

Lab 03

XML Schema

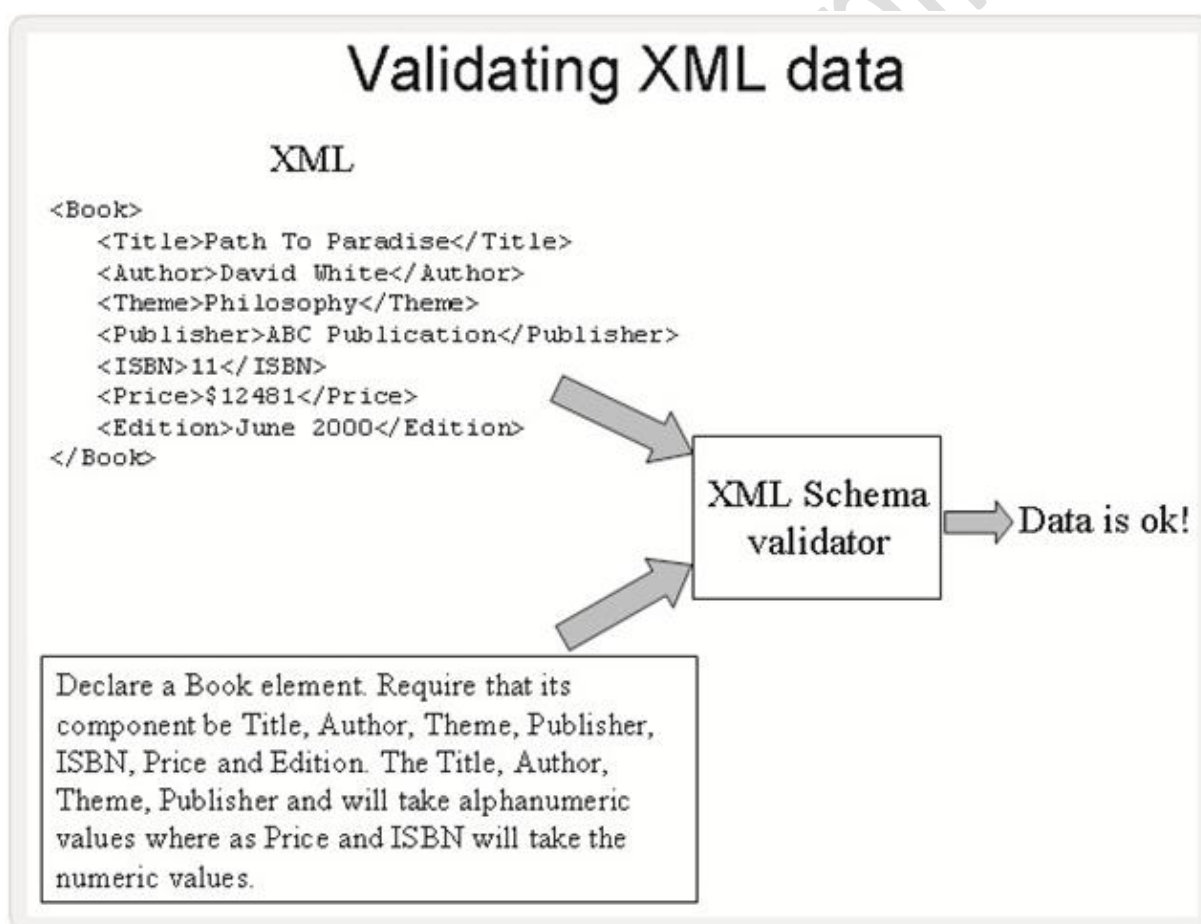
Phần I - Thực hiện trong 120 phút

1.1 Mục tiêu

- ✓ Hiểu biết về schema.
- ✓ Ý nghĩa các thẻ schema.
- ✓ Xây dựng tài liệu schema để định dạng tài liệu XML.

1.2 Thực hiện

Ôn tập kiến thức



1. Schema là gì?

Tài liệu XML sử dụng các cặp thẻ tự định nghĩa nên do đó có thể sẽ bị sai sót trong quá trình viết ví dụ như sai tên thẻ, sai thuộc tính, giá trị lưu trữ không phù hợp quy

tác. Do vậy XML đưa ra khái niệm tài liệu Schema để định nghĩa dữ liệu trong tài liệu XML. Schema được viết với cú pháp XML.

2. Schema vs DTD

- Tương tự như DTD, Schema cũng có vai trò định nghĩa dữ liệu lưu trữ, thẻ cho phép trong tài liệu XML.
- Một tài liệu khi xác thực với Schema phải “Well-formed” và “Valid”.
- Schema là một sự thay thế cho DTD.

```
<xs:element name="note">

  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

</xs:element>
```

Giải thích:

<xs:element name="note"> định nghĩa element được gọi là "note"

<xs:complexType> một element "note" element bên trong có kiểu dữ liệu complex

<xs:sequence> kiểu complex bên trong lại có tuần tự các elements

<xs:element name="to" type="xs:string"> element "to" có kiểu dữ liệu **string** (text)

<xs:element name="from" type="xs:string"> element "from" có kiểu dữ liệu **string**

<xs:element name="heading" type="xs:string"> element "heading" có kiểu dữ liệu **string**

<xs:element name="body" type="xs:string"> element "body" có kiểu dữ liệu **string**

- Schema mạnh mẽ hơn DTD do:
 - ✓ Viết như tài liệu XML

- ✓ Được mở rộng, bổ sung.
- ✓ Hỗ trợ kiểu dữ liệu
- ✓ Hỗ trợ namespace

3. Cấu trúc tài liệu Schema:

```
<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">
...
...
</xs:schema>
```

4. Chỉ định tài liệu XML xác thực bởi tài liệu Schema (*.xsd)

```
<?xml version="1.0"?>

<note xmlns="http://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3schools.com note.xsd">

  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

5. Định nghĩa một Element cơ bản

XML Schema định nghĩa các element (các thẻ) cho tài liệu XML, một element cơ bản có thể chỉ chứa text (nếu nó không chứa bất kỳ THẺ hay THUỘC TÍNH nào bên trong). Cú pháp sẽ như sau:

```
<xs:element name="xxx" type="yyy"/>
```

Trong đó **xxx** là tên của element còn **yyy** là kiểu dữ liệu. XML Schema có các kiểu dữ liệu phổ biến là:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Ví dụ:

```
<lastname>Refsnes</lastname>  
<age>36</age>  
<dateborn>1970-03-27</dateborn>
```

```
<xs:element name="lastname" type="xs:string"/>  
<xs:element name="age" type="xs:integer"/>  
<xs:element name="dateborn" type="xs:date"/>
```

Định nghĩa giá trị MẶC ĐỊNH:

```
<xs:element name="color" type="xs:string" default="red"/>
```

Định nghĩa giá trị CỐ ĐỊNH:

```
<xs:element name="color" type="xs:string" fixed="red"/>
```

6. Thuộc tính của Element

Để định nghĩa thuộc tính của Element thì cần chèn định nghĩa thuộc tính trong cặp thẻ Element (lưu ý bọc các thẻ thuộc tính trong thẻ complexType):



```
<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ten" type="xs:string" maxOccurs="unbounded"/>
      <xs:element name="lang">
        <xs:complexType>
          <xs:attribute name="location1" type="xs:string" default="vietnam" />
          <xs:attribute name="location2" type="xs:string" fixed="vietnam" />
          <xs:attribute name="language">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:pattern value="[A-Z][A-Z]"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Định nghĩa giá trị phải tuân thủ theo quy ước định dạng:

```
<xs:attribute name="code">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z][A-Z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

Giá trị mặc định:

```
<xs:attribute name="lang" type="xs:string" default="EN"/>
```

Giá trị cố định:

```
<xs:attribute name="lang" type="xs:string" fixed="EN"/>
```

Tùy chọn thuộc tính (có hoặc không):

```
<xs:attribute name="lang" type="xs:string" use="optional"/>
```

Bắt buộc phải có:

