Data Management – exam of 11/06/2021

Problem 1
It is well-known that the scheduler called "Serialization Graph Tester" aims at ensuring conflict serializability, based on the precedence graph associated to the current schedule. In order to prevent the precedence graph to grow indefinitely, consider the following strategy: when a transaction t commits, remove from the precedence graph the node corresponding to the transaction t, as well as its ingoing and outgoing edges. Prove or disprove that such strategy is correct with respect to the goal of checking the current schedule for conflict serializability.

Problem 2
Consider the following schedule S (where we have relaxed the condition that no transaction contains more than one occurrence of the same action):

$$B(T_1)\ r_1(A)\ B(T_2)\ w_2(A)\ B(T_3)\ r_3(A)\ r_3(B)\ c_2\ r_1(A)\ r_1(B)\ B(T_4)\ r_4(A)\ c_1\ c_3\ c_4$$

where the action B means "begin transaction", the initial values of A and B are 10 and 50, respectively, and every write action increases the value of the element on which it operates by 10.

1. Suppose that S is executed by PostgreSQL, and tell which are the values read by all the "read" actions in the schedule, in both the following two cases: (1) all the transactions are defined with the isolation level "read committed"; (2) all the transactions are defined with the isolation level "repeatable read".
2. Tell whether S is a 2PL schedule with both exclusive and shared locks, motivating the answer in detail.

Problem 3
Let $R(A, B)$ and $S(C, D)$ be two relations with $B(R)$ and $B(S)$ pages, respectively, where R is stored in a file sorted on B, and S is stored in a heap. Let us denote by $\sigma(R, S)$ the operator computing the following result:

$$\sigma(R, S) = \{\ x \in R[A] \mid \text{for all } y \text{ such that } (x, y) \in R,\ \text{ we have that } (x, y) \in S\ \}$$

Assuming that $M > 2$ free buffer frames are available, describe the most efficient algorithm that you think we can use to compute the result of $\sigma(R, S)$, and tell which is the cost (as a function of M, $B(R)$ and $B(S)$) of such algorithm in terms of number of page accesses.

Problem 4
Consider the relations `R(A,B)` with `B` as the key and 1.000 pages, `S(C,D)` with 3.000 pages and `T(E,F)` with 250.000 pages, each of them stored in a heap. We know that each page of every relation has space for 100 tuples. We also know that V`(S,C)` = 1.000, V`(S,D)` = 10, V`(T,E)` = 1.000 and that we have 302 free buffer frames available. Consider the query ("minus" is the difference operator):

```
select A,D from R, S where B=C and D=100
 minus
select E,F from T where E = 200
```

Show the logical query plan associated to the query, as well as the logical query plan and the physical query plan you would choose for executing the query efficiently. Also, tell which is the cost (in terms of number of page accesses) of executing the query according to the chosen physical query plan.

Problem 5
Let $\mathtt{R}_1$`(A,C)` be a relation with $B(\mathtt{R}_1)$ pages stored in a heap, and let $\mathtt{R}_2$`(A,C)` be a relation with `A` as the key and with $B(\mathtt{R}_2)$ pages also stored in a heap. We know that $B(\mathtt{R}_1) < B(\mathtt{R}_2)$, and we have to compute the difference between $\mathtt{R}_1$ and $\mathtt{R}_2$, in two different scenarios: (a) $\mathtt{R}_1$ has no duplicates; (b) $\mathtt{R}_1$ has duplicates. For each of the above scenarios, tell whether the "block-nested loop algorithm" can be used or not; if the answer is negative, then motivate the answer in detail, and if the answer is positive, then describe the algorithm and characterize its cost in terms of number of page accesses, assuming that we have $M > 2$ buffer frames available.