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):

T_____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.

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

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

T_____Bond Energy method is a matrix sequencing method

F_____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.

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

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

F_____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

T_____ 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

T_____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>
```

Correct Answer: A

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)

5	10	15	20	25	30	35	40	45
200	50	100	300	350	250	150	-	-

b) Draw the corresponding prefix sum cube. (3points)

5	10	15	20	25	30	35	40	45
200	250	350	650	1000	1250	1400	1400	1400

c) Draw the corresponding space-efficient relative prefix sum (SRPS) cube assuming a block size of 4 (2 points)

5	10	15	20	25	30	35	40	45
-	-	-	650	-	-	-	1400	1400

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

Product (PID, Name, Description)
Store (SID, Name, Address)
Location (PID, SID, 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). (10 points)

	PID	SID	Quantity	Startdate
Q1	1	1	1	0
Q2	1	0	0	1

b) Construct the affinity matrix AA containing all attributes of the relation **Location**. (5 points)

Frequency matrix

	S1	S2	S3
Q1	1	2	0
Q2	0	2	1

Affinity matrix

	PID	SID	Quantity	Startdate
PID	6	3	3	3
SID	3	3	3	0
Quantity	3	3	3	0
Startdate	3	0	0	3

c) Using any clustering technique you know partition the attributes in **Location** into two vertical fragments. (5 points)

Option 1: Using BEA algorithm

It is enough to do the first 3 columns of the CA matrix using the BEA algorithm, and guess where the 4th column goes because otherwise there will be too much computation.

CA matrix

	Quantity	PID	SID	Startdate
Quantity	3	3	3	0
PID	3	6	3	3
SID	3	3	3	0
Startdate	0	3	0	3

Visually Partition into two partitions

Partition 1. Quantity, PID and SID

Partition 2. Startdate

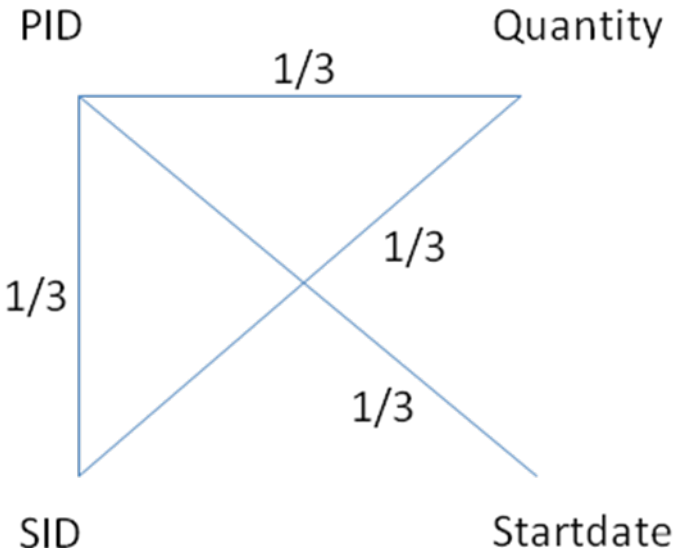
Since **PID and SID** are part of the primary key and they have to be part of each vertical fragment, then the fragments will be

Fragment 1: Quantity, **PID and SID**

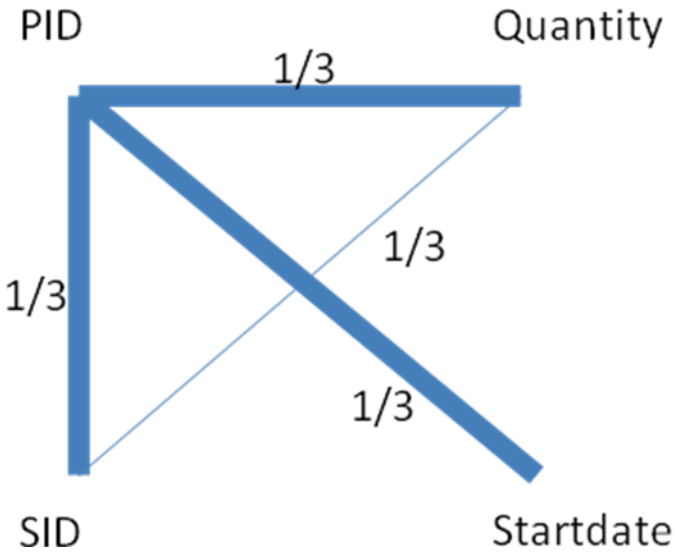
Fragment 2: Startdate, **PID and SID**

Option 2: Using the MST

Create undirected graph



Create MST (note, in this particular problem, the MST is not unique, so any spanning tree that uses a subset of the graph's edges is a good MST)



Split the graph into two groups of nodes by dropping the highest cost edge:
Again, since all edges on the MST have the same cost, we could come up with any of the following splits:

Group 1	**Group 2**
SID	PID, startedate, quantity
Quantity	PID, SID, Startdate
Startedate	PID, SID, Quantity

Again, similar to the BEA method, since **PID and SID** are part of the primary key and they have to be part of each vertical fragment, then the fragments will be

Fragment 1: Quantity, **PID and SID**
Fragment 2: Startdate, **PID and SID**

7) 10 pts (2.5 points each)
Briefly answer the following questions

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

Better than dynamic query optimization, since query evaluation can be done offline and used at run time.

