XML Technology Assignment

Due date: 1st June, 2007

Released on: 1st May, 2007

Weighting: 20%

Submission at: OAS (http://www.fit.qut.edu.au/current/oas/oas.jsp)

Creator and Moderator : Dr Richi Nayak

Introduction

This assignment is intended to allow you to display your knowledge and understanding of XML related technologies such as XQuery and XSLT. In this assignment, you will use the Saxon-B 8.9 (Java version) processor to execute the assignment tasks. The purpose of this assignment is to give you an appreciation of benefits that various XML technologies bring to data and document processing.

Instructions

1. The assignment is **due on 1st June**. It is a firm deadline.

2. You should submit your assignment via **OAS**.

3. This assignment will be undertaken by a group of **2 people**. The group is ARRANGED and MANAGED by you. As in real life, the performance of the individuals in the team shall be judged by the performance of the team together, so choose your partners carefully. The academic staff will NOT participate in group formation or dispute resolution.

4. Group should be formed within your cohorts only. ITB students should form groups with their fellow ITB students only. Similarly, ITN students should form groups with their fellow ITN students only.

5. Once you have formed your team, it needs to be registered on Teamworker. This should to be done by end of week 12. The link to the TeamWorker application is:
   https://www.talssweb.qut.edu.au/teamworker

   Please make sure that your team is registered on Teamworker so that marks can be assigned to all members of the team. The reason is that OAS would allow only one submission per group. Only one member of a group should submit the assignment.

6. Of course, the work you (group) hand in must be your own; no collaboration or borrowing from others groups is permitted. We will use the usual methods of detecting plagiarism.

7. Use the XML data files provided (**assign2data.zip**).

8. Name each XQuery, XSLT, HTML, XML file according to the instruction given. You should create/generate:
   - researchnet.xml
   - task1.xq, task1.xml, task1.xsl and task1.html for Task1.
   - task2.xq, task2.xml, task2.xsl and task2.html for Task2.
   - task3.xq and task3.xml for Task3.
   - task4.xq, task4.xml, task4.xsl and task4.html for Task4.

9. After finishing all tasks, you should combine all files into one file i.e. **assign2.zip**. This is the file you are required to submit on **OAS**. Please read the OAS instructions if you have not used it before. The file size is 5 MB, over this limit or/and any file name other than **assign2.zip**, OAS would not accept.

10. Read the Assessment Policies on Blackboard or QUT Website.

Distribution of Marks

Total marks: 20

Tasks	Marks
Task 1 • Xquery • XSLT	Total 5 marks • 4 marks • 1 marks
Task 2 • Xquery • XSLT	Total 5 marks • 1 Marks • 4 Marks
Task 3	4 marks
Task 4 • Xquery • XSLT	Total 6 marks • 2 marks • 4 mark

Failed to Execute Queries/Programs: Markers will be executing your queries/programs on Saxon-B 8.7.1. You loose half of the allocated marks straight away if your query/program fails to successfully run on SAXON.

Poor Quality Queries/Programs: We will be grading your solutions primarily on correctness. However, solutions will mark down for contorted queries or inefficient solutions (which for XQuery tend to go hand in hand). So please pay attention on how your queries/programs are expressed or how efficiently they execute.

You loose some of the marks (a small fraction) if you are using unnecessary variables, duplications and expressions. So use the appropriate variables, functions and expressions. For example, if you are asked to rewrite an element name in a particular format, do not do it manually, but do it automatically with using functions and expressions. Another example is inappropriate use of *distinct-value* command. Use the "functions" if you intend to repeat the same process more than once or need to use a function within a method.

In particular, if you are asked to conduct a task in Xquery, perform it with Xquery rather than XSLT and vice-versa.

Description

Consider the following data related to a university:

1. staff: name, position, faculty, email, homepage;
2. project: id, name, description, members, fundings, products, publications, start-date, end-date;
3. funding: id, source, amount, start-date, end-date;
4. product: id, name, type, date, cost, authors;
5. publication: id, name, authors, type, detail, publisher, date.

The complete database or application might have many constraints, different kinds of transactions and GUIs, as treated in a standard database; we ignore those in this assignment and focus on XML data manipulation using Xquery and XSLT. For this assignment, you only need to work on queries and XSLT programs, not on the DTD and XML Schema.

A sample data about the university research data collection is provided with this assignment. The researchNetData.txt file contains this data. Do not modify the content of the data file.

Create a XML document, labelled as "researchnet.xml", to store the given information. Retain the (structural) relationships given as above while populating the data file "researchnet.xml".

You should adopt the "XML shredding" approach while mapping these relations into a XML document. An example of this approach is given at Slide no 63 of the XQuery lecture slides. The "staff", "project", "funding", "product" and "publication" should all be immediate descendent of the root node. A "row" node should be used to hold each instance of these nodes. Use the given names only as elements or attributes of the file. Use your discretion for deciding the data as an attribute or as an element.

Following are the specific tasks. Please adhere to the instructions given in each task.

Task 1

The task is to produce an HTML document that displays a list of faculties and their staff involved with the projects in the format as below using XQuery and XSLT.

