CHAPTER 15
DISCUSSIONS

Discussion 15-1

Execute the two transactions by alternating the instructions observing the *two phase locking protocol* (*2PL*). Whenever there is a *deadlock*, abort the transaction of the last lock request and restart it. Start with T_1.

T_1:	T_2:
read(X)	read(X)
write(X)	read(Y)
read(Y)	write(X)
write(Y)	write(Y)

Discussion 15-2

Explain and compare *strict* 2PL and *rigorous* 2PL.

Discussion 15-3

Both *strict* 2PL and *rigorous* 2PL ensure conflict serializability and cascadelessness (thus, recoverability) while both are not free from deadlocks.

Since rigorous 2PL holds both shared locks and exclusive locks until the transaction commits, it allows less concurrency than strict 2PL.

Then why do DBMSs implement rigorous 2PL at all?

Discussion 15-4

The *lock manager* is a module within a DBMS that handles lock requests. How would you implement a lock manager?

1) data structure

2) APIs

Lock Table