Fall, 2005 CIS 550

Database and Information Systems

Homework 4

October 27, 2005; Due November 3 at 1:30 PM

For this homework, you should test your answers using Galax, an XQuery processor. See course home page for information about where to download the Galax system and Galax manual. Alternatively, you can ssh to *eniac-l.seas.upenn.edu* and run *~zives/galax/bin/galax-run* on your query source file.

Problem 1 [70 points]: The XML data files used for this problem are **proc.xml** and **inproc.xml** in *~zives/galax/* on eniac-l. Based on the two XML files, write the following queries in XQuery with the output delimited by the tags ⟨answer⟩ . . . ⟨/answer⟩:

1. Find the distinct **titles** of proceedings in **proc** that have paper information in **inproc**. Result should be nested in **title** tag.

```
<answer>
{
  for $d in distinct-values
  (
     for $u in document("proc.xml")/dblp/proceedings,
         $v in document("inproc.xml")/dblp/inproceedings
     where $u/@key = $v/crossref
     order by $u/title/text()
     return
        $u/title/text()
  )
  return
     <title>{$d}</title>
}
</answer>
```

2. Find the distinct **titles** of proceedings in **proc** that don't have any paper information in **inproc**. Result should be nested in **title** tag.

```
<answer>
{
   for $u in document("proc.xml")/dblp/proceedings
```

```
      let $inprocs:=document("inproc.xml")/dblp/inproceedings
      let $p := count($inprocs[/crossref/text()=$u/@key])
      return
         if ($p=0) then
           <title> {$u/crossref/text() } </title>
         else
           ()
}
</answer>
```

3. Find the **authors** whose papers were most cited. Result should be nested in **author** tag.

```
<answer>
{
  let $inprocs:=document("inproc.xml")/dblp/inproceedings
  let $d := fn:max
  (
     for $a in $inprocs
        return fn:count($a/cite/text())
  )
  for $a in $inprocs
     where fn:count($a/cite/text()) = $d
     return
        for $b in $a/author/text()
           return
             <author> {$b} </author>
}
</answer>
```

4. Find the **authors** who published most papers.

```
<answer>
{
  let $authors:=document("inproc.xml")/dblp/inproceedings/author/text()
  let $d := fn:max
  (
     for $a in fn:distinct-values($authors)
        for $c in fn:count
        (
           for $b in $authors
             where $b = $a
             return $b
        )
```

```
        return $c
  )
  for $a in fn:distinct-values($authors)
    for $c in fn:count
    (
       for $b in $authors
        where $b = $a
        return $b
    )
    where $c=$d
    return $a
}
</answer>
```

5. Find the **year** in which the author in (4) published most paper.(assume answer to (4) is known).

```
<answer>
{
  let $inprocs:=document("inproc.xml")/dblp/inproceedings
  let $years:= for $a in $inprocs
               where $a/author/text() = 'A. R. Dasgupta'
               return $a/year
  let $d := fn:max
  (
     for $a in fn:distinct-values($years)
        for $c in fn:count
        (
           for $b in $inprocs
             where $b/year = $a
               and $b/author/text()='Hector Garcia-Molina'
             return $b
        )
        return $c
  )
  for $a in fn:distinct-values($years)
     for $c in fn:count
     (
       for $b in $inprocs
        where $b/year = $a
          and $b/author/text()='Hector Garcia-Molina'
        return $b
     )
     where $c=$d
```

```
     return $a
}
</answer>
```

6. Find the authors who had papers published in two continuous years. Results should be in the following format:

```
  <answer>
     <publication>
        <author>name1</name>
        <paper>title11</paper>
        <paper>title12</paper>
     </publication>
     <publication>
     ....
     <publication>
  </answer>
```

```
<answer>
{
 let $inprocs:=document("inproc.xml")/dblp/inproceedings
 let $years:= fn:distinct-values($inprocs/year)

 for $a in $years
    let $aut1 := for $b in $inprocs
                 where $b/year = $a
                 return $b
    let $aut2 := for $b in $inprocs
                 where $b/year = $a+1
                 return $b
    for $c in $aut1
       for $d in $aut2
         where $c/author = $d/author
         return
           <publication>
               <author> {$c/author/text()} </author>
               <paper>  {$c/title/text()}</paper>
               <paper>  {$d/title/text()}</paper>
           </publication>

}
</answer>
```

7. Sort papers in **inproc** in ascending order of **year** (use tag ⟨year⟩). When two papers were published on same year, further sort them in ascending order of **booktitle** (use tag ⟨booktitle⟩). Don't lose any data in inproc after sorting. Results should look like

```
<answer>
 <dblp>
   sorted inproceedings here
 </dblp>
</answer>
```

```
<answer>
  <dblp>
  {
    let $inprocs:=document("inproc.xml")/dblp/inproceedings
    for $a in $inprocs
       order by $a/year ascending, $a/booktitle ascending
       return $a
  }
  </dblp>
</answer>
```

Problem 2 [30 points]: Design XML schema for PBAY system.

Sellers(*sellerID*: int, *rating*: char, *email*: string)
Items(*itemID*: int, *type: string*)
Buyers(*buyerID*: int, *email*: string, *address*: string)
Stock(*itemID*: int, *sellerID*: int, *startBid*: float, *quantity*: int, *endingTime*: int)
Purchases(*itemID*: int, *buyerID*: int, *sellerID*: int, *price*: float, *purchaseQuantity*: int, *bidTime*: int)

The entity sets are **Sellers**, **Items** and **Buyers**. Make up some appropriate attributes for each entity set. Each of the entity sets should have 7 attributes. Explain how to encode **Stock** and **Purchases** relationship sets in XML schema.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <xsd:element name="Sellers">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="rating" type="xsd:char"/>
         <xsd:element name="email" type="xsd:string"/>
      </xsd:sequence>
      <xsd:key name="sellerID">
         <xsd:selector xpath=".//Sellers"/>
         <xsd:field xpath="./email"/>
      </xsd:key>
      <xsd:keyref name="itemRef" type="itemID">
        <xsd:selector xpath=".//Items"/>
      </xsd:keyref>
   </xsd:complexType>
 </xsd:element>

 <xsd:element name="Items">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="type" type="xsd:string"/>
      </xsd:sequence>
      <xsd:key name="itemID">
         <xsd:selector xpath=".//Items"/>
      </xsd:key>
   </xsd:complexType>
 </xsd:element>

 <xsd:element name="Buyers">
   <xsd:complexType>
      <xsd:sequence>
```

```
        <xsd:element name="email" type="xsd:string"/>
        <xsd:element name="address" type="xsd:string"/>
      </xsd:sequence>
      <xsd:key name="buyerID">
        <xsd:selector xpath=".//Buyers"/>
      </xsd:key>
      <xsd:keyref name="itemRef" type="itemID">
        <xsd:selector xpath=".//Items"/>
      </xsd:keyref>
      <xsd:keyref name="sellerRef" type="sellerID">
        <xsd:selector xpath=".//Sellers"/>
      </xsd:keyref>
  </xsd:complexType>
</xsd:element>


</xsd:schema>
```