Data Management – exam of 12/07/2019

Problem 1

A *transition point* in a schedule S over transactions $\{T_1, \ldots, T_n\}$ is a sequence of two consecutive actions a_i, b_j in S such that a_i is a read or a write action of transaction T_i, b_j is a read or a write action of transaction T_j with $i \neq j$, and a_i is not the final read or write action of T_i. A schedule is said to be *quasi-serial*, if it contains exactly one transition point. By referring to the transactions $T_1 = r_1(x)\, w_1(y)$, $T_2 = w_2(x)\, w_2(y)$, and $T_3 = w_3(x)\, w_3(y)$, prove or disprove the following statements, motivating each of the answers in detail.

1.1 There is no quasi-serial schedule on transactions $\{T_1, T_2, T_3\}$ that is conflict-serializable.

1.2 There is no quasi-serial schedule on transactions $\{T_1, T_2, T_3\}$ that is view-serializable.

1.3 There is no quasi-serial schedule on transactions $\{T_1, T_2, T_3\}$ that is strict.

1.4 There is no quasi-serial schedule on transactions $\{T_1, T_2, T_3\}$ that is rigorous.

Problem 2

The relation Student(id,school,gender,grade,year) contains information about the final grades of a set of high-school students in the last 30 years. We know that there are 12.000 schools, for each school 200 students per year took the final exam, 50% of the students are female, and 100 tuples of the relation fit in one page. The relation is stored as a heap file in node 1 of a system with 30 nodes in total, where each node has 420 frames available in its buffer. Consider the query Q_1 shown on the right, and tell both which is the algorithm you would use to answer the query, and its cost in terms of number of page accesses.

Query Q_1:

```
select school, year, avg(grade)
from Student
where gender = 'female'
group by school, year
order by school
```

Problem 3

The *quotient* between two relations, denoted by $\div$, is a binary operator of the extended relational algebra, defined as follows: if R(A,B,C,D) and S(C,D) are two relations, the result of R $\div$ S is the set

$$\{\ \langle a, b\rangle\ \in\ \mathsf{R[A,B]} \mid \forall c, d:\ \langle c, d\rangle\ \in\ \mathsf{S}\ \rightarrow\ \langle a, b, c, d\rangle\ \in\ \mathsf{R}\ \}$$

In other words, R $\div$ S contains all the tuples of R[A,B], i.e., the projection of R on attributes A,B, that are combined in R with all the tuples of S. Let R and S be stored as heap files with 160.000 pages, and 450 pages, respectively, and let the buffer have 500 frames available. Describe the algorithm you would use to compute the result of R $\div$ S, and tell which is the cost of the algorithm in terms of number of page accesses.

Problem 4

Consider the following schedule on transactions $\{T_1, T_2, T_3\}$:

$$S = r_1(x)\, w_3(x)\, r_2(x)\, r_1(y)\, w_1(x)\, r_2(y)\, w_2(y)\, w_3(y)$$

4.1 Show the precedence graph associated to S, and tell whether S is conflict-serializable or not, explaining the answer in detail.

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

4.3 If the answer to 4.2 is positive, then show a serial schedule on transactions $\{T_1, T_2, T_3\}$ that is view-equivalent to S. If the answer to 4.2 is negative, then tell whether there is a way to add to S suitable operations on local variables carried out by the transactions $\{T_1, T_2, T_3\}$ in such a way that the resulting schedule is serializable. We remind the student that when we consider the operations on local variables carried out by transactions, the read and write actions are expressed in the notation $r_i(x, Z)$ – meaning that the value of the database element x is read by transaction i and then stored in the local variable Z – and $w_i(x, Z)$ – meaning that the value of the local variable Z is written by transaction i in x.

4.4 Tell whether S is ACR (Avoiding Cascading Rollbacks) or not, explaining the answer in detail.

Problem 5

Assume that the relation R(A,B,C,E,F,G) has 960.000 tuples, each attribute and each pointer in the system occupy 40 Bytes, each page has space for 240 Bytes, B has 1.000 values uniformly distributed among the tuples of R, and there is an unclustering B$^+$-tree index on R with search key B using alternative 2. Consider the query Q_2 shown on the right, describe the algorithm you would use to compute the result of evaluating the query, and tell which is the cost of the algorithm in terms of number of page accesses.

