Introduction to Databases (winter term 2005/2006)

Assignment 11

hand in on January 30, 2006 during the tutorial

This is the last assignment for Introduction to Databases in this semester. It will be discussed and returned to you on February 6, which will be the date of the last tutorial session.

To be able to plan the written examination that takes place on February 15, we would like to ask you to register informally by an online form linked from our homepage.

Task 11.1 (XML University Database) (2 Points)

Consider the XML Schema document Lectures.xsd linked from our homepage which constraints the format of XML documents that describe a collection of lectures. These lectures can be part of several courses and can end with an oral or written examination.

a) Decide for the two sample XML documents linked from our homepage (Lectures1.xml and Lectures2.xml), whether they are valid according to the given schema. For the invalid document(s), give all reasons why they violate the schema constraints.

b) Describe the following query in XQuery:

List all lectures with an oral examination grouped by their instructors in the following format:

```xml
<lectures>
  <instructor name=INSTRUCTOR_NAME email=INSTRUCTOR_EMAIL>
    <lecture lang=LECTURE_LANG>LECTURE_TITLE</lecture>
  </instructor>
</lectures>
```

The instructors should be sorted in alphabetic order. Formulate your query against an XML document `lectures.xml` that is considered to be valid according to the given schema.

Task 11.2 (XQuery) (4 Points)

Although Oracle’s support for XML features can been seen as quite complete since the version 10g (XML schema, XML Query and XSL Transformations can be used), the handling of XML sometimes still is a bit cumbersome and slow. We therefore encourage you to develop the following XML Queries with the help of a public web frontend of a well-known LGPL licensed native XML server called eXist. This frontend can be accessed via the URL http://demo.exist-db.org/sandbox/sandbox.xql.

As a data basis for the queries we employ the Mondial database, that can be inspected at http://www.dbis.informatik.uni-goettingen.de/Mondial/mondial-europe.xml. It stores a lot of information about countries, cities, lakes, etc. This data basis is accessible from the eXist front end under the path /mondial. To view the stored information about Germany we could e.g. use the XPath expression /mondial/country[name='Germany']. Please also take a look at the instructive saved XML query examples available in the combo box of the XQuery Sandbox frontend (starting with the 6th entry).

Formulate the following information demands as XML queries:

a) List the name and the inflation of all countries that have common border with Germany in the following format:

```xml
<GermanyNeighbours>
  <country>
    <name>...</name>
    <inflation>...</inflation>
  </country>
</GermanyNeighbours>
```

b) List all countries and their population where the percentage of Muslims is above 50 in the following format:

```xml
<MuslimCountries>
  <country name = "...">
    <population>...</population>
  </country>
</MuslimCountries>
```
c) List all cities (do not forget cities inside province elements) sorted by their population in descending order. Use the following output format:

```xml
<Cities>
  <city>
    <name>...</name>
    <population>...</population>
  </city>
</Cities>
```

Please note that in the current XML Query version that is implemented by eXist, the Sortby directive is replaced by order by. Since eXist does not support XML Schema yet, it treats numerical values as text which affects the sorting. Use the expression order by xs:integer(Expr) descending to cast Expr to an integer and sort the current node set according to it in descending order.

d) List all islands group by the country to which they belong. The sequence of countries should be sorted in descending order by the area the islands cover in this country. Use the following output format:

```xml
<countryinfo>
  <country name="..." islandArea="...">
    <island name="..." area="...">
    </island>
  </country>
</countryinfo>
```

Task 11.3 (Structural Recursion with XSLT) (2 Points)
Linked from our homepage you find a small sample XML document ShopData.xml that contains some shop information about DVDs and Books.

Write an XSLT stylesheet that transforms an XML document of this kind to a simple price list of DVDs in the following format:

```xml
<DVDPrices>
  <DVD>
    <Title>...</Title>
    <Price>...</Price>
  </DVD>
</DVDPrices>
```

To develop and test your stylesheet you can e.g. use the freely available Home Edition of Altova’s XMLSpy¹.

---
¹download: http://www.altova.com/support_freexmlspyhome.asp

For additional information about XML Schema, XSLT and XML Query, we recommend to visit the official W3C site on the web at www.w3.org.