

**Problem 1**

Consider the following schedule

$$S = w_1(u) r_1(x) w_3(x) r_2(y) r_1(y) c_1 r_4(u) w_2(y) w_3(y) c_3 w_4(z) c_4 r_2(z) c_2.$$

- 5.1 Tell whether  $S$  is a 2PL schedule (with shared and exclusive locks) or not, explaining the answer in detail.
- 5.2 Tell whether  $S$  is view-serializable or not, explaining the answer in detail.
- 5.3 Describe the behavior of the timestamp-based scheduler when processing the schedule  $S$ , assuming that, initially,  $rts(\alpha)=wts(\alpha)=0$ , and  $wts-c(\alpha)=cb(\alpha)=\mathbf{true}$  for each element  $\alpha$  of the database, and assuming that the subscript of each action denotes the timestamp of the transaction executing such action.
- 5.4 Tell whether  $S$  is ACR (Avoiding Cascading Rollback) or not, explaining the answer in detail.

**Problem 2**

Assume that relation  $R(A,B)$  has 10.000 tuples, relation  $Q(C,D,E,F)$  has 400.000 tuples, attribute  $D$  has 2.000 values uniformly distributed on the tuples of  $Q$ , each page of our system contains 400 Bytes, every attribute value or pointer requires 20 Bytes, and we have 252 free buffer frames. Consider the query

```
select A
from R
where not exists (select * from Q where Q.D = R.B)
```

and tell which is the algorithm you would use and the corresponding cost (in terms of number of page accesses) for executing such query for each of the following methods for representing  $Q$ : (1) heap file; (2) sorted file with sorting key  $D$ ; (3) heap file with unclustering, dense sorted index with duplicates with search key  $D$  (strongly dense index); (4) sorted file with clustering, dense sorted index without duplicates with search key  $D$ ; (5) sorted file with clustering, sparse sorted index with search key  $D$ .

**Problem 3**

The SQL table  $R(A,B,C)$  has 1.400.000 tuples stored in a heap file, the SQL table  $Q(E,F,G,H,L)$  has 2.400.000 tuples stored in a heap file, and there is a  $B^+$ -tree index on  $(E,F)$ , where  $(E,F)$  is the key of  $Q$ . We know that each attribute or pointer occupies 20 Bytes, the size of each page is 600 Bytes, and the buffer has 400 free frames. Consider the following SQL query (we remind the student that the `minus` clause in SQL computes the difference of two tables by eliminating duplicates):

```
select A,B from R order by A,B
minus
select E,F from Q order by E,F
```

Describe in detail the algorithm you would use to compute the answer to the above query, and tell which is the cost of the algorithm in terms of number of page accesses.

**Problem 4**

A schedule  $S$  is said to be *parsimonious* if it satisfies the following two conditions: (i) every transaction in  $S$  that does not contain any “read” action contains exactly one “write action”, and every transaction that contains at least one “read” action does not contain any “write” action; (ii) no element of the database is read more than once in  $S$ , and no element of the database is written more than once in  $S$ . Prove or disprove the following propositions:

1. Every parsimonious schedule is a 2PL schedule (with shared and exclusive locks).
2. Every parsimonious schedule is conflict serializable.
3. Every parsimonious schedule is view serializable.

**Problem 5**

Suppose that we have only 3 buffer frames available, and we have to compute the union (producing a result without duplicates) of two SQL tables (it is well known that an SQL table may contain duplicates). Describe in detail the algorithm you would use, and tell which is the cost of the algorithm in terms of number of page accesses.