Data Management – exam of 13/06/2019

Problem 1

It is well-known that the schedulers based on locking can recognize deadlock situations when processing schedules. Provide the definition of deadlock, and prove or disprove the following statement: *If a schedule S is not conflict serializable, then the scheduler based on locking with both shared and exclusive locks recognizes a deadlock situation when processing S.*

Problem 2

Relations R and S are stored in heap files with 3.000 and 20.000 pages, respectively, and we have to compute the set union of R and S by using k frames available in the buffer, where $150 < k < 1.500$. If we measure efficiency in terms of number of page accesses, tell which of the following sentences is true, explaining the answer in detail.

2.1 For no value of k the block-nested loop algorithm is more efficient than the two-pass algorithm.

2.2 For no value of k the two-pass algorithm is more efficient than the block-nested loop algorithm.

2.3 There are values of k for which the block-nested loop algorithm is more efficient than the two-pass algorithm, and there are values of k for which the converse holds.

Problem 3

Let R and S be two relations having A as common attribute. The *semijoin* between R and S, denoted as $R \ltimes S$, is defined as follows $R \ltimes S = \{\ t \in R \mid t.A \in S[A]\ \}$. Suppose that we have to compute the semijoin between R and S.

3.1 Can we use the block-nested loop technique?

3.2 Can we adapt the simple sort-based join technique?

3.3 Can we adapt the sort-merge join technique?

In all the above three cases, if the answer is negative, then motivate the answer in detail, otherwise describe an appropriate algorithm for computing the semijoin between R and S using the corresponding technique. Also, among the three techniques, tell which is, in general, the best option in terms of efficiency, motivating the answer in detail.

Problem 4

Consider the following schedule

$$S = r_1(x)\, w_3(x)\, w_3(z)\, c_3\, w_2(x)\, w_2(y)\, c_2\, r_4(x)\, w_1(v)\, c_1 r_4(v)\, c_4.$$

4.1 Tell whether S is view-serializable or not, explaining the answer in detail.

4.2 Tell whether S is a 2PL schedule with both shared and exclusive locks or not, explaining the answer in detail.

4.3 Describe the behavior of the timestamp-based scheduler when processing the schedule S, assuming that, initially, for each element α of the database, we have rts(α)=wts(α)=wts-c(α)=0, and cb(α)=`true`, and assuming that the subscript of each action denotes the timestamp of the transaction executing such action.

4.4 Tell whether S is strict or not, and whether S is rigorous or not, explaining the answer in detail.

Problem 5

Assume that the relation Player(code,name,yearOfBirth,prizelevel) (where code is a key), has 325.000 tuples, each attribute and each pointer in the system occupies 100 Bytes, each page has space for 2.000 Bytes, the values for prizelevel are in the range [1..1000], equally distributed among the various tuples of the relation, and the most frequent query on Player asks for all players whose prizelevel falls into a given range.

5.1 Tell which method would you use to represent the relation in secondary storage.

5.2 Tell which logical plan and physical plan (assume that 40 free buffer frames are available) would you use for the following query Q:

```
select t.code, t.yearOfBirth, t.prizelevel
from (select code, yearOfBirth, prizelevel from Player order by yearOfBirth) as t
where t.prizelevel >= 925 and t.prizelevel < 950
```

5.3 Tell which is the cost of executing the physical query plan you have defined for item 5.2 in terms of the number of page accesses.