Exercices in XQuery
Querying DBLP Data

DBLP[1] is an online bibliographical database for computer science containing around 1 million references. Its content is publically available in XML format[2]. Since this content is more than 400 MB a small excerpt of this data will be used for this exercice.

The DBLP collection follows the BibTeX format and contains the following types of references: article, inproceedings, proceedings, book, incollection, phdthesis, mastersthesis, and www. The fields describing the above types of references are the following: author, editor, title, booktitle, pages, year, address, journal, volume, number, month, url, ee, cdrom, cite, publisher, note, crossref, isbn, series, school, and chapter. Notice that the not all fields are allowed in all reference types; please refer to the DTD file for this information.

Define in XQuery the following set of queries. For each query the output format is specified.

(1) Give the types of publications in the file.

```
<publications>
    <type>article</type>
    <type>book</type>
    <type>incollection</type>
    <type>inproceedings</type>
    <type>mastersthesis</type>
    <type>phdthesis</type>
    <type>proceedings</type>
</publications>
```

(2) Give the number of publications of each type.

```
<publications>
    <reference type="article">
        <number>263</number>
    </reference>
    <reference type="book">
        <number>9</number>
    </reference>
    ...
</publications>
```

(3) Give the list of authors' names.

```
<authors>
    <name>A. A. Majid</name>
    <name>A. B. M. Shawkat Ali</name>
    <name>A. Bernoussi</name>
    ...
</authors>
```

[1] http://www.informatik.uni-trier.de/~ley/db/

[2] http://dblp.uni-trier.de/xml/

(4) Give the number of authors.

```
<nb_authors>1478</nb_authors>
```

(5) Give the names of authors who are also editors.

```
<authors_editors>
    <name>Ana Paiva</name>
    <name>Chris George</name>
    <name>Du Zhang</name>
    <name>Ed Dawson</name>
    ...
</authors_editors>
```

(6) Give the number of publications by author.

```
<author>
    <name>Ahmed Sameh</name>
    <number>3</number>
</author>
...
```

(7) Give the authors ordered by the number of publications, in descending order.

```
<author>
  <name>Christoph Meinel</name>
  <number>47</number>
</author>

<author>
  <name>Dieter Baum</name>
  <number>21</number>
</author>
...
```

(8) Give the author(s) having the highest number of publications.

```
<prolific_authors>
    <author>
        <name>Morshed U. Chowdhury</name>
        <number>5</number>
    </author>
</prolific_authors>
```

(9) Give for each author the number of publication types.

```
<author>
    <name>Mazeyar E. Makoui</name>
    <pubtypes>1</pubtypes>
</author>
...
```

(10) Give for each author the total number of publications and the number of publications by type.

```
<author>
    <name>A. B. M. Shawkat Ali</name>
    <total>3</total>
    <pubtype type="inproceedings">
        <number>3</number>
    </pubtype>
</author>
```

```
<author>
    <name>A. Bernoussi</name>
    <total>1</total>
    <pubtype type="article">
        <number>1</number>
    </pubtype>
</author>
...
```

(11) Give the list of proceedings that have at least one editor that is also author of at least one article in the proceedings.

```
<proc_editor_author>
    <title>Proceedings of the International (...)</title>
    <title>Advanced Data Mining and Applications,  (...)</title>
    <title>Proceedings of the 5th International  (...)</title>
</proc_editor_author>
```

(12) Give for each author the number of co-authors and the number of joint publications with each of them.

```
<authors_coauthors>
    <author>
        <name>A. B. M. Shawkat Ali</name>
        <coauthors number="4">
            <coauthor>
                <name>M. Delowar Hossain</name>
                <nb_joint_pubs>1</nb_joint_pubs>
            </coauthor>
            ...
        </coauthors>
    </author>
    ...
<authors_coauthors>
```

(13) For each proceedings give its title and the titles of articles appearing in it.

```
<proceedings>
    <proc_title>6th Annual IEEE/ACIS International Conference (...)</proc_title>
    <title>Understanding Consumer Search Activity and Online (...)</title>
    <title>Approximate Element Computational Time for Domain (...)</title>
    <title>Towards a Table Driven XML QoS Aware Transmission Framework.</title>
    ...
</proceedings>
```

(14) Define a function that pretty-print the references of type article in HTML format.

