CS561: Advanced Topics In Database Systems
Spring-2012

Final Exam

Total Points: 170
Release Date: 04/19/2012 @4:00pm
Due Date: 04/24/2012 @4:00pm

Submission: Submit a single zip (or tar) file containing your answers on Blackboard system by the due date.

Question 1 [50 Points]

Given the following database schema:

Department(ID, name, address) -- short name D
Faculty(ID, deptID, firstName, lastName, joinYear) --------------- short name F
Course(ID, deptID, name, numOfCredits) ---------------------------- short name C
Student(ID, deptID, firstName, LastName, joinYear) -------------- short name S
Registration(studentID, courseID, semester, facultyID, grade) -- short name R

The primary key in each relation is underlined. Each faculty member and each course belongs to exactly one department. Each student also belongs to one department. However, a student can take courses outside his/her department.

1) **[10 Points]** Write an SQL query that reports for each semester and department the number of course offered. The output schema should look like:

Semester DepartmentId CountCourses
----

2) **[10 Points]** For each department, report the year in which the largest number of student joined that department. Note that the "Student" relation has the student's department Id and the join year. The output schema should look like:

DepartmentId YearWithMaxEnrollment EnrollmentCount
----

3) **[10 Points]** Give an object-relation design of the schema above. Create the appropriate types and tables and give column the suitable data types.

4) **[10 Points]** Use the SQL-99 syntax (given in class) to query the object-relational model you designed in 3) and report the first name and last name of faculty members who joined in year 2010.

5) **[10 Points]** Write database trigger(s) to enforce the following integrity constraint: The number of students registered in a certain course in a certain semester cannot exceed 50.

Question 2 [50 Points]

Given the attached XML document (recipes.xml), provide the XQuery of the following queries.

1) **[10 Points]** Report the title and nutrition of all recipes.

2) **[10 Points]** For each ingredient, report the recipes (the title only) that uses this ingredient

3) **[10 Points]** Report the title of recipes that have complex ingredients

4) **[10 Points]** For each recipe, report the recipe title, the number of ingredients used, and the number of steps needed in preparation. If there recipe contains 'ComplexIngredient", then you count what's inside them. For example, the output of the first two recipes should look like:

```
<recipe>
        <title>Beef Parmesan with Garlic Angel Hair Pasta</title>
         <numOfIngredient>  11 </numOfIngredient>
         <numOfSteps>  4 </numOfSteps>
</recipe>

<recipe>
        <title>Ricotta Pie</title>
         <numOfIngredient>  11 </numOfIngredient>
         <numOfSteps>  6 </numOfSteps>
</recipe>
```

5) **[10 Points]** Report the recipe with the lowest "fat" number (attribute inside nutrition) and the recipe with the highest "protein" number. Report only the recipe title and the nutrition element. The output should look like:

```
<recipe>
        <title>….</title>
         <nutrition … />
         <Type>  Has the highest protein number </Type>
</recipe>
<recipe>
        <title>….</title>
         <nutrition … />
         <Type>  Has the lowest fat number </Type>
</recipe>
```

Question 3 [40 Points]

Given the following set of transactions, each transaction consists of a set of items. Answer the following questions.

TID	List of Items
T1	I2, I3
T2	I2, I4
T3	I1, I2, I5
T4	I1, I2, I3
T5	I1, I3
T6	I1, I2 ,I3, I5
T7	I2, I3
T8	I1, I3
T9	I3, I5
T10	I1, I2, I4

1) **[20 Points]** Using the Apriori algorithm given in class, show the different scans of the algorithm and in each scan show the candidate itemsets and the frequent itemsets. Assume the minimum support for a frequent itemset = 2 (20%), i.e., above or equal to 20% is frequent.

2) **[10 Points]** Compute the support and confidence of the following association rules

   a. I2 → I3
   b. I3, I5 → I1

3) **[10 Points]** Give two association rules where their confidence is 100%. Report the support of each of these rules as well.

Question 4 [30 Points]

1) **[10 Points]** In distributed databases, the 2 Phase Commit (2PC) algorithms is used to ensure that atomicity of a distributed transactions (that is the transaction is done entirely or not done at all). Describe what will happen in the following case:

a. Assume that in Phase 1 of the algorithm, all clients sent back to the coordinator a "ready T" message except one client sent "do not commit T" and then this client failed. When this client recovers from the failure, what should it do? Does it need to contact the coordinator again or not?

b. Assume in Phase 1, the coordinator sends a "prepare T" message and then crashes. What will happen after it recovers from the crash?

2) **[10 Points]** Assume a customer table that has attributes (ID, Name, Age, CountryCode, Salary) where the records are hash-based partitioned over a distributed system based on the ID column. Now given the following query:

```
Select *
From Customer
Where CountryCode betweem (x, y);
```

Describe how this query will execute in parallel? Will we need to re-partition the data or not?

3) **[10 Points]** Assume a customer table that has attributes (ID, Name, Age, CountryCode, Salary) where the records are hash-based partitioned over a distributed system based on the Name column. Now given the following query:

```
Select Age, Name, sum(Salary)
From Customer
Group By Age, Name;
```

Describe how this query will execute in parallel? Will we need to re-partition the data or not?

Bonus Question (Optional) [30 Points]

This question is a continuation to the Hadoop project that you delivered before (Refer to the project released on 02/16/2012). Using the same datasets described in the project, write a MapReduce code corresponding to the following queries.

1) Report the transactions whose total number of items above 5 and their total cost above 400

2) For every customer, report the customer Id, the total number of transactions done by this customer, and the sum of "TransTotal" of these transactions.