```
<xs:attribute name="lang" type="xs:string" use="required"/>
```

7. Định nghĩa kiểu dữ liệu phức hợp (Complex) cho Element

Kiểu dữ liệu phức hợp là kiểu có thể chứa các element hoặc/và các thuộc tính. Có 4 kiểu dữ liệu phức hợp:

1. Element rỗng.
2. Element chỉ chứa một element khác.
3. Element chỉ chứa text.
4. Element chứa đồng thời cả element khác và text.

Định nghĩa kiểu dữ liệu Complex đơn giản:

```
<employee>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</employee>
```

Thì định nghĩa schema như sau:

```
<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Cách viết trên là trực tiếp nhưng để tái sử dụng nên viết tham chiếu như sau:

```
<xs:element name="employee" type="personinfo"/>
<xs:element name="student" type="personinfo"/>
<xs:element name="member" type="personinfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Thậm chí có thể phối hợp nhiều complex như sau:

```
<xs:element name="employee" type="fullpersoninfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="fullpersoninfo">
  <xs:complexContent>
    <xs:extension base="personinfo">
      <xs:sequence>
        <xs:element name="address" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="country" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

8. XSD Indicators

Với Indicators, Schema kiểm soát cách mà các element được sử dụng trong tài liệu. Có 7 loại indicators chia nhóm như sau:

Order indicators:

- All
- Choice
- Sequence

Occurrence indicators:

- maxOccurs
- minOccurs

Group indicators:

- Group name
- attributeGroup name

All Indicator:

Quy định các element có thể xuất hiện **thứ tự bất kỳ** và mỗi element chỉ xuất hiện **1 lần**.

```
<xs:element name="person">
  <xs:complexType>
    <xs:all>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:all>
  </xs:complexType>
</xs:element>
```

Choice Indicator:

Quy định chỉ chọn 1 element trong số các element đưa ra. (VD: chỉ employee hoặc member).

```
<xs:element name="person">
  <xs:complexType>
    <xs:choice>
      <xs:element name="employee" type="employee"/>
      <xs:element name="member" type="member"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Sequence Indicator:

Quy định các element có thể **xuất hiện theo thứ tự** và mỗi element chỉ xuất hiện **1 lần**.

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

maxOccurs Indicator:

Quy định các element có thể xuất hiện **tối đa bao lần**. Nếu để **maxOccurs="unbounded"** thì số lần KHÔNG GIỚI HẠN.

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name" type="xs:string"/>
      <xs:element name="child_name" type="xs:string" maxOccurs="10"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

minOccurs Indicator:

Quy định các element có thể xuất hiện **tối thiểu bao nhiêu lần**.

```
<xs:element name="person">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="full_name" type="xs:string"/>
      <xs:element name="child_name" type="xs:string"
        maxOccurs="10" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Group Indicator:

Nhóm các element có liên quan và có thể gọi sử dụng khi cần qua dạng tham chiếu.

```
<xs:group name="persongroup">
  <xs:sequence>
    <xs:element name="firstname" type="xs:string"/>
    <xs:element name="lastname" type="xs:string"/>
    <xs:element name="birthday" type="xs:date"/>
  </xs:sequence>
</xs:group>

<xs:element name="person" type="personinfo"/>

<xs:complexType name="personinfo">
  <xs:sequence>
    <xs:group ref="persongroup"/>
    <xs:element name="country" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Attribute Indicator:

Nhóm các thuộc tính và gọi sử dụng theo dạng tham chiếu

```
<xs:attributeGroup name="personattrgroup">
  <xs:attribute name="firstname" type="xs:string"/>
  <xs:attribute name="lastname" type="xs:string"/>
  <xs:attribute name="birthday" type="xs:date"/>
</xs:attributeGroup>

<xs:element name="person">
  <xs:complexType>
    <xs:attributeGroup ref="personattrgroup"/>
  </xs:complexType>
</xs:element>
```

9. Định nghĩa giới hạn

Giới hạn miền giá trị:

```
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn giá trị cài đặt:

```
<xs:element name="car">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Audi"/>
      <xs:enumeration value="Golf"/>
      <xs:enumeration value="BMW"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Hoặc

```
<xs:element name="car" type="carType"/>

<xs:simpleType name="carType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Audi"/>
    <xs:enumeration value="Golf"/>
    <xs:enumeration value="BMW"/>
  </xs:restriction>
</xs:simpleType>
```

CarType được viết như là phân tử thuộc tính cho element car

Định nghĩa giới hạn dữ liệu bằng mẫu (pattern):

```
<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn **MỘT** ký tự từ **a** -> **z**

```
<xs:element name="initials">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z][A-Z][A-Z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn **BA** ký tự từ **A** -> **Z**

```
<xs:element name="initials">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z][a-zA-Z][a-zA-Z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn **BA** ký tự từ **A** -> **Z** hoặc **a** -> **z**

```
<xs:element name="prodid">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:pattern value="[0-9][0-9][0-9][0-9][0-9]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn chuỗi **NĂM** (5) số

```
<xs:element name="choice">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[xyz]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn **MỘT** trong **BA** ký tự $x - y - z$

```
<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="([a-z])*"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn chuỗi có ít nhất **KHÔNG** ký tự đến **NHIỀU** từ $a-z$

```
<xs:element name="letter">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="([a-z][A-Z])+"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn chuỗi bắt đầu bằng ký tự **THƯỜNG**, sau đó có **ÍT NHẤT MỘT** ký tự **VIẾT HOA**

```
<xs:element name="gender">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="male|female"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Chỉ chấp nhận **MỘT** trong **HAI** giá trị male hoặc female

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9]{8}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn chuỗi gồm **TÁM (8)** ký tự chữ và số

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

Giới hạn chuỗi tối đa 8 ký tự


```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="5"/>
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

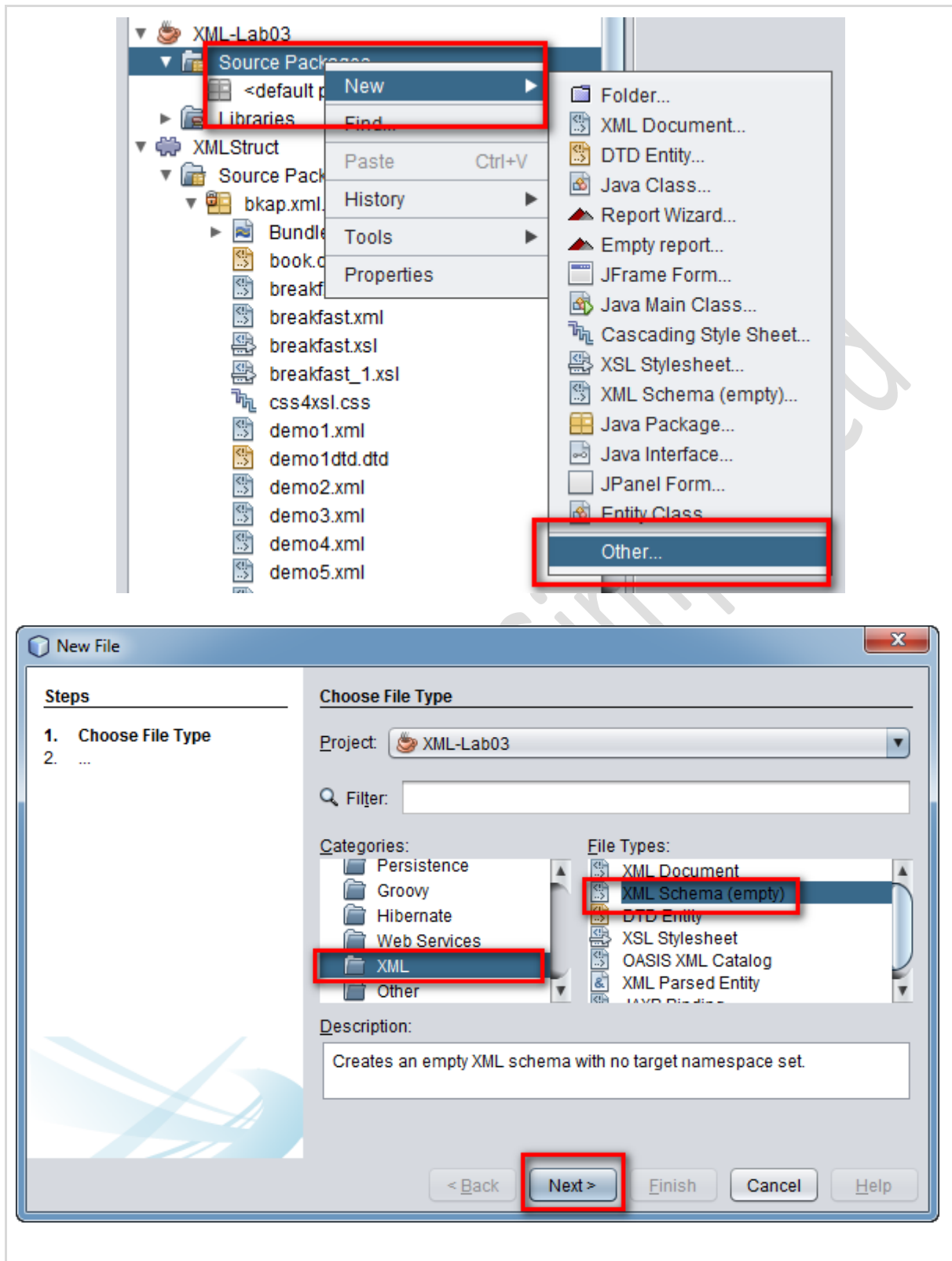
Giới hạn chuỗi tối thiểu 5 và tối đa 8 ký tự

