XML Data Management: Exam

Werner Nutt

13/09/2012

- The exam is computer based. This means, you will work with a Windows desk top on which the following tools are installed:

  - Eclipse with a validator for DTDs and XML schemas and an editor for XML schema
  - Kernow for Saxon.

  In addition, there are the standard text editors under Windows and gVim, a version of vi.

- The exam is a **closed book** exam. The Web pages of the course are accessible from the exam machines.

- After the exam, all files on the `Y:` drive will be collected. Please, leave all your answers on the `Y:` drive (for details, see inside the booklet).

- The exam comprises 3 questions, some of which consist of several subquestions. There is a total of 60 points that can be achieved in this exam. You will have 2 hours time to answer the questions.

- **No questions** will be answered during the exam. If you are not sure about interpreting a question, you may write down additional assumptions you made in order to proceed with your solution.

- For students that did not submit assignments, the final mark will consist solely of the mark for the exam. For students that submitted *n* assignments for the labs the final mark for the course will be a weighted average of

  $$70\% \text{ exam mark} + n \times 10\% \text{ lab mark}$$

  or the exam mark, whichever is higher. (In other words, if a student submitted 2 courseworks, the weight is 20%, if he/she submitted 3 courseworks, the weight is 30%, etc.)

In the exam you will use two documents that I have extracted from the `mondial` database, originally compiled by Wolfgang May:

- `mondial-small.xml`
- `mondial.xml`.

You will work with `mondial-small.xml` for your XML schema question and with `mondial.xml` for the XQuery and XSLT questions.

1 XML Schema

Write an XML schema "`mondial-small.xsd`" that as tightly as possible captures the `mondial-small.xml` document. In other words:

- the document should validate against your XML schema;
- whenever one tightens the constraints of your XML schema, the document should no more validate.

It is sufficient if all your simple types are built-in types.

How to proceed: Start Eclipse. There is a folder "XML Exam", which contains two files, `mondial-small.xml` and `mondial-small.xsd`. Create the XML schema in `mondial-small.xsd`.

(20 Points)

2 XQuery

You are asked to write queries in XQuery that implement the questions below.

How to proceed: In the top directory on your `Y:` drive, there are two files, `mondial.xml` and `answers.txt`.

Start Kernow to develop queries that answer the questions. Note that within Kernow, you can refer to the `mondial.xml` file with

```
doc("Y:/mondial.xml")
```

When you finish a query, copy it into the corresponding slot of `answers.txt`.

Here are the questions:

1. How many countries are there in the file `mondial.xml`?

   (2 Points)

2. Which is the country with the smallest population in Europe?
   (Check the result. Saxon may show a bug here if you use pure XPath.)

   (4 Points)

3. Return an alphabetically sorted list of all continents, where a continent element has an attribute “name” and no other content.

   (2 Points)

4. Extend the previous query such that also the number of countries on a continent is returned.

   (2 Points)

5. Modify the previous query so that it returns the continents in alphabetical order and, for each continent, the name of the country with the largest population and the population of that country.

   (4 Points)

6. Return an alphabetically ordered list of religions, together with the number of members of that religion.

   (6 Points)

7. Refine the previous query so that for each religion the total number of members is given as well as the number on each continent, provided the number is not 0.

   (6 Points)

8. Return for each continent the names of those organizations, where the name of the continent is contained in the name of the organization.

   (3 Points)

3 XSLT

You are asked to write XSLT stylesheets that fulfill the requirements below. Proceed as for the XQuery part.

1. Write a script that returns a list of all cities in the document, ordered alphabetically.
   Each city element in the result will contain the name of the city and the name of the country to which it belongs, but no other information.
   Note that the cities in a country are not always immediately below the country element but may as well be nested in a province element.

   (5 Points)

2. Write a script that returns a list of all cities in the document for which the last population count was greater than 100,000.
   (Note that some cities have no population elements while others have more than one, if data are available for several years.) Each city element in the result will contain the name of the city, the last population number, and the name of the country to which it belongs, but no other information. The cities will be ordered according to the population number, in descending order. Cities with identical population will be ordered alphabetically.

   (6 Points)