

CS582 – Midterm 2

11:45 am – 1:00 pm

6 questions (100 points for 25% of final grade)

April 4, 2005

Note: The questions are on page 3, 4, 5, 6, 7, and 8.

Name:

Signature:

Let suppose that we are developing a database for the library. After several discussions, we come up with a database schema which contains the following tables

- BOOKS(ISBN: Integer(9), Title : Char(50), PublisherCode: integer(19), Year: integer(4))
- AUTHORS(SSN: integer(9), Name: Char(50), Address : Address)
- AUTHORSOFF(SSN: integer(9), ISBN: integer(9))

As usual, we underline the primary key for each table. The type of each attribute is written using the following convention:

- $att : Integer(n)$ means that the domain of the attribute att is the set of integers with n digits;
- $att : Char(m)$ means that the domain of the attribute att is the set of strings whose length less than or equal m ;
- $Address$ is a special type that we would like to contain the following information:
 - Street : String(50)
 - Number : Integer(4)

Besides the primary keys, the constraint “Each book has at least one author” should be maintained.

Suppose that we create the following XML file for storing information of the given tables:

```
<Library xmlns="http://www.cs.nmsu.edu/lib.xml">

<Books>
<Book ISBN="123456789">
  <Title> Gone With The Soul </Title>
  <PublisherCode> 001122334 </PublisherCode>
  <Year> 2001 </Year>
</Book>
<Book ISBN="123456789">
  <Title> Gone With The Soul </Title>
  <PublisherCode> 001122334 </PublisherCode>
  <Year> 2004 </Year>
</Book>
<Book ISBN="234567890">
  <Title> XML for Dummy </Title>
  <PublisherCode> 001122334 </PublisherCode>
  <Year> 2001 </Year>
</Book>
<Book ISBN="345678901">
  <Title> XML for Everyone </Title>
  <PublisherCode> 001122335 </PublisherCode>
  <Year> 2005 </Year>
</Book>
</Books>

<Authors>
<Author SSN="912345678">
  <Name> Harry J. Mid </Name>
  <Address><Street>Main St. </Street><Number>1515</Number></Address>
</Author>
<Author SSN="923456789">
  <Name> Amazing Lue </Name>
  <Address><Street>Amador </Street><Number>1771</Number></Address>
</Author>
</Authors>

<AuthorOfs>
<AuthorOf SSN="912345678" ISBN="123456789"/>
<AuthorOf SSN="912345678" ISBN="234567890"/>
<AuthorOf SSN="923456789" ISBN="234567890"/>
<AuthorOf SSN="923456789" ISBN="345678901"/>

```

</AuthorOfs>
</Library>

1. (20 points) Suppose that we would like to create XML schema, which we will refer to as lib.xsd, for the above XML file. Suppose also that only Integer and String are the built-in types. Define the types that we need to have in order to create the schema.

We will need the following types: Book, Author, AuthorOf, Integer9, Char50, Integer19, Integer4, Address, Library

```
<simpleType name=''Integer4''>
  <restriction base=''Integer''>
    <minInclusive value=''1000''/>
    <maxInclusive value=''9999''/>
  </restriction>
</simpleType>

<simpleType name=''Integer19''>
  <restriction base=''Integer''>
    <minInclusive value=''1000000000000000000''/>
    <maxInclusive value=''9999999999999999999''/>
  </restriction>
</simpleType>

<simpleType name=''Integer9''>
  <restriction base=''Integer''>
    <minInclusive value=''100000000''/>
    <maxInclusive value=''999999999''/>
  </restriction>
</simpleType>

<simpleType name=''Char50''>
  <restriction base=''String''>
    <length value=''50''/>
  </restriction>
</simpleType>

<complexType name=''Address''>
  <all>
    <element name=''Street'' type=''Char50''/>
    <element name=''Number'' type=''Integer4''/>
  </all>
</complexType>

<complexType name=''Book''>
  <all>
    <element name=''Title'' type=''Char50''/>
```

```

        <element name='PublisherCode' type='Integer19' />
        <element name='Year' type='Integer4' />
    </all>
    <attribute name='ISBN' type='Integer9' />
</complexType>

<complexType name='Book'>
    <all>
        <element name='Name' type='Char50' />
        <element name='Address' type='Address' />
    </all>
    <attribute name='SSN' type='Integer9' />
</complexType>

<complexType name='AuthorOf'>
    <attribute name='SSN' type='Integer9' />
    <attribute name='ISBN' type='Integer9' />
</complexType>

<complexType name='Library'>
    <sequence>
        <element name='Books' type='Book'
minOccurs='0' maxOccurs='unbounded' />
        <element name='Authors' type='Author'
minOccurs='0' maxOccurs='unbounded' />
        <element name='AuthorOfs' type='AuthorOf'
minOccurs='0' maxOccurs='unbounded' />
    </sequence>
</complexType>

