

Student ID: 33114064
Student Name: Harshath Muruganantham
Unit Code: FIT3171
Applied Class No: 1

Comments for your marker:

(a)

TIME	TRANS	ACTION	A	B	C	D
0	T1	UPDATE A	X(T1)			
1	T1	UPDATE B		X(T1)		
2	T2	READ C			S(T2)	
3	T2	READ D				S(T2)
4	T3	UPDATE A	T3 wait T1			
5	T2	UPDATE C			X(T2)	
6	T1	ROLLBACK	X(T3)			
7	T3	UPDATE C			T3 wait T2	
8	T2	UPDATE B		X(T2)		
9	T2	UPDATE A	T2 wait T3			

- Does a deadlock exist in this transaction sequence? Yes
- Explain why you came to this conclusion.

The two highlighted transactions in the table above show where the deadlock exists. At Time 7, T3 tries to 'Update' C, which requires T3 to put an exclusive lock on C. However, T2 already has an exclusive lock on C from time 5. So here, T3 will have to wait for T2 to relinquish its shared lock. Now taking a look at time 9, T2 tries to 'Update' A, which requires for T2 to have an exclusive lock on A. However, T3 currently has an exclusive lock on A, which was initiated at time 6 when T1 had relinquished its exclusive lock on A to T3 after a rollback statement. So, T2 will have to wait for T3 from time 9 in order to update A.

Here, T3 is waiting for T2 (from time 7) and at the same time T2 is waiting for T3 (at time 9). Neither action can proceed without the other either committing or aborting, causing a stalemate, and thereby creating a deadlock between T2 and T3.

(b)

TRL ID	TRX NUM	PREV PTR	NEXT PTR	OPERATION	TABLE	ROW ID	ATTRIBUTE	BEFORE VALUE	AFTER VALUE
101	601	Null	108	START	****Start Transaction				
108	601	101	113	UPDATE	PRODUCT	ABC	PROD_QOH	1205	1206
113	601	108	115	UPDATE	PART	A	PART_QOH	567	566
115	601	113	120	UPDATE	PART	B	PART_QOH	98	97
120	601	115	122	UPDATE	PART	C	PART_QOH	549	548
122	601	120	Null	COMMIT	****End of Transaction				