TD8 – XQuery

Considérons le document suivant. Donnez le résultat de l'exécution de chacune des sept requêtes suivantes.

```
<?xml version="1.0"?>
<bib>
  <book year="1994">
    <title>TCP/IP Illustrated</title>
    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price>65.95</price>
  </book>

  <book year="1992">
    <title>Advanced Programming in the Unix environment</title>
    <author><last>Stevens</last><first>W.</first></author>
    <publisher>Addison-Wesley</publisher>
    <price>65.95</price>
  </book>

  <book year="2000">
    <title>Data on the Web</title>
    <author><last>Abiteboul</last><first>Serge</first></author>
    <author><last>Buneman</last><first>Peter</first></author>
    <author><last>Suciu</last><first>Dan</first></author>
    <publisher>Morgan Kaufmann Publishers</publisher>
    <price>39.95</price>
  </book>

  <book year="1999">
    <title>The Economics of Technology and Content for Digital TV</title>
    <editor>
      <last>Gerbarg</last><first>Darcy</first>
      <affiliation>CITI</affiliation>
    </editor>
    <publisher>Kluwer Academic Publishers</publisher>
    <price>129.95</price>
  </book>
</bib>
```

Requête 1

```
declare variable $bib := "biblio.xml" ;
<bib>
  {
  for $b in doc($bib)//book
  where $b/publisher = "Addison-Wesley" and $b/@year > 1991
  return <book year="{$b/@year}"> {$b/title} </book>
  }
</bib>
```

Requête 2

```
let $bib := "biblio.xml"
return
<results>
  {
  for $b in doc($bib)//book,  $t in $b/title, $a in $b/author
  return <result> {$t,$a} </result>
  }
</results>
```

Requête 3

```
let $bib := "biblio.xml"
return
<results>
  {
  let $a := doc($bib)//author
  for $last in distinct-values($a/last),
      $first in distinct-values($a[last=$last]/first)
  order by $last, $first
  return
    <result>
      <author><last>{$last}</last><first>{$first}</first></author>
      {
      for $b in doc($bib)/bib/book
      where some $ba in $b/author satisfies ($ba/last = $last and $ba/first=$first)
      return $b/title
      }
    </result>
  }
</results>
```

Requête 4

```
declare function local:books-by-author ($root, $last, $first) {
  for $b in $root/book
  where some $ba in $b/author satisfies ($ba/last = $last and $ba/first=$first)
  return $b/title
};
let $bib := "biblio.xml"
return
  <results>
    {
    let $a := doc($bib)//author
    for $last in distinct-values($a/last),
        $first in distinct-values($a[last=$last]/first)
    order by $last, $first
    return
    <result>
      <author> <last>{$last}</last><first>{$first}</first> </author>
      {local:books-by-author(doc($bib)/bib, $last, $first)}
   </result>
   }
  </results>
```

Requête 5

```
let $bib := "biblio.xml"
return
  <bib>
    {
    for $b in doc($bib)//book
    where count($b/author) > 0
    return
      <book>
      {$b/title}
      { for $a in $b/author[position() <= 2]
        return $a }
      { if (count($b/author) > 2)
        then <et-al/> else () }
      </book>
    }
  </bib>
```

Requête 6

```
declare function local:books-by-author ($root, $last, $first) {
  for $b in $root/book
  where some $ba in $b/author satisfies ($ba/last = $last and $ba/first=$first)
  return $b/title
};
let $bib := "biblio.xml"
return
  <results>
    {
    let $a := doc($bib)//author
    for $last in distinct-values($a/last),
        $first in distinct-values($a[last=$last]/first)
    order by $last, $first
    return
      <result>
        <author><last>{$last}</last><first>{$first}</first></author>
        <number>{count(local:books-by-author(doc($bib)/bib,$last, $first))}</number>
      </result>
   }
  </results>
```

Requête 7

```
let $bib := "biblio.xml"
return
  <data>
    {
    for $year in distinct-values(doc($bib)//book/@year)
    let $avg := avg(doc($bib)//book[@year=$year]/price/text())
    return <year value="{$year}" avgprice="{$avg}"/>
    }
  </data>
```