Query Q_2:

```
select A,E
from R
where B = 10 and C = 20
```

Problem 1

A *transition point* in a schedule S over transactions $\{T_1, \ldots, T_n\}$ is a sequence of two consecutive actions a_i, b_j in S such that a_i is a read or a write action of transaction T_i, b_j is a read or a write action of transaction T_j with $i \neq j$, and a_i is not the final read or write action of T_i. A schedule is said to be *quasi-serial*, if it contains exactly one transition point. By referring to the transactions $T_1 = r_1(x)\, w_1(y)$, $T_2 = w_2(x)\, w_2(y)$, and $T_3 = w_3(x)\, w_3(y)$, prove or disprove the following statements, motivating each of the answers in detail.

1.1 There is no quasi-serial schedule on transactions $\{T_1, T_2, T_3\}$ that is conflict-serializable.

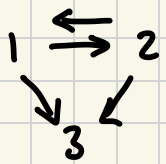
1.2 There is no quasi-serial schedule on transactions $\{T_1, T_2, T_3\}$ that is view-serializable.

1.3 There is no quasi-serial schedule on transactions $\{T_1, T_2, T_3\}$ that is strict.

1.4 There is no quasi-serial schedule on transactions $\{T_1, T_2, T_3\}$ that is rigorous.

T_1: $r_1(x)$ $w_1(y)$
T_2: $w_2(x)$ $w_2(y)$
T_3: $w_3(x)$ $w_3(y)$

1) S: $r_1(x)$ $w_2(x)$ $w_2(y)$ $w_1(y)$ $w_3(x)$ $w_3(y)$



S IS A QUASI SERIAL SCHEDULE AND IT'S CONFLICT-SER

2) S: $r_1(x)$ $w_2(x)$ $w_2(y)$ $w_1(y)$ $w_3(x)$ $w_3(y)$

VIEW BUT NO CONFLICT

3) S: $r_1(x)$ $w_2(x)$ $w_2(y)$ c_2 $w_1(y)$ c_1 $w_3(x)$ $w_3(y)$ c_3

4)

Problem 4

Consider the following schedule on transactions $\{T_1, T_2, T_3\}$:

$$S = r_1(x)\, w_3(x)\, r_2(x)\, r_1(y)\, w_1(x)\, r_2(y)\, w_2(y)\, w_3(y)$$

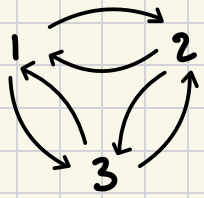
4.1 Show the precedence graph associated to S, and tell whether S is conflict-serializable or not, explaining the answer in detail.

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

4.3 If the answer to 4.2 is positive, then show a serial schedule on transactions $\{T_1, T_2, T_3\}$ that is view-equivalent to S. If the answer to 4.2 is negative, then tell whether there is a way to add to S suitable operations on local variables carried out by the transactions $\{T_1, T_2, T_3\}$ in such a way that the resulting schedule is serializable. We remind the student that when we consider the operations on local variables carried out by transactions, the read and write actions are expressed in the notation $r_i(x, Z)$ – meaning that the value of the database element x is read by transaction i and then stored in the local variable Z – and $w_i(x, Z)$ – meaning that the value of the local variable Z is written by transaction i in x.

4.4 Tell whether S is ACR (Avoiding Cascading Rollbacks) or not, explaining the answer in detail.

1) $S = r_1(x)\, w_3(x)\, r_2(x)\, r_1(y)\, w_1(x)\, r_2(y)\, w_2(y)\, w_3(y)$



THE GRAPH IS CYCLIC, SO S IS NOT CONFLICT.

2) READ FROM = $\langle r_2(x), w_3(x) \rangle$

LAST WRITE = $w_1(x)$, $w_3(y)$

THERE IS NO S' SERIAL THAT RESPECT THE SAME READ FROM AND LAST WRITE

3)