

# XML

## **Class**

ECE 568 - Software Engineering of Web Applications

## **Instructor**

Prof. Shiyu Zhou

## **Semester**

Spring 2018

## **Student**

Tina Drew

035006375

## **Assignment**

HW4

## **Submission Date**

February 9, 2018

<b>Question 1</b>	<b>3</b>
Part a:	3
Part b:	4
Part c:	5
Part d:	6
<b>Question 2</b>	<b>6</b>
Part a	6
Part b	6
Part c	6
Part d	7
<b>Question 3</b>	<b>7</b>
Part a	7
<b>References</b>	<b>8</b>

# Question 1

## Part a:

(a) We want to export this data into an XML file. Write a DTD describing the following structure for the XML file:

- there is one root element called products
- the products element contains a sequence of product sub-elements, one for each product in the database
- each product element contains one name, one price, and one description sub-element, and a sequence of store sub-elements, one for each store that sells that product
- each store element contains one name, one phone, and one markup subelement.

Based on an example from [https://www.w3schools.com/xml/xml\\_dtd\\_intro.asp](https://www.w3schools.com/xml/xml_dtd_intro.asp)

Have indicated the specifications based on the parameters set in the question According to the XMT\_DTD lecture from class if e refers to an element

e - exactly one occurrence

e\* - 0 or more occurrences of e

e+ - 1 or more occurrences

So, I have noted the elements with one occurrence such as name and phone with no additional characters after.

```
<?xml version="1.0"?>
<!DOCTYPE products [
<!ELEMENT product (name, price, description, stores*)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT stores (name, phone, Sells)>
<!ELEMENT sid (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT sells (markup)>
<!ELEMENT markup (#PCDATA)>
]>
```

Upon a conversation with the TA, I was informed that I should include all elements in the relational tables. So I have provide a full DTD with all the information below:

```
<?xml version="1.0"?>
<!DOCTYPE products [
<!ELEMENT product (pid, name, price, description, store*)>
<!ELEMENT pid (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT store (sid, name, phones)*>
<!ELEMENT sid (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT sells (sid*, pid*, markup)>
<!ELEMENT sid (#PCDATA)>
<!ELEMENT pid (#PCDATA)>
<!ELEMENT markup (#PCDATA)>
]>
```

## Part b:

(b) Assume the relational database above is accessible through an XML interface that exports it as:

```
<db>
<products>
<row> <pid>p123</pid>
<name>gizmo</name>
<price>22.99</price>
<description>great</description>
</row>
...
</products>
<stores>
<row> ... </row>
...
</stores>
<sells>
<row> ... </row>
```

...

</sells>

</db>

**Write an XQuery expression that, when given an input with this structure, constructs an XML document with the structure described in part (1a)**

```
LET $z = document("products.xml")/Stores
FOR $x IN document("products.xml")/Product
RETURN $x/name, $x/price, $x/description
  {
    FOR $y IN document("products.xml")/Sells
    WHERE $y/pid = $x/pid
    AND $y/sid = $z/sid
    RETURN $z/name, $z/phone, $y/markup
  }
```

## Part c:

**(c) Assuming that you have XML documents with the structure given in part (1a), write an XQuery expression that returns the names and prices of all products that are sold at least at one store with a markup of 25%.**

Based off of Slide 53 of the XQUERY lecture...

```
FOR $y IN doc("products.xml")/sells
  $x IN doc("products.xml")/product
WHERE
  $y/markup = 25%
  AND $x/pid = $y/pid
RETURN {$x/name and $x/price}
```

## ALTERNATIVELY

```
FOR $y in /(products.xml)/sells[markup = 25%],
  $x in (products.xml)/product[pid = $y/pid]
RETURN $x/name and $x/price
```

## Part d:

**(d) Write the same query in SQL over the original relational database schema.**

```
SELECT P.name, P.price
FROM Products P, Sells Se
WHERE Se.markup = 25% AND Se.pid = P.pid
```

## Question 2

Consider XML data given by the following DTD:

```
<!ELEMENT Broadway
((theater | concert | opera)*)>
<!ELEMENT theater (title, address, date, price*)>
<!ELEMENT concert (title, type, date, price*)>
<!ELEMENT opera (title, date, price*)>
```

