Data Management – exam of 09/09/2022

Problem 1

Suppose that for every schedule S on transactions $\{T_1, \ldots, T_n\}$ and for every element x of the database, we have a sequence $\sigma_{S,x}$ that is a total order on the transactions $\{T_1, \ldots, T_n\}$. Consider a scheduler R that behaves as follows when analysing action $a_i(y)$: if there is an action $b_j(y)$ (with $j \neq i$) preceeding $a_i(y)$ in S such that T_j appears after T_i in $\sigma_{S,y}$, then S is not accepted, otherwise the action $a_i(y)$ is executed and the schedule proceeds. Question: is it true that every schedule S accepted by R is conflict serializable? If your answer to the question is positive, then provide a proof. Otherwise, (i) provide a proof that the answer is negative and (ii) specify a condition on the various $\sigma_{S,x}$ under which the answer becomes true. If you do not know how to solve the problem in general, try to solve it in the case where S is on just two transactions, i.e., the case of $n = 2$.

Problem 2

Let S be the schedule $\boxed{r_1(X)\, r_3(Z)\, w_1(Y)\, w_2(X)\, w_1(Z)\, w_3(U)\, w_1(V)\, w_2(U)\, w_2(V)\, r_3(T)}$ and answer the following questions: (2.1) Tell whether S is accepted by the 2PL scheduler with exclusive and shared locks. If the answer is yes, then show the 2PL schedule obtained from S by adding suitable lock and unlock commands. If the answer is no, then motivate the answer. (2.2) Tell whether we can insert into S the commit commands for all the transactions in S such that the resulting schedule S' is commit order preserving conflict serializable. If the answer is yes, then show S', otherwise motivate the answer.

Problem 3

Let R(A,B,C) and S(C,D) two relations stored in two heap files with 1.000 and 5.000 pages, respectively. We have to compute the natural join between the two relations. If M is the number of frames, are there values of M for which the block nested loop algorithm is more efficient than the multi-pass algorithm based on sorting for executing the operation? If the answer is negative, then motivate the answer. If the answer is positive, then tell which are the values of M for which the block nested loop algorithm is more efficient than the multi-pass algorithm, again motivating the answer.

Problem 4

In this exercise we assume that every value and every pointer occupies the same number of bytes and the buffer has 100 frames. Let R(A,B,C) and S(D,E,F,G,H,L) be two relations stored as heap files. R has 1.000 pages, where each page has 100 tuples, and the attribute A has 500 values uniformly distributed over the tuples of R. S has 2.000 pages and has an associated B^+-tree index using alternative 2 with search key equal to the primary key ⟨D,E⟩. Consider the query `select A, F from R,S where A=20 and B=D and C=E` and answer the following questions: (4.1) Determine which is the most efficient between the "index-based" and the "two-pass" strategies for the above query, showing and comparing the cost of the algorithms based on the two strategies. (4.2) Would the answer change if the number of pages of R were 20.000?

Problem 5 (A.Y. 2021/22)

We refer to a data warehouse on car accidents. An accident (having an associated cost) is caused by a driver (with an age) driving a vehicle (with a certain power) in a date (day, month and year) and in a city (belonging to a province, which is part of a region). Sometimes the accident has a witness (again, of a certain age, but also with an address) and requires an emergency vehicle (again, with a certain power). The student is asked to (5.1) show the conceptual representation of the above domain in terms of the Dimensional Fact Model, (5.2) produce the corresponding "star schema", and (5.3) write the SQL query that computes the average cost of the accidents that required an emergency vehicle and occurred in 2021 in the Lazio region.

Problem 5 (A.Y. before 2021/22) Consider the relation `OPERA(ocode,author,duration)` and the relation `CONCERT(ccode,artist,date,city)`, recording the operas (with their duration) and the concerts (with date and city) where the artists have performed such operas. We know that `CONCERT` has 2.000.000 tuples, each page contains 50 such tuples in the average, and has a dense B^+-tree index with search key ⟨`ccode,artist`⟩ using alternative 2. We also know that `OPERA` is stored in a file with 400.000 pages sorted on ⟨`ocode,author`⟩, that there are 100 buffer frames available, that each `OPERA` is performed in 500 concerts at most, and that each value or pointer occupies the same space in memory. With regard to the query

```
select artist,author from OPERA join CONCERT on ocode = ccode
where author = 'Mozart' or author = 'Verdi'
```

(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.

Problem 1

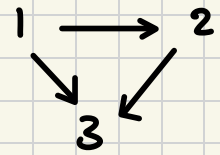
Suppose that for every schedule S on transactions $\{T_1, \ldots, T_n\}$ and for every element x of the database, we have a sequence $\sigma_{S,x}$ that is a total order on the transactions $\{T_1, \ldots, T_n\}$. Consider a scheduler R that behaves as follows when analysing action $a_i(y)$: if there is an action $b_j(y)$ (with $j \neq i$) preceeding $a_i(y)$ in S such that T_j appears after T_i in $\sigma_{S,y}$, then S is not accepted, otherwise the action $a_i(y)$ is executed and the schedule proceeds. Question: is it true that every schedule S accepted by R is conflict serializable? If your answer to the question is positive, then provide a proof. Otherwise, (i) provide a proof that the answer is negative and (ii) specify a condition on the various $\sigma_{S,x}$ under which the answer becomes true. If you do not know how to solve the problem in general, try to solve it in the case where S is on just two transactions, i.e., the case of $n = 2$.

