Fall, 2004 CIS 550

Database and Information Systems

Homework 5 Solutions

Problem 1 [25 points]: Write an XML Schema capturing an integrated view of the two schemas.

Answer: Here is one possibility:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType name="authorType">
    <xs:sequence>
      <xs:element name="authorId" type="xs:string"/>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="gender" type="xs:string"/>
      <xs:element name="country" type="xs:string"/>
      <xs:element name="birthDate" type="xs:string"/>
      <xs:element name="deathDate" type="xs:string" minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="email" type="xs:string" minOccurs="0"
                  maxOccurs="1"/>
      <xs:element name="note" type="xs:string" minOccurs="0"
                  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="bookType">
    <xs:sequence>
      <xs:element name="isbn" type="xs:string"/>
      <xs:element name="title" type="xs:string"/>
      <xs:element name="price" type="xs:string" minOccurs="0"/>
      <xs:element name="author" type="xs:string" maxOccurs="unbounded"/>
      <xs:element name="publicationDate" type="xs:string"/>
      <xs:element name="genre" type="xs:string" minOccurs="0"/>
      <xs:element name="salesRank" type="xs:integer" minOccurs="0"/>
      <xs:element name="publisher" type="xs:string" minOccurs="0"/>
      <xs:element name="note" type="xs:string" minOccurs="0"
```

```
                  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="catalogType">
    <xs:sequence>
      <xs:element name="author" type="authorType" minOccurs="0"
                  maxOccurs="unbounded"/>
      <xs:element name="book" type="bookType" minOccurs="0"
                  maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="catalog" type="catalogType"/>
</xs:schema>
```

Problem 2 [25 points]: Write a view in XQuery over the two schemas outputting XML that conforms to your integrated schema.

Answer:

```
define function view() as element* {
  <catalog>
  {  document("/home6/t/tjgreen/public/data-a.xml")/catalog/author
  }
  {  for $a in document("/home6/t/tjgreen/public/data-b.xml")/bookstore/author
     return <author>
        <authorId> { fn:string($a/@authorid) } </authorId>
        <name> { fn:concat($a/fname/text(), $a/lname/text()) } </name>
        { $a/gender,
          $a/country }
        <birthDate> { $a/birthday/text() } </birthDate>
        { $a/email,
          $a/note }
     </author>
  }
  { document("/home6/t/tjgreen/public/data-a.xml")/catalog/book
  }
  { for $b in document("/home6/t/tjgreen/public/data-b.xml")/bookstore/book
    return <book>
      { $b/isbn,
        $b/title,
        $b/price }
      { for $a in $b/authorRef/text()
```

```
        return <author> { $a } </author>
      }
      <publicationDate> { $b/publishbdate/text() } </publicationDate>
    </book>
  }
  </catalog>
}
```

Problem 3 [25 points]: Write the following query in XQuery over your view from the previous problem: Find all books written by authors named Jim Gray.

Answer:

```
for $a in view()/author,
    $b in view()/book,
    $a2 in $b/author
where fn:normalize-space($a/authorId) = fn:normalize-space($a2) and
      fn:normalize-space($a/name) = "Jim Gray"
return $b;
```

Problem 4 [25 points]: Manually write the unfolding of the previous query over Schema *A*.

Answer: Substituting the view body for the `view()` function call, we have:

```
for $a in (
  <catalog>
  {  document("/home6/t/tjgreen/public/data-a.xml")/catalog/author
  }
  {  document("/home6/t/tjgreen/public/data-a.xml")/catalog/book
  }
  </catalog>
  )/author, $b in (
  <catalog>
  {  document("/home6/t/tjgreen/public/data-a.xml")/catalog/author
  }
  {  document("/home6/t/tjgreen/public/data-a.xml")/catalog/book
  }
  </catalog>
  )/book,
  $a2 in $b/author
  where fn:normalize-space($a/authorId) = fn:normalize-space($a2) and
```

```
        fn:normalize-space($a/name) = "Jim Gray"
return $b;
```

Answer: This can be further simplified to:

```
for $a in document("/home6/t/tjgreen/public/data-a.xml")/catalog/author,
    $b in document("/home6/t/tjgreen/public/data-a.xml")/catalog/book,
    $a2 in $b/author
where fn:normalize-space($a/authorId) = fn:normalize-space($a2) and
      fn:normalize-space($a/name) = "Jim Gray"
return $b;
```