

# INFO-H-509 : Technologies XML

## TP 4 - Correction

Professor : Stijn Vansummeren

Teaching Assistant : Michaël Waumans

<http://cs.ulb.ac.be/public/teaching/infoh509>

---

### Exercise 1.1

Give the types of publications in the file.

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

### Exercise 1.2

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>
```

### Exercise 1.3

Give the list of author's names.

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

## Exercise 1.4

Give the number of authors.

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

## Exercise 1.5

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>
```

## Exercise 1.6

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>
      }
```

## Exercise 1.7

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
```

## Exercise 1.8

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>
```

## Exercise 1.9

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>
}
```

## Exercise 1.10

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>
    }
}

```

## Exercise 1.11

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>

```

## Exercise 1.12

Define a function that pretty-prints 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>
)

```

```
    </body>
  </html>
)
```