## Part a

**(a) Return all titles in the XML document**

```
FOR $x IN document(broadway) Broadway//title
RETURN $x
```

## Part b

**(b) Find the addresses of all theaters that have some tickets under \$35 on 11/9/2008 and the titles of their show on that night.**

```
FOR $x IN document(broadway) Broadway//theater
WHERE $x/@price < $35 AND $x/@date = 11/9/2008
RETURN $x/address, $x/title
```

## Part c

**(c) Retrieve all concert titles whose type is chamber orchestra where the average ticket price is at least \$50.**

```
LET $a := avg(document(broadway) Broadway//concert/@price)
FOR $x IN document(broadway) Broadway//theater/concert
WHERE $x/@price >= $a AND $x/@type = chamber orchestra
RETURN $x/title
```

## Part d

(d) Write a query that constructs a new XML document with the following structure:

```
<!ELEMENT groupedByDate (day*)>
<!ELEMENT day (date, show*)>
<!ELEMENT show (title, price*)
```

```
FOR $x in (broadway.xml)
  $d in $x//date
RETURN
  <groupByDate>
    <day>{$d}
      <show>{$x//title}
        {$x//price}
      <show/>
    <day/>
  </groupByDate>
```

## Question 3

The original XML, DTD, and XSL file are provided in the homework instructions.

### Part a

1) Modify the XSL (and maybe the XML) file so that books are displayed in a style like that of the following:

Lamport, Leslie. Latex: **A Document Preparation System** (Addison-Wesley 1986).

Please note that the modified XML, DTD, and XSL files for this question are provided in the attached zip file. To run the XML file:

- 1) Download the zip file
- 2) Extract or unzip files
- 3) Right click on the bib3a.xml
- 4) Select open with <web browser>

I was able to run the file in Microsoft Edge, Internet Explorer, and Safari.

## Part b

2) Add two books and two journals to the XML file, where two of the new items have some information missing.

Below are the two books added to the XML file:

```
<book>
  <author>
    <first>Tina</first>
    <last>Drew</last>
  </author>
  <title>Life, Love, and Understanding</title>
  <year>2014</year>
  <publisher>Causing Elevation Publishing, LLC </publisher>
</book>
```

```
<book>
  <title>Life Application Study Bible</title>
  <year>2012</year>
  <publisher>Tyndale House Publishers, Inc</publisher>
</book>
```

```
<article>
  <author>
    <first>W.G.</first>
    <last>Jacoby</last>
  </author>
  <title>Public attitudes toward government spending</title>
  <year>1994</year>
  <volume>38</volume>
  <page>
    <from>336</from>
    <to>361</to>
  </page>
  <journal>American Journal of Political Science</journal>
</article>
```

```
<article>
  <author>
    <first>U.G.</first>
    <last>Poiger</last>
  </author>
  <title>Rock 'n' roll, female sexuality, and the Cold War Battle over German Identities.</title>
  <year>1996</year>
  <volume>68</volume>
</article>
```



Please note that the modified XML, DTD, and XSL files for this question are provided in the attached zip file. To run the XML file:

- 5) Download the zip file
- 6) Extract or unzip files
- 7) Right click on the bib3b.xml
- 8) Select open with <web browser>

I was able to run the file in Microsoft Edge, Internet Explorer, and Safari.

## Part c

**3) Define a new type of bibliography item for PhD theses in XSL, add two such items to the XML file, and add the appropriate declarations to the DTD code.**

Below is the modified DTD code

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT bib ( (book | article | thesis)+)>
<!ELEMENT book (author, title, year, (address)?, publisher )>
<!ELEMENT article (author, title, year, volume, page, journal) >
<!ELEMENT thesis ( author, title, year)>
<!ELEMENT page (from, to)>
<!ELEMENT author (first, last)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT last (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT journal (#PCDATA)>
<!ELEMENT volume (#PCDATA)>
```

Below are the thesis that were added to the XML code:

```
<thesis>
  <author>
    <first>K.A.</first>
    <last>Knight</last>
  </author>
  <title>Media epidemics: Viral structures in literature and new media</title>
  <year>2011</year>
</thesis>
```

```
<thesis>
  <author>
    <first>Michael P.</first>
    <last>Mendenhall</last>
  </author>
  <title>Measurement of the neutron beta decay asymmetry using ultracold neutrons</title>
  <year>2015</year>
</thesis>
```

Please note that the modified XML, DTD, and XSL files for this question are provided in the attached zip file. To run the XML file:

- 9) Download the zip file
- 10) Extract or unzip files
- 11) Right click on the bib3c.xml
- 12) Select open with <web browser>