```

2. (20 points) Add to lib.xsd the integrity constraints so that the XML files, if valid with respect to lib.xsd, will contain only valid data about the library.

```
<key name='PKeyForBooks'>
  <selector xpath='Books/Book' />
  <field xpath='@ISBN' />
</key>
```

```
<key name='PKeyForAuthors'>
  <selector xpath='Authors/Author' />
  <field xpath='@SSN' />
</key>
```

```
<key name='PKeyForAuthorOf'>
  <selector xpath='AuthorOfs/AuthorOf' />
  <field xpath='@SSN' />
  <field xpath='@ISBN' />
</key>
```

```
<key name='BookNeedAuthors' refer='PKeyForAuthors'>
  <selector xpath='AuthorOfs/AuthorOf' />
  <field xpath='@SSN' />
</key>
```

3. (10 points)

Draw the XPath document tree for `lib.xml`.

See part of it on the picture on the last page

4. (10 points) What will be returned by the expressions

(a) `//Books` Node 2

(b) `/[1]` Node 1

(c) `//Book[PublisherCode/text()>200000000]` None

(Assign each node of the tree a number and answer this question by listing the nodes).

5. (20 points) Give the result of the following stylesheet when applied on `lib.xml`.

```
<?xml version="1.0">
<xml:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xsl:version="1.0">

<xsl:template match="/">
  <Extract>
    <xsl:apply-templates/>
  </Extract>
</xsl:template>

<xsl:template match="//Books">
  <xsl:for-each select="Book">
    <xsl:value-of select="Title"/>
    <xsl:value-of select="Year"/>
  </xsl:for-each>
</xsl:template>

</xml:stylesheet>

<Extract>
<Book>
  Gone With The Soul 2001
  Gone With The Soul 2004
  XML for Dummy 2001
  XML for Everyone 2005
</Book>
</Extract>
```

6. (20 points) Write XQuery for the following queries:

- (a) List the name and address of authors who have written at least one book.
- (b) List all publisher followed by the titles published by them, i.e., the query should return a document with the following form:

```
<PublisherBooks>

<PublisherBook>
<PublisherCode> publisher 1 </PublisherCode>
  <Title> title 1 of 1 </title>
  <Title> title 2 of 1 </title>
  .....
  <Title> title n of 1 </title>
</PublisherBook>

.....

<PublisherBook>
  <PublisherCode> publisher 2 </PublisherCode>
  <Title> title 1 of 2 </title>
  <Title> title 2 of 2 </title>
  .....
  <Title> title m of 2 </title>
</PublisherBook>

.....

</PublisherBooks>
```

where “publisher j” is the code of a publisher and “title 1 of j,” ..., “title k of j” is the list of all books published by the publisher.

Make sure that your query returns a well-formed XML document.

```
<ListOfAuthorOfAtLeastOneBook>
for $a in document(''http://www.cs.nmsu.edu/library.xml'')
           //Authors
where $a/@ssn in distinct(document(''http://www.cs.nmsu.edu/library.xml'')
                          //AuthorOfs/AuthorOf/@ssn)
return
  ( <AuthorOfAtLeastOneBook>
```

```

        @a/Name @a/Address
    </AuthorOfAtLeastOneBook>
)
</ListOfAuthorOfAtLeastOneBook>

```

A different way:

```

<ListOfAuthorOfAtLeastOneBook>
for $at in distinct(document(''http://www.cs.nmsu.edu/library.xml'')
                    //AuthorOfs/AuthorOf/@ssn)
return
(
    for $a in document(''http://www.cs.nmsu.edu/library.xml'')
                //Authors
    where $a/@ssn = $at
    return
        (<AuthorOfAtLeastOneBook>
         @a/Name @a/Address
         </AuthorOfAtLeastOneBook>
        )
)
</ListOfAuthorOfAtLeastOneBook>

