Lecture 6: Transactions, Concurrency & Recovery

1. Consider the following transaction **T** consisting of **T1** and **T2** : Transfer of $100 from account **X** to account **Y** .

Before: X : 500	Y : 200
Transaction T	
T1	T2
Read (X) X: = X - 100 Write (X)	Read (Y) Y: = Y + 100 Write (Y)
After: X : 400	Y : 300

2. Let's consider two transactions:Consider two transactions T and T".
   X = 500, Y = 500

X = 500 Rs	Y = 500 Rs
T	T''
Read (X) X :=X*100 Write (X) Read (Y) Y:= Y - 50 Write (Y)	Read (X) Read (Y) Z:= X + Y Write (Z)

3. **Temporary Update Problem:**
   Temporary update or dirty read problem occurs when one transaction updates an item and fails. But the updated item is used by another transaction before the item is changed or reverted back to its last value.

Example:

T1	T2
read_item(X) X = X - N write_item(X) read_item(Y)	 read_item(X) X = X + M write_item(X)

4. **Incorrect Summary Problem:**

Consider a situation, where one transaction is applying the aggregate function on some records while another transaction is updating these records. The aggregate function may calculate some values before the values have been updated and others after they are updated.

Example:

T1	T2
   read_item(X) X = X - N write_item(X) read_item(Y) Y = Y + N write_item(Y)	sum = 0 read_item(A) sum = sum + A read_item(X) sum = sum + X read_item(Y) sum = sum + Y

5. **Lost Update Problem:**

In the lost update problem, an update done to a data item by a transaction is lost as it is overwritten by the update done by another transaction.

Example:

T1	T2
read_item(X) X = X + N	X = X + 10 write_item(X)

6. **Unrepeatable Read Problem:**

The unrepeatable problem occurs when two or more read operations of the same transaction read different values of the same variable.

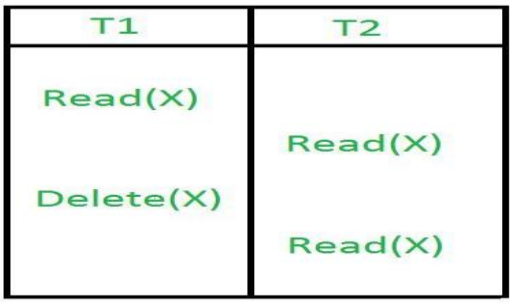
Example:

T1	T2
Read(X)	
	Read(X)
Write(X)	
	Read(X)

7. **Phantom Read Problem:**

The phantom read problem occurs when a transaction reads a variable once but when it tries to read that same variable again, an error occurs saying that the variable does not exist.

Example:



8. Consider a database with a single data item X = 10.
   Transactions:
   T1: Wants to read and update X.
   T2: Wants to read X.

9. Let's see a transaction implementing 2-PL.

	T1	**T2**
1	lock-S(A)	
2		lock-S(A)
3	lock-X(B)	
4	.....	.....
5	Unlock(A)	
6		Lock-X(C)
7	Unlock(B)	
8		Unlock(A)
9		Unlock(C)
10	.....	.....

10. This is a basic outline of a transaction that demonstrates how locking and unlocking work in the Two-Phase Locking Protocol (2PL).

Suppose there is two schedules one is non-serial and another one is serial schedule.

Schedule 1		Schedule 2	
T1	T2	T1	T2
r1(A)		r1(A)	
A=A+10		A=A+10	
w1(A)		w1(A)	
r1(B)			r2(A)
B=B*10			A=A+10
w1(B)			w2(a)
	r2(A)	r1(B)	
	A=A+10	B=B*10	
	w2(A)	w2(B)	
	r2(B)		r2(B)
	B=B*10		B=B*10
	w2(B)		w2(B)

11. Consider the following schedule:

S:R1(A), W2(A), Commit2, W1(A), W3(A), Commit3, Commit1

Which of the following is true? (A) The schedule is view serializable schedule and strict recoverable schedule (B) The schedule is non-serializable schedule and strict recoverable schedule (C) The schedule is non-serializable schedule and is not strict recoverable schedule. (D) The Schedule is serializable schedule and is not strict recoverable schedule.

12. **Example of View Serializable Schedule**

Step	Transaction	Operation	Explanation
1	T1	W(A)	T1 writes to A.
2	T2	R(A)	T2 reads the value of A written by T1.
3	T2	W(B)	T2 writes to B.
4	T1	R(B)	T1 reads the value of B written by T2.