BÀI THỰC HÀNH

Bài thực hành 0: Khởi tạo project sử dụng tài liệu schema để quy định các thành phần trong tài liệu XML. Viết file schema **bookstore.xsd** và file xml **bookstore.xml**. Tài liệu XML có các thành phần như mô tả sau:

```
<!-- bookstore.xml -->
<bookstore>
  <book ISBN="0123456001">
    <title>Java For Dummies</title>
    <author>Tan Ah Teck</author>
    <category>Programming</category>
    <year>2009</year>
    <edition>7</edition>
    <price>19.99</price>
  </book>
</bookstore>
```

Bước 1: tạo project và file schema.



Bước 2: viết file schema

```

1  <?xml version="1.0"?>
2  <!-- Create by MinhVT FC 8.2016 -->
3  <xs:schema version="1.0"
4  |           xmlns:xs="http://www.w3.org/2001/XMLSchema"
5  |           elementFormDefault="qualified">
6  |
7  </xs:schema>

```

File schema tạo bằng tool Netbeans

Định nghĩa thẻ root gốc tên là bookstore:

```

<?xml version="1.0"?>
<!-- Create by MinhVT FC 8.2016 -->
<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <!-- Định nghĩa thẻ root gốc tên là bookstore -->
  <xs:element name="bookstore">

    </xs:element>
  </xs:schema>

```

Định nghĩa tiếp trong thẻ bookstore có các thẻ khác bằng cách đặt trong **xs:complexType**:

```

<?xml version="1.0"?>
<!-- Create by MinhVT FC 8.2016 -->
<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <!-- Định nghĩa thẻ root gốc tên là bookstore -->
  <xs:element name="bookstore">
    <!--

```

Trong thẻ bookstore có chứa các thẻ khác do vậy schema cần phải định nghĩa các thẻ bên trong xs:complexType

```

-->
    <xs:complexType>
      <xs:sequence>

```

```

        </xs:sequence>
      </xs:complexType>

```

```

    </xs:element>
  </xs:schema>

```

Bước 3: định nghĩa thẻ book:

