

### Part 3

A)

T1	T2	T3
L <sub>1</sub> (A)		
R <sub>1</sub> (A)		
W <sub>1</sub> (A)		
L <sub>1</sub> (B)		
U <sub>1</sub> (A)		
R <sub>1</sub> (B)		
W <sub>1</sub> (B)		
U <sub>1</sub> (B)		
		L <sub>3</sub> (A)
		R <sub>3</sub> (A)
		W <sub>3</sub> (A)
		L <sub>3</sub> (B)
		U <sub>3</sub> (A)
		R <sub>3</sub> (B)
		W <sub>3</sub> (B)
		U <sub>3</sub> (B)
	L <sub>2</sub> (A)	
	R <sub>2</sub> (A)	
	L <sub>2</sub> (B)	
	R <sub>2</sub> (B)	
	U <sub>2</sub> (A)	
	U <sub>2</sub> (B)	

B)

2PL ensures conflict-serializability by holding all the locks until the transaction ends, preventing inconsistencies. However, 2PL does not prevent cascading aborts, where the failure of a transaction could lead to the failure of several dependent transactions.

Strict 2PL is an enhancement over 2PL that requires transactions to hold all exclusive locks until after they commit. This ensures recoverability and cascadeless transactions, as other transactions can only access the data after it has been fully committed. This means that even if a transaction fails and aborts, it will not affect transactions that have already committed.