Homework 4

1.Consider the following relational data:

Products:

pid	Name	Price	Description
p123	gizmo	22.99	great
p231	gizmoPlus	99.99	more features
p312	gadget	59.99	good value

Stores:

sid	Name	Phones
s323	Wiz	555-1234
s521	Econo-Wiz	555-6543

Sells:

sid	pid	Markup
s323	p231	10%
s323	p123	25%
s323	p123	15%

(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 sub-element.

(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).

(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%.

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

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*)>
```

(Elements that are not defined are PCDATA.) For each of the questions below write an XPath or an XQuery query.
(In case you need any XQuery built-in functions, you can reference: http://www.xqueryfunctions.com/)

(a) Return all titles in the XML document

(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.

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

(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*)>

3. This homework problem uses the following XML file, DTD file and XSL file:

The XML File

Note that one item is repeated, and that the use of spaces is not entirely uniform.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?xml-stylesheet type="text/xsl" href="bib.xsl"?>
<!DOCTYPE bib SYSTEM "bib.dtd">

<bib>
 <book>
    <author>Leslie Lamport</author>
    <title>Latex: A Document Preparation System </title>
    <year>1986</year>
    <publisher>Addison-Wesley</publisher>
 </book>

 <article>
   <author>David Marr</author>
   <title>Visual information processing</title>
   <year>1980</year>
   <volume>290</volume>
   <page>
     <from>199</from>
     <to>218</to>
   </page>
   <journal>Phil. Trans. Roy. Soc. B</journal>
 </article>

 <article>
    <author>R. K. Clifton </author>
    <title>Breakdown of echo suppression in the precedence
effect</title>
    <year>1987</year>
    <volume>82</volume>
```

```
    <page>
      <from>1834</from>
      <to>1835</to>
    </page>
    <journal>J. Acoust. Soc. Am. </journal>
  </article>

  <book>
    <author>David Marr</author>
    <title>Vision</title>
    <year>1982</year>
    <address> NY </address>
    <publisher>Freeman</publisher>
  </book>

 <article>
   <author>David Marr</author>
   <title>Visual information processing</title>
   <year>1980</year>
   <volume>290</volume>
   <page>
     <from>199</from>
     <to>218</to>
   </page>
   <journal> Phil. Trans. Roy. Soc. B</journal>
 </article>

</bib>
```

The DTD Code

```
<?xml version="1.0" ?>
<!ELEMENT bib ( (book | article)+)>
<!ELEMENT book ( author, title, year, (address)?, publisher )>
<!ELEMENT article ( author, title, year, volume, page, journal) >
<!ELEMENT page (from, to)>
<!ELEMENT author (#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)>
```

The XSL Code

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"
                result-ns="http://www.w3.org/TR/REC-html">
```

```
<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:value-of select="author"/>,
                <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:value-of select="author"/>,
                <b><xsl:value-of select="title"/></b>,
                <em><xsl:value-of select="journal"/></em>,
                <xsl:value-of select="volume"/>,
                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>
```

Answer the following questions.

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).

and journal articles are displayed in a style like that of the following:

> Marr, David. Visual information processing, **Phil. Trans. Roy. Soc. B, 290**, pp.199-218, 1980.

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

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.