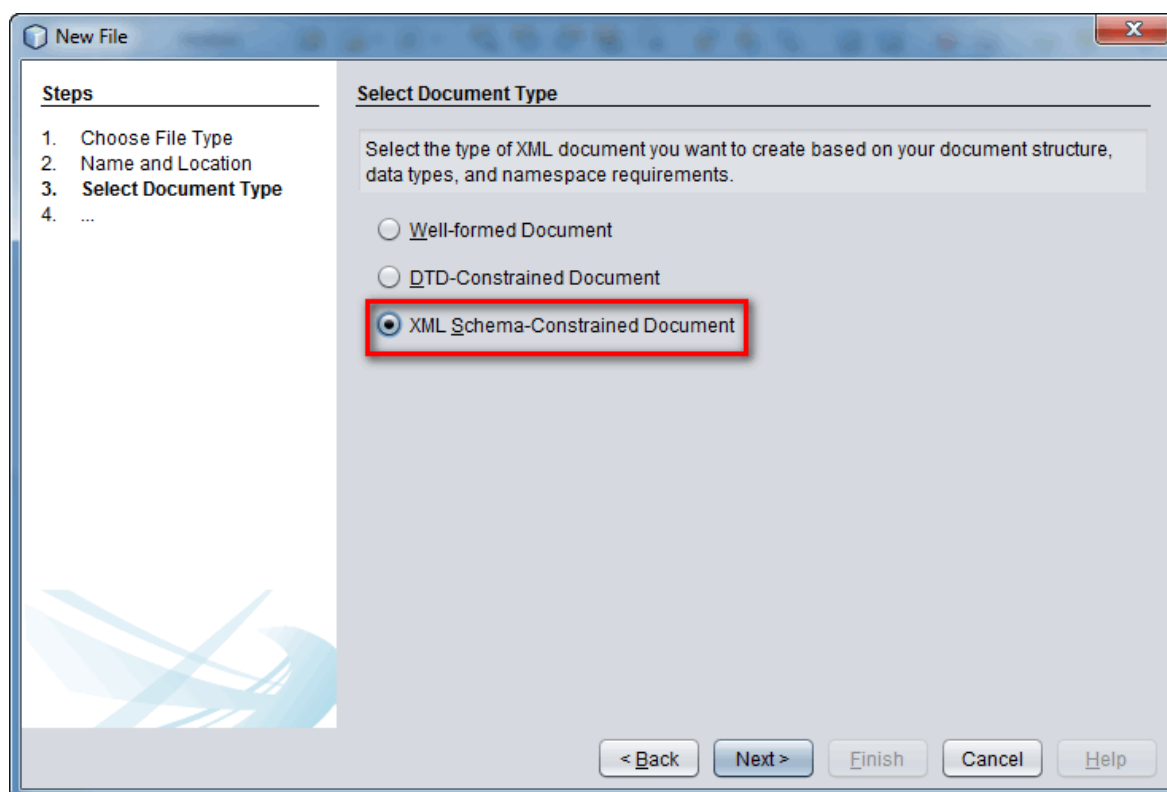
```

<?xml version="1.0"?>

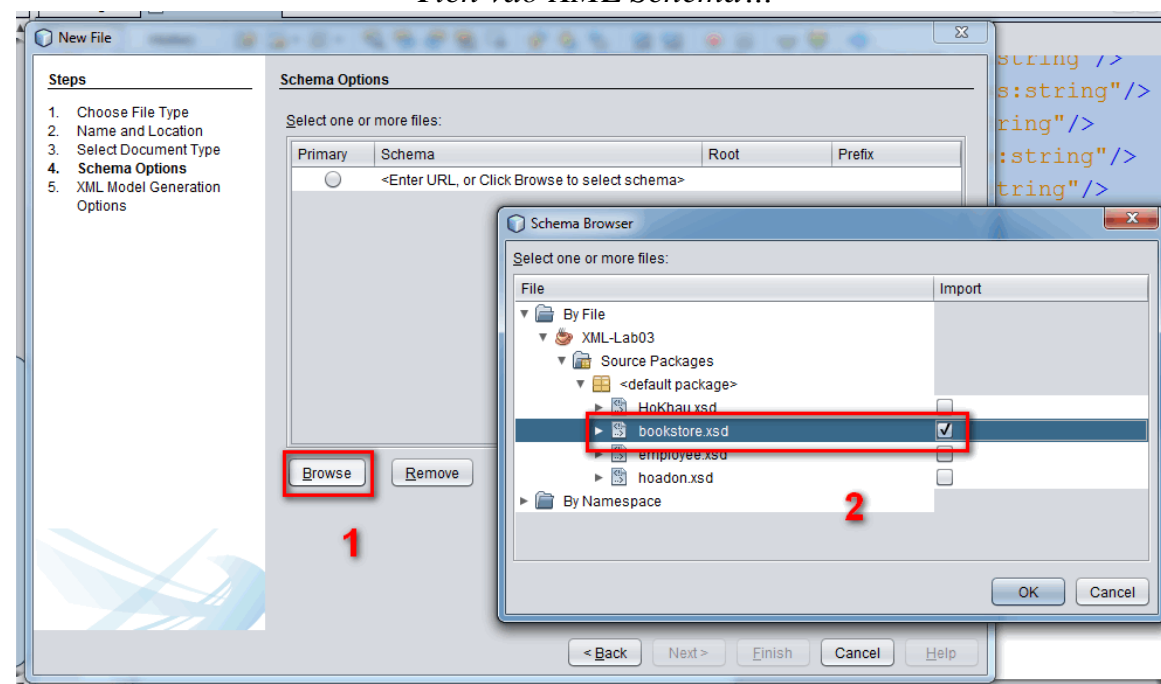
```

```
<!-- Create by MinhVT FC 8.2016 -->
<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <!-- Định nghĩa thẻ root gốc tên là bookstore -->
  <xs:element name="bookstore">
    <!--
      Trong thẻ bookstore có chứa các thẻ khác do vậy schema cần phải
      định nghĩa các thẻ bên trong xs:complexType
    -->
    <xs:complexType>
      <xs:sequence>
        <xs:element name="book">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title" type="xs:string"/>
              <xs:element name="author" type="xs:string"/>
              <xs:element name="category" type="xs:string"/>
              <xs:element name="year" type="xs:string"/>
              <xs:element name="edition" type="xs:string"/>
              <xs:element name="price" type="xs:string"/>
            </xs:sequence>
            <xs:attribute name="ISBN" type="xs:string"/></xs:attribute>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

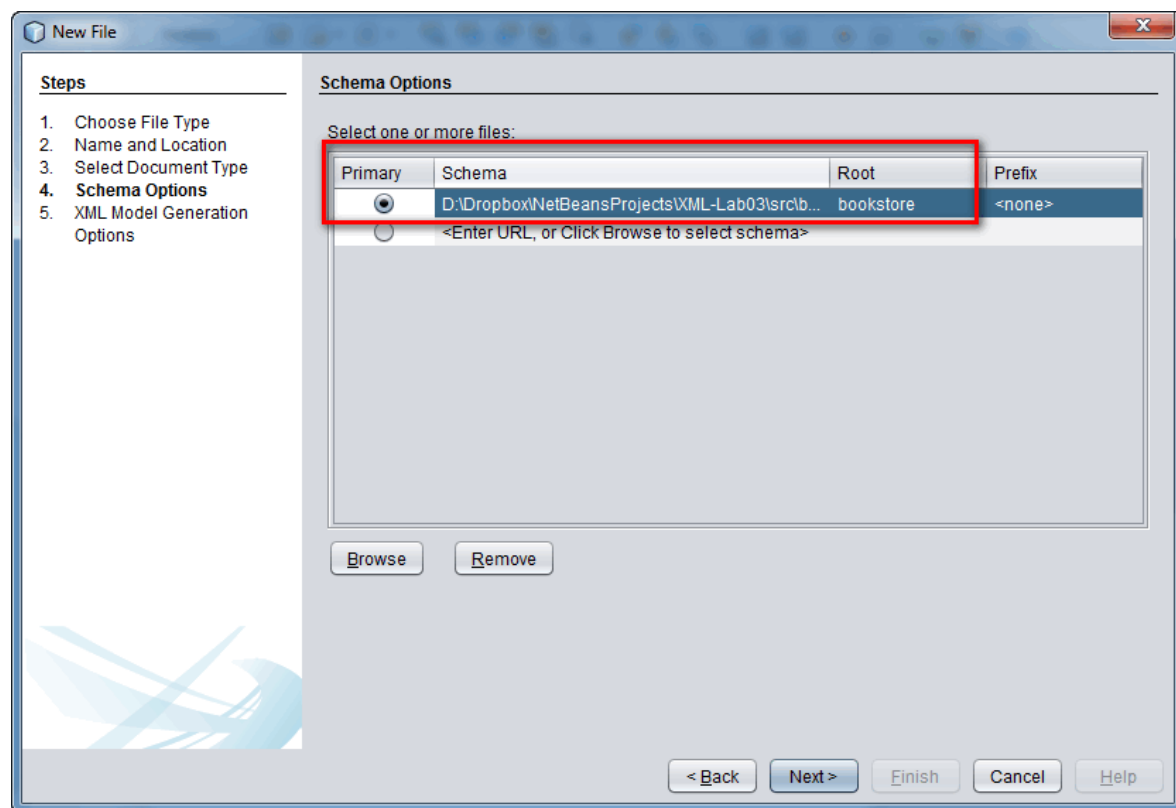
Bước 4: tạo file xml liên kết với schema tạo ở trên



Tích vào XML Schema...



Chọn file schema muốn gắn kết



Tích chọn file rồi bấm next

Viết code xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Create by MinhVT FC 8.2016 -->
<bookstore
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xsi:noNamespaceSchemaLocation='bookstore.xsd'>
  <book ISBN="123456">
    <title>XML Đơn giản</title>
    <author>Quế Ngọc Hải</author>
    <category>Lập trình</category>
    <year>2016</year>
    <edition>Đoàn Nguyên Đức</edition>
    <price>$69</price>
  </book>
</bookstore>
```

