Homework 4

Software Engineering of Web Applications

Dhruv Dogra

RUID: 174001491

NetID: dd798

Answer 1-

(a)
```
<!ELEMENT products (product*)>
<!ELEMENT product (name, price, description, store*)>
<!ELEMENT store (name, phone, markup)>
<!ELEMENT name #PCDATA>
<!ELEMENT price #PCDATA>
<!ELEMENT description #PCDATA>
<!ELEMENT phone #PCDATA>
<!ELEMENT markup #PCDATA>
```

(b) Assuming that input.xml is the name of input xml file.
```
<products>
{
        FOR $x IN doc("input.xml")/db/products/row
        RETURN
        <product>
                {$x/name}
                {$x/price}
                {$x/description}
                {
                        FOR $a IN doc("input.xml")/db/sells/row
                        WHERE $x/pid = $a/pid
                        RETURN
                        FOR $s IN doc("input.xml")/db/stores/row
                        WHERE $a/sid=$s/sid
                        RETURN
                        <store>
                                {$s/name}
                                {$s/phones}
                                {$a/markup}
                        </store>
                }
        </product>
}
</products>
```

(c)
```
FOR $a IN doc("input.xml")/products/product
WHERE $x/store/markup >= 25
RETURN <product>
                {$x/name}
                {$x/price}
        <product>
```

(d)
```
SELECT names, prices FROM products
WHERE pid IN
{
        SELECT * FROM sells
        WHERE markup>=25
}
```

Answer 2- Assuming the input xml file is called input.xml

(a)
```
FOR $x IN doc(“input.xml”)/broadway
RETURN //title
```

(b)
```
FOR $x IN document(“input.xml”)broadway/theater/
WHERE $x/price<35 AND $x/date = “11/9/2008”
RETURN
        <theater>
                {$t/address}
                {$t/title}
        <theater>
```

(c)
```
FOR $x IN document(“input.xml”)/broadway/concert/
LET $avg = avg($x/price)
WHERE $x/type = “chamber orchestra” AND $avg >= 50
RETURN
        <concert>
                {$c/title}
        <concert>
```

(d)
```
<groupedByDate>
{
        LET $dates:= FOR $date IN doc("input.xml") return //date
        FOR $d IN distinct-values($dates)
        RETURN
```

```
                <day>
                        <date>{$d}</date>
                        {
                                FOR $b IN doc("input.xml")/broadway/*
                                WHERE $b/date=$d
                                RETURN
                                <show>
                                        {$b/title}
                                        {$b/price}
                                </show>
                        }
                </day>
        }
        </groupedByDate>
```

Answer 3-

(a) Updating XSL only we get the desired results

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        exclude-result-prefixes="xs"
        version="2.0">

<xsl:template name="SplitString">
        <xsl:param name="stringtosplit" />
        <xsl:variable name="words" select="tokenize($stringtosplit, '\s+')"/>
        <xsl:variable name="first" select="for $i in 1 to count(tokenize($stringtosplit, '\s+'))-1
return concat($words[$i], '')"/>
        <xsl:variable name="remaining" select="$words[count(tokenize($stringtosplit, '\s+'))]"/>
        <xsl:value-of select="$remaining" />, <xsl:value-of select="$first"/>.
</xsl:template>

<xsl:template match="/">
        <html>
                <head>
                        <title>Bibliography</title>
                </head>
                <body background="antiquewhite">
                        <center><h2>Bibliography</h2><hr width="90%"/></center>
                        <ul>
                        <xsl:for-each select="bib/book">
                        <p/><li>
```

```
                                        <xsl:call-template name="SplitString">
                                        <xsl:with-param name="stringtosplit" select
="normalize-space(author)"/>
                                        </xsl:call-template>
                                        <b><xsl:value-of select="title"/></b>,
                                        <xsl:value-of select="publisher"/>
                                        <xsl:value-of select="address"/>,
                                        <xsl:value-of select="year"/>.
                              </li>
                    </xsl:for-each>

                    <xsl:for-each select="bib/article">
                              <p/><li>
                                        <xsl:call-template name="SplitString">
                                        <xsl:with-param name="stringtosplit" select
="normalize-space(author)" />
                                        </xsl:call-template>
                                        <xsl:value-of select="title"/>,
                                        <b><xsl:value-of select="journal"/></b>,
                                        <b><xsl:value-of select="volume"/></b>,
                                        pages<xsl:apply-templates select="page"/>
                                        <xsl:value-of select="year"/>.
                              </li>
                    </xsl:for-each>
                    </ul>
          </body>
</html>
</xsl:template>

<xsl:template match="page">
          <xsl:value-of select="from"/>-<xsl:value-of select="to"/>,
</xsl:template>

</xsl:stylesheet>
```

(b) Adding 2 books and 2 journals-

```
     <book>
               <author>Jane Doe</author>
               <title>ABC</title>
               <year>2017</year>
               <publisher>Pearson</publisher>
     </book>
     <book>
```

```
            <author>John Doe</author>
            <title>Test 123</title>
            <year>1998</year>
            <publisher>LabRats</publisher>
    </book>
    <article>
            <author>John Doe</author>
            <title>Paper 1</title>
            <year>2017</year>
            <volume>10</volume>
            <page>200</page>
            <journal>IEEE</journal>
    </article>
    <article>
            <author>Jane Doe</author>
            <title>Paper 2</title>
            <year>1988</year>
            <volume>2</volume>
            <page>100</page>
            <journal>AICTE</journal>
    </article>
```

(c)

Updated DTD:

```
<!ELEMENT bib ( (book | article)+)>
<!ELEMENT book ( author, title, year, (address)?, publisher )>
<!ELEMENT article ( author, title, year, volume, page, journal) >
<!ELEMENT thesis ( author, title, department, university, year) >
<!ELEMENT page (from, to)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT department (#PCDATA)>
<!ELEMENT university (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT journal (#PCDATA)>
<!ELEMENT volume (#PCDATA)>
```

Updated XML:

```
<thesis>
        <author>M Rao</author>
        <title>k-node trees</title>
        <year>2016</year>
        <department>Computer Science</department>
        <university>XYZ</university>
</thesis>
<thesis>
        <author>Joe Samoe</author>
        <title>Sports Head-locks</title>
        <year>2017</year>
        <department>Physical Education</department>
        <university>ABC</university>
</thesis>
```

Updated XSL:

```
<xsl:for-each select="bib/thesis">
<p/><li>
        <xsl:value-of select="author"/>,
        <b><xsl:value-of select="title"/></b>,
        <xsl:value-of select="year"/>,
        <xsl:value-of select="department"/>,
        <xsl:value-of select="university"/>
        </li>
</xsl:for-each>
```