CS585

Database Systems

Spring 2009

Final Exam

Name: _____

Student ID: _____

	Maximum	Received
Problem 1	20	
Problem 2	15	
Problem 3	10	
Problem 4	20	
Problem 5	15	
Problem 6	20	
Total	100	

Problem 1 (20 points)

Briefly answer the following questions:

a) 4pts
Define a view and a table in DBMS. And specify the similarity and the difference between a view and a table.

- view - a stored query accessible as a virtual table composed of the result set of a query. A view does not form part of the physical schema
- table – a part of physical schema

b) 4pts
Explain the difference between data mining and decision support using OLAP.

OLAP -> higher level views of business data, info representation
DM -> meaningful patterns and rules

c) 4 pts
Name 2 properties that are not common between XML elements and XML attributes.

- Element names are not unique – can have 2 price elements for a product element
- Elements can be nested

d) 4 pts
Can you write an automated tool to convert a DTD document into an XML schema without losing information? Can you do this in the reverse order? Explain your answers.

- can do DTD to XML schema easily. The data types in DTD are a small subset of the data types in XML schema. No information is lost.
- Cannot do this in reverse. A lot of information on data types will be lost

e) 4 pts
Explain a declaration and a definition in XML schema.

- You *declare* elements and attributes. Schema components that are *declared* are those that have a representation in an XML instance document.
- You *define* components that are used just within the schema document(s). Schema components that are *defined* are those that have no representation in an XML instance document.
- **Declarations:**
- - element declarations
- - attribute declarations
- **Definitions**:
- - type (simple, complex) definitions
- - attribute group definitions
- - model group definitions

Problem 2 (15 points)

Given the XML documents “bstore1.com/bib.xml”, “bstore2.com/bib.xml”, “bstore1.com/reviews.xml”, and “bstore2.com/reviews.xml” with the following sample content:

```
<bib>
    <book year="1994">
        <title>TCP/IP Illustrated</title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
    </book>

    <book year="1992">
        <title>Advanced Programming in the Unix environment</title>
        <author><last>Stevens</last><first>W.</first></author>
        <publisher>Addison-Wesley</publisher>
        <price>65.95</price>
    </book>

    <book year="2000">
        <title>Data on the Web</title>
        <author><last>Abiteboul</last><first>Serge</first></author>
        <author><last>Buneman</last><first>Peter</first></author>
        <author><last>Suciu</last><first>Dan</first></author>
        <publisher>Morgan Kaufmann Publishers</publisher>
        <price>39.95</price>
    </book>

    <book year="1999">
        <title>The Economics of Technology and Content for Digital TV</title>
        <editor>
               <last>Gerbarg</last><first>Darcy</first>
                <affiliation>CITI</affiliation>
        </editor>
            <publisher>Kluwer Academic Publishers</publisher>
        <price>129.95</price>
    </book>
</bib>
```

```
<reviews>
    <entry>
        <title>Data on the Web</title>
        <price>34.95</price>
        <review>
               A very good discussion of semi-structured database
               systems and XML.
        </review>
    </entry>
    <entry>
        <title>Advanced Programming in the Unix environment</title>
        <price>65.95</price>
        <review>
               A clear and detailed discussion of UNIX programming.
        </review>
    </entry>
    <entry>
        <title>TCP/IP Illustrated</title>
        <price>65.95</price>
        <review>
               One of the best books on TCP/IP.
        </review>
    </entry>
</reviews>
```

a) 7 pts

Write an Xquery to find the list of all books at bookstore 1, published by Addison-Wesley after 1996, including their year and title.

```
for $b in doc("http://bstore1.com/bib.xml")/bib/book
  where $b/publisher = "Addison-Wesley" and $b/@year > 1996
  return
    <book year="{ $b/@year }">
     { $b/title }
    </book>
```

b) 8 pts

For each book found at both stores bstore1 and bstore2, list the title of the book and its price from each source.

```
for $b in doc("http://bstore1.example.com/bib.xml")//book,
        $a in doc("http://bstore2.example.com/bib.xml")//book
    where $b/title = $a/title
    return
        <book-with-prices>
            { $b/title }
            <price-bstore2>{ $a/price/text() }</price-bstore2>
            <price-bstore1>{ $b/price/text() }</price-bstore1>
        </book-with-prices>
```

Problem 3: (10 points)