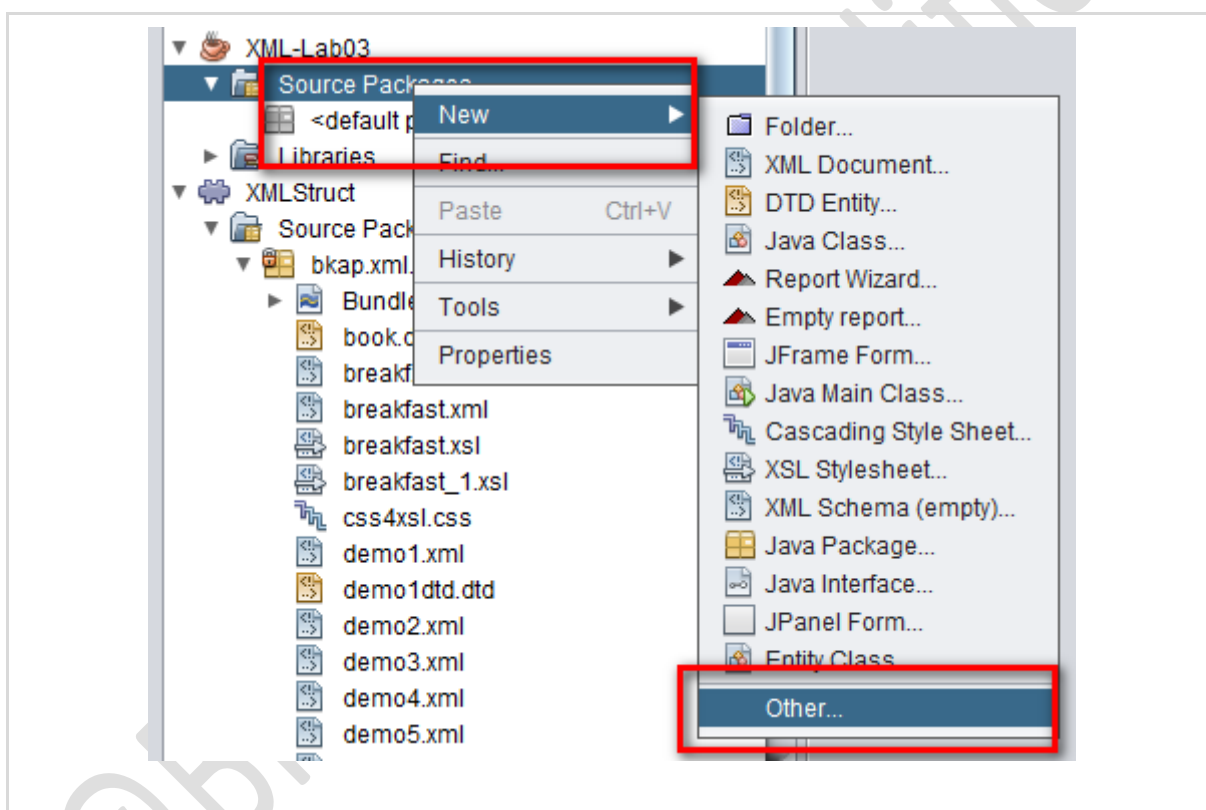
Bước 5: hiển thị lên trình duyệt

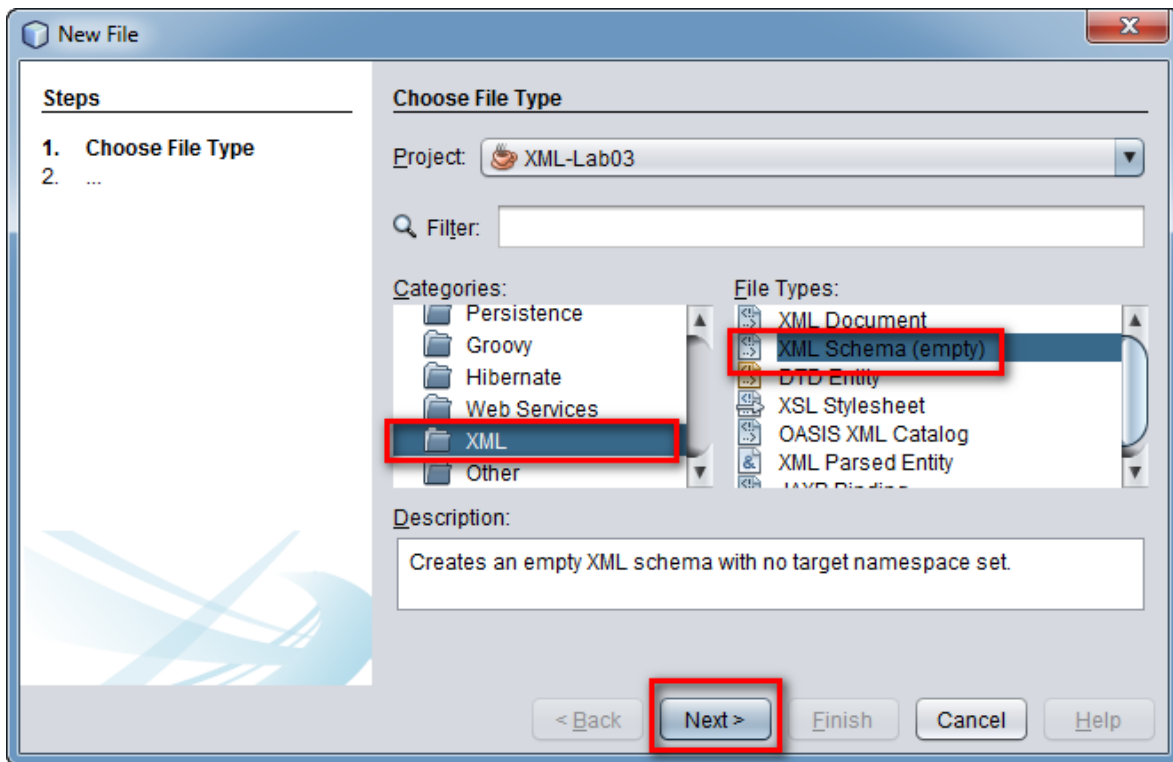
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<!-- Create by MinhVT FC 8.2016 -->
<bookstore xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="bookstore.xsd">
  <book ISBN="123456">
    <title>XML Đơn giản</title>
    <author>Quế Ngọc Hải</author>
    <category>Lập trình</category>
    <year>2016</year>
    <edition>Đoàn Nguyên Đức</edition>
    <price>$69</price>
  </book>
</bookstore>
```

Bài thực hành 1: Khởi tạo project sử dụng tài liệu schema để định dạng tài liệu XML.
Viết file schema hoadon.xsd và file xml hoadon.xml.

Bước 1: tạo project và file schema.





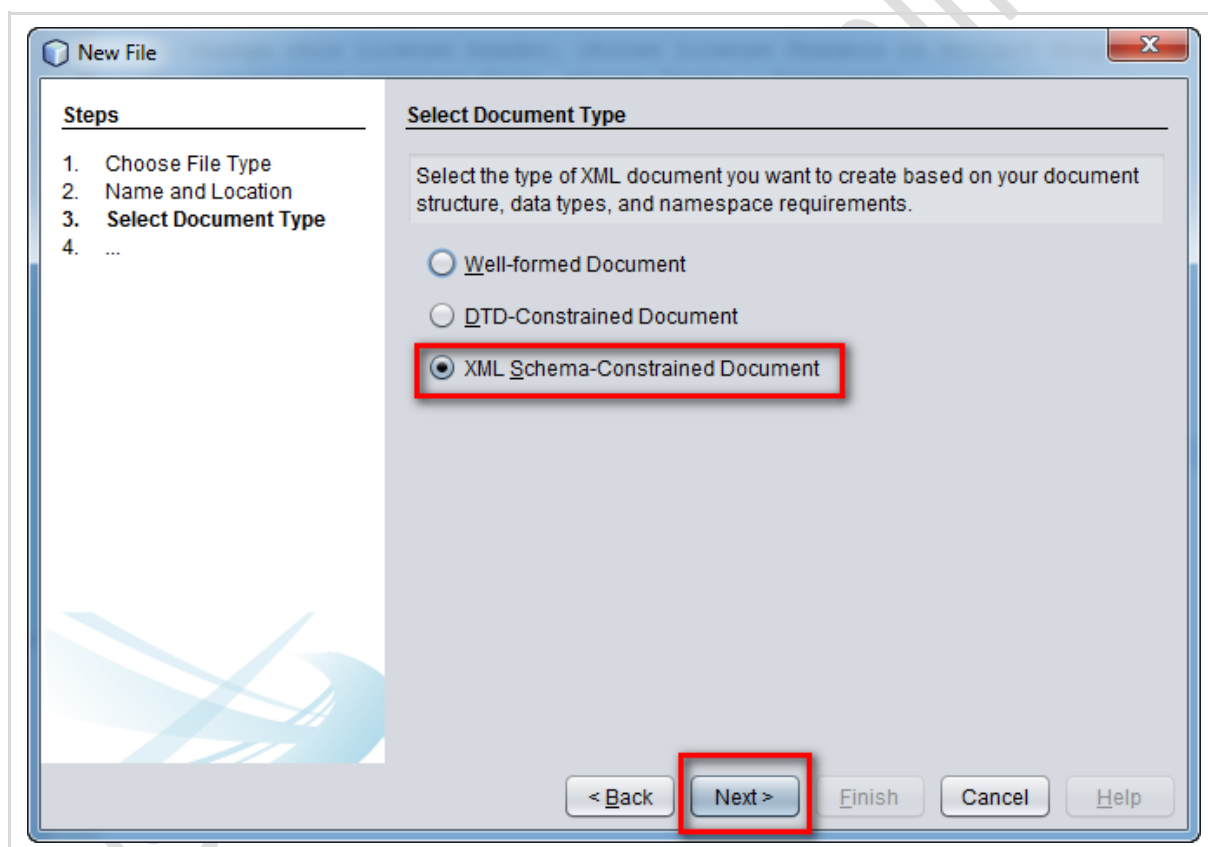
