CS585

Database Systems

Spring 2013

Exam II

Name: _____

Student ID: _____

	Maximum	Received
Problem 1	10	
Problem 2	10	
Problem 3	10	
Problem 4	10	
Problem 5	10	
Problem 6	20	
Problem 7	10	
Problem 8	20	
Total	100	

2hr exam. One 8.5X11 cheat sheet allowed

1) 10 pts
Indicate whether each of the following statements is true or false (T/F):

_____The advantage of the prefix sum matrix is that independent of how many terms appear in a given range, one would need to fetch a constant number of elements from the prefix sum matrix to evaluate a range query.

_____For relations A, B, and C, (A X B) X C = A X (B X C)

_____For relations A, B, and C, (A X B) X C has same performance as A X (B X C)

_____Bond Energy method is a matrix sequencing method

_____In dynamic query optimization, we use the most up-to-date statistics on database relations including relation sizes, index ranges, etc. In other words, the statistics used correspond to the relations at that instance on the database.

_____ In static query optimization, we use statistics pulled from catalog tables

_____Consider the following definition: <!ELEMENT books (title|authors)>. This means that we are allowed to have a title or authors or both.

_____In an OLAP system, a Prefix Sum matrix can be created to optimize the performance of range queries to find the maximum value in a range

_____ In an OLAP system, a Prefix Sum matrix can be created to optimize the performance of range queries to find the average value in a range

_____XML tags are case sensitive.

2) 10 pts
Consider the following XML document:

```
<?xml version="1.0"?>
<!DOCTYPE PARTS SYSTEM "parts.dtd">
<?xml-stylesheet type="text/css" href="xmlpartsstyle.css"?>

   <TITLE>Computer Parts<TITLE>
   <PART>
      <ITEM>Motherboard</ITEM>
      <MANUFACTURER>ASUS</MANUFACTURER>
      <MODEL>P3B-F</MODEL>
      <COST> 123.00</COST>
   </PART>
   <PART>
      <ITEM>Video Card</ITEM>
      <MAKER>ATI</MAKER>
      <MODEL>All-in-Wonder Pro</MODEL>
      <COST> 160.00</COST>
   </PART>
```

a- Is the XML file well-formed? If yes, why, and if no, modify the xml file so that it is well-formed. (5 points)

b) Is the well-formed XML file valid with respect to the following DTD file? If yes, why, and if no, modify the DTD file so that it is valid. (10 points)

Parts.dtd

```
<!ELEMENT PARTS (TITLE?, PART*)>
<!ELEMENT TITLE (#PCDATA)>
<!ELEMENT PART (ITEM, MANUFACTURER, MODEL, COST)+>
<!ATTLIST PART
   type (computer|auto|airplane) #IMPLIED>
<!ELEMENT ITEM (#PCDATA)>
<!ELEMENT MANUFACTURER (#PCDATA)>
<!ELEMENT MODEL (#PCDATA)>
<!ELEMENT COST (#PCDATA)>
```

3) 10 pts

You wish to execute an XQuery on the following [example.xml] to obtain [Output Result]. Select the correct XQuery (A thru D) to obtain [Output Result]. Assume that the code attribute value of the log element noted in [example.xml] matches the code attribute value of the list element. Select a single answer.

[example.xml]

```
<logList>
 <list code="w001" message="Warning1"/>
 <list code="w002" message="Warning2"/>
 <list code="e001" message="Error1"/>
 <list code="e002" message="Error2"/>
 <day date="2007-12-01">
  <log time="10:00:00" code="w001"/>
  <log time="14:00:00" code="e001"/>
 </day>
 <day date="2007-12-02">
  <log time="13:00:00" code="e002"/>
  <log time="15:00:00" code="e001"/>
 </day>
</logList>
```

[Output Result]

```
<result>
 <log date="2007-12-01" time="10:00:00" message="Warning1"/>
 <log date="2007-12-01" time="14:00:00" message="Error1"/>
 <log date="2007-12-02" time="13:00:00" message="Error2"/>
 <log date="2007-12-02" time="15:00:00" message="Error1"/>
</result>
```

A.

```
<result>{
 let $doc := fn:doc("example.xml")
 for $log in $doc//log
 return
   <log>{
    $log/../@date,
    $log/@time,
    $doc//list[@code eq $log/@code]/@message
   }</log>
}</result>
```

B.

```
<result>{
 let $doc := fn:doc("example.xml")
 for $log in $doc//log
 return
   <log>{
    ../@date,
    @time,
    ../../list[@code = $log/@code]/@message
   }</log>
}</result>
```

C.

```
<result>{
 let $doc := fn:doc("example.xml")
 for $day in $doc//day
 return
   <log>{
    $day/@date,
    $day/log/@time,
    $doc//list[@code eq $day/log/@code]/@message
   }</log>
}</result>
```

