Data Management – exam of 08/06/2022 (Compito B)

Problem 1

Let τ indicate the ternary operator such that $\tau(R, S, T) = R - (S \cup_b T)$, where R, S and T are three relations with the same schema and without duplicates, $\cup_b$ indicates bag union and $-$ indicates bag difference.

1.1 Design and describe in detail a two pass algorithm that, given R, S, T, each one stored as a heap, computes $\tau(R, S, T)$.

1.2 Tell under which condition the algorithm can be used and illustrate the cost of the algorithm in terms of number of page accesses.

Problem 2

Given a schedule S, a serial schedule S_1 on the same transactions as S is said to be "begin-order preserving with respect to S" if it satisfies the following property: for every pair of transactions T_i, T_j in S, if the first action of T_i precedes the first action of T_j in S, then T_i precedes T_j in S_1. A schedule S is called *begin-order preserving view serializable* if there exists a serial schedule S_1 on the same transactions that is both view equivalent to S and begin-order preserving with respect to S.

2.1 Prove or disprove the following claim: every conflict serializable schedule is "begin-order preserving view serializable".

2.2 Is the problem of checking whether a schedule is "begin-order preserving view serializable" decidable? If the answer is negative, then motivate the answer; if the answer is positive, then exhibit an algorithm for the problem, provide evidence of the correctness of the algorithm and illustrate its computational conplexity.

Problem 3

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

$$B(T_0)\ r_0(E)\ c_0\ B(T_1)\ r_1(A)\ r_1(E)\ w_1(E)\ B(T_2)\ r_2(E)\ w_2(E)\ B(T_3)\ r_3(A)\ w_3(A)\ r_1(A)\ c_3\ r_1(A)\ c_1\ c_2$$

where the action B means "begin transaction", the initial values of A and E are 10 and 30, respectively, and every write action increases the value of the element on which it operates by 10. Suppose that S is executed by PostgreSQL, and describe what happens when the scheduler analyzes each action (illustrating also which are the values read and written by all the "read" and "write" actions) 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".

Problem 4

Consider the relations `Bus(code,stopcode,passengers)` with 15.000.000 tuples, and `Stop(stopcode,district,city)` occupying 290 pages, each page with 100 tuples (keys are underlined). We assume that 150 values are in the attribute `city`, that each value (regardless of the type) requires the same number of bytes and that we have 300 frames available in the buffer. Consider the query

```
select city, sum(passengers)
from Bus b, Stop s where b.stopcode = s.stopcode
group by city
```

and describe the algorithm you would use to execute the query, illustrating the number of page accesses required by the execution of the algorithm.

Problem 5

Consider a graph database with nodes of type `Shop` with properties `name` (identifying the shop) and `city`, nodes of type `Product` with property `category`, and edges of type `Sold` connecting each shop with the products sold in that shop. In such database, for example, one may represent the node `s1` of type `Shop` with `name`:"Bakery100" and `city`:"Roma" connected to node `p1` of type `Product` with `category`:"book" by means of an edge of type `Sold`.

5.1 Illustrate how you would represent the above database as a schema in the relational model.

5.2 Assuming that the three most relevant queries are (1) given the name of a shop, compute the products sold by the shop, (2) produce the sorted list of ⟨name of shop, name of product⟩ for shops and the products they have sold and (3) compute all the shops of a given city, describe the file organizations you would choose for the relations in the relational database mentioned above, and then describe the algorithms for executing the queries on the basis of such representation.