Bước 2: viết mã lệnh schema.

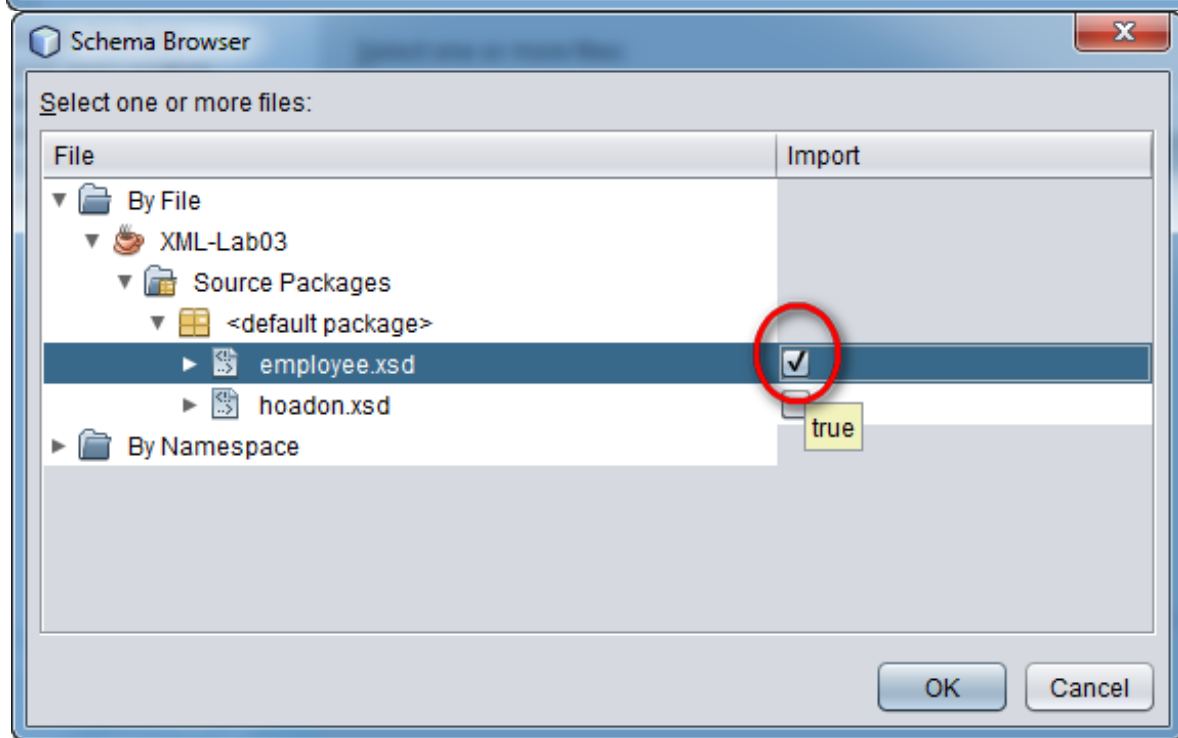
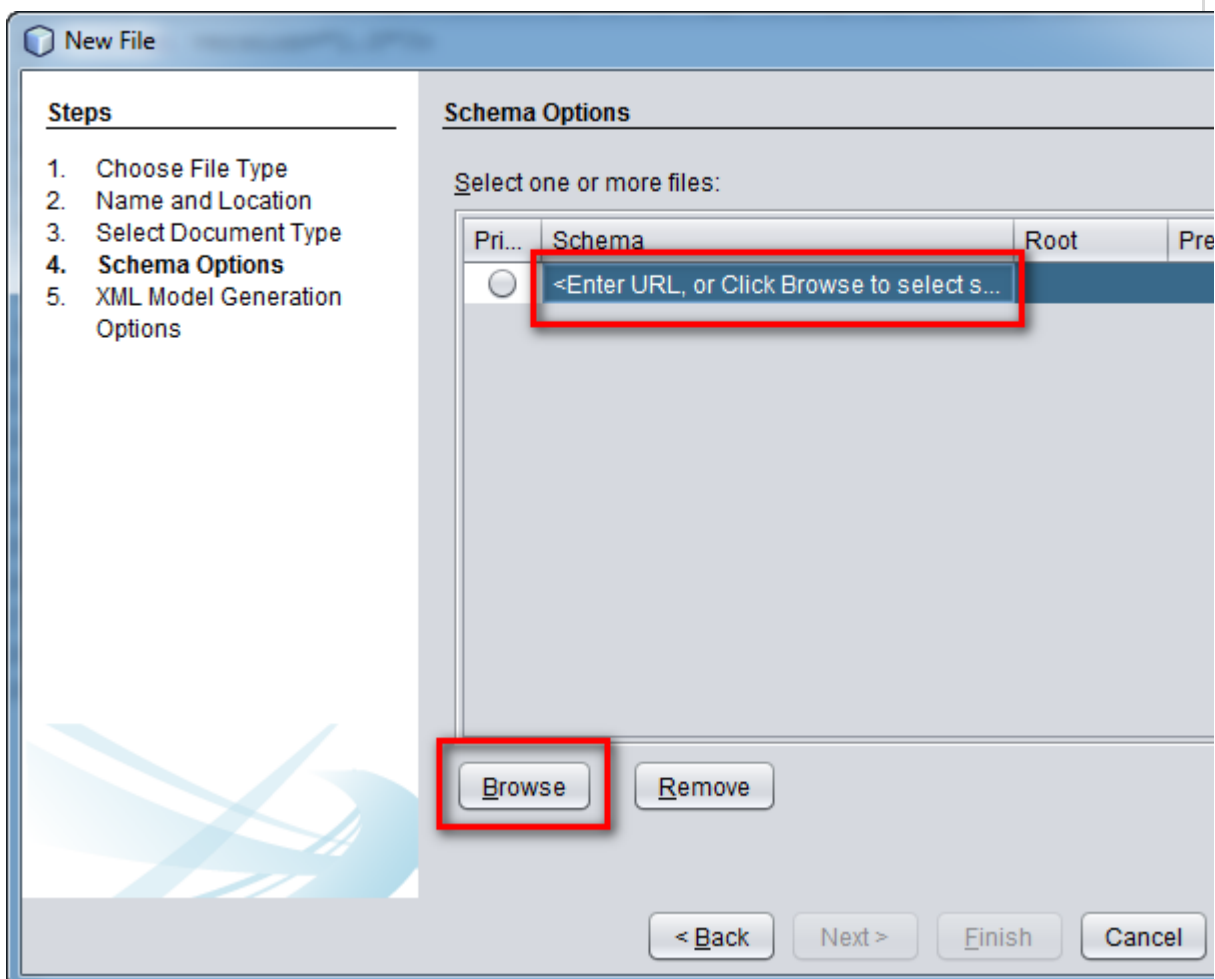
```
<?xml version="1.0"?>
<!--
Create by MinhVT FC ver 6.2016
-->
<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="tblhoadon">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="id" type="xs:integer" />
        <xs:element name="ten">
          <!-- Chỉ nhận dữ liệu chuỗi thông thường (ko ký tự đặc biệt) -->
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:pattern value="([a-zA-Z])*"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="soluong">
          <!-- Chỉ nhận số nguyên có giá trị 0 > 100 -->
          <xs:simpleType>
```

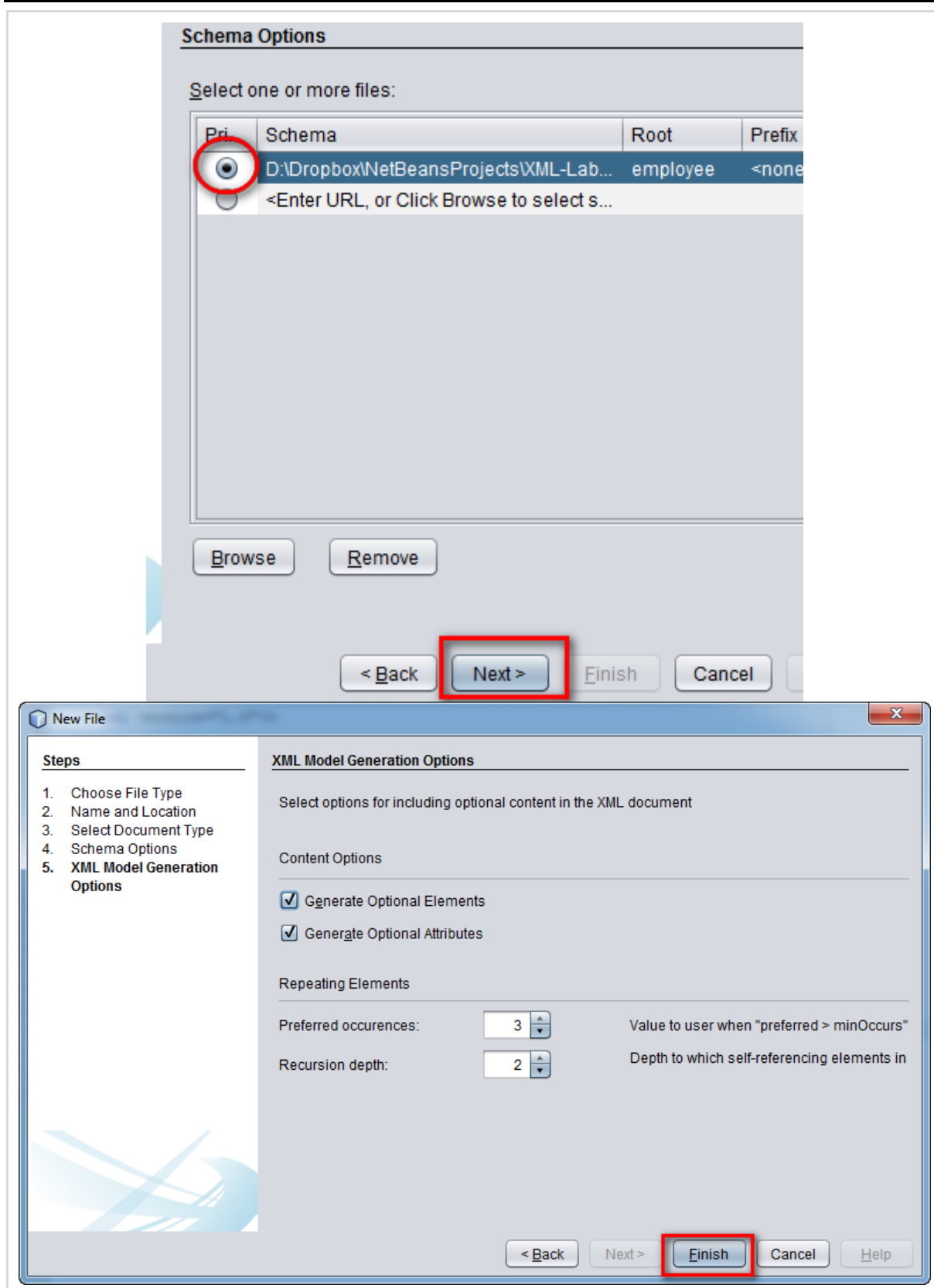