D.

```
<result>{
 let $doc := fn:doc("example.xml")
 for $day in $doc//day
 return
   <log>{
   @date,
    log/@time,
    ../list[@code = $day/log/@code]/@message
   }</log>
}</result>
```

4) 10 pts

Fill out the 5 blank spaces in the XSD file

XML FILE:

```
<?xml version="1.0"?>
   <x:books xmlns:x="urn:books">
      <book id="bk001">
         <author>Writer</author>
         <title>The First Book</title>
         <genre>Fiction</genre>
         <price>44.95</price>
         <pub_date>2000-10-01</pub_date>
         <review>An amazing story of nothing.</review>
      </book>
      <book id="bk002">
         <author>Poet</author>
         <title>The Poet's First Poem</title>
         <genre>Poem</genre>
         <price>24.95</price>
         <review>Least poetic poems.</review>
      </book>
   </x:books>
```

XSD FILE:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="urn:books"
            xmlns:bks="urn:books">
  <xsd:element name="books" type="bks:BooksForm"/>
  <xsd:complexType name="BooksForm">
    <xsd:sequence>
      <xsd:space1)_____ name="book"
                        type="bks:BookForm"
            space2)_____="0"
            space3)_____="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="BookForm">
    <xsd:sequence>
      <xsd:element name="author"   type="xsd:string"/>
      <xsd:element name="title"    type="xsd:string"/>
      <xsd:element name="genre"    type="xsd:string"/>
      <xsd:element name="price"    type="xsd:float" />
      <xsd:element name="pub_date"type=spce4)_____/>
      <xsd:element name="review"  type=spce5)_____/>
    </xsd:sequence>
    <xsd:attribute name="id"   type="xsd:string"/>
  </xsd:complexType>
</xsd:schema>
```

5) 10 pts
Consider the following relation and assume that the domain of age is {5, 10, 15, 20, 25, 30, 35, 40, 45} and the domain of salary is {50, 100, 150, 200, 250, 300, 350, 400, 450}

Age	Salary
10	$50
5	$200
20	$300
35	$150
30	$250
15	$100
25	$350

a) If "Age" is dimension attribute and "Salary" is the measure attribute, draw the corresponding one dimensional cube for this relation. (5 points)

b) Draw the corresponding prefix sum cube.

c) Draw the corresponding space-efficient relative prefix sum (SRPS) cube assuming a block size of 4

6) 20 pts
Consider the following three relations with their keys underlined:

Product (<u>PID</u>, Name, Description)
Store (<u>SID</u>, Name, Address)
Location (<u>PID</u>, <u>SID</u>, Quantity, Startdate)

Furthermore, assume the following two queries:

• Query 1 (q1):
select Product.PID, Store.Name, Location.Quantity
from Product, Location, Store
where Product.PID = Location.PID **and** Location.SID = Store.SID **and** Location.Quantity>1000

Suppose q1 is executed by an application that is located at sites S1 and S2, with frequencies 1 and 2, respectively.

• Query 2 (q2):
select Location.PID, Location.Startdate
from Location
where Location.Startdate > "Nov 6, 2011"

Suppose q2 is executed by an application that is located at sites S2 and S3, with frequencies 2 and 1, respectively.

a) Construct the usage matrix UA for the attributes of the relation **Location**. (Reminder: element eij of the UA matrix is *use(qi, Aj),* the usage value for the attribute Aj by the query qi).

b) Construct the affinity matrix AA containing all attributes of the relation **Location**.

c) Using any clustering technique you know partition the attributes in **Location** into two vertical fragments.

7) 10 pts

Briefly answer the following questions

a) What is the advantage of hybrid query optimization over static and dynamic query optimization

b) Give two reasons why we should implement a data warehouse

c) Describe how an RDBMS can help speed up the recovery of information stored in XML documents

d) Describe two scenarios where XML would be a more suitable method of data storage as compare to the relational format.

8) 20 pts
Given the following relations

Students (SID, Name, DOB, field_of_study)
Athletes (SID, year, sport, rating)

And the following assignment of fragments to sites:

Students1 = fragment of Students with SID $\leq$ 1M	at site 1
Students2 = fragment of Students with SID > 1M	at site 2
Athletes1 = fragment of Athletes with SID $\leq$ 2M	at site 3
Athletes2 = fragment of Athletes with SID > 2M	at site 4

And following assumptions
- Size of the Students table is 40,000
- Size of the Athlete's table is 50,000
- Students table holds list of current USC Students
- Athletes table holds list of all USC Athletes from 1960 on
- 2% of athletes are current students at USC
- 3% of current USC students are athletes
- SID's range between 1 to 3M and have a uniform distribution
- size of a Students tuple = size of an Athletes tuple
- size of the field SID is $1/8^{th}$ of the size of the a Students tuple
- We only consider the communication cost of the query

a) If we were to form a natural join of EMP and ASG and have the results of the join at site 4, describe the best join strategy without use of semi-join.

b) Can the performance in a) be improved using semi-join? If no, why? If yes, how?

Additional space

Additional space