Assignment 5

Reading – Read Sections 6.3, 6.4, and 6.10 (up to the second paragraph after Fig. 6.20, beginning “so far”) of the revised Chapter 6 of the textbook that was handed out in class.

Problem 1

Suppose a transaction sets an intention-write lock on a file and later sets a write lock on a record of the file. Is it safe for the transaction to release the intention-write lock before it commits? Why?

Problem 2

Consider a database system that uses two-phase locking where locks are held until after a transaction commits. In the following scenarios, throughput (i.e. transactions per second) is the same before and after the change.

a. The code of the transactions running on a particular system is changed, but the number of locks (all of which are write locks) required by a transaction is unaffected. The change results in an increase in response time. Give three possible reasons.

b. A system is running a mixture of queries and updates. (Queries only set read locks, whereas updates set write locks.) Whenever the proportion of queries increases, overall response time becomes worse. Give three possible reasons.

c. A certain portion of a database is identified as a high contention area, so the granularity of locks for this portion was refined. However, response time becomes worse. Give two possible reasons.

Problem 3

Consider a data manager that uses two-phase locking. Suppose all transactions are single-threaded sequential programs, so no transaction can have more than one outstanding read or write request that is blocked. Could a transaction be involved in more than one deadlock? Explain your answer.