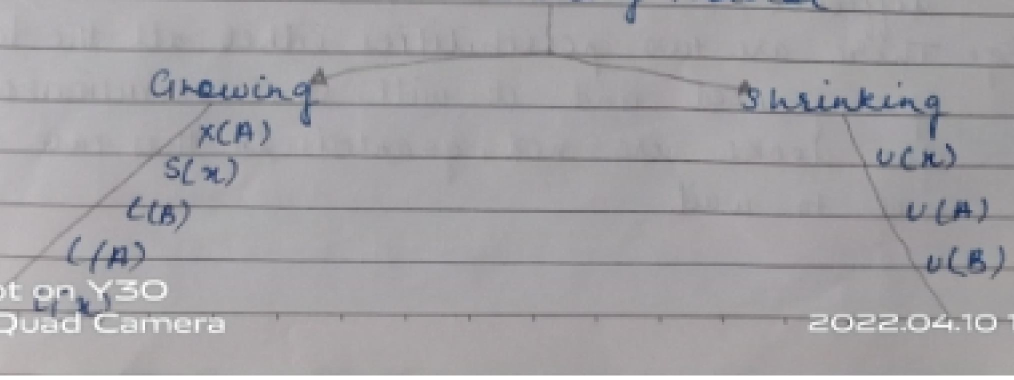
Eg. There are two possibilities either all the locks are granted and it will strat execution or the locks are not granted in this case it has to wait

2. Again there is a new transaction T_4 requesting for shared lock and it has been granted, T_2 is still waiting for T_4.
3. There is possibitily that sequence of transaction will request for shared lock and T_2 will never get exclusive lock.
4. The entire scenario where a particular transaction is selected as victim and never getting chance to execute is called as Starvation.

- Two Phase Locking Protocol (2PL)

1. Rules to be followed while requesting and ~~realising~~ releasing locks.
2. Two phases are : Growing phase; Shrinking Phase.
3. In Growing phase locks can be obtained but not released
4. In Shrinking phase locks can be released but no new lock can be aquired

Two Phase Locking Protocol.



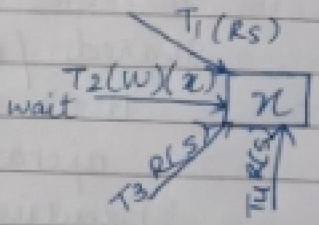
2. Compatibility condition:

If transaction T_1 is holding shared lock some other transaction T_j request for shared lock and it can be granted that we can say that transaction T_i is compatible with T_j

- If the requested lock can't be granted then the Transaction T_i and T_j are not compatible with each other.

$T_j \backslash T_1$	S	E
S	✓	×
E	×	×

3. Starvation: Waiting for getting lock.



1. Let Transaction T_1 started execution with shared lock on data x some other transaction T_2 request for execlusive locks on same data x. Then T_2 has to wait until T_1 release its lock. Now transaction T_3 request for shared lock on some data x and it has granted, still T_2 has to wait for T_3 to release its lock

- Ensuring serializibility by Locks:

Algorithms for concurrency control [Unit:3]

There are different algorithms

1. Lock based protocol: 15 mks.

a. Locking ensures serializibility or concurrency

b. In this process, we restrict the transaction to access the data or to lock access to the particular data.

c. Eg: let x is some data in database T_1 transaction wants to read the data and at the same time T_2 also wants to access the same data. So depending on what operation transaction T_2 wants to perform, locks are applied or restrictions will be made on T_2.

Locks.

Binary
[locked
unlocked

Shared/Exculsive
It is used when operation is read.
Exculsive is used when operation is write.