Data Management – exam of 17/04/2018

Problem 1 Consider the following schedule

$$S = r_1(A)\, r_1(C)\, w_2(A)\, r_2(B)\, r_1(B)\, r_4(B)\, w_3(A)\, w_3(B)\, w_5(D)\, w_4(D).$$

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.
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.
3. Tell whether S is strict, explaining the answer.

Problem 2 Consider the following schedule

$$S = r_2(A)\, w_2(D)\, w_3(C)\, r_1(B)\, w_2(B)\, r_1(C)\, w_3(A).$$

and illustrate in the most detailed way possible all the steps executed by the 2PL scheduler with exclusive and shared locks when it analyzes such schedules.

Problem 3 Describe in the most detailed way possible the parallel algorithm for computing the bag difference between two tables, and tell which is the cost of the algorithm, both in terms of number of page accesses, and in terms of elapsed time.

Problem 4 Consider the relations `R(A,B,C)` and `Q(D,E,F,G)`, where `D,E` form the key of the relation `Q`, and consider the join of the two relations on the condition

$$(\texttt{B} = \texttt{D} \text{ and } \texttt{C} = \texttt{E})$$

We know that `R` has 8000 pages, `Q` has 9000 pages, the buffer has 92 pages free, and there is a tree-based index on `Q` with search key `D,E` that support the equality search query with a cost of 3 page accesses. Answer the following questions:

1. Is the "block nested loop" algorithm applicable for computing the join? If yes, which is the cost of the join in this case? Motivate the answer.
2. Is the "two-pass sort-merge join" algorithm applicable for computing the join? If yes, which is the cost of the join in this case? Motivate the answer.
3. Is the "index nested loop" algorithm applicable for computing the join? If yes, which is the cost of the join in this case? Motivate the answer.

Problem 5 Consider the relation `Sailor(code,name,cityOfBirth)`, that stores the data about a set of sailors, and the relation `Sails(scode,boat)`, each of whose tuple $\langle s, b\rangle$ tells that the sailor whose code is s can sail boat b. We know that the number of sailors is 2.000.000, that each boat can be sailed by an average of 3 sailors, that each value of each attribute, as well as each page address in our DBMS require 10 bytes, and that the size of every page in our DBMS is 410 bytes. We also know that we have a hash-based index on the attribute `code` of the relation `Sailor`, and a B^+-tree unclustered index using alternative 3 on the attribute `boat` of the relation `Sails`. Consider the query

```
select name, cityOfBirth
from Sailor, Sails
where boat = b and scode = code
```

where `b` denotes a particular boat, and

5.1 Describe the logical query plan associated to the query code, and illustrate both the logical and the physical query plan you would select, motivating the choices.

5.2 Tell which is the cost (in terms of number of page accesses) of executing the query according to the selected physical query plan.