```

```

<PublisherBooks>
for $pcode in distinct(document(''http://www.cs.nmsu.edu/library.xml'')
                       //Books/Book/PublisherCode)
return
(
    <PublisherBook>
    $pcode
    for $b in document(''http://www.cs.nmsu.edu/library.xml'')
                //Books/Book
    where $b/PublisherCode = $pcode
    return
        (<title> $b/title </title>)
    <PublisherBook>
)
</PublisherBooks>

```

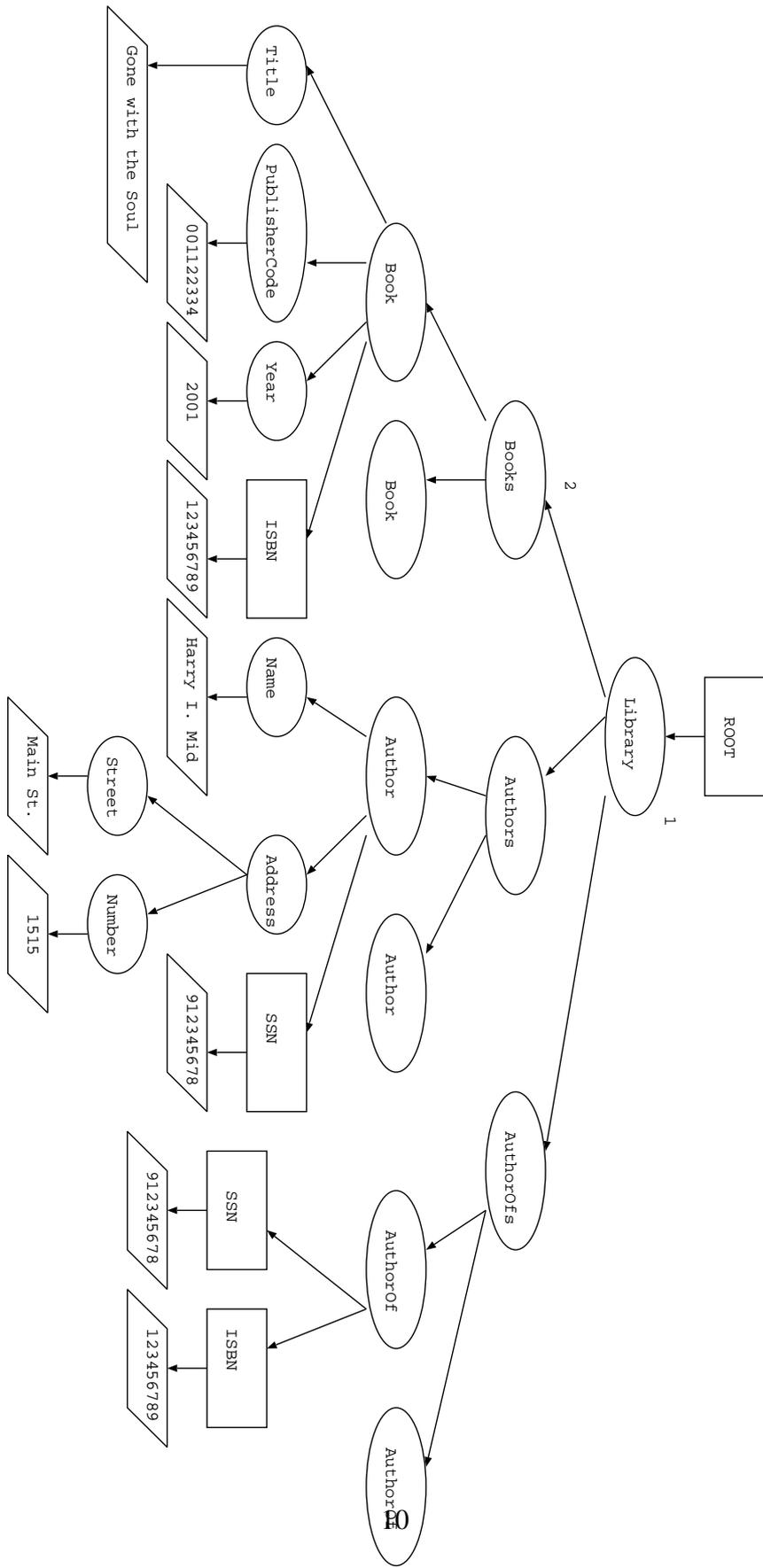


Figure 1: Part of the tree