T_1: $r_1(x)$ $w_1(y)$

T_2: $r_2(y)$ $w_2(x)$

T_3: $r_3(x)$ $w_3(y)$

T_1 T_2 T_3 : $r_1(x)$ $w_1(y)$ $r_2(y)$ $w_2(x)$ $r_3(x)$ $w_3(y)$



THE PRECEDENCE GRAPH IS ACYCLIC, SO IT'S CONFLICT-SER

T_1: $r_1(x)$ $w_1(y)$

T_2: $r_2(y)$ $w_2(x)$

Problem 2

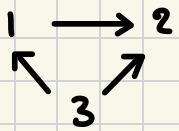
Let S be the schedule $r_1(X)\, r_3(Z)\, w_1(Y)\, w_2(X)\, w_1(Z)\, w_3(U)\, w_1(V)\, w_2(U)\, w_2(V)\, r_3(T)$ and answer the following questions: (2.1) Tell whether S is accepted by the 2PL scheduler with exclusive and shared locks. If the answer is yes, then show the 2PL schedule obtained from S by adding suitable lock and unlock commands. If the answer is no, then motivate the answer. (2.2) Tell whether we can insert into S the commit commands for all the transactions in S such that the resulting schedule S' is commit order preserving conflict serializable. If the answer is yes, then show S', otherwise motivate the answer.

1) $r_1(X)\, r_3(Z)\, w_1(Y)\, w_2(X)\, w_1(Z)\, w_3(U)\, w_1(V)\, w_2(U)\, w_2(V)\, r_3(T)$

S': $sl_1(x)\ r_1(x)\ sl_3(z)\ r_3(z)\ xl_1(y)\ w_1(y)\ xl_3(u)\ sl_3(T)\ u_3(z)$
$xl_1(z)\ xl_1(v)\ u_1(x)\ xl_2(x)\ w_2(x)\ w_1(z)\ w_3(u)\ w_1(v)\ u_3(u)$
$xl_2(u)\ w_2(u)\ u_1(v)\ xl_2(v)\ w_2(v)\ r_3(T)\ u_1(y)\ u_1(z)$
$u_2(x)\ u_2(u)\ u_2(v)\ u_3(T)$

2) $r_1(X)\, r_3(Z)\, w_1(Y)\, w_2(X)\, w_1(Z)\, w_3(U)\, w_1(V)\, w_2(U)\, w_2(V)\, r_3(T)$

S': $r_1(x)\ r_3(z)\ w_1(y)\ w_2(x)\ w_1(z)\ w_3(u)\ w_1(v)\ w_2(u)\ w_2(v)\ r_3(T)$



CONFLICT SER

WE SHOULD HAVE

c_1 BEFORE c_2, c_3 BEFORE c_1, c_3 BEFORE c_2, c_1 BEFORE c_2

S': $r_1(x)\ r_3(z)\ w_1(y)\ w_2(x)\ w_1(z)\ w_3(u)\ c_3\ w_1(v)\ c_1\ w_2(u)\ w_2(v)\ c_2\ r_3(T)$

Problem 3

Let R(A,B,C) and S(C,D) two relations stored in two heap files with 1.000 and 5.000 pages, respectively. We have to compute the natural join between the two relations. If M is the number of frames, are there values of M for which the block nested loop algorithm is more efficient than the multi-pass algorithm based on sorting for executing the operation? If the answer is negative, then motivate the answer. If the answer is positive, then tell which are the values of M for which the block nested loop algorithm is more efficient than the multi-pass algorithm, again motivating the answer.

MULTI PASS

WE HAVE TO SORT R AND S.

THE RUNS MUST FIT IN M OR $M(M-1)$ OR $M(M-1)(M-1)$ ...

$$\text{RUNS}(R) + \text{RUNS}(S) \leq M(M-1)\ldots$$

THE COST IS $((K \cdot 2) - 1)\,(B(R) + B(S))$

NESTED

WE NEED 1000 FRAMES FOR R, 1 FOR S, 1 FOR THE VALUES OF THE CARTISIAN PRODUCT, AND 1 FOR THE OUTPUT

1003 FRAMES

COST = $B(R) + B(S)$ ✓

Problem 4

In this exercise we assume that every value and every pointer occupies the same number of bytes and the buffer has 100 frames. Let R(A,B,C) and S(D,E,F,G,H,L) be two relations stored as heap files. R has 1.000 pages, where each page has 100 tuples, and the attribute A has 500 values uniformly distributed over the tuples of R. S has 2.000 pages and has an associated B^+-tree index using alternative 2 with search key equal to the primary key ⟨D,E⟩. Consider the query `select A, F from R,S where A=20 and B=D and C=E` and answer the following questions: (4.1) Determine which is the most efficient between the "index-based" and the "two-pass" strategies for the above query, showing and comparing the cost of the algorithms based on the two strategies. (4.2) Would the answer change if the number of pages of R were 20.000?

1) TWO PASS

PASS 1:

R: 1000/100 = 10 RUNS
S: 2000/100 = 20 RUNS
} 30 RUNS < 99

PASS 2:

WE READ ONCE R AND S TO COMPARE

COST = 3(B(R) + B(S)) = 9000 PAGE ACCESSES

B^+ TREE

THE INDEX IS DENSE, A DATA ENTRY FOR EACH TUPLE

S HAS 50 · 2000 = 100 000 TUPLES

A PAGE OF S HAS 50 · 6 = 300 VALUES

300/3 (⟨KEY, RID⟩) = 100 DATA ENTRIES —67%→ 67 ENTRIES

F = 0.75 · 100 = 75 N = 100 000/67 = 1493 LEAVES

$\log_F N = 2$ COST = $B(S) + 200 \cdot (\log_F N + 1)$ = 1600 PAGE ACCESSES

READ F

100 000 (TUPLES R)/500 (VALUES A) = 200