Better than static query optimization, since static optimization results may become suboptimal over time because the relation sizes and other related statistics change

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

To keep historical information of events/transactions and to find trends and patterns in them

To bring information from disparate systems into one place where we can get a uniform view of events/transactions

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

We can store XML documents in relational tables and use **index** structures to index the rows of the table based on the element or attribute values in the XML document. For example, each customer order can be stored in one row of a relational table as an XML document. Such an order can be found much faster (using index structures) than if we had all orders stored in one large XML document.

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

- If data is of hierarchic nature
- If data represents a business object
- If the schema is volatile
- If data is to be shared across multiple systems with different OS, database, etc.

8) 20 pts
Given the following relations

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

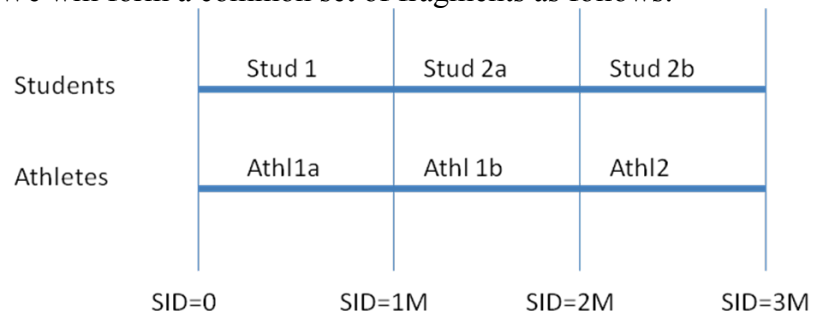
And the following assignment of fragments to sites:

Students1 = fragment of Students with SID ≤ 1M	at site 1
Students2 = fragment of Students with SID > 1M	at site 2
Athletes1 = fragment of Athletes with SID ≤ 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 0 to 3M and have a uniform distribution
- size of a Students tuple = size of an Athletes tuple
- size of the field SID is 1/8th 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. (10 points)

We will form a common set of fragments as follows:



Based on assumption that the SID's re uniformly distributed, we can say that each fragment sizes are as follows:
Site 1 - Stud 1: 1/3 * 40K tuples
Site 2 - Stud 2a: 1/3 * 40K tuples

Site 2 - Stud 2b: 1/3 * 40K tuples
Site 3 – Athl 1a: 1/3 * 50K tuples
Site 3 – Athl 1b: 1/3 * 50K tuples
Site 4 – Athl 2: 1/3 * 50K tuples

Need to do 3 joins between corresponding fragments, send results to site 4 (if the result of the join is not already there) and then do a merge of these results at site 4:

To join Stud 1 and Athl 1a – Since Stud 1 is smaller, then
- send Stud 1 to site 3 to join with Athl 1a; send results of join to site 4

To join Stud 2a and Athl 1b – Since Stud 2a is smaller, then
- send Stud 2a to site 3 to join with Athl 1b; send results of join to site 4

To join Stud 2b and Athl 2 – Since results are needed at site 4
- send Stud 2b to site 4 to join with Athl 2

b) Can the performance in a) be improved using semi-join? If no, why? If yes, how? (10 points)

Since only a small percentage of each of the two relations participate in the join, we can replace all three joins in part **a,** with semi-joins. For example, to join Stud 1b and Athl 2:
- Send SID's from Stud 2b to site 4
- At site 4, join these SID's with Athl 2, and send reduction to site 2
- At site 2, do a join of reduction with Stud 2b and send results of join back to site 4

For the other two joins we do the same process before sending the results to site 4 to be merged with other parts of the results.