UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

INFR11199 ADVANCED DATABASE SYSTEMS PG AND INFR11217 UG

Thursday 11 th May 2023

13:00 to 15:00

INSTRUCTIONS TO CANDIDATES

1. **Note that ALL QUESTIONS ARE COMPULSORY.**
2. **DIFFERENT QUESTIONS MAY HAVE DIFFERENT NUMBERS OF TOTAL MARKS. Take note of this in allocating time to questions.**
3. **This is a NOTES AND CALCULATORS PERMITTED examination: candidates may consult up to THREE A4 pages (6 sides) of notes.**

   CALCULATORS MAY BE USED IN THIS EXAMINATION.

Year 4 Courses

Convener: K.Etessami

External Examiners: J.Bowles, A.Pocklington, R.Mullins

THIS EXAMINATION WILL BE MARKED ANONYMOUSLY

Joins

1. Consider a variant of the relational join operator called "left anti join", which outputs only those tuples in the left relation that do not join with any tuple in the right relation. For example, consider two relations $R(A, B)$, and $S(B, C)$, where R consists of the tuples $(1, 10), (1, 20), (2, 30), (2, 40)$, and S consists of the tuples $(10, 60), (10, 70), (30, 80)$. Then, R `LeftAntiJoin` S on the B attribute produces two tuples: $(1, 20)$ and $(2, 40)$. There are duplicates

   For the following questions, assume that R consists of M pages, S consists of N pages, the buffer pool consists of B pages, and the cost model only counts the number of disk I/O operations, as discussed in class.

   (a) Adapt the *block nested-loops join* algorithm to process left anti joins. Provide the pseudocode of your algorithm, and estimate its cost. [*5 marks*]

   (b) Adapt the *sort-merge join* algorithm to process left anti joins. Provide the pseudocode of your algorithm, and estimate its cost. [*5 marks*]

Hardware & Disk Space Management

2. Consider a disk-based database system where the storage device (i.e., hard-disk drive) has a controller that uses its own memory for page prefetching.

   (a) Explain what page prefetching is, and what its benefits are. [*3 marks*]

   (b) Give reasons why the database system may not want to rely on prefetching controlled by the disk. [*3 marks*]

Tree based Indexing

3. Consider an unclustered B+-tree index built on the primary key of a relation with 100 000 records. Assume the following:

   - Each primary key is a 12-byte string.
   - Pointers (i.e., record IDs, page IDs) are 8-byte values.
   - The size of one page (node) is 1000B.

   (a) Assume that all nodes in the index are filled up as much as possible. What is the height of this tree? How many nodes are at each level of the tree? Explain your answer. [*5 marks*]

   (b) Assume now that all nodes in the index are filled up as much as possible but no more than 70% of the page size. What would be the height of this tree? Explain your answer. [*2 marks*]

Locking

4. (a) Consider the following sequence of lock requests made by five transactions. Here S(·) and X(·) stand for "shared lock" and respectively "exclusive lock".

time	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
T_1	X(A)									S(D)
T_2		S(A)								
T_3			X(B)					S(C)	X(A)	
T_4				S(C)			X(A)			
T_5					S(D)	X(B)				

Give the wait-for-graph for this sequence of lock requests. Determine if there exists a deadlock in the lock requests, and explain why. [*3 marks*]

(b) Argue whether the following claim is true or not:
The "wait-die" deadlock avoidance strategy avoids all deadlocks. [*2 marks*]

(c) Argue whether the following claim is true or not:
The "wait-die" deadlock avoidance strategy never aborts transactions unnecessarily. [*2 marks*]

CQ Evaluation and Static Analysis

5. Let Q be a conjunctive query of the form

$$Q(x_1, \ldots, x_k) :- \mathit{body},$$

where *body* mentions only variables (no constant values). Consider two databases D and D', and two tuples $(a_1, \ldots, a_k) \in \mathit{terms}(D)^k$ and $(b_1, \ldots, b_k) \in \mathit{terms}(D')^k$. Recall that $\mathit{terms}(D)$ is the set of all constants occurring in D.

Assume that the following statements hold:

- $(a_1, \ldots, a_k) \in Q(D)$, i.e., $(a_1, \ldots, a_k)$ belongs to the answer to Q over D.
- There exists a function $g : \mathit{terms}(D) \to \mathit{terms}(D')$ such that $g(D) \subseteq D'$ and $g(a_i) = b_i$ for each $1 \leq i \leq k$. In other words, for every atom $R(c_1, \ldots, c_m) \in D$, $R(g(c_1), \ldots, g(c_m)) \in D'$, and $(g(a_1), \ldots, g(a_k)) = (b_1, \ldots, b_k)$.

Prove that $(b_1, \ldots, b_k) \in Q(D')$. [*6 marks*]

CQ Fast Evaluation

6. Consider the Boolean conjunctive query

$$Q :- R(x_1, x_4), P(x_6, x_4, x_7), P(x_1, x_2, x_3), P(x_3, x_5, x_6), S(x_2, x_7)$$

where $x_1, x_2, x_3, x_4, x_5, x_6, x_7$ are variables.

(a) Give the hypergraph $H(Q)$ of the conjunctive query Q.

(b) Is $H(Q)$ acyclic? If yes, then give a join tree of $H(Q)$; otherwise, prove that $H(Q)$ is indeed not acyclic.

[4 marks]

Tutorial

7. Consider the following plan generated by PostgreSQL using the EXPLAIN command on an unknown input query.

```
QUERY PLAN
------------------------------------------------------------------------------
Sort
  Sort Key: station.city DESC
  -> HashAggregate
       Group Key: station.city, trip.start_station_name
       -> Hash Join
            Hash Cond: (weather.zip_code = station.zip_code)
            -> Index Only Scan using weather_pkey on weather
                 Index Cond: ((date >= '2013-01-01'::date) AND
                              (date <= '2013-12-31'::date))
            -> Hash
                 -> Hash Join
                      Hash Cond: (trip.start_station_id = station.station_id)
                      -> Bitmap Heap Scan on trip
                           Recheck Cond: ((bike_id = 200) AND (duration < 100))
                           -> BitmapAnd
                                -> Bitmap Index Scan on idx_bike_id
                                     Index Cond: (bike_id = 200)
                                -> Bitmap Index Scan on idx_duration
                                     Index Cond: (duration < 100)
                      -> Hash
                           -> Seq Scan on station
```

(a) What access methods are used in the query plan? Explain your answer. *[3 marks]*

(b) Based on the query plan, what indices exist in the database? For each index, specify its type (hash, B+ tree, or could be either) and the attributes on which the index was built. *[4 marks]*

(c) Based on the query plan, give one possible SQL query for which PostgreSQL could produce such a plan. *[3 marks]*