Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Querying Data on the Web

Date: Tuesday 17th January 2017

Time: 14:00 - 16:00

Please answer ALL Questions provided. They amount to a total of 50 marks.

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text

1. a) Briefly draw one contrast between database languages and general-purpose languages. (1 mark)

   b) Briefly draw one contrast regarding the time at which integrity constraint specification and integrity constraint checking take place. (1 mark)

   c) Briefly explain what is meant by *generalized projection*. (1 mark)

   d) Let *BBD* be a database schema on beers (made by some brewer) that are served in bars (at some address) for a certain price, and on drinkers (at some address) that frequent bars a certain number of times per week, where the beers that drinkers like are also recorded, as follows:

```
Beer(name,brewer)
Serves(bar[fk Bar.name],beer[fk Beer.name],price)
Bar(name,address)
Drinker(name,address)
Frequents(drinker[fk Drinker.name],bar[fk Bar.name],time_a_week)
Likes(drinker[fk Drinker.name],beer[fk Beer.name])
```

   Let the following be the current state of the `Bar` and `Serves` relations in *BBD*:

```
Bar                         Serves
=============               ================
name   address              bar   beer   price
----   -------              ---   ----   -----
D      W                    D     B      2.2
E      O                    E     B      2.5
S      W                    S     A      2.5
=============               ================
```

   Let *Q* be the following SQL query:

```
SELECT    b.name, b.address, MIN(s.price) AS best_value
FROM      Serves s, Bar b
WHERE     s.bar = b.name AND s.price < 3
GROUP BY  b.name, b.address
HAVING    best_value > 2.2;
```

   As described in this course unit, apply the seven-step procedure to characterize the answer to an SQL query to *Q* as it evaluates over the *BBD* schema and relation instances above. Your answer must state each of the seven steps and, at every step, you must show, in tabular form (as appropriate), the outcome of each specific step. (7 marks)

2. a) State the three main issues in query optimization that were cited in this course unit. (3 marks)

b) Assume the *BBD* schema and relation instances given in Question (1d) above. Apply one relational-algebraic equivalence studied in this course unit to each of the expressions below and write the expression that results from such application.

i) $\sigma_{\texttt{name}='\texttt{E}' \land \texttt{name}='\texttt{S}'}(\texttt{Bar})$ (1 mark)

ii) $\pi_{\texttt{name}}(\pi_{\texttt{address}}(\texttt{Bar}))$ (1 mark)

c) Again, assume the *BBD* schema and relation instances given in Question (1d) above. State whether each rewriting below is valid, or invalid, and state one relational-algebraic equivalence studied in this course unit that validates, or invalidates, that rewriting:

i) $\pi_{\texttt{name}}(\sigma_{\texttt{address}='\texttt{O}'}(\texttt{Bar})) \Leftrightarrow \sigma_{\texttt{address}='\texttt{O}'}(\pi_{\texttt{name}}(\texttt{Bar}))$ (1 mark)

ii) $\sigma_{\texttt{name}='\texttt{E}' \land \texttt{price}>2}(\texttt{Bar} \bowtie \texttt{Serves}) \Leftrightarrow \sigma_{\texttt{name}='\texttt{E}'}(\texttt{Bar}) \bowtie \sigma_{\texttt{price}>2}(\texttt{Serves})$ (1 mark)

d) Once more, assume the *BBD* schema and relation instances given in Question (1d) above. Let E be the following relational-algebraic expression:

$$\pi_{\texttt{name},\texttt{price}}(\sigma_{\texttt{name}=\texttt{bar}}(\texttt{Bar} \times \texttt{Serves}))$$

Apply a sequence of two relational-algebraic equivalences studied in this course unit that result in a final, rewritten relational-algebraic expression that is likely to be more efficient to evaluate according to the heuristics studied in this course unit. You must state the equivalence and show the outcome of applying it in each of the two steps. (3 marks)

3. a) State <u>two</u> of the goals of the working group that set out to design XQuery. (2 marks)

b) XQuery Core has a static and a dynamic semantics. Briefly draw a contrast between the two kinds of semantics. (1 mark)

c) Translate the following XPath selection expression into an XQuery FLWOR expression:

```
$root/A/B[@C > 1]/D
```

(3 marks)

d) Consider the following FLWOR expression F:

```
for $x in (<composer>Bach</composer>,
           <composer>Mozart</composer>)
return <creator>{data($x)}</creator>
```

Now, consider the following trace of the evaluation of F using the `sequence`, `right unit`, `let` and `data` equivalence laws, where `E1`, `E2`, `E3` and `E4` act as place-holders. Using your knowledge of those equivalence laws, write down the XQuery expressions that instantiate `E1`, `E2`, `E3` and `E4`.

```
  for $x in (<composer>Bach</composer>,
             <composer>Mozart</composer>)
  return <creator>{data($x)}</creator>
= (sequence)
  E1
= (right unit)
  E2
= (let)
  E3
= (data)
  E4
```

(4 marks)

4. a) Briefly explain the role of `PREFIX` declarations in SPARQL. (1 mark)

   b) Briefly explain the role of the `GRAPH` keyword in SPARQL. (1 mark)

   c) You have learned that, in essence, SQL is based on a tuple calculus. Briefly explain whether this is also the case for SPARQL. (1 mark)

   d) Assume the following RDF graph:

```
G = { (R1, a, v1),
      (R2, a, v2),
      (R3, a, v3),
      (R1, b, v4),
      (R3, b, v5),
      (R3, c, v6) }
```

   Now, consider the following graph pattern expression:

$$[\![(?\mathtt{X}, \mathtt{a}, ?\mathtt{I})\ \mathtt{OPT}\ (?\mathtt{X}, \mathtt{b}, ?\mathtt{J})]\!]_{\mathtt{G}}$$

   i) Write the SA-expression that corresponds to the graph pattern expression above. (1 mark)

   ii) Write the tabular version of the result of evaluating the left-hand operand in the SA-expression over G. (2 marks)

   iii) Write the tabular version of the result of evaluating the right-hand operand in the SA-expression over G. (2 marks)

   iv) Write the tabular version of the final result of evaluating the SA-expression over G. (2 marks)

5. a) State two of the four main types of irregularly-structured stores (e.g., NOSQL ones). (1 mark)

   b) In the course unit, some observations were made that suggest that SQL and NOSQL systems are complementary. State two of them. (1 mark)

   c) State two of the ten rules for scalable on-line transaction processing proposed by Stonebraker and Cattell. (1 mark)

   d) Briefly explain in what sense `map` and `reduce` are said to be second-order functions then write one example call for either one of them. (1 mark)

   e) Assume the *BBD* schema and relation instances given in Question (1d) above. Assume that the `serves` relation is partitioned as follows: partition P1 contains beers with prices at most 2.00, partition P2 contains beers with prices above 2.00 and at most 2.50, and partition P3 contains beers with prices above 2.50.

   Briefly explain how the following SQL query over the distributed database that results from the partitioning described above could be evaluated in a map-reduce engine. Your answer must have three parts: one that states what relational-algebraic expression the mapper implements, one that states what relational-algebraic expression the reducer implements, and, finally, one that states the set-theoretic expression that represents the assembly of the final result.

```sql
SELECT   bar, COUNT(*)
FROM     serves
WHERE    price < 3.0
GROUP BY bar;
```

(6 marks)