BAHRIA UNIVERSITY
(Karachi Campus)

Department of Software Engineering

ASSIGNMENT#03 – Spring 2023

COMPLEX ENGINEERING PROBLEM

Based on CLO-3

COURSE TITLE:	**Database Management System**	COURSE CODE:	**CSC-220**
Class:	**BSE – 4A & B**	Shift:	**Morning**
Course Instructor:	**ENGR. LARAIB SIDDIQUI**	Max. Marks:	**06 Marks**
Assignment Date :	6th **JUNE 2023**	Due Date :	13th **JUNE 2023**

This assignment is based on the following CEP attributes:

- Cannot be resolved without in-depth engineering knowledge.
- Have no obvious solution and require abstract thinking and originality in analysis to formulate suitable models.

Question

Consider three transactions given below to create one serial schedule and two concurrent schedules with different time slices. Also, assume different orders for the transaction's timestamps e.g. the order $T_1 < T_2 < T_3 < T_4$ (T_1 is the oldest). Then check for view serializability and conflict serializability.

- T_1 : I_1 R(Y) I_2 R(Z) I_3 R(B) I_4 B:=B+(Y*.1)+Z*.2 I_5 Y:=Y-Y*.1 I_6 Z:=Z-Z*.2 I_7 W(B) I_8 W(Y) I_9 W(Z)
- T_2 : I_1 R(D) I_2 R(Y) I_3 D:=D-50 I_4 W(D) I_5 Y:=Y+50 I_6 W(Y)
- T_3 : I_1R(Z) I_2 R(D) I_3 Z:=Z-5 I_4 W(Z) I_5 D:=D+5 I_6 W(D)

a) Apply shared lock (L_s), exclusive lock (L_x) and unlock (U_s or U_x) on any one of the schedules produced in part a) for T_1, T_2 and T_3 on database items B, D, Y and Z. Then simulate how these transactions are executed under the wait-die and wound-wait prevention strategy. For the simulation, assume that these transactions are executed in a round-robin fashion.

When it is a transaction's turn, it executes its next lock or unlock step if it can, and otherwise dies or waits or wounds the holder of the lock, as appropriate. When a waiting transaction is restarted (due to the action of some other transaction) it becomes eligible to run at its very next turn.

Make a table to show the sequence of steps that these transactions make under each policy. Use $L_x(Y)$ to denote exclusive locking an object Y, $U_x(Y)$ to denote unlocking Y, similarly $L_s(Y)$ and $U_s(Y)$ for locking and unlocking shared lock on Y, “Die” to denote the transaction dying, “Wound” being wounded, and “Wait” to denote the transaction waiting. If transaction A is wounded by some other transaction B, transaction A’s next action is ’Die’. In the grid, the cell for [i, j] (row i column Tj) represents the i th action of Tj . We have filled out a few cells to illustrate how you should fill out the rest of the table if cycle is for two instructions. Simulate actions until all three transactions complete. If a transaction is already complete, its turn is skipped.

T_1	T_2	T_3	T_4
Ls(Y)			
R(Y)			
	$L_s(D)$		
	R(D)		
		$L_x(Z)$	
		W(Z)	

Execute for both schedules and apply both wait-die and wound-wait strategy. Also explain which is better and why.