TUTORIAL-II
DBMS

Exercise 1: Give an example of a transaction schedule that is conflict-serializable, but not possible under 2PL.

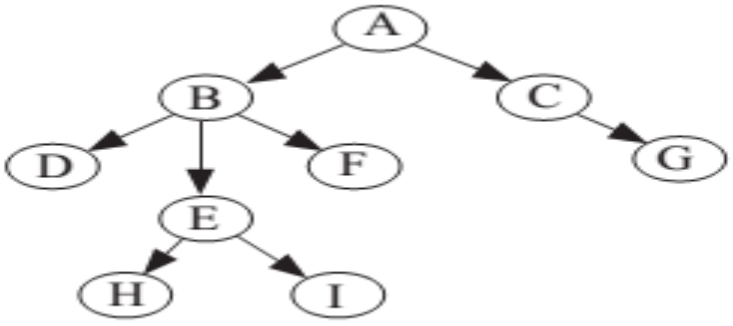
Exercise 2: The lost update anomaly is said to occur if a transaction T_j reads a data item, then another transaction T_k writes the data item (possible based on previous read), after which T_j writes the data item. The update performed by T_k has been lost, since the update done by T_j ignored the value written by T_k.
2(a): Give an example of schedule showing the lost update anomaly.
2(b) Give an example schedule to show that the lost update anomaly is possible with the read committed isolation level.
2(c) Explain why the lost update anomaly is not possible with the repeatable read isolation level.

Exercise 3: Consider the following two transactions:
T1: read(A);
read(B);
If A = 0, then B = B+1;
Write(B);
T2: read(B);
read(A);
If B = 0, then A = A+1;
Write(A);
Add lock and unlock instructions to transactions T1 and T2, so that they observe the two-phase locking protocol. Can execution of these transactions result in a deadlock?

Exercise 4: Consider a database organized in the form of a rooted tree. Suppose that we insert a dummy vertex between each pair of vertices. Show that, if we follow the tree protocol on the new tree, we get better concurrency than if we follow the tree protocol on the original tree.

Exercise 5: Show by example that there are schedules possible under the tree protocol that are not possible under the two-phase locking protocol, and vice versa.

Exercise 6 Consider the following tree and locking sequences:



(1) Lock-X(A), Lock-X(B), Lock-X(D), unlock(D), Lock-X(F), unlock(F), unlock(B), unlock(A)
(2) Lock −X(A), Lock-X(E), Lock-X(H), unlock(H), unlock(E), unlock(A)
(3) Lock −X(B), Lock-X(F) ,Lock-X(E) ,unlock(F) ,unlock(E) ,unlock (B)

Which of the above is (are) the valid sequence(s) for tree based protocol?

Exercise 7: Give a schedule which is conflict serializable but not dead lock free?