Data Management (A.A. 2023/24) – exam of 05/06/2024

Problem 1 A transaction T is called *eager* if all its requests for an exclusive lock appear just after the "begin" action of T (and, therefore, before every other action of T). A lock-based scheduler with both shared and exclusive locks is called *disciplined* if it behaves like a 2PL scheduler with the only difference that it aborts all the transactions that are not eager.

1.1 Show a schedule that is accepted by the 2PL scheduler (with both shared and exclusive locks) and is not accepted by a disciplined scheduler.

1.2 Prove or disprove that a disciplined 2PL sheduler can avoid to deal with deadlock management.

Problem 2 Consider the relations `R(A,B)` and `S(B,C)`, where the attributes forming the key are underlined. Each of them is stored in a heap with 1.000 pages. We have to compute the equi join of `R` and `S` on the condition `R.B` = `S.B`, knowing that we have M free frames available in the buffer. For the two cases of $M = 3$ and $M = 32$, answer the following questions. (2.1) Is the block-nested loop applicable? If the answer is negative, provide a motivation; if the answer is positive, illustrate the algorithm and tell which is its cost in terms of number of page accesses. (2.2) Illustrate what you think is the most efficient algorithm, and tell which is its cost in terms of number of page accesses.

Problem 3 Consider the relations `R(A,B,C,D,E,F)` and `S(A,B,G,H)`, where `R` contains 260.000 tuples (20 tuple per page), `R` contains 1.000 distinct values uniformly distributed in the integer attribute `A`, there is a hash-based index for `S` on the integer attribute `A`, there is a B^+-tree index for `R` on attribute `A`, and the buffer has 205 free frames available. Consider the query type Q (where `x` and `y` are integer constants):

```
select A,B from R where A ≥ x and A < x + 20
union
select A,B from S where A = y
```

for which we remind the reader that the SQL `union` operator performs the union of two sets (or bags) by eliminating duplicates in the result. Illustrate the algorithm you would use to answer queries of type Q, and tell which is the cost of the algorithm in terms of number of page accesses.

Problem 4 Consider the relations `R(A,B,C)` and `S(A,B,C)`, where (*i*) `R` has ⟨`A,B,C`⟩ as the key and contains 100.000 tuples stored in a heap with 1.000 pages; (*ii*) `S` contains 12.000 tuples stored as a heap of 120 pages. Our buffer has 100 free frames available, and we have to compute the difference between `R` and `S` (i.e., `R - S`), with the goal of *maximizing efficiency*, obviously in terms of number of page accesses. Illustrate the algorihm you would choose and tell which is its cost in terms of number of page accesses.

Problem 5 (only for students who do **not** do the project)
Let B be a relational database with relations `Agency(id,nation)`, `Travel(tcode,agencyid,duration)`, `Visit(tcode,citycode)`, `Participate(tcode,person)`, `Likes(person,citycdode)`, where (*i*) each travel agency is located in a nation, (*ii*) each travel is offered by an agency, has a duration, includes a set of at most 10 cities to visit (relation `Visit`) and accepts at most 50 participants (relation `Participate`); (*iii*) each person likes a set of at most 20 cities (relation `Likes`).

5.1 Describe how would you organize a property graph database G in order to represent the relational database B. In particular, (*i*) specify how nodes, edges, labels, etc. of G are used in order to capture the information stored in the tables of B and (*ii*) choose a few tuples for the relations in B, and show the specific property graph database G obtained by applying the chosen representation method.

5.2 Consider the query Q *on the property graph database* that, given the `tcode` of a specific travel T, returns each person who participates to T and likes at least one city visited by T. Choose *one* of the following questions and answer it. (*i*) Assuming the graph database organized as decided for point (5.1), formulate the query on such graph database using a Cypher-like syntax. (*ii*) Assuming that every page of our system contains 1.000 values and there are 10 free buffer frames, illustrate a possible algorithm that the graph database system can use to answer the query and tell which is its cost in terms of number of page accesses.