CS561 – 2005 Elke A. Rundensteiner

Examination

Date: 20 April 2005
Allocated Time: 90 minutes (in class exam)
Maximum Points: 100

STUDENT NAME: _____

STUDENT NAME: _____

Please place your signature below to confirm that you have completed this examination on your own without help from others, and using only material explicitly permitted for this examination.

SIGNATURE: _____

General Instructions:

- If a question seems vague, make reasonable assumptions, state those and answer the question under those assumptions. *Make sure to state any assumptions you make! Make sure to show your work!*

PROBLEMS:	MAX SCORE	YOUR SCORE
problem 1 (XML/RD/ORDB:	30	
problem 2 (constraints+queries):	20	
problem 3 (algebra operators):	25	
problem 4 (query optimization):	25	
total:	100	

Problem 1: XML, Relational and Object-Relational Databases [30pts]

For the DTD, XML, and XQUERY given below, answer the questions listed next.

```
< !ELEMENT myfriends (person*)>
< !ELEMENT person (id, name?, cell-phone*, children?)>
< !ELEMENT children (child*)>
< !ELEMENT child (name,toys*)>
< !ELEMENT name ( #PCDATA)>
< !ELEMENT toys ( #PCDATA)>
< !ELEMENT id ( #PCDATA)> ...  ]

< !ELEMENT employees (emp*)>
< !ELEMENT emp (id, work-phone, (contact|address)>
< !ELEMENT address (city,zip,street)>
< !ELEMENT id ( #PCDATA)>]
< !ELEMENT contact ( #PCDATA)> ... ]
```

The xml file friends.xml:

```
<myfriends>
  <person>
     <id>  1 </id>
     <name> ‘‘jack’’  </name>
     <cellphone> 2222  </cellphone>
  </person>
  <person>
     <id>  2 </id>
     <cellphone> 3333 </cellphone>
     <children> <child><name> c1 </name> </child>

     <children> <child><name> c2 </name> <toys> t1 </toys> </child>
     <child> <name> c2 </name> </child> ... </children>
  </person>
</myfriends>
```

The xml file employees.xml:

```
<employees>
  <emp>
     <id>  1 </id>
     <workphone> 9999 </workphone>
     <contact> ‘‘me’’ </contact>
  </emp>
```

```
  <emp>
     <id>  2 </id>
     <workphone> 8888 </workphone>
     <address> <city> c </city> <zip> z </zip> <street> s </street> </address>
  </emp>
</employees>
```

The XQUERY expression:

```
<contact-info>
FOR $outer in (friends.xml)//person,
LET  $child :=  $outer/children
WHERE  ($outer/cellphone >  2000 )
RETURN
     $outer/id
     FOR $inner IN  (employees.xml)/employees/emp[id=$outer/id]
     RETURN
        {<contact>
             $outer/cellphone
             $child/child
             $inner/workphone
             $inner/address/city
         </contact>
        }
</contact-info>
```

1. List the **XML output** that the XQUERY expression would generate when applied to the given XML input documents.

2. Design a **relational schema** to store the two given XML data files. Explain your design choices. Briefly explain your algorithm for mapping XML data to relational data model.

3. Design an **object-relational schema** to store the two given XML data files. Utilize and explain any special OO features that could be used here for this design. Explain your algorithm for mapping XML data to OR data model.

4. List the **SQL query** that you would generate to execute the given XQUERY expression on your relational database. State what final computations would remain to be done by the XQUERY processor beyond executing your SQL statement, if any.

5. Briefly explain how the XPATH and the XQUERY language relate. In particular, list three key features of the XQUERY language that cannot be supported the XPATH language, or vice versa.

Problem 2: Constraints and Query Processing [20pts]

1. Can static constraints such as foreign keys be used to aid stream query processing? Explain.

2. If the X-Join algorithm had more constraints or meta knowledge available about the data streams as indicated below, then explain which aspects (stages, algorithms, storage, space, time) of the X-Join algorithm would you modify to exploit this knowledge, and how?

   (a) Assume we knew that the join was a 1-to-many join and the many side of the join would arrive clustered on values of the join attribute. For example, all tuples with attribute STREAM2.COMPANY="IBM" would arrive together without any other tuples interleaved, then all tuples with STREAM2.COMPANY="MS", etc. After the "IBM" stream2 cluster is completed as indicated by a punctuation, then no tuple with STREAM2.COMPANY="IBM" will appear in the future on that stream.

   (b) Assume that we knew that both input streams would be arriving in a strict order sorted on the respective join attribute, STREAM1.DATE and STREAM2.DATE. But the relative arrival of the two streams is not synchronized, i.e., one stream may arrive faster than the other.

Problem 3: Algebra Query Operators [25pts]

1. Consider the following properties of query operators:
   (1) stateless and non-blocking
   (2) small bounded state and non-blocking
   (3) unbounded-state and non-blocking
   (4) stateless and blocking
   (5) small bounded state and blocking
   (6) unbounded-state and blocking

   Give an example of an (algebra) query operator for each of the above properties, if one exists. If not, indicate so.

2. Which of the above 6 operator types can (cannot) be plugged into a continuous query environment such as CAPE or EDDIES. Explain why or why not.

Problem 4: Query Optimization [25pts]

1. Given a query with two filter query operators OP1 and OP2 that operate on the same input. Assume you could determine several statistics about the two operators, including their *selectivity* (number of tuples outputted versus number of tuples taken in) and *time* (processing cost per tuple). Explain under which of the conditions below would you adapt the query plan so that the operator OP1 would be placed below OP2, or vice versa.

   (a) selectivity(OP1)=selectivity(OP2) and time(OP1)=time(OP2).

   (b) selectivity(OP1)=selectivity(OP2) and time(OP1)>time(OP2).

   (c) selectivity(OP1)<selectivity(OP2) and time(OP1)=time(OP2).

   (d) selectivity(OP1)<selectivity(OP2) and time(OP1)<time(OP2).

   (e) selectivity(OP1)<selectivity(OP2) and time(OP1)>time(OP2).

2. Describe the main steps of an on-line migration algorithm that would correctly migrate the query plan (OP1 $->$ OPS2) into the plan (OP2 $->$ OPS1) at run time, if that plan is part of a larger plan structure.

3. Discuss what, if anything, may need to be changed about your above proposed algorithm if OP1 and OP2 were join operators?

.

Done with Exam? Let's Go Home !