



Universität Freiburg  
Institut für Informatik  
Prof. Dr. G. Lausen  
Alexander Schätzle  
Martin Przyjaciel-Zablocki

Georges-Köhler Allee, Geb. 51  
D-79110 Freiburg  
lausen@informatik.uni-freiburg.de  
schaetzle@informatik.uni-freiburg.de  
zablocki@informatik.uni-freiburg.de

## Data Models and Query Languages Summerterm 2013

### 5. Exercise Sheet: XQuery & XML SQL

Discussion: 21.06.2013

#### Exercise 1 (XQuery)

Given the XML document “bib.xml”<sup>1</sup>. Formulate the following queries in XQuery.

- Output all books where the last name of the author and the last name of the editor is the same.
- Output the title and number of authors for every book of author Peter Buneman. Also output the price of the book, if it is above 30.
- Output all pairs of different books from the same publisher without duplicates. You may assume that book titles are unique.
- For each author, output his/her last name, first name, and the sum of prices of those books he/she has (co-)authored. Order the output by price sum.
- Output an HTML document with title “Number of books:  $X$ ” where  $X$  is the number of all books in the XML document. There should be a headline with text “The document contains books from  $Y$  authors” where  $Y$  is the number of distinct authors. Finally, there should be a table of all book titles and prices, sorted by title.

#### Exercise 2 (XQuery)

Which of the following queries are pairwise equivalent? For query pairs that are not equivalent, give an XML document that verifies this.

- a) Query 1.1:

```
<q1> { for $a in /a, $b in $a//b return <match/> } </q1>
```

Query 1.2:

```
<q1> { for $a in /a return  
  for $b in $a//b return <match/> } </q1>
```

Query 1.3:

```
<q1> { for $b in /a//b return <match/> } </q1>
```

- b) Query 2.1:

```
<q2> { let $x := (1, 2, 3)  
  return <a>{ $x }</a> } </q2>
```

---

<sup>1</sup><http://tinyurl.com/w3c-bib-xml>

Query 2.2:

```
<q2> { let $x := (1, 2, 3)
      for $y in $x return <a>{ $y }</a> } </q2>
```

Query 2.3:

```
<q2> { for $x in (1, 2, 3) return <a>{ $x }</a> } </q2>
```

c) Query 3.1

```
<q3> { for $book in //book return
      for $article in //article
      where $article/author=$book/author
      return $book } </q3>
```

Query 3.2

```
<q3> { for $book in //book return
      for $article in //article return
      if ($article/author=$book/author)
      then $book else () } </q3>
```

Query 3.3

```
<q3> { for $book in //book return
      for $author in //article/author
      where $book/author=$author
      return $book } </q3>
```

### Exercise 3 (XML SQL)

Consider the following XML document:

```
<bib><book><title>DBMS</title><authors><author>Ramakrishnan</author>
<author>Gehrke</author></authors></book></bib>
```

- Draw the tree representation of the document and give for every node  $n$  its preorder rank  $pre(n)$ , postorder rank  $post(n)$ , depth  $level(n)$  and size of its subtree  $size(n)$ .
- Is there a correlation between preorder/postorder rank and start/end tags?
- Verify the following axis characterizations for the authors node of the tree:
  - $n'$  is descendant of  $n \iff pre(n) < pre(n')$  and  $pre(n') \leq pre(n) + size(n)$
  - $n'$  is preceding of  $n \iff pre(n') + size(n') < pre(n)$
- Refute the following **wrong** characterization for the parent axis. How do you have to change it such that the characterization is correct?

$$n' \text{ is parent of } n \iff pre(n) > pre(n') \text{ and } pre(n) \leq pre(n') + size(n')$$