Date: __/__/____

MON TUE WED THR FRI SAT SUN

ASSIGNMENT-4

[A]

1. Consistency
2. Only S_2 is conflict serializable
3. A schedule that is not conflict serializable
4. II only [Time-Stamp ordering]
5. $T_1 \rightarrow\rightarrow T_3 \rightarrow\rightarrow T_2$
6. D. $r_2(x) ; W_2(x) ; r_3(x) ; r_1(x) ; W_1(x)$
7. S is both conflict serailizable and recoverable.
8. We ~~need to redo~~ must undo log ~~rec~~ record 6 to set B to 10000 and then redo log records 2 and 3.
9. Schedule S is non-recoverable and cannot ensure transaction atomicity.
10. Guarantee serializability and deadlock-freedom.

[B]

1. What do you mean by dirty read problem? When a transaction is allowed to read a row that has been modified by another transaction that is not been committed yet that time Dirty Reads occurred. It is mainly occurred because of multiple transactions at a time which is not committed.

Date: __/__/____
MON TUE WED THR FRI SAT SUN

2. What is the necessary condition of deadlock?
A deadlock will only occur if the four Coffman conditions hold true. These conditions are:
Mutual Exclusion
Hold and Wait
No-preemption
Circular Wait

3. What is meant by locks in concurrency control?
Locks are an integral part to maintain concurrency control in DBMS. A transaction in any system implementing lock based concurrency control cannot read or write a statement until it has obtained the required locks.

4. What is 2PL?
A transaction is said to follow the Two-Phase locking protocol if Locking and Unlocking can be done in two phases.
1. Growing Phase:
New locks on data items may be acquired but none can be released.

2. Shrinking Phase:
Existing locks may be released but no new locks can be acquired.

Date: __/__/____

MON TUE WED THR FRI SAT SUN

[c]

1. Explain Thomas write rule in detail.

To Thomas Write Rule provides the guarantee of serializability order for the protocol. It improves the Basic Timestamp Ordering Algorithm.

The basic Thomas Write rules are as follows

- If TS(T) < R_TS(x) then transaction T is aborted and rolled back, and operation is rejected.

- If TS(T) < W_TS(x) then don't execute the W_item(x) operation of the transaction and continue processing.

- If neither condition 1 nor condition 2 occure, then allowed to execute the WRITE operation by transaction Ti and set W_TS(x) to TS(T).

2. Explain ACID properties with example.

1. Atomicity:

By this, we mean that either the entire transaction takes place at once or doesn't happen at all. There is no midway i.e. transactions do not occur partially. Each transaction is considered as one unit and either runs to completion or is not executed at all. It involves the following two operations.

Date: __/__/____
MON TUE WED THR FRI SAT SUN

Commit
~~Abort~~ : If a transaction commite, changes made are visible.
Abort : If a transaction aborts, changes made to the databases are not visible.

Atomicity is also known as the "All or nothing rule".

2. Consistency :
This means that integrity constraints must be maintained so that the database is consistent before and after the transaction. It refers to the correctness of a database.

Ex: The total amount before and after the transaction must be maintained.

Total before T occurs = 500 + 200 = 700
Total after T occure = 400 + 300 = 700
Therefore, the database is consistent.
Inconsistency occurs in case T_1 completes but T_2 fails. As a result, T is incomplete.

3. Isolation :
This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of the database state. Transactions occur independently without interference. Changes occurring in a

Date: __/__/____
MON TUE WED THR FRI SAT SUN

particulas transaction will not be visible to any other transaction until that particulas change in that transaction is written to memory or has been committed. This property ensures that the execution of transactions concurrently will result in a state that is equivalent to a state achieved these were executed serially in some order.

4. Durability:
This property ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if a system failure occurs. These updates now become permanent and are stored in non-volatile memory. The effects of the transaction, thus, are never lost.

D.

1. Explain recoverability in detail.
Sometimes a transaction may not execute completely due to a software issue, system crash or hardware failure. In that case, the failed transaction has to roll back. But some other transaction has to be rollback. But some other

Date: __/__/____
MON TUE WED THR FRI SAT SUN

transaction may also have used value produced by the failed transaction so we also have to rollback those transaction.

T_1	T_2
Read (A) ;	
A = A - 500	
write (A) ;	
	Read (A)
	A = A + 1000 ;
	write (A) ;
	commit ;
Failure point.	
commit ;	

In the above table, T_1 fails after T_2 is committed which reads the value written by T_1. T_2 can't be rollback and hence this type of schedule is called Irrecoverable schedule.

T_1	T_2
Read (A) ;	
A = A - 500 ;	
write (A) ;	
	Read (A) ;
	A = A + 1000
	write (A) ;
failure Port	
commit	
	commit ;

Date: __/__/____

MON TUE WED THR FRI SAT SUN

In the above table T_2 reads the value written by T_1 and when T_1 fails T_2 can be rolled back. Hence this type of schedule is called recoverable schedule with Cascade rollback.

2. Differentiate between conflict serializable and view serializable schedule with example.

Conflict Serializability	View Serializability
1. Two schedules are said to be conflict equivalent if all the conflicting operations in both the schedule get executed in the same order. If a schedule is a conflict equivalent to its serial schedule then it is called Conflict Serializable Schedule.	1. Two schedules are said to be view equivalent if the order of initial read, final write and update operation is the same in both the schedules. If a schedule is view equivalent to its serial schedule then it is called View Serializable Schedule.
2. If a schedule is view serializable then it may or may not be conflict serializable.	2. If a schedule is conflict serializable then it is also view serializable schedule.

Date: __/__/____
MON TUE WED THR FRI SAT SUN

3. Conflict equivalence can be easily achieved by reordering the operations of two transactions therefore, Conflict Serializability is easy to achieve.	3. View equivalence is rather difficult to achieve as both transactions should perform similar actions in a similar manner. Thus View Serializability is difficult to achieve.
4. For a transaction T_i writing a value A that no one else reads but later some other transactions say T_2 write its own value of A, W(A) cannot be placed under positions where it is never read.	4. If a transaction T_i writes a value A that no other transaction reads (because later some other transactions say T_2 writes its own value of A) W(A) can be placed in positions of the schedul where it is never read

Risky