Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

January 15, 2018

XML and Databases
— Exercise Sheet 10 —

You only have to submit the parts marked as "Homework Exercises", i.e. Part c). But please think about the questions in Part a) before the meeting! Send your homework solutions to the instructor via EMail: `brass@informatik.uni-halle.de` (with "`xml17`" in the subject line). The official deadline is January 18, 10:00 (before the lecture time).

Repetition Questions

a) Answer the following questions about XQuery:

- What is the most important restriction of XPath compared to XQuery (i.e. what can be done in XQuery that is not possible in XPath)?
- What are FLWOR-expressions? Name the keywords and explain the purpose of each clause.
- Compare FLWOR-expressions with SQL. Which XQuery-keywords correspond to the SQL-keywords SELECT and FROM?
- How do direct constructors look like? In which places can one insert expressions that should be evaluated? How are these expressions marked?
- Consider the direct constructor <E>{Q}</E> gegeben. Let the result of evaluating Q be the sequence S. What is the result of this constructor if S is a sequence of atomic values? What happens if S contains a prefix of attribute nodes?
- What is "Boundary Whitespace" in a direct constructor? Why is this notion important?
- Name a system that can be used to evaluate XQuery expressions.
- Consider a simple SQL query such as

```
SELECT E.ENAME, E.SAL
FROM   EMP E, DEPT D
WHERE  E.DEPTNO = D.DEPTNO
AND    D.DNAME = 'RESEARCH'
```

  Discuss various possibilities to translate this to XQuery (of course, this depends also on how the data from the relational tables is represented in XML, and how the output should look like — you can also discuss alternatives for that).

- Consider the following FLWOR-expression:

```
for    $x in //a
where  exists($x/b) and $/@c > 0
return $x/d
```

  Give some equivalent versions of this query. In which other places except under `where` can conditions be written?
- Discuss the statement "XQuery queries are often longer and more complex than the corresponding SQL queries".
- Suppose that the result of the query "`//a/b[c and d]`" is empty, but you know that this is not the expected answer. What would you do to find the error?
- Compare the error handling or debugging support of typical XQuery implementations (such as BaseX) to the treatment of errors in typical relational databases (such as Oracle).
- How can LIKE conditions be expressed in XQuery?
- How can one specify in XQuery that one wants output in "descending" order, i.e. the largest value first? Consider the following example:

```
SELECT E.ENAME, E.SAL
FROM   EMP E
ORDER  BY E.SAL DESC
```

- Suppose the element that contains the sorting value does not exist for some variable assignments. The generated answers of these variable assignments should appear last in the result sequence. How can this be specified (for the normal "ascending" order)?
- Which aggregation functions exist in XQuery? How can a simple aggregation query be translated to XQuery? Consider the following example:

```
SELECT SUM(E.SAL)
FROM   EMP E
WHERE  E.JOB = 'CLERK' OR E.JOB = 'MANAGER'
```

- How can GROUP BY queries be translated to XQuery? Explain this with the following example:

```
SELECT D.DEPTNO, D.DNAME, MIN(E.SAL), MAX(E.SAL)
FROM   EMP E, DEPT D
WHERE  E.DEPTNO = D.DEPTNO
GROUP  BY D.DEPTNO, D.DNAME
```

- How can duplicates be eliminated in XPath/XQuery?

In-Class Exercises

b) We will continue to work on last year's exam, available at this web address:

[http://www.informatik.uni-halle.de/˜brass/xml17/exam16.pdf]

Homework Exercises

c) Consider again the XML file for the classical music CDs:

- Data file:
  [http://www.informatik.uni-halle.de/˜brass/xml17/cd.xml]
- XML Schema definition:
  [http://www.informatik.uni-halle.de/˜brass/xml17/cd.xsd]
- DTD:
  [http://www.informatik.uni-halle.de/˜brass/xml17/cd.dtd]

The document has the following elements:

- `CDDB`: `(composers, cds?, soloists?)`
- `composers`: `(composer*)`
- `composer`: `(pieceOfMusic*)`, attributes: `cno`, `firstName`, `name`, `born`, `died`.
- `pieceOfMusic`: `(recording*)`, attributes: `pno`, `title`, `key`, `opus`.
- `recording`: empty content, attributes: `rno`, `orchestra`, `conductor`.
- `cds`: `(cd*)`.
- `cd`: `(track*)`, attributes: `cdno`, `name`, `producer`, `numDiscs`, `totalTime`.
- `track`: empty content, attribute: `rno`.
- `soloists`: `(soloist*)`.
- `soloist`: `(performance*)`, attribute: `name`.
- `performance`: empty content, attributes: `rno`, `instrument`.

Please write the following queries in XQuery and test it with an actual implementation, e.g. BaseX:

[http://basex.org/home/]

It is important that you gain "hands-on experience" with XQuery. If there were any difficulties or other interesting observations, please share your experiences in our meeting.

Please write the following queries in XQuery (or its subset XPath):

- Generate a document that contains all composers, but only ith the attributes `firstName`, `name`, `born`. The `pieceOfMusic`-elements that are nested in the input file in the `composer`-elements should not appear in the output. The out should look like this:

```
<homework10>
    <composer firstName='Georg Friedrich' name='Händel'
              born='1685' />
    <composer firstName='Serge' name='Prokofiev'
              born='1891' />
    ...
</homework10>
```

  Of course, the line breaks and indentation is selected by the implementation and can be different than shown here.

- Please print a list of composers that were born before 1700, ordered by year of birth (descending). The output should be an unordered list in HTML and structured as follows:

```
<ul>
    <li>Lean-Marie Leclair: 1697</li>
    <li>Pietro Locatelli: 1695</li>
    <li>Benedetto Marcello: 1686</li>
    <li>Georg Friedrich Händel: 1685</li>
    <li>Johann Sebastian Bach: 1685</li>
    <li>Domenico Scarlatti: 1685</li>
    <li>Georg Philipp Telemann: 1681</li>
    ...
</ul>
```