

# Homework 4 of ECE568 Web Application

---

Name: Chaoji Zuo RUID: 190003416 netID: CZ296 Date: 3.7

1

(a) export data into XML file

```
<?xml version="1.0"?>
<!DOCTYPE products [
  <!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) XQuery expression to construct 1a

```
for $x in document("db.xml")/db
return
<products>
{
  for $p in $x/products/row
  return
  <product pid="{ $p/pid }">
    { $product/name }
    { $product/price }
    { $product/description }
    {
      for $st in $x/stores/row
      for $se in $x/sells/row
      where $p/pid = $se/pid and $st/sid = $se/sid
      return
      <store sid="{ $st/sid }">
        { $st/name }
        { $st/phones }
        { $se/markup }
      </store>
    }
  </product>
}
</products>
```

(c) XQuery expression for 1a

```
for $p in document("1a.xml")/products/product
let $s := document("1a.xml")/products/product/store[markup = "25%"]
where count($s)>=1
return
<result>
  {$p/name},{ $p/price}
</result>
```

(d) same query in SQL

```
SELECT P.name, P.price
FROM products P, sells S
WHERE P.pid = S.pid AND EXISTS (S.markup = "%25")
```

2

(a) return all titles in the XML document

```
for $x in doc("db.xml")/boordway//title
return $x
```

(b) find the addresses of all theaters

```
for $x in doc("db.xml")/broadway/theater[date = "11/9/2008"]
where some $b in $x/price satisfies data($b) < 35
return
<theater>
  {$x/title}{$x/address}
</theater>
```

(c) retrieve all concert titles

```
for $x in doc("db.xml")/broadway/concert[type = "chamber orchestra"]
where avg(data($x/price)) >=50
return
  $x/title
```

(d) construct a new XML document

```

for $b in document("db.xml")/broadway/
return
<groupByDate>
  {for $x in $b/*
  let $dd :=distinct-values($x/date)
  for $d in $dd
  return
  <day>
    {$d}
    {for $s in document("db.xml")/broadway/*[date = $d]
    return
    <show>
      {$s/title}
      {$s/price}
    </show>}
  </day>}}
</groupByDate>

```

3

1) modify XSL

books:

```

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

<xsl:for-each select="bib/book">
  <p>
    <xsl:value-of select="author"/>.
    <b><xsl:value-of select="title"/></b>
    (<xsl:value-of select="publisher"/>
    <xsl:value-of select="year"/>).
  </xsl:for-each>

```

journal articles:

```

<article>
  <author>Marr, David</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>

<xsl:for-each select="bib/article">
    <p/>
    <xsl:value-of select="author"/>.
    <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"/>.
</xsl:for-each>

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

```

2) add two books and two journals

```

<book>
    <!-- year and address missing-->
    <author>Zhihu Zou</author>
    <title>Machine Learning</title>
    <publiser>Tsuinghua University Publishers</publisher>
</book>

<book>
    <author>Thomas H. Cormen</author>
    <year>2009</year>
    <address>US</address>
    <title>Introduction to Algorithms(Third Edition)</title>
    <publisher>springer</publisher>
</book>

<article>
    <author>Manfred Jaeger</author>
    <title>Counts-of-counts similarity for prediction and search in
relational data</title>
    <year>2019</year>
    <!-- volume missing -->
    <page>
        <from>1</from>
        <to>44</to>
    </page>
    <journal>Data Mining and Knowledge Discovery</journal>
</article>

<article>
    <author>Hassan Ismail Fawaz</author>

```

```
<title>Deep learning for time series classification: a review</title>
<year>2019</year>
<volume>33</volume>
<page>
  <from>1</from>
  <to>47</to>
</page>
<journal>Data Mining and Knowledge Discovery</journal>
</article>
```

### 3) Define new type of bibliography item

add to XSL:

```
<xsl:for-each select="bibliography/PHD_theses"> <p/>
  <li>
    <xsl:value-of select="author"/>,
    <b><xsl:value-of select="title"/></b>,
    <xsl:value-of select="year"/>,
    <xsl:value-of select="department"/>
    <em><xsl:value-of select="university"/></em>.
  </li>
</xsl:for-each>
```

two such item:

```
<PHD_theses>
  <author>Chaoji Zuo</author>
  <title>A deeplearning approach to make the world better</title>
  <year>2019</year>
  <department>School of Engineering</department>
  <university>Rutgers University</university>
</PHD_theses>

<PHD_theses>
  <author>Sun F. L.</author>
  <title>Hello World</title>
  <year>2018</year>
  <department>Scool of Art</department>
  <university>Tongji University</university>
</PHD_theses>
```

add to DTD:

```
<!ELEMENT bibliography( (book | article | PHD_theses)+)>
<!ELEMENT PHD_theses (author, title, year, department, university)>
```

```
<!ELEMENT department (#PCDATA)>  
<!ELEMENT university (#PCDATA)>
```