

IN3020&4020 – Database Systems Exam, 2022

30 May 2022, 3:00–7:00pm (Inspira)

Question 1 of 8 (14%)

Part a (7%)

Consider the following relations:

Table1:				Table2:	
<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>D</i>	<i>E</i>
1	1	a	k	m	30
2	NULL	NULL	k	p	20
2	NULL	NULL	m		
1	1	a	n		
2	NULL	NULL	n		
1	1	a	m		

Evaluate the following (bag) RA expression over these relations:

$$(\gamma_{A,B,\text{COUNT}(C) \rightarrow \text{Count}C}(\text{Table1} \triangleright \text{Table2})) \bowtie \text{Table2}.$$

Note: here, $\triangleright$ is applied to set relations, and you can evaluate $\triangleright$ using the definition for sets.

Part b (7%)

In the lectures it was erroneously claimed that left outer join is associative (this error is corrected in the slides below). Demonstrate, by giving an example, that left outer join (over set relations) is not associative.

Note: you can use $=|X|$ (or another similar sequence of symbols) instead of $\bowtie$ in your answer to mitigate the difficulties with writing special symbols in text.

Question 2 of 8 (8%)

Consider two relations with the schemas `NorwegianCities` [`CID`, `CName`, `Population`] and `SwedishCities` [`CID`, `CName`, `Population`]. Compile the following query to a (logical) query plan:

```
SELECT CName FROM NorwegianCities WHERE CName <> 'Oslo'
      AND Population > ANY (SELECT Population FROM SwedishCities);
```

Note: as in Question 1, you may introduce and then use reasonable sequences of symbols instead of RA operators.

Question 3 of 8 (14%)

Part a (7%)

Recall that we defined the cost of a (logical) query plan as the estimate of the sum of the sizes (in bytes) of the intermediate results of the plan. Give an example when our approach fails to decide the better plan—that is, give an example of two equivalent query plans Expr and Expr' and relations relevant to Expr and Expr' (with their schemas and sizes of each attribute) such that the cost of Expr is less than the cost of Expr' , but the (real) sum of the sizes of the intermediate results of Expr is greater than the sum of the sizes of the intermediate results of Expr' when evaluated over the relations.

Part b (7%)

Recall that in the lectures we omitted the definition of $\text{Vals}((\sigma_{\text{Cond}}(R)).A)$ with attribute A mentioned in condition Cond (we did this for brevity; however, such definition is necessary for the full definition of our cost function). Suggest a reasonable definition of $\text{Vals}((\sigma_{\text{Cond}}(R)).A)$ for Cond of the form $A = c$ with c a constant.

Question 4 of 8 (14%)

Part a (7%)

In the lectures, we have considered dense and sparse primary indexes, dense and sparse clustering indexes, but only dense secondary indexes. Discuss the possibilities for sparse secondary indexes; in particular, give a definition of such an index if you think it is possible (without violating the ideas of secondary indexes and sparse indexes), or briefly (1–2 sentences) explain the crucial obstacles if you think it is impossible.

Part b (7%)

Consider a relation with N tuples, where each tuple takes 330B. Consider also a (one-level) *clustering* sparse index of variant 2 (as defined in the lectures) for this relation, which has 10B per search key and 8B per pointer. Let both the table and the index be stored in a continuous unspanned manner in blocks of size 8K (i.e., 8000B). How many block accesses, as a function of N , do we need to retrieve all the tuples with the given search key using this index in the worst case? (Here, we may assume that the index relies on binary search to find the first tuple with the search key.)

Question 5 of 8 (14%)

Part a (7%)

Demonstrate the effect of cascading rollbacks by giving an example of a schedule with a cascade of at least three rollbacks.

Part b (7%)

Consider the following transactions:

```
 $T_1$  : read_item( $X$ ), write_item( $Y$ ), write_item( $Z$ )  
 $T_2$  : write_item( $Y$ ), read_item( $U$ ), read_item( $X$ ), write_item( $U$ )  
 $T_3$  : read_item( $X$ ), write_item( $V$ ), read_item( $Y$ ), write_item( $W$ )
```

Form a non-serialisable schedule of T_1 , T_2 , T_3 or explain why it is not possible.

