Winter 2009 **CSC 468:** DBMS Implementation Alexander Dekhtyar

Lab 5: XPath Practice

Due date: Monday, February 9, midnight.

Lab Assignment

The purpose of this assignment is for you to learn XPath. XPath is an XML path expression language. The course project will have you develop an XPath processor on top of the native XML index structures you have built. This requires some preliminary knowledge of XPath.

In the lab you will use eXist, an open-source native XML DBMS, which includes full XPath (and XQuery) support. You will download eXist, install it (it is distributed as a Java .jar file), and will work with it in interactive mode.

This is an ¡b¿individual lab¡/b¿: each student is expected to submit his/her own solutions. Only one aspect of this lab, pertaining to data preparation, can be done in teams.

The Tasks

There are two basic tasks in this lab: (1) download, installation and learning your way around eXist and (2) design of XPath expressions for XML data and information needs provided below. Additionally, each team will construct a single XML repository file out of its Lab 2 XML files.

Preparation of data

The lab will use four datasets: Food, Movies, Money and Music. Each dataset shall consist of a single XML file, `FoodRepository.xml`, `MovieRepository.xml`, `MoneyRepository.xml` and `MusicRepository.xml`.

Each of the repository files is organized as follows. The root of each file is an XML element `<repository>`. The children of the `<repository>` element are the top-level objects from the Lab 2 XML files for each respective team:

- For the `Food` repository, the children of the `<repository>` element are the `<restaurant>` elements corresponding to menus of all restaurants described by The Food Group in Lab 2.

- For the `Money` repository, the children of the `<repository>` element are the `<currency>` elements corresponding to descriptions of individual bank notes and coins created by The Money Group in Lab 2.

- For the `Movies` repository, the the children of the `<repository>` element are the `<tv>` and `<movie>` elements describing the DVDs, as created by The Pointers in Lab 2.

- For the `Music` repository, the the children of the `<repository>` element are the `<album>` elements describing the CDs, as created by Group Name in Lab 2.

Each group shall prepare the appropriate repository file by merging the content of all their Lab 2 XML files as described above. The resulting files shall be loaded onto the course wiki. Links to the files shall be placed on the front page of the wiki, to ensure that everyone has easy access to the files. Feel free to add a link to the raw file (`"?format=raw"`) in addition to the regular link.

eXist Installation

eXist is an open-source light-weight native XML DBMS. It is distributed in the from of a Java .jar file.

eXist comes with a variety of access modes: client-server architecture, embedded mode (eXist provides API that can be directly queried from Java code) and interactive mode. In this lab, you will be using eXist's GUI client application, and will perform all actions within it.

eXist implements XQuery as the main method of access to stored XML data. XPath is the subset of XQuery designed to provide access to XML data. In the class we discuss XPath version 1.0, and this is the language you will be using in this lab.

Prior to start work on XPath you need to:

1. Download eXist. On CSL machines, you will be using eXist version 1.2.1, revision 7764. This revision ships in a .jar file `eXist-setup-1.2.1-rev7764.jar`.

   This file can be obtained

   - from the course home page:

     http://www.csc.calpoly.edu/~dekhtyar/468-Winter2009

     (scroll all the way down to **"eXist"**).

   - from the Sourceforge eXist page:

     http://sourceforge.net/project/showfiles.php?group_id=17691&package_id=13430

Scroll down to find `eXist-setup-1.2.1-rev7764.jar`

On your own PCs/Macs, you can use eXist version 1.2.4, revision 8072. Download the .jar (for Macs) or .exe file (for PCs) at

http://exist.sourceforge.net/download.html

(note, this page is linked to from the course web page.)

2. Install eXist. Run the downloaded .exe file or the .jar file. Select necessary installation options (you do not need sources and javadocs, but feel free in include them). Select the admin password when prompted.

3. Run eXist. eXist comes with a client-server setup. On CSL machines, to run the server change to the directory in which eXist is installed and enter the command

```
bin/startup.sh &
```

To run the client, enter

```
bin/client.sh&
```

To shut down the server, enter

```
bin/shutdown.sh
```

On Windows systems, you can use `bin`
`startup.bat`, `bin`
`client.bat` and `bin`
`shutdown.bat`, or you can find eXist programs in your system's "Start" menu.

4. Become familiar with the client's interface. You need to learn how to

   - Load an XML file from disk.
   - Browse an existing XML file.
   - Enter query mode.
   - Enter queries and observe answers[1].
   - Exit program.

[1]Note, that since the project involves implementing a(n almost) proper subset of XPath, you can use eXist later on for debugging purposes - it will allow you to establish the correct answer to any XPath query.

XPath queries

Each file loaded on eXist is treated as a separate (unique) XML document/repository. Because eXist supports handling multiple documents, any XPath expression must be prefaced with the designation of the XML document over which this expression shall range. This is done in eXist using XQuery's built-in `document()` function. The format of a full XPath expression understood by eXist is thus:

```
document(<documentName>)/<Relative-XPath-Expression>
```

Here, <documentName is the name of the document — typically the name of the file from which the XML document was read (in double quotes). <Relative-XPath-Expression> is any relative XPath expression.

We use four XML files for this lab. You need to load each of them as a separate document into eXist's database. Please take some time to study the structure of the XML files — without knowing it, it may be hard to compose XPath expressions.

FoodRepository.xml

For this file, you need to write XPath expressions retrieving the following information:

1. List the names of all restaurants in the repository.
2. Find all dishes in the repository that cost exactly $2.99.
3. Find all prices for a dish called `"Hamburger"`.
4. Find all restaurants that sell tacos (hint: use `contains()` built-in function). Return the names of the restaurants.
5. Find all breakfast items at `Louisa's Palace` which cost less than $8. Return their names.

MoneyRepository.xml

1. List all the mints found in the repository.
2. For each coin that is no longer valid, list its year of release.
3. Report the denominations of all the USA coins/bank notes.
4. Report the mint of the 5 Euro coin.
5. Report the names and the denominations of all coins/bank notes released after 1990.

MovieRepository.xml

1. List the titles of all movies in the repository.
2. List the titles of all Eureka episodes that follow episode named `"Duck Duck Goose"`.
3. List the first three stars of `"Memento"`.
4. List the titles of all **R**-rated movies.
5. List the genres of all `Sony Pictures` movies.

MusicRepository.xml

1. Report the names of all the albums in the repository.
2. Report all the tracks of the album on which `"Kronos Quartet"` is playing.
3. List all bass players from the repository.
4. Find all tracks on `"Present"` that preceed `"Double Bass"`.
5. List titles of all 1-disk CDs.

Submission

Put all XPath queries into a single file. Name the file `lab03-<userName>.xq`, where `<userName>` is your Cal Poly login Id. Use `"(:"` and `":)"` symbols to place comments in the file. E.g.,

```
(: Lab 5.  Alex Dekhtyar :)
(: Movie Repository      :)

(: test query            :)

document("MovieRepository.xml")//genre

(: End of File           :)
```

Submit the `.xq` file you have created using handin:

```
$ handin dekhtyar lab05 lab03-<userName>.xq
```