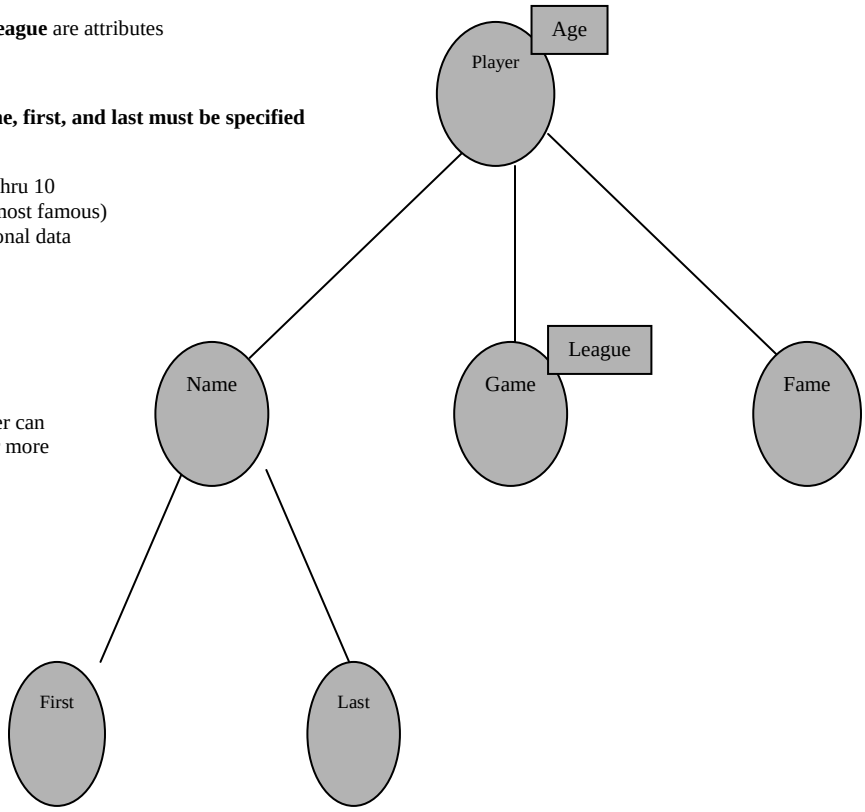
Write an XML schema that captures the element hierarchy and rules shown below:

Age and **League** are attributes

Name:
Rule: Name, first, and last must be specified

Fame:
Integers 1 thru 10
(10 being most famous)
Rule: Optional data

Game:
Football
Soccer
Basketball
Cricket
Rule: Player can play one or more games



Additional space

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">

   <xs:element name="player">
          <xs:complexType>
                 <xs:sequence>
                        <xs:element ref="name" minOccurs="1"/>
                        <xs:element ref="game" minOccurs="1"
maxOccurs="unbounded"/>
                        <xs:element ref="fame" minOccurs="0"/>
                 </xs:sequence>
                 <xs:attribute name="age" use="optional"/>
          </xs:complexType>
   </xs:element>

<xs:simpleType name="gameType">
   <xs:restriction base="xs:string">
          <xs:enumeration value="Football"/>
          <xs:enumeration value="Soccer"/>
          <xs:enumeration value="Basketball"/>
          <xs:enumeration value="Cricket"/>
   </xs:restriction>
</xs:simpleType>

   <xs:element name="name">
          <xs:complexType>
                 <xs:sequence>
                        <xs:element ref="first" minOccurs="1"/>
                        <xs:element ref="last" minOccurs="1"/>
                 </xs:sequence>
          </xs:complexType>
   </xs:element>

   <xs:element name="fame">
          <xs:simpleType>
                 <xs:restriction base="xs:int">
                        <xs:minInclusive value="1"/>
                        <xs:maxInclusive value="10"/>
                 </xs:restriction>
          </xs:simpleType>
   </xs:element>

   <xs:element name="first" type="xs:string"/>
   <xs:element name="last" type="xs:string"/>