The result should be grouped by faculty and displayed in sorted order.

For each faculty, report the staff name, project names that they are members of and the total amount of fundings of all projects. (Assuming that the amount is divided equally among multiple projects with the same funding, and the amount is divided equally among multiple participants of the same project).

Entries should be sorted according to funding amount in decreasing order.

Staff members without any project should also be listed in the end for each faculty in the ascending order of names.

A sample output:

Following is the list of projects for each faculty.

Information Technology		
Mary	Project 1, Project 2, Project 3	$500,000
Alan	Project 1	$100,000
Mike		
Law		
John		
Lucy		

Notes: The format of the output report should be as above.

The faculty names should start with uppercase and with no punctuation in middle.

There should be a '$' sign preceding the total amount of funding with no space in between.

You should first write the XQuery query that groups the staff according to faculty, orders the display list as asked above, and formats the tags and values as asked.
And then, use the XSLT program to display the output as HTML only.

Instruction:

- Name the Xquery query as task1.xq.

- Run this query on SAXON. Save the query output as task1.xml.
- Name the XSLT program as task1.xsl.
- Execute this program on SAXON. Save the output HTML file as task1.html
- Check that the HTML file is displaying the required information.

Task 2

The task is to determine names of all products of any project that have a funding with the most recent start date. For each product, list the name of author, cost, date of making and faculty of the author.

Retrieval of the related information should be done via XQuery.
And then, XSLT should be used for processing the tags, sorting the display order by product names, and display the information as required.

Hardware 1			
Mary	$432	03/02/2007	Information Technology
Software 1			
Alan	$100	01/07/2005	Information Technology
Mary	$100	01/07/2005	Information Technology

Notes:

For each product, list the names of authors in sorted order.

The faculty name should start with uppercase and with no punctuation in middle.

Date should be formatted as dd/mm/yyyy.

There should be a '$' sign preceding the cost with no space in between.

XQuery should be used to determine products that have a funding with the most recent start date only.

XSLT should be then used for displaying the report in the required format as above.

Instruction:

- Name the Xquery query as task2.xq.
- Run this query on SAXON. Save the query output as task2.xml.
- Name the XSLT program as task2.xsl.
- Execute this program on SAXON. Save the output html file as task2.html
- Check that the html file is displaying the required information.

Task 3

Write a XQuery query for reporting the number of senior staff (professors and associate professors) for each faculty if the proportion of senior staff is higher than 50% of the total staff that work in the faculty.

The query should produce a XML document that displays the total number of senior staff only if the condition is met. If a faculty does not have more senior staff than junior staff, print the count as zero.

Additionally, display the name and position of each senior staff for each faculty (that meets the condition) listed in sorted order of their names.

The output format should look like as:

```
<Faculties>
        <Senior-staff-Information-Technology>
                <Count> ….   </Count>
                <Staff>
                        <name> …. </name>
                        <position> …. </price>
                <Staff>
                <Staff> ….</Staff>
        </ Senior-staff-Information-Technology>

        <Senior-staff-Law>
                        …..
        </Senior-staff-Law >

</ Faculties >
```

Notes:

The above example is for giving you a hint on **tags** that are required for this task.

Your answer would offcourse replace the "….." with right values and tags.

The tag such as "<Senior-staff-Information-Technology >" that includes a faculty name should be created automatically not statically (or manually).

If there exist no such faculties, display the name of the faculty tag as asked with the Count as 0.

Instruction:

- Name the Xquery query as task3.xq.
- Run the query on SAXON. Save the query output as task3.xml.

Task 4

This task includes the generation of ordering reports based on the information contained within three files: `products.xml`, `shippers.xml` and `members.xml`.

Write a XQuery for returning a XML document that contains information about the following two orders that are placed via telephone:

1. Sue has ordered two products of microsoft-technologies: VB (40 copies → number) and PowerPoint (2 copies). She always uses 'fedex' for delivery.
2. Bob wants 5 copies of opensource-technologies product Java, 40 copies of microsoft-technologies: VB, and 5 copies of operating-systems product Solaris. He will like to use the 'ship mail' shipper for delivery.

The 'gold' status customer gets the first preference in receiving orders among the orders placed on the same day.

A warning is issued if enough stock is not available for a product but the available stock is delivered. And the process is continued with remaining orders.

These two above tasks should be performed with XQuery.

Finally, write a XSLT program that displays these order information in a browser in a form similar to the form as below.

If there is more than one product ordered then the products should be listed in alphabetic order.

Report for each customer should be clearly separated from others.

A sample report:

Report 1:

Name	
Email	
Shipping Address	
City	
Zip	

Product	Price	Quantity Ordered	Quantity Delivered	Total Price	Warning

Choose the type of shipping ◉ 4-14 days ○ 1-2 days

Instruction:

- Name the Xquery query as task4.xq.
- Run this query on SAXON. Save the query output as task4.xml.
- Name the XSLT program as task4.xsl.
- Execute this program on SAXON. Save the output html file as task4.html
- Check that the HTML file is displaying the required information.