```
<xs:restriction base="xs:integer">  
  <xs:minInclusive value="0"/>  
  <xs:maxInclusive value="100"/>  
</xs:restriction>  
</xs:simpleType>  
</xs:element>  
<xs:element name="giatri" type="xs:float"/>  
<xs:element name="ngaylap" type="xs:date"/>  
</xs:sequence>  
</xs:complexType>  
</xs:element>  
</xs:schema>
```

Bước 3: tạo file xml ràng buộc với Schema.







Bước 4: viết file xml.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!--
Create by MinhVT FC ver 6.2016
-->
<tblhoadon
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xsi:noNamespaceSchemaLocation='hoadon.xsd'>
  <id>123</id>
  <ten>DSFDS</ten>
  <soluong>50</soluong>
  <giatri>80</giatri>
  <ngaylap>2016-06-15</ngaylap>
</tblhoadon>
```

Bước 5: validate file xml.

The screenshot shows the NetBeans IDE interface. The XML file 'hoadon.xml' is open, displaying the XML code. A context menu is open over the code, with the 'Validate XML' option selected. The output window at the bottom shows the XML validation process.

XML Code:

```
<tblhoadon
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xsi:noNamespaceSchemaLocation='hoadon.xsd'>
  <id>123</id>
  <ten>DSFDS</ten>
  <soluong>50</soluong>
  <giatri>80</giatri>
  <ngaylap>2016-06-15</ngaylap>
</tblhoadon>
```

Context Menu Options:

- Format (Alt+Shift+F)
- View
- Check XML (Alt+F9)
- Validate XML (Alt+Shift+F9)**
- Generate DTD...
- XSL Transformation...
- Send To...
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Tools
- Select in Projects

Output - XML check:

```
XML validation started.
Checking file:/D:/Dropbox/NetBeansProjects/XML-Lab03/src/hoadon.xml...
Referenced entity at "file:/D:/Dropbox/NetBeansProjects/XML-Lab03/src/hoadon.xsd".
XML validation finished.
```

Bài thực hành 2: Tạo tài liệu XML lưu trữ thông tin hồ sơ hồ khẩu của các hộ dân trong tổ dân phố. Trong hồ khẩu của từng nhà chúng ta cần lưu các thông tin sau:

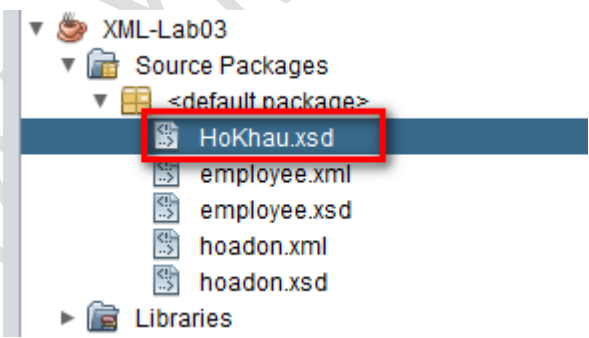
- Thông tin về căn nhà bao gồm: **số nhà, tên đường, phường (xã), quận (huyện), thành phố (Tỉnh)**
- Thông tin về các thành viên trong nhà bao gồm: **chủ hộ, thành viên**. Trong đó chủ hộ cần lưu các thông tin sau: **số CMND, họ, tên, giới tính, ngày sinh, nguyên quán**, các thành viên ngoài các thông tin như chủ hộ, chúng ta cần lưu thêm thông tin về **mối quan hệ** với chủ hộ.

Xác định các ràng buộc, giới hạn dữ liệu như sau:

- Số nhà: chuỗi tối đa 20 ký tự
- Tên đường, phường (Xã), quận (huyện), thành phố (tỉnh): chuỗi tối đa 50 ký tự
- Số CMND: 9 ký tự số
- Họ: chuỗi tối đa 15 ký tự
- Tên: tối đa 35 ký tự
- Giới tính: Nam hay nữ
- Ngày sinh: kiểu ngày tháng
- Nguyên quán: chuỗi tối đa 30 ký tự
- Quan hệ: chuỗi tối đa 30 ký tự

Viết file xml **HoKhau.xml** có ràng buộc với file Schema **HoKhau.xsd**

Bước 1: tạo project và file schema và xác định kiểu dữ liệu dùng chung.



Code:

```
<?xml version="1.0"?>
<!--
Create by Minh Vũ FC 6.2016
-->
<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
```

</xs:schema>

Bài toán trên cần quy định dữ liệu **ĐỊA CHỈ** và **THÔNG TIN THÀNH VIÊN**, do đó cần tạo riêng 2 kiểu dữ liệu này để tiện sử dụng, tránh viết lại nhiều lần.

<?xml version="1.0"?>

<!--

Create by Minh Vũ FC 6.2016

-->

<xs:schema version="1.0"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

elementFormDefault="qualified">

<!-- Định nghĩa kiểu dữ liệu ĐỊA CHỈ có các thẻ -->

<xs:complexType name="diachi">

<xs:sequence>

</xs:sequence>

</xs:complexType>

<!-- Định nghĩa kiểu dữ liệu THÔNG TIN THÀNH VIÊN có các thẻ -->

<xs:complexType name="ttThanhVien">

<xs:sequence>

</xs:sequence>

</xs:complexType>

</xs:schema>

Bước 2: định nghĩa cụ thể các thẻ trong kiểu dữ liệu **diachi**, dữ liệu nhập giới hạn 20 ký tự.