   <xs:element name="game">
          <xs:complexType>
                 <xs:simpleContent>
                        <xs:extension base="gameType">
                               <xs:attribute name="league" type="xs:string"/>
                        </xs:extension>
                 </xs:simpleContent>
          </xs:complexType>
   </xs:element>
</xs:schema>
```

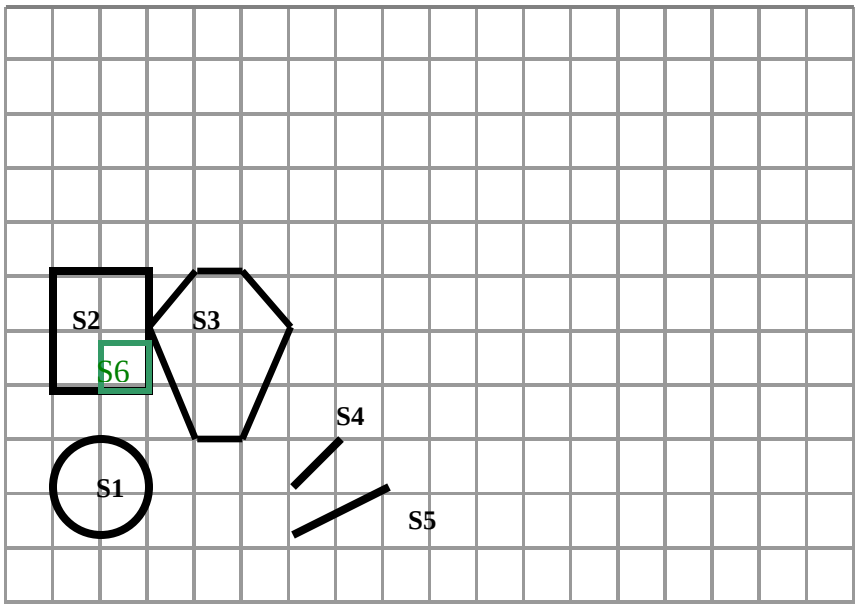
Problem 4: (20 points)

Assume the following table for the position of some of shapes in a painting which is stored in a database with spatial support.

ID	Shape	Spatial Data
S1	Circle	center: (2,2), radius:1
S2	Rectangle	(1,4), (1,6), (2,6), (2,4)
S3	Hexagon	(3,5), (4,6), (5,6), (6,5), (5,3), (4,3)
S4	Line	(6,2), (7,3)
S5	Line	(6,1), (8,2)

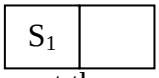
Assume that the shapes are inserted in to the table in the ascending order of ID (i.e., S1, S2, S3, S4, S5). Also assume that (m,M)=(1,2).

(a) Draw the R-Tree index generated for the above table after each insertion. In other words, you should draw five R-Trees. Show your computation for the intermediate steps (e.g., how you decide to split nodes). You can use the following chart to draw the shapes. (15 points)

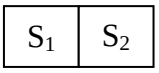


Additional space

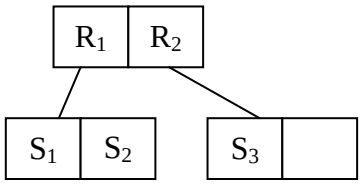
The resulting index will be:
1) Insert the first one into the empty tree:



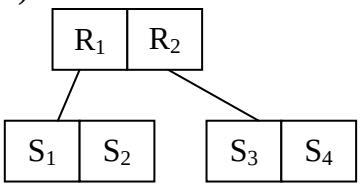
2) Insert the second one into the previous one (still have room):



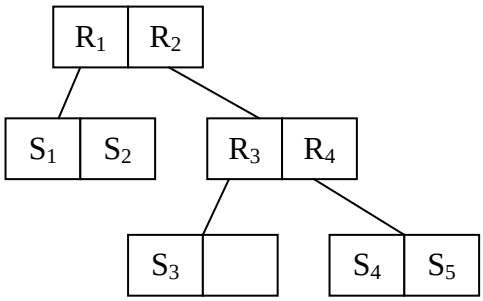
3) Insert the third one into the previous one (don't have room, need to split the node)
(either this combination or (S_2,S_3) , S_1 is fine. Both have the same amount of coverage)



4) Insert the 4th one into the previous one (still have room):

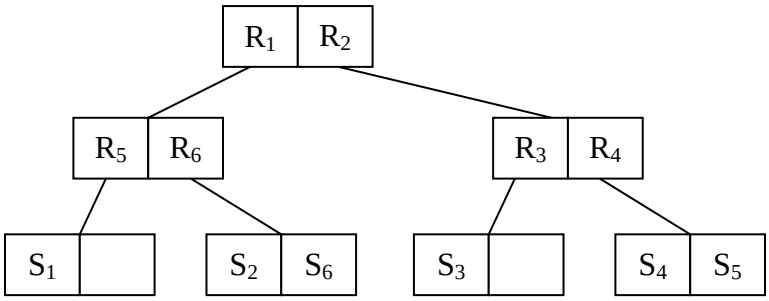


5) Insert the 5th one into the previous one (don't have room, need to split the node)



(b) Suppose the following new shape is inserted. Update the R-Tree to include this new shape. (3 points)

ID	Shape	Spatial Data
S6	Square	(2,4), (2,5), (3,5), (3,4)



(c) In order to search for an object in R-Tree, you might have to go through several rectangles or the entire database in the worst case. Explain why? How does R+ Tree address this problem? (2 points)

1) In R-Tree, the bounding rectangles could overlap each other, and an object is only associated with one bounding rectangle.
2) In R+-Tree, the space is decomposed into disjoint cells. Moreover, if a node overlaps with several rectangles, it would appear in all overlapping rectangles.

