

IN3020&4020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 1 of 12 (8%)

Consider the following relations:

Table1:			Table2:	
<i>A</i>	<i>B</i>	<i>C</i>	<i>C</i>	<i>D</i>
1	1	2	1	30
2	NULL	2	3	30
2	NULL	3	3	40
1	1	4		
1	1	3		

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

$$(\gamma_{A, \text{SUM}(B) \rightarrow \text{Sum}B, \text{COUNT_DISTINCT}(D) \rightarrow \text{Cnt}DstD}(\pi_{A,B,D}^b(\text{Table1} \bowtie \text{Table2}))).$$

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 2 of 12 (10%)

This question concerns relational algebra over set relations (i.e., not bags). For simplicity, you may assume that *NULL* behaves as a usual constant in selection conditions and joins (e.g., $\sigma_{A=NULL}(R)$ selects all tuples in R where attribute A is *NULL*).

Consider the following set RA operations: selection, projection, union, difference, renaming, left semi-join, right semi-join.

- (a) Is natural join expressible via these operations (over sets)? If your answer is yes give a justifying formula, otherwise explain why it is not possible.
- (b) Answer the same question as in (a) for the case when left semi-join and right semi-join are replaced by left outer join and right outer join, respectively.

Note: you can first introduce and then use your own notation for RA operators in the answer; for example, you can first say "I use $X|$ for right semi-join below" and then use $X|$ instead of $\bowtie$.

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 3 of 12 (8%)

Consider relations with schemas `Table1 [A, B]`, `Table2 [C, D]`, and `Table3 [E, F]`. Compile the following SQL query to a (logical) query plan that does not mention anti-join:

```
SELECT A FROM Table1 WHERE  
      B NOT IN (SELECT C FROM Table2) AND  
      EXISTS (SELECT E FROM Table3 WHERE E = A);
```

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

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 4 of 12 (8%)

Recall that, by our definition, the cost of a (logical) query plan does not include the estimate of the size of the final result (i.e., the result of the overall plan) as a summand. Explain, in no more than 5 sentences, what would happen with our query optimiser if we include this estimate into the cost. In particular, state whether the optimiser based on the modified cost will usually produce better or worse plans than the optimiser based on the non-modified cost and explain why.

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 5 of 12 (8%)

In the lectures, we have considered two variants of sparse clustering indexes and argued that in different situations it may be better to use one or another. Give an example when Variant 1 (as in the lectures) is more efficient than Variant 2. In particular, your example should include the following (where all sizes are in bytes):

- a relation with the size of each attribute,
- one of the attributes selected as the search key,
- a search key value,
- the size of a pointer,
- the size of a block, and
- the numbers of block reads that are needed to retrieve all the tuples with the search key value from the relation when using each of these two indexes so that the number of reads for Variant 1 is less than for Variant 2.

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 6 of 12 (10%)

Consider the following relations, where each *EID* and each *Age* is an integer taking 2B, and each *Name* is a string taking 10B:

SalesEmployees:			ProductionEmployees:		
<i>EID</i>	<i>Name</i>	<i>Age</i>	<i>EID</i>	<i>Name</i>	<i>Age</i>
1	Eirik	50	1	Hanna	40
2	Eirik	30	2	Eirik	30
3	Linda	30			

Consider also the following logical query plan:

$$\pi_{Name, Age}^b(\sigma_{Age < 25}(\text{SalesEmployees} \cup^b \text{ProductionEmployees}))$$

Compute the statistics of the relations relevant to our cost function (i.e., $Size(R)$, $Tups(R)$, $TupSize(R)$, and $Vals(R.A)$ for each relation R and its each attribute A). Then, using the cost function introduced in the lectures, compute the cost of this query plan based on the statistics.

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 7 of 12 (8%)

Recall that Cartesian product is a special case of natural join when the set of common (i.e., join) attributes is empty, and so we can use algorithms for natural join to evaluate Cartesian product.

Let us have 1000 buffers in the main memory and two relations, R and S , without common attributes. Let also R take 1000 blocks and S take 2000 blocks on the disk. Which of the natural join algorithms considered in the lectures should we use to compute the Cartesian product of R and S , and why? (Here, we assume that we should use the algorithm that is the most efficient according to our cost—that is, that minimises the number of disk reads and writes.)

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 8 of 12 (8%)

Give an example of two non-serial schedules that are not conflict equivalent, but view equivalent.

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 9 of 12 (8%)

Consider the following transactions:

```
 $T_1$  : write( $X$ ), read( $U$ ), read( $Y$ ), write( $U$ ), commit  
 $T_2$  : read( $Y$ ), write( $V$ ), read( $X$ ), write( $W$ ), commit  
 $T_3$  : read( $Y$ ), write( $X$ ), write( $Z$ ), commit
```

Form a schedule of (the prefixes of) these transactions that has to abort when following the timestamp-ordering protocol without the Thomas write rule, but can continue if the rule is used.

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 10 of 12 (8%)

Consider the following schedule:

$\mathbf{b}_1, \mathbf{b}_2, \mathbf{r}_1(Y), \mathbf{r}_2(X), \mathbf{w}_2(X), \mathbf{r}_1(Z), \mathbf{c}_2, \mathbf{b}_3, \mathbf{r}_3(Z), \mathbf{c}_1, \mathbf{b}_4, \mathbf{w}_4(U), \mathbf{c}_3$

Let the deferred-update recovery protocol be used, and let the last (usual, not fuzzy) checkpoint be evaluated between operations $\mathbf{c}_2$ and $\mathbf{b}_3$. Let also a system crash happen at the end of the schedule, and all log buffers be flushed by the crash moment. Give the log state at the crash moment and discuss what the system has to do to recover after the crash (you may write *NewValue* and *OldValue* instead of the non-given concrete item values in the log).

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 11 of 12 (8%)

Consider a relation

`XCResults` [*Number*, *Name*, *Club*, *Age*, *Result*],

which stores the results of a cross-country skiing competition (where the names of attributes are self-explanatory). Write SQL commands with DAC mechanism that give (read) access privileges to *UserA* to all the information in this relation except the results (i.e., the values of *Result* attribute) of all competitors under 14 years old.

IN3020 – Database Systems Exam, 2023

5 June 2023, 3:00–7:00pm (Inspira)

Question 12 of 12 (8%)

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 CQ-SPARQL be the fragment of SPARQL where all queries have the form

```
SELECT DISTINCT Vars WHERE { Pattern }
```

where *Vars* is a list of variables, and *Pattern* is a list of triples of URIs and variables such that all the variables in *Vars* are mentioned in *Pattern*. When evaluated over an RDF graph, each CQ-SPARQL query returns a set of mappings from its variables *Vars* to URIs (i.e., assignments of URIs to variables *Vars*). Let $g(M)$ be a direct translation of such a set of mappings M over *Vars* to a relation over schema with an attribute X for each variable $?X$ in *Vars*. Note that such translation does not need to introduce NULL.

Let us say that a SQL query Q_{sql} is *equivalent* to a CQ-SPARQL query $Q_{\text{cq-sparql}}$ if for every RDF graph G (using only URIs) Q_{sql} evaluated over $f(G)$ equals $g(M)$, where M is the result of $Q_{\text{cq-sparql}}$ evaluated over G .

Give an SQL query equivalent to the following CQ-SPARQL query:

```
SELECT DISTINCT ?x WHERE { ?x ?p ?y . ?y ?p ?z }
```