Data Management – exam of 05/07/2013

Problem 1 Consider the following schedule

$$S = r_1(A)\, r_1(B)\, w_2(A)\, w_3(B)\, w_1(B)\, w_3(A).$$

1. Tell whether S is accepted by the 2PL scheduler with exclusive and shared locks. If the answer is yes, then show the schedule obtained from S by adding suitable lock and unlock commands. If the answer is no, then explain the answer, and then tell whether there is a single action in the schedule whose removal brings the schedule in the class of 2PL with exclusive and shared locks.
2. Tell whether S is conflict-serializable. If the answer is yes, then show a serial schedule that is conflict-equivalent to S. If the answer is no, then explain the answer.

Problem 2 Answer the following three questions.

1. Describe in detail an algorithm that, given two schedules on the same set of transactions, checks whether the two schedules are view-equivalent.
2. Describe in detail an algorithm that, given a schedule, checks whether the schedule is view-serializable, by using the algorithm illustrated above for item 1.
3. Discuss in detail the computational complexity of the two algorithms illustrated for items 1 and 2 above.

Problem 3 Suppose the buffer of our DBMS contains 100 frames, where each frame at address i (where $1 \leq i \leq 100$) contains the page whose address is i, all frames from address 1 to address 50 have pin-count > 0, all frames from address 51 to 100 with an odd address have pin-count > 0, all frames from address 51 to 100 with an even address have pin-count=0 and have referenced=true, and, finally, the value of current is 5. Describe all the actions performed by the buffer manager when the page of address 70 is asked with a fix operation, and illustrate the configuration of the buffer at the end of the execution of such operation.

Problem 4 Consider the relations `Student(code,age,name)` (where `code` is the primary key), and `Enrolled(code,course)` (where again `code` is the primary key), that tells in which course the student is enrolled in. Consider the query that, given a specific student code, asks for the age and the name of the corresponding student, as well as the course in which (s)he is enrolled. Explain which is the method you would choose for storing the two relations with the goal of optimizing the performance of the system in executing the above mentioned query.

Problem 5 Consider the relation `Earthquake(code,location,date,mgnt)` with 400.000 tuples, in which every tuple stores code, location, date and magnitude of an earthquake. We know that each value of each attribute, as well as each page address in the system require 10 bytes. We also know that the size of every page in our DBMS is 500 bytes, and that we have a B^+-tree index on the attribute `date`; the index is unclustered and uses alternative 3. We assume that in the average there are 3 earthquakes every day. Consider the query asking for code, location and magnitude of all the earthquakes occurring in a specific week (given as interval of date values), and tell how many page accesses are needed in order to answer such query.