Problem 5: (15 points)

Consider the following dataset for three categories (Cellphones, CONSOLEs, and Monitors):

product	store	total sale in million $
Cellphone iPhone	BB	4
Cellphone iPhone	CC	2
Cellphone iPhone	AA	1
Cellphone G1	BB	3
Cellphone G1	CC	2
Cellphone G1	AA	0
Cellphone N95	BB	2
Cellphone N95	CC	1
Cellphone N95	AA	3
Monitor Dell	BB	2
Monitor Dell	CC	1
Monitor Dell	AA	1
Monitor LG	BB	4
Monitor LG	CC	3
Monitor LG	AA	0
CONSOLE XBOX	BB	6
CONSOLE XBOX	CC	3
CONSOLE XBOX	AA	7
CONSOLE PS2	BB	5
CONSOLE PS2	CC	5
CONSOLE PS2	AA	8
CONSOLE wii	BB	9
CONSOLE wii	CC	9
CONSOLE wii	AA	8
CONSOLE PS3	BB	8
CONSOLE PS3	CC	0
CONSOLE PS3	AA	9

a) Generate the corresponding 2-dimensional datacube. Assume that the dimension attributes are alphabetically sorted in ascending order in the resulting cube(3 pts).

	AA	BB	CC
Cellphone G1	0	3	2
Cellphone iPhone	1	4	2
Cellphone N95	3	2	1
CONSOLE PS2	8	5	5
CONSOLE PS3	9	8	0
CONSOLE wii	8	9	9
CONSOLE XBOX	7	6	3
Monitor Dell	1	2	1
Monitor LG	0	4	3

b) Compute the corresponding Prefix Sum cube (5 pts).

	AA	BB	CC
Cellphone G1	0	3	5
Cellphone iPhone	1	8	12
Cellphone N95	4	13	18
CONSOLE PS2	12	26	36
CONSOLE PS3	21	43	53
CONSOLE wii	29	60	79
CONSOLE XBOX	36	73	95
Monitor Dell	37	76	99
Monitor LG	37	80	106

c) Answer the following query using your prefix sum cube. Specifically show the cells you are using to answer this query (4 pts).

"Total sales for all CONSOLE products in all stores except BB."

Sum(0:0,3:6) + sum(2:2,3:6)
= P[0,6] – P[0,2]
+ P[2,6] – P[2,2] – P[1,6] + P[1,2]
= 36 – 4
+ 95 – 18 – 73 + 13
= 32
+ 17 = 49

d) We want to update the data by <" Cellphone N95","CC",5>. Show the cells which need to be updated (3 pts).

	AA	BB	CC
Cellphone G1	0	3	5
Cellphone iPhone	1	8	12
Cellphone N95	4	13	18
CONSOLE PS2	12	26	36
CONSOLE PS3	21	43	53
CONSOLE wii	29	60	79
CONSOLE XBOX	36	73	95
Monitor Dell	37	76	99
Monitor LG	37	80	106

Problem 6: (20 points)

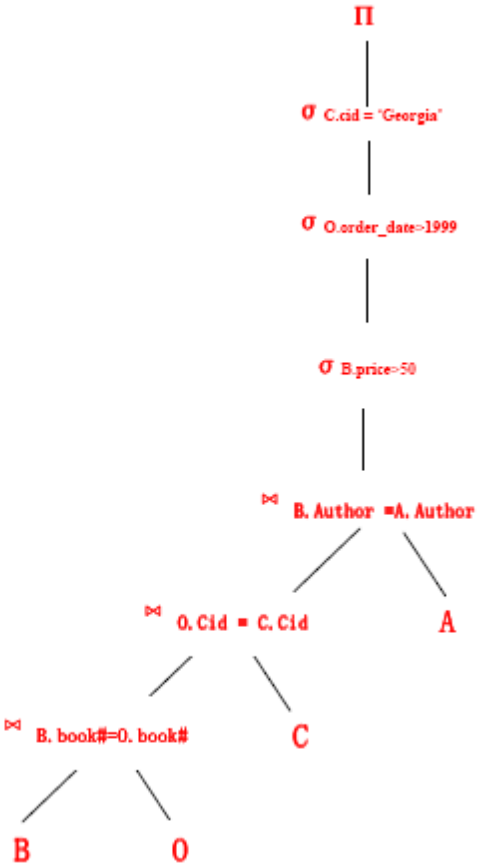
Consider the following database at BooksDotCom:
- CUSTOMER (Cid, Cname, Ccity, Cstate)
- BOOK (Book #, Author, Price, Topic, ...)
- ORDER (Order#,Cid,Book#, Order_date,Payment_type, ...)
- AUTHOR (Author, Affiliation,Author-type, ...)

Each table has some additional fields that we are not interested in.

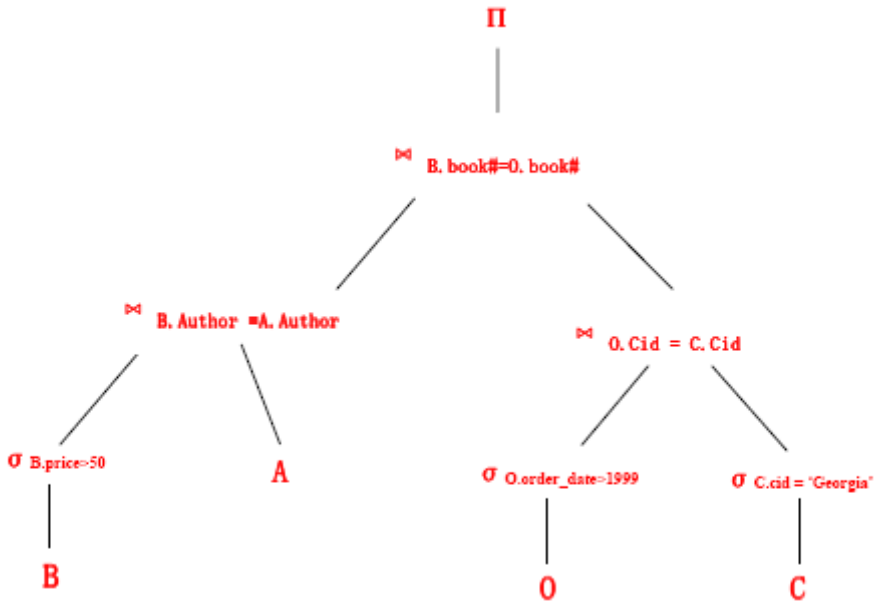
a) Write the following query in SQL.
"List the Book #, Price, Author-type, Customer_city, Date of order for books ordered by customers in Georgia after 1998 (order_date > "01-JAN-1999"); only include books that cost more than $50.00" (2 pts)