Question 6 of 8 (14%)

Part a (7%)

Demonstrate the advantages of shared/exclusive locks over binary locks by giving an example of transactions T_1, T_2, T'_1, T'_2 , as well as schedules S and S' of T'_1, T'_2 and of T_1, T_2 , respectively, such that

- T_1, T_2 follow the two-phase locking protocol with binary locks,
- T'_1, T'_2 follow the two-phase locking protocol with shared/exclusive locks,
- the usual (i.e., non-locking and non-unlocking) operations and their order in T_1 and T'_1 are the same, and the same holds for T_2 and T'_2 ,
- the order of usual operations in S and S' are the same,
- S is in deadlock while S' is not.

Comment: in the question, "... S and S' of T'_1, T'_2 and of T_1, T_2 ..." should be read as "... S and S' of T_1, T_2 and of T'_1, T'_2 ...". This typo was found during the exam, the students were notified and given extra time for this.

Part b (7%)

Let us assume that transactions T_1, T_2, T_3, T_4 are evaluated aiming to follow the multiversion timestamp-ordering protocol. Let the evaluation start with the following sequence of operations:

```
read(Z) by  $T_1$ 
write(Y) by  $T_2$ 
read(X) by  $T_3$ 
read(X) by  $T_4$ 

write(Z) by  $T_1$ 
write(Z) by  $T_2$ 
write(Z) by  $T_4$ 

read(Z) by  $T_3$ 
```

Can this sequence satisfy the multiversion timestamp-ordering protocol? If no, explain which transaction should be rolled back and after which operation. If yes, then

1. state which transaction wrote the value of Z that is read by T_3 in the last operation of the sequence, and
2. continue the sequence by operations of the transactions so that your last operation cannot be evaluated and should be rolled back (under the rules of the protocol).

Question 7 of 8 (8%)

In one sentence, explain what is the main difference between the immediate update recovery protocol (with fuzzy checkpoints) and ARIES recovery protocol. In another sentence, explain why this difference is advantageous.

Question 8 of 8 (14%)

Part a (7%)

For brevity, let us assume in this question that RDF graphs use only URIs (i.e., mention no blank nodes and no literals). Let $f(G)$ be the (direct) translation of an RDF graph G to a relation over schema `Triple [Subject, Predicate, Object]`.

Let us say that a SPARQL query Q is equivalent to a SQL query Q' if for every RDF graph G (using only URIs) Q evaluated over G gives the same answer as Q' evaluated over $f(G)$. Note here, that Q may leave some variables undefined in its answers, and we assume that such missing values can be simulated by NULL in the answer to Q' (additionally, we assume that ? in front of a variable name is immaterial); for example, we consider the SPARQL answer

?person	?spouse
<http://example.org/#mary>	
<http://example.org/#john>	<http://example.org/#mary>

and SQL answer

Q':	
person	spouse
<http://example.org/#mary>	NULL
<http://example.org/#john>	<http://example.org/#mary>

as the same. Write an SQL query equivalent to the following SPARQL query:

```
SELECT DISTINCT ?student, ?course
      WHERE { ?student <http://example.org/#year> ?year .
      OPTIONAL { ?student <http://example.org/#enrolledIn> ?course } }
```

Part b (7%)

Recall that the distinctive feature of Cypher is the reach library for graph tasks (and algorithms), such as computing the length of the shortest path between two nodes. Is SQL capable to solve this task? More formally, let us consider the encoding of a directed graph G with weighted edges as a relation `Edge [source, target, length]` that has a tuple $\langle a, b, n \rangle$ if G has an edge from node a to node b of length n . Then, can we write an SQL query that, when evaluated over a database with such a table, returns all tuples $\langle c, d, k \rangle$ such that there is a (directed) path from c to d in the encoded graph, and the length of a shortest path from c to d is k ? Note that the query should work for graphs of arbitrary size and paths of arbitrary length. If your answer is no, explain the source of this limitation in SQL. If your answer is yes, explain, which SQL operators such a query will use (but it is not necessary to write this query in full).