Homework 7 for CSC 742: Database Management Systems

Collaborative Work

You may form teams of 1 - 4 members (of students in this class) to cooperate on this problem set. After discussing the problem, please write up your answers individually. Indicate the names of the other members in your team, if any.

Question 1 (50 points)

Problems 21.28 – 21.37.

Independent Work

You must solve this problem set individually without any assistance from anyone.

Question 2 (30 points)

Consider the following sequences of actions, listed in the order they are submitted to the DBMS:

- **Sequence S1:** T1:R(X), T2:W(X), T2:W(Y), T3:W(Y), T1:W(Y), T1:Commit, T2:Commit, T3:Commit
- **Sequence S2**: T1:R(X), T2:W(Y), T2:W(X), T3:W(Y), T1:W(Y), T1:Commit, T2:Commit, T3:Commit

For each sequence and for each of the following concurrency control mechanisms, describe how the concurrency control mechanism handles the sequence.

Assume that the timestamp of transaction T*i* is *i*. For lock-based concurrency control mechanisms, add lock and unlock requests to the above sequence of actions as per the locking protocol. The DBMS processes actions in the order shown. If a transaction is blocked, assume that all of its actions are queued until it is resumed; the DBMS continues with the next action (according to the listed sequence) of an unblocked transaction.

1. Strict 2PL with timestamps used for deadlock prevention.
2. Strict 2PL with deadlock detection. (Show the waits-for graph if a deadlock cycle develops.)
3. Conservative (and strict, i.e., with locks held until end-of-transaction) 2PL.
4. Optimistic concurrency control.
5. Timestamp concurrency control with buffering of reads and writes (to ensure recoverability) and the Thomas Write Rule.
6. Multiversion concurrency control.

Question 3 (45 points)

Consider a database that is organized in terms of the following hierarchy of objects: The database itself is an object (*D*), and it contains two files (*F*1 and *F*2), each of which contains 1000 pages (*P*1…*P*1000 and *P*1001…*P*2000, respectively). Each page contains 100 records, and records are identified as *p : i*, where *p* is the page identifier and *i* is the slot of the record on that page.

Multiple-granularity locking is used, with *S*, *X*, *IS*, *IX* and *SIX* locks, and database-level, file-level, page-level and record-level locking. For each of the following operations, indicate the sequence of lock requests that must be generated by a transaction that wants to carry out (just) these operations:

1. Read record *P*1200 : 5.
2. Read records *P*1200 : 98 through *P*1250 : 2.
3. Read all (records on all) pages in file *F*1.
4. Read pages *P*500 through *P*520.
5. Read pages *P*10 through *P*980.
6. Read all pages in *F*1 and modify about 10 pages, which can be identified only after reading *F*1.
7. Delete record *P*1200 : 98. (This is a blind write.)
8. Delete the first record from each page. (Again, these are blind writes.)
9. Delete all records

Question 4 (10 points)

Problem 20.23 in Elsmasri & Navathi.

Question 5 (10 points)

Problem 20.25 in Elsmasri & Navathi

Question 6 (10 points)

Problem 20.26 in Elsmasri & Navathi