```
Select B.Book#, B.Price, A.Author-type, C.Ccity, O.orade_date
From Customer C, Book B, Order  O, Author A
Where B.Book# = O.Book# AND
        O.Cid = C.Cid AND
        B.Author = A.Author AND
        B.Price > 50 AND
        O.order_data > "01-JAN-1999" AND
        C.Ccity = "Georgia"
```

b) Draw the relational algebra query tree for the query. (4 pts)



c) Use the heuristics for algebraic query optimization to transform/restructure the query-tree you generated at part (b) above into a more efficient query-tree. (4 pts)



d) Given the following database catalog information and the fragmentation of relations across sites, show how this query can be optimally executed in a distributed environment. Present your solution by generating a list of operations that must be performed in sequence to execute the query. Each operation is either a local query performed at a particular site or a data transfer between sites. In your list, number the operations in chronological order, and use the same number for the operations that can happen in parallel. (10 pts)

Catalog information:
- Cardinality of BOOK is 10 million
- Cardinality of ORDER is 200 million
- Cardinality of CUSTOMER is 30 million
- Cardinality of AUTHOR is 3 million
- Orders are saved from Jan-01-1972 to Nov-27-2006
- Book prices range from $5 to $60.
- 5% of all customers are from Georgia
- Assume uniform distributions

Fragmentation:
- Site 1: Customers with Cid ≤ C5000000
- Site 2: Customers with Cid > C5000000
- Site 3: Orders corresponding to books with Cid ≤ C5000000
- Site 4: Orders corresponding to books with Cid > C5000000
- Site 5: Books with Book #'s ≤ B2000000
- Site 6: Books with Book #'s > B2000000
- Site 7: Authors of books with Book #'s ≤ B2000000
- Site 8: Authors of books with Book #'s > B2000000

a- Select customers with cid = 'georgia' in site 1, Select orders with order_date > '01-JAN-1999' in site 3 → join in site 1 or 3 (consider 1)

a- Select customers with cid = 'georgia' in site 2, Select orders with order_date > '01-JAN-1999' in site 4 → join in site 2 or 4 (consider 2)

a- Select Books with price > 50 in site 5, Select authors in site 7 → join in site 5 or 7 (consider 5)