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
        <title>DBLP Articles</title>
    </head>
    <body>
        <h2>DBLP Articles</h2>
        <p>P. Berthon, C. B. Williams. Stages of e-democracy: (...)</p>
        ...
    </body>
</html>
```

(15) Give the distance of author "Lizhu Zhou" with respect to other authors. Two authors that write together a publication have distance 0. If an author a write a publication with author b and if author b write a publication with author c, then a is at distance 1 from c if a and c have not published together.

```
<distance name="Lizhu Zhou">
    <author name="Dengfeng Zhang" distance="2"/>
    <author name="Zhiquan Wang" distance="2"/>
    <author name="Shousong Hu" distance="2"/>
    <author name="Dianhong Wang" distance="1"/>
    <author name="Zhihua Cai" distance="1"/>
    <author name="Xuesong Yan" distance="1"/>
    <author name="Hang Guo" distance="0"/>
    <author name="Liangxiao Jiang" distance="0"/>
</distance>
```

Exercices in XQuery
Querying DBLP Data
Answers

(1) Give the types of publications in the file.

```
<publications>
  {
  for $type in distinct-values(/dblp/*/name())
  order by $type
  return
    <type>{$type}</type>
  }
</publications>
```

(2) Give the number of publications of each type.

```
<publications>
  {
  for $type in distinct-values(/dblp/*/name())
  let $number := count(/dblp/*[name()=$type])
  order by $type
  return
    element reference
      {
      attribute type {$type},
      <number>{$number}</number>
      }
    }
</publications>
```

(3) Give the list of authors' names.

```
<authors>
  {
  for $author in distinct-values(/dblp//author)
  order by $author
  return
    <name>{$author}</name>
  }
</authors>
```

(4) Give the number of authors.

```
let $nb_authors := count(distinct-values(/dblp//author))
return
  <nb_authors>{$nb_authors}</nb_authors>
```

(5) Give the names of authors who are also editors.

```
<authors_editors>
  {
  for $author in distinct-values(/dblp//author)
  where exists (/dblp//editor[.=$author])
  order by $author
  return
    <name>{$author}</name>
  }
</authors_editors>
```

(6) Give the number of publications by author.

```
for $author in distinct-values(/dblp//author)
let $number := count(/dblp/*[author=$author])
order by $author
return
element author
  {
  <name>{$author}</name>,
  <number>{$number}</number>
  }
```

(7) Give the authors ordered by the number of publications, in descending order.

```
let $result := (
  for $author in distinct-values(/dblp//author)
  let $number := count(/dblp/*[author=$author])
  return
  element author
    {
    <name>{$author}</name>,
    <number>{$number}</number>
    }
)
for $i in $result
order by $i/number(number) descending
return
  $i
```

(8) Give the author(s) having the highest number of publications.

```
<prolific_authors>
  {
  let $result := (
    for $author in distinct-values(/dblp//author)
    let $number := count(/dblp/*[author=$author])
    return
    element author
      {
      <name>{$author}</name>,
      <number>{$number}</number>
      }
  )
  let $max := max($result//number)
  for $i in $result[number=$max]
  return $i
  }
</prolific_authors>
```

(9) Give for each author the number of publication types.

```
for $author in distinct-values(/dblp//author)
let $refs := /dblp/*[author=$author]
let $number := count(distinct-values($refs/name()))
order by number($number) descending
return
element author
  {
  <name>{$author}</name>,
```

```
  <pubtypes>{$number}</pubtypes>
  }
```

(10) Give for each author the total number of publications and the number of publications by type.

```
let $pubtypes := distinct-values(/dblp/*/name())
for $author in distinct-values(/dblp//author)
let $total := count(/dblp/*[author=$author])
order by $author
return
element author
  {
  <name>{$author}</name>,
  <total>{$total}</total>,
  for $type in $pubtypes
  let $number := count(/dblp/*[name()=$type and author=$author])
  where $number > 0
  order by $type
  return
    element pubtype
    {
    attribute type {$type},
    <number>{$number}</number>
    }
  }
```

(11) Give the list of proceedings that have at least one editor that is also author of at least one article in the proceedings.

