BT5110 XML Project

Sem 1, 2018/19

Instructions:

1. This project consist of four questions with a maximum score of 15 marks.
2. The project is due on **November 5 (Monday, 8pm)**.
3. Download the file named `project3-files.zip` from IVLE's Projects/Project3 folder. The unzipped folder contains the following 7 files:
   - item.sql, stock.sql, warehouse.sql, q1.sql
   - q2.xml
   - q3.xml
   - border-area.xml
4. Each team is to submit a zip file named `A1234567X.zip` into the IVLE workbin folder named Project 3 Submissions, where *A1234567X* is the student number of one of the team members. The submitted zip file should contain only these 6 files:
   - q1.sql
   - q1.xml
   - q2.xml
   - q3a.xsl
   - q3b.xsl
   - q4.xsl

   For late submissions, upload your zip file into the IVLE workbin folder named Project 3 Late Submissions. There will be a penalty of -1 mark for each late day until a maximum of three late days. Submissions after November 8 will not be graded and will receive 0 marks.
5. Questions 3 and 4 require writing XSLT stylesheets. If you're a Windows user, we recommend using the **XML Copy Editor** http://xml-copy-editor.sourceforge.net/. If you're a MacOS user, we recommend using the command-line tool **xsltproc**. If you're a Linux user, you can use XML Copy Editor or xsltproc.

---

1. (3 points) Based on your knowledge of SQL and XML, you will learn how to use the SQL/XML features supported in `PostgreSQL` to export relational data into a XML document. Refer to the following documentations of `PostgreSQL`'s SQL/XML features:

   - https://www.postgresql.org/docs/current/static/datatype-xml.html
   - https://www.postgresql.org/docs/current/static/functions-xml.html

   First, create and populate three database tables named `warehouse`, `item`, and `stock` using the provided SQL scripts (warehouse.sql, item.sql, stock.sql).

   Edit the provided SQL script named q1.sql to export the contents of the three populated relational tables into an XML document file named q1.xml. You may use any of the following XML functions for this question: `xmlagg`, `xmlconcat`, and/or `xmlelement`.

   The exported XML document in q1.xml must satisfy the following properties:

   - The document's root element is named `warehouses`.
   - Each warehouse record is represented by an element `warehouse` that is a sub-element of `warehouses`.
   - Each `warehouse` element consists of a sequence of six sub-elements named `id`, `name`, `street`, `city`, `country`, and `items`. The first five sub-elements correspond to the five attributes of the warehouse record. The `items` element contains one `item` sub-element for each of the items stocked by the warehouse.
   - Each `item` element consists of a sequence of five sub-elements named `id`, `im_id`, `name`, `price`, and `qty`. The first four sub-elements correspond to the four attributes of the item record and the `qty` element represents the quantity of that item stocked by the warehouse.
   - The `warehouse` elements are ordered in increasing order of their `id` sub-element values.
   - For each `items` element, its `item` sub-elements are ordered in increasing order of their `id` sub-element values.

   The correct output XML document should be the same as the provided file `q3.xml`.

   Submit the files q1.sql and q1.xml for this question.

2. (3 points) Edit the provided file q2.xml to specify a tight DTD for its XML contents.

   You should validate your completed solution using the tool `xmllint` (http://xmlsoft.org/) or **XML Copy Editor**. For `xmllint`, the command

   ```
   xmllint --noout --valid q2.xml
   ```

   will not return any output if q2.xml is valid with respect to its DTD.

   Submit the file q2.xml for this question.

3. (6 points) This question examines generating HTML output using XSLT stylesheets from the provided XML file `q3.xml`.

   (a) (3 points) Create a XSLT stylesheet named q3a.xsl to generate an HTML document from `q3.xml` that contains the following contents: for each warehouse in Singapore, display the warehouse's id and name, and list the items that are stocked by the warehouse in quantity larger than 975. Display the item's name and the quantity stocked by the warehouse for that item. The HTML document should have the following structure:

```
<html>
        <body>
                <h2>Query 3a</h2>
                <h3>Warehouse warehouseId, warehouseName</h3>
                <h4>Item Lists</h4>
                <ol>
                        <li>ItemName, ItemQty</li>
                        ....
                        <li>ItemName, ItemQty</li>
                </ol>
                ...
                <h3>Warehouse warehouseId, warehouseName</h3>
                <h4>Item Lists</h4>
                <ol>
                        <li>ItemName, ItemQty</li>
                        ....
                        <li>ItemName, ItemQty</li>
                </ol>
        </body>
</html>
```

   (b) (3 points) Create a XSLT stylesheet named q3b.xsl to generate an HTML document from `q3.xml` that contains the following contents: for each warehouse in Singapore, display the id and name of the warehouse and the name of the items available in the largest quantity in that warehouse. The HTML document should have the following structure:

```
<html>
        <body>
                <h2>Query 3b</h2>
                <h3>Warehouse warehouseId, warehouseName</h3>
                <h4>Largest Quantity Items</h4>
                <ol>
                        <li>ItemName, ItemQty</li>
                        ....
                        <li>ItemName, ItemQty</li>
                </ol>
                ....
                <h3>Warehouse warehouseId, warehouseName</h3>
                <h4>Largest Quantity Items</h4>
                <ol>
                        <li>ItemName, ItemQty</li>
                        ....
                        <li>ItemName, ItemQty</li>
                </ol>
        </body>
</html>
```

   Submit the files q3a.xsl and q3b.xsl for this question.

4. (3 points) This question examines how to transform an XML document to a CSV file. Write a XSLT stylesheet named q4.xsl to transform the tabular data contained in the provided XML file named `border-area.xml`, which is derived from https://en.wikipedia.org/wiki/List_of_countries_and_territories_by_border/area_ratio, into a five-column CSV file partially shown below.

```
Rank,Country or territory,Total length of land borders (km),Total surface area (km2),Border/area ratio (km/km2)
1,Vatican City,3.2,0.44,7.2727273
2,Monaco,4.4,2,2.2000000
                ⋮
245,Vanuatu,0,12189,0
246,Wallis and Futuna (France),0,142,0
```

For data transformations involving deletion/insertion of whitespace characters (e.g., line feed, carriage return), you may use XSLT's `<xsl:text> element` and XPath's `translate()` function. For more information about these features, you may refer to the discussions in Jeni Tennison's book on "Translating Strings" (page 197), "Whitepace in Node Trees" (page 290), and "Generating Text Nodes" (page 359).

Submit the file q4.xsl for this question.