CS585
Database Systems
Summer 2010
Final Exam

Name: _____
Student ID: _____

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

Note: Students can bring a one-sided 8.5”x11” sheet of notes to the exam.

1) 20 pts
   Short answer questions
   a. 2 pts
      Dscribe the fundamental difference between the two technologies: embedded SQL and JDBC

   b. 2 pts
      Describe when using stored procedures can be advantageous

   c. 2 pts
      When designing an XML document, state two reasons why you would declare information as an attribute as opposed to an element.

   d. 2 pts
      When designing an XML document, state why you would store information as part of a tag as opposed to the contents of an element.

e. 2 pts
Describe how a distributed DBMS could be more economical than a central one.

f. 10 pts
Describe how semi join is performed and when it can result in better performance. Give a complete example.

2) 20 pts
Describe how you can perform an AVERAGE range query from a blocked prefix-sum array. Give a complete example for a 2D cube.

3) 15 pts
Write an XML schema which is equivalent to the DTD element declarations below. Make your attributes reusable if possible.

```
<!ELEMENT SalesOrder (Customer,Item+,Contract?,Promotion*)>
<!ELEMENT Customer (#PCDATA)>
<!ELEMENT Item (#PCDATA)>
<!ELEMENT Contract (#PCDATA)>
<!ELEMENT Promotion (#PCDATA)>
<!ATTLIST SalesOrder Quatity CDATA #REQUIRED>
```

4) 15 pts
Given the XML documents users.xml, items.xml, and bids.xml with the abbreviated set of data showing the XML format of the instances,

```
<items>
  <item_tuple>
    <itemno>1001</itemno>
    <description>Red Bicycle</description>
    <offered_by>U01</offered_by>
    <start_date>1999-01-05</start_date>
    <end_date>1999-01-20</end_date>
    <reserve_price>40</reserve_price>
  </item_tuple>
  <!-- !!! Snip !!! -->

<users>
  <user_tuple>
    <userid>U01</userid>
    <name>Tom Jones</name>
    <rating>B</rating>
  </user_tuple>
  <!-- !!! Snip !!! -->

<bids>
  <bid_tuple>
    <userid>U02</userid>
    <itemno>1001</itemno>
    <bid>35</bid>
    <bid_date>1999-01-07</bid_date>
    </bid_tuple>
  <bid_tuple>
  <!-- !!! Snip !!! -->
```

a- Write an Xquery expression to find the number of items offered by Tom Jones in 2001.

b- Write an XQuery expression to find cases where a user with a rating worse (alphabetically, greater) than "C" is offering an item with a reserve price of more than 1000. For each case create a warning that includes user name, rating, and reserve price.

c- Write an XQuery expression to find the userid, name, number of bids, and average bid, for each user who has placed a bid, in order by userid.

5) 15 pts

Consider the following database at PetSuppliesDotCom:

- Supplier (Supplier#, Sname, Scity, Sstate)
- Item (Item#, Supplier#, Price, category, ...)
- PurchaseOrder (PO#, Item#, Warehouse#, PO_date, ...)
- Warehouse (Warehouse#, Wstate, ...)

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

Given the following database catalog information and the fragmentation of relations across sites, show how the following query can be optimally executed in a distributed environment.

```
Select I.Item#, I.Price, W.warehouse#
From Supplier S, Item I, PurchaseOrder P, Warehouse W
Where I.Item# = P.Item#
AND I.Supplier# = S.Supplier#
AND P.Warehouse# = W.Warehouse#
AND P.PO_date > "01-JAN-2010"
AND I.price > 100
AND S.Scity = "Chicago"
AND W.Wstate = "CA"
```

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 points)

Catalog information:

- Cardinality of Supplier is 400
- Cardinality of Item is 200,000
- Cardinality of PurchaseOrder is 15 million
- Cardinality of Warehouse is 156
- Orders are saved from Jan-01-1985 up to now
- Item prices range from $10 to $110.
- 2% of suppliers are based in Chicago
- Warehouses exist in all 52 states
- Assume uniform distributions

Fragmentation given on next page…

Fragmentation:

- Site 1: Suppliers with Supplier# $\leq$ S200
- Site 2: Suppliers with Supplier# > S200
- Site 3: Items corresponding to suppliers with Supplier# $\leq$ S200
- Site 4: Items corresponding to suppliers with Supplier# > S200
- Site 5: Warehouses with warehouse# $\leq$ W70
- Site 6: Warehouses with warehouse# > W70
- Site 7: Purchase orders corresponding to warehouses with warehouse# $\leq$ W70
- Site 8: Purchase orders corresponding to warehouses with warehouse# > W70

6) 15 pts
Consider the following three relations with their keys underlined:

Bus (BId, Company)
Driver (DId, Name, Age, Gender)
Schedule (BId, DId, Date, Route)

Furthermore, assume the following four queries:

• Query 1 (q1):

```
Select Driver.Name, Schedule.Route
From  Driver, Schedule
Where Driver.Gender = "Male"
        and Driver.DId = Schedule.DId
```

Suppose q1 is executed by an application that is located at sites S1 and S2, with frequencies 6 and 10, respectively.

• Query 2 (q2):

```
Select Bus.BId,  Count(Distinct Schedule.Date)
From Bus, Schedule
Where Bus.BId = Schedule.BId
Group By Bus.BId
```

Suppose q2 is executed by an application that is located at sites S3 and S4, with frequencies 10 and 8, respectively.

• Query 3 (q3):

```
Select Driver.Name, Bus.BId, Schedule.Route
From  Driver, Bus, Schedule
Where Driver.DId= Schedule.DId and Bus.BId = Schedule.BId
        and Driver.Gender = "Female"
```

Suppose q3 is executed by an application that is located at sites S3, with frequency 4.

• Query 4 (q4):

```
Select Driver.Name, Bus.BId,
From  Driver, Bus, Schedule
Where Driver.DId = Schedule.DId and Bus.BId = Schedule.BId
        and Driver.age < 25
```

Suppose q4 is executed by an application that is located at sites S4, with frequency 2.

a) 2 pts

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

b) 3 pts

Construct the affinity matrix AA containing all attributes of the relation *Schedule*. Assume each query accesses the attributes once during each execution.

c) 10 pts
Use any of the techniques described in class to find a clustering (into two groups) of the attributes such that it represents the best vertical fragmentation of the relation *Schedule* given the usage described above.

Additional space

Additional space