NO. Date: / / ①

Transaction processing.

- Transaction is a sequence / series of read and writes

Properties of transaction

① Atomicity
② Consistency
③ Isolation
④ Durability.

} ACID short term.

Atomicity - While transaction happening succesfully it will update db. if transaction fail not going to update db.

Consistency - After transaction totall values should be updated. without any lose. (data integrity) Start with correct data end with correct data.

Isolation - Executing several transactions at a same time without errors by Schedular. Concurrency Control Sub system.

one after otte Parrerel - Serial Schedular.
Parrerel - Parrerel Schedular

- executed parrerally by indipendent.

Durability - Db should be survive after the crash. log e maintain recovery manager responsible for ensuring durability of a db.

NO. Date.

Db architecture.

Transaction manager. Cordinate transaction.
Schedular - Schedul transaction.
lock manager - Works with Schedular.
recovery manager. take help of in failiers
Buffer manager - to transfer data from disk storage and to main memory.

Transactions & Shedules.

read → $R_T(O)$ - modification to acc
Write → $W_T(O)$ - read the data
↓
Commit → $Commit_T$ or C_{T_i}
Abort → $Abort_T$

Interleade - do little of t_1 and do t_2 and go to t_1 complete and t_2.

Scheduling transaction.

- give priority to shorter transaction

3 main typesof Schedles.
(61) Serial schedule.
(62) Equivalent Schedule
(63) Serializable Schedule.

Serial - transaction happen one after others. t_1 to t_2 after completion of t_1. not parrerol.

Equivalent - if the net effect Schedular 1 and Sechelar 2 are same there are equivalent.

Serializable - different transaction Concurrently at a Same time. pick one Schedule and Compare with other Schedule. if it matches to eithe one or more it is serializability.

Anomalies with Interleaved Execution

(01) Reading Uncommited data
(02) Unrepeatable reads
(03) Overwriting Uncommited data

(01) reading Uncommited data.
(WR conflicts - dirty reads)

- T_2 reads the data that uncommited by t_1. before commiting value going to read by the anothe transaction.

(02) Unrepeatable reads.
(RW conflicts)

- 2 reads of a Some value gives you different values.

(03) Overwriting Uncommited data
(WW conflicts)

- overwriting, uncommited data.

NO.

Lock based Concurrency Control.

(i) Strict Two-phase locking (Strict 2PL)
- gradually have the lock and gradually release lock.

2 type of lock.
(a) S Lock - Shared lock.
(b) X Lock - Exclusive lock.

- Shared Lock - it prevent transaction editing details only available to read, only by commit.

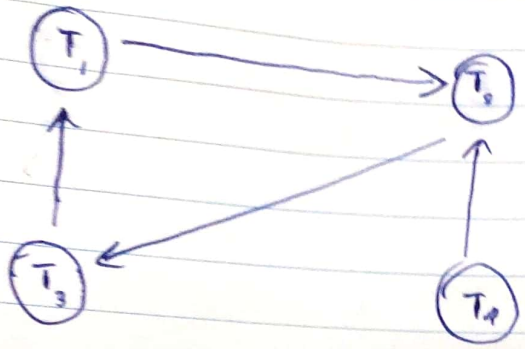
- exclusive lock - if t_1 has X lock it will not accesible by other transactions. t_2 has to wait until t_1 finish, after the commit only.

Read locks

T_1	T_2	T_3	T_4
S(A)			
R(A)			
	X(B)		
	W(B)		
S(B)			
R(B)		S(C)	
		R(C)	
	X(C)		
	R(C)		
			X(B)
		Ex(A)	

Date: / /

Wait - for graph cycle



→ its a deadlock situation

Aborting transaction

- all the details should be removed

2 type:

(01) Cascading Aborts

(02) Unrecoverable Aborts

(01) Cascading abort.

- aborting of t_1 will effect t_2 process need to abort t_2 too.

(02) Unrecoverable schedule.

- before t_1 abort t_2 has done and Commited the transaction.