```
<proc_editor_author>
  {
  for $proc in /dblp/proceedings
  where some $editor in $proc/editor satisfies
  exists (/dblp/inproceedings[crossref=$proc/@key]/author[. = $editor])
  return
    $proc/title
  }
</proc_editor_author>
```

(12) Give for each author the number of co-authors and the number of joint publications with each of them.

```
<authors_coauthors>
  {
  for $author in distinct-values(/dblp//author)
  let $refs := /dblp/*[author=$author]
  let $coauthors := distinct-values($refs/author[. ne $author])
  order by $author
  return
  element author
    {
    <name>{$author}</name>,
    element coauthors
      {
      attribute number {count($coauthors)},
      for $coauthor in distinct-values($coauthors)
      let $count := count($refs[author=$coauthor])
```

```
      order by $coauthor
      return
        element coauthor
          {
          <name>{$coauthor}</name>,
          <nb_joint_pubs>{$count}</nb_joint_pubs>
          }
      }
    }
  }
</authors_coauthors>
```

(13) For each proceedings give its title and the titles of articles appearing in it.

```
for $proc in doc('dblp_exc1.xml')/dblp/proceedings
let $inproc := doc('dblp_exc1.xml')/dblp/inproceedings[crossref=$proc/@key]
return
  (
  <proceedings>
    {
    <proc_title>{data($proc/title)}</proc_title>,
    for $i in $inproc
      return $i/title
    }
  </proceedings>
  )
```

(14) Define a function that pretty-print the references of type article in HTML format.

```
declare function local:format-article(
  $article as element(article)?)
  as element()?
{
  let $authors :=
    if (not(empty($article/author))) then
      concat(string-join($article/author, ", "),". ")
    else
      if (count($article/editor)>1) then
        concat(string-join($article/editor, ", "),", editors. ")
      else
        concat(string-join($article/editor, ", "),", editor. ")
  return
    <p>
      {
      concat($authors,$article/title,", ",$article/journal,", ",$article/volume,
        if (not(empty($article/number))) then concat("(",$article/number,"): ") else "",
        if (not(empty($article/pages))) then concat($article/pages,", ") else ", ",
        $article/year,".")
      }
    </p>
};
let $result := (
  for $article  in /dblp/article
  return
    local:format-article($article)
)
return
```

```
(
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <title>DBLP Articles</title>
  </head>
  <body>
    <h2>DBLP Articles</h2>
    {$result}
  </body>
</html>
)
```

(15) Give the distance of author "Lizhu Zhou" with respect to other authors.

```
declare function local:distinct-authors(
  $authors as element()*)
  as element()*
{
  for $i in distinct-values($authors/@name)
  return ($authors[@name=$i])[1]
};
declare function local:distance(
  $authors as element()*, $sofar as element()*)
  as element()*
{
  let $result := (
    for $author in $authors
    let $refs := /dblp/*[author=$author/@name]
    let $coauthors := distinct-values($refs/author[. ne $author])
    let $one_step := (
      for $coauthor in $coauthors
      where not(exists($sofar[@name=$coauthor]))
      return
        element author {
          attribute name {$coauthor},
          attribute distance {$author/@distance+1}
          }
      )
    return local:distance($one_step, ($sofar,$one_step))
    )
  return ($authors, local:distinct-authors($result) )
};

let $author := "Lizhu Zhou"
let $author_elem := element author { attribute name {$author}, attribute distance {-1} }
let $distance := local:distance(($author_elem),($author_elem))
let $distance := remove($distance,1)
return
element distance {
  attribute name {$author},
  $distance
}
```

Another version of the recursive function

```
declare function local:distance(
  $authors as element()*, $sofar as element()*)
  as element()*
{
  let $one_step := local:distinct-authors(
    for $author in $authors
    let $refs := /dblp/*[author=$author/@name]
    let $coauthors := distinct-values($refs/author[. ne $author])
    return
      for $coauthor in $coauthors
      where not(exists($sofar[@name=$coauthor]))
      return
        element author {
          attribute name {$coauthor},
          attribute distance {$author/@distance+1}
          }
    )
  return
    if (count($one_step)>0) then
      ($authors, local:distance($one_step, ($sofar,$one_step)) )
    else $authors
};
```