

Data Management – exam of 13/07/2022

Problem 1

In a schedule S on transactions $\{T_1, \dots, T_n\}$ we say that two transactions T_i, T_j share the element X of the database if there exist actions $\alpha(X)$ in T_i and $\beta(X)$ in T_j such that α is either r_i or w_i and β is either r_j or w_j . Moreover, S is called “chary” if (i) no transaction in S uses the same element twice, and (ii) for every $T_i, T_j \in \{T_1, \dots, T_n\}$, T_i and T_j share at most one element. Prove or disprove the following claims:

- 1.1 Every chary schedule is view-serializable.
- 1.2 Every chary schedule on two transactions is conflict-serializable.
- 1.3 Every chary schedule on two transactions is a 2PL schedule with exclusive and shared locks.

Problem 2

Let S be the schedule: $r_1(Z) w_3(Y) w_3(V) r_1(Y) r_2(V) w_2(Y) w_3(X) r_2(X) r_2(Z) r_3(Z) w_4(Z) w_4(X) w_2(X)$

- 2.1 Tell whether S is accepted by the 2PL scheduler with exclusive and shared locks. If the answer is yes, then specify the 2PL schedule obtained from S by adding suitable lock and unlock commands. If the answer is no, then explain the answer.
- 2.2 Tell whether S is view-serializable. If the answer is yes, then illustrate a serial schedule which is view-equivalent to S . If the answer is no, then explain the answer.
- 2.3 Answer all the following questions, motivating the answers: (i) Is S recoverable? (ii) Is S ACR? (iii) Is S strict?

Problem 3

Let $R(A,B,C)$, $S(A,D,E)$, $T(A,B,C)$ be three tables (where T is a bag) and let τ indicate the ternary operator such that $\tau(R,S,T) = \delta(T \cup_b \pi_{A,B,C}(R \bowtie S))$, where δ denotes duplicate elimination, \cup_b denotes bag union and \bowtie denotes natural join.

- 3.1 Design and describe in detail a one pass algorithm that, given R,S,T as above, each one stored as a heap, computes $\tau(R,S,T)$.
- 3.2 Tell what is the weakest condition under which the algorithm can be used and illustrate the cost of the algorithm in terms of number of page accesses.
- 3.3 Tell what does it change if all the tables have A as key and are stored as sorted file with search key A .

Problem 4

Consider the relations $\text{Flight}(\underline{\text{code}}, \text{company}, \text{type})$ with 1.000 pages and 10.000 tuples, and $\text{Ticket}(\underline{\text{number}}, \text{code}, \text{company}, \text{type})$ with 2.000 pages and an associated index on Ticket with search key $\langle \text{company}, \text{type} \rangle$, for which we know that the cost of retrieving the records with a given value of attribute company is 3 page accesses. Assume a buffer with 50 frames, and consider the two queries shown below.

Query Q_1 :

```
select code, company from Flight
except all -- not removing duplicates
select code, company from Ticket
```

Query Q_2 :

```
select company, type from Flight
except all -- not removing duplicates
select company, type from Ticket
```

where “except all” denotes bag difference. For both queries Q_1 and Q_2 , tell (i) whether it is possible to process the query by using a block-nested loop algorithm, and (ii) whether it is possible to process the query by using an index-based algorithm. In all four cases, if the answer is positive, then describe the algorithm and tell which is its cost in terms of number of page accesses. If the answer is negative, then motivate the answer in detail.

Problem 5 (A.Y. 2021/22)

Describe in detail the notion of “star schema” in data warehousing and illustrate the difference between such a notion and the notion of “snowflake schema”.

Problem 5 (A.Y. before 2021/22)

Consider the relations $R(A,B,C)$ (with 1.500 pages and with one duplicate, in the average, for each tuple) and $S(\underline{D},E)$ (with 7.000 pages), and the query `select distinct B,E from R, S where A <> 3`. Assume that the buffer contains 260 frames and 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 as efficiently as possible. Also, tell which is the cost (in terms of number of page accesses) of executing the query according to the chosen physical query plan.