CSE 4/560 Project2: XML and XQuery

Deadline: 23:59 Dec. 18, 2020 EST

December 7, 2020

This is an individual project about XML and XQuery. The official XQuery platform is eXistDB (`http://exist-db.org/exist/apps/homepage/index.html`). There is no offline grader in this project, however, partial points will be given in each problems.

1 Creating a XML from a DTD, 2 points

Given following XML document, create a valid XML based on the DTD. Use the website, `https://www.xmlvalidation.com/`, to validate the answer. Save the solution as *q1.xml*

A sample output:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE biblio[
  <!ELEMENT biblio (author*)>
  <!ELEMENT author (name,book+)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT book (title, price)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT price (#PCDATA)>
  <!ATTLIST book year CDATA #REQUIRED>
]>
<!--Write the XML solution here -->
```

2 Creating a DTD from a XML, 2 points

Given following XML document, create a DTD that validate the XML. Use the website, `https://www.xmlvalidation.com/`, to validate the answer. Save the solution as *q2.xml*

A sample output:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Write the DTD solution here -->
<biblio>
```

```
  <book year="...">
     <author>
       <name>...</name>
     </author>
     <title>...</title>
     <price>...</price>
  </book>
  <book year="...">
     <author>
       <name>...</name>
     </author>
     <author>
       <name>...</name>
     </author>
     <title>...</title>
     <price>...</price>
  </book>
</biblio>
```

Note that we treat the ellipsis (...) as a PCDATA.

3 XQuery Problem, 2 points

Given *books.xml*, return all the books and sort the price from lowest to highest. Save the solution as *q3.xq*

A sample output:

```
<results>
  <book year="....">
      <title>Book 1</title>
      <category>...</category>
      <rating>...</rating>
      <price>150</price>
  </book>
  <book year="....">
      <title>Book 2</title>
      <category>...</category>
      <rating>...</rating>
      <price>200</price>
  </book>
  ...
</results>
```

4 XQuery Problem, 2 points

Given *books.xml*, find the names of all Jeff's co-authors and list them together with the titles of books that were coauthored. Save the solution as *q4.xq*

A sample output:

```
<results>
  <book>
      <title>Big data analytics</title>
      <name>Jeff</name>
      <name>Jane</name>
  </book>
  ...
</results>
```

Do NOT return a book if there is no co-authors, e.g. a book is only written by Jeff.

5 XQuery Problem, 3 points

Given *books.xml*, return all the author pairs who have co-authored two or more books together, list their co-authored books' information. Save the solution as *q5.xq*

A sample output:

```
<results>
 <pair>
   <name>author1</name>
   <name>author2</name>
   <book year="the book 1 year">
     <title>the book title 1</title>
     <category>the book 1 category</category>
     <rating>the book 1 rating</rating>
     <price>the book 1 price</price>
   </book>
   <book year="the book 2 year">
     <title>the book title 2</title>
     <category>the book 2 category</category>
     <rating>the book 2 rating</rating>
     <price>the book 2 price</price>
   </book>
 </pair>
 ...
</results>
```

Note that if a pair of authors has been coauthor more than one book, then the output pair needs to contain all the co-authored books.

6 XQuery Problem 4, 3 points

Given *books.xml*, find the average price of each category and the global average price. If a category has higher than global average book price, list the most expensive book(s), for each of those categories. Save the solution as *q6.xq*

A sample output:

```
<results>
 <global_average>10.00</global_average>
 <category id="Name of the Category">
   <category_average>12.00</category_average>
   <book year="the book 1 year">
      <title>the book title 1</title>
      <price>15.00</price>
   </book>
 </category>
 ...
</results>
```

7 From a DTD to another DTD, 6 points

Given the DTDs in question 1 and 2 contain the same information, the main difference is DTD A, the DTD in question 1, uses author nodes to group books they written while the DTD B, the DTD in question 2, uses books nodes to group authors. Write an XQuery accepts a valid XML document respect to the DTD A, the query will translate the input XML document into another output XML document which can be validated by DTD B. Save the solution as *q7.xq*

8 Submission

Failure to comply with the submission specifications will incur penalties for EACH violation.

- What to submit: A zip file has to be submitted through the 'submit_cse460' (if you are CSE460 student) or 'submit_cse560' (if you are CSE560 student) submit script by 12/18/2020 11:59PM EST. Only zip extension will be accepted, please **don't** use any other compression methods such as tar or 7zip. You can submit multiple times, note that **only** the last submission will be kept on the server.
- Zip file naming: Use *ubit_proj2* (**NO SPACE!**) for the filename, for example: *jsmith_proj2.zip*, where *jsmith* is the ubit of submitter.
- Sub-structure of zip file: On unzipping the zip file, there should be a folder named with your ubit *ubit_proj2*, under the folder *ubit_proj2*, there should be seven files: q1.xml, q2.xml, q3.xq, q4.xq, ..., and q7.xq.