<?xml version="1.0"?>

<!--

Create by Minh Vũ FC 6.2016

-->

<xs:schema version="1.0"

xmlns:xs="http://www.w3.org/2001/XMLSchema"

elementFormDefault="qualified">

<!-- Định nghĩa kiểu dữ liệu ĐỊA CHỈ có các thẻ -->

<xs:complexType name="diachi">

<xs:sequence>

<xs:element name="sonha" type="length20"/>

<xs:element name="duong" type="length20"/>

<!-- Phường/Xã -->

```

    <xs:choice>
      <xs:element name="phuong" type="length20"/>
      <xs:element name="xa" type="length20"/>
    </xs:choice>

    <!-- Quận/Huyện -->
    <xs:choice>
      <xs:element name="quan" type="length20"/>
      <xs:element name="huyen" type="length20"/>
    </xs:choice>

    <!-- Thành phố/Tỉnh -->
    <xs:choice>
      <xs:element name="thanhpho" type="length20"/>
      <xs:element name="tinh" type="length20"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<!-- Định nghĩa kiểu dữ liệu chuỗi giới hạn 20 ký tự -->
<xs:simpleType name="length20">
  <xs:restriction base="xs:string">
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>

<!-- Định nghĩa kiểu dữ liệu THÔNG TIN THÀNH VIÊN có các thẻ -->
<xs:complexType name="ttThanhVien">
  <xs:sequence>

    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

Lưu ý: để tái sử dụng code, tạo thêm kiểu dữ liệu simpleType định nghĩa giới hạn chuỗi chỉ tối đa 20 ký tự

Bước 3: định nghĩa cụ thể các thẻ trong kiểu dữ liệu **ttThanhVien**, dữ liệu nhập giới hạn với các yêu cầu sau:

- soCMND: là chuỗi số gồm 9 chữ số
- ho, ten, nguyenquan, moiquanhe: giới hạn 20 ký tự
- gioitinh: chỉ một trong 2 giá trị Nam/Nữ (dùng enumeration hoặc pattern)
- ngaysinh: kiểu date

Định nghĩa kiểu dữ liệu cơ bản cho thành viên:

```
<!-- Chuỗi số gồm 9 số -->
<xs:simpleType name="soCMND">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{9}" />
  </xs:restriction>
</xs:simpleType>

<!-- Giới hạn giá trị là Nam/Nữ -->
<xs:simpleType name="gioiTinh">
  <xs:restriction>
    <xs:pattern value="Nam|Nữ"/>
  </xs:restriction>
</xs:simpleType>
```

Định nghĩa dữ liệu thành viên:

```
<!-- Định nghĩa kiểu dữ liệu TT thành viên (Chủ hộ) -->
<xs:complexType name="ttCoBan">
  <xs:sequence>
    <xs:element name="soCMND" type="soCMND"/>
    <xs:element name="ho" type="length20"/>
    <xs:element name="ten" type="length20"/>
    <xs:element name="nguyenquan" type="length20"/>
    <xs:element name="gioiTinh" type="gioiTinh"/>
    <xs:element name="ngaySinh" type="xs:date"/>
  </xs:sequence>
</xs:complexType>
```

Đối với thành viên thì cần bổ sung thêm thẻ mối quan hệ (với chủ hộ)

```
<!-- Định nghĩa kiểu dữ liệu TT thành viên đầy đủ (thành viên khác) -->
<xs:complexType name="ttDayDu">
  <xs:complexContent>
    <xs:extension base="ttCoBan">
      <xs:sequence>
        <xs:element name="moiQuanHe" type="length20"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Bước 4: định nghĩa các thẻ cho một element hộ khẩu.

Do có nhiều hộ khẩu trong khu vực dân cư nên cần tạo thẻ DSHoKhau có nhiều thẻ

HoKhou bên trong:

```
<!-- Định nghĩa root gốc là thẻ DSHoKhou có nhiều HoKhou -->
<xs:element name="DSHoKhou">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="HoKhou" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="DiaChi" type="diachi"/>
            <xs:element name="ChuHo" type="ttCoBan"/>
            <xs:element name="ThanhVien" type="ttDayDu"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Bước 5: toàn bộ file xsd.

```
<?xml version="1.0"?>
<!--
Create by Minh Vũ FC 6.2016
-->
<xs:schema version="1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <!-- Định nghĩa kiểu dữ liệu ĐỊA CHỈ có các thẻ -->
  <xs:complexType name="diachi">
    <xs:sequence>
      <xs:element name="sonha" type="length20"/>
      <xs:element name="duong" type="length20"/>

      <!-- Phường/Xã -->
      <xs:choice>
        <xs:element name="phuong" type="length20"/>
        <xs:element name="xa" type="length20"/>
      </xs:choice>

      <!-- Quận/Huyện -->
      <xs:choice>
        <xs:element name="quan" type="length20"/>
```

```
<xs:element name="huyen" type="length20"/>
</xs:choice>

<!-- Thành phố/Tỉnh -->
<xs:choice>
  <xs:element name="thanhpho" type="length20"/>
  <xs:element name="tinh" type="length20"/>
</xs:choice>
</xs:sequence>
</xs:complexType>

<!-- Định nghĩa kiểu dữ liệu chuỗi giới hạn 20 ký tự -->
<xs:simpleType name="length20">
  <xs:restriction base="xs:string">
    <xs:maxLength value="20"/>
  </xs:restriction>
</xs:simpleType>

<!-- Định nghĩa kiểu dữ liệu THÔNG TIN THÀNH VIÊN có các thẻ -->
<xs:complexType name="ttThanhVien">
  <xs:sequence>

    </xs:sequence>
  </xs:complexType>

  <!-- Chuỗi số gồm 9 số -->
  <xs:simpleType name="soCMND">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{9}" />
    </xs:restriction>
  </xs:simpleType>

  <!-- Giới hạn giá trị là Nam/Nữ -->
  <xs:simpleType name="gioiTinh">
    <xs:restriction base="xs:string">
      <xs:pattern value="Nam|Nữ"/>
    </xs:restriction>
  </xs:simpleType>

  <!-- Định nghĩa kiểu dữ liệu TT thành viên (Chủ hộ) -->
  <xs:complexType name="ttCoBan">
    <xs:sequence>
      <xs:element name="soCMND" type="soCMND"/>
      <xs:element name="ho" type="length20"/>
    </xs:sequence>
  </xs:complexType>

```

```

    <xs:element name="ten" type="length20"/>
    <xs:element name="nguyenquan" type="length20"/>
    <xs:element name="gioiTinh" type="gioiTinh"/>
    <xs:element name="ngaySinh" type="xs:date"/>
  </xs:sequence>
</xs:complexType>

<!-- Định nghĩa kiểu dữ liệu TT thành viên đầy đủ (thành viên khác) -->
<xs:complexType name="ttDayDu">
  <xs:complexContent>
    <xs:extension base="ttCoBan">
      <xs:sequence>
        <xs:element name="moiQuanHe" type="length20"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<!-- Định nghĩa root gốc là thẻ DSHoKhau có nhiều HoKhau -->
<xs:element name="DSHoKhau">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="HoKhau" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="DiaChi" type="diachi"/>
            <xs:element name="ChuHo" type="ttCoBan"/>
            <xs:element name="ThanhVien" type="ttDayDu"
maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>

```

Bước 6: tạo file HoKhau.xml ràng buộc với file schema trên.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
Create by Minh Vũ FC 6.2016

```

```
-->
<DSHoKhou
  xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
  xsi:noNamespaceSchemaLocation='HoKhou.xsd'>
  <HoKhou>
    <DiaChi>
      <sonha>238</sonha>
      <duong>Hoàng Quốc Việt</duong>
      <phuong>Thanh Xuân Nam</phuong>
      <quan>Cầu Giấy</quan>
      <thanhpho>Hà Nội</thanhpho>
    </DiaChi>
    <ChuHo>
      <soCMND>123456789</soCMND>
      <ho>Vu</ho>
      <ten>Minh</ten>
      <nguyenquan>Hoàng Mai</nguyenquan>
      <gioiTinh>Nam</gioiTinh>
      <ngaySinh>1984-10-23</ngaySinh>
    </ChuHo>
    <ThanhVien>
      <soCMND>123456789</soCMND>
      <ho>Vu</ho>
      <ten>Tung Lam</ten>
      <nguyenquan>Hoàng Văn Thụ</nguyenquan>
      <gioiTinh>Nam</gioiTinh>
      <ngaySinh>2012-11-15</ngaySinh>
      <moiQuanHe>Con</moiQuanHe>
    </ThanhVien>
    <ThanhVien>
      <soCMND>123456789</soCMND>
      <ho>Vu</ho>
      <ten>Thuy Lam</ten>
      <nguyenquan>Hoàng Văn Thụ</nguyenquan>
      <gioiTinh>Nữ</gioiTinh>
      <ngaySinh>2017-02-15</ngaySinh>
      <moiQuanHe>Con</moiQuanHe>
    </ThanhVien>
  </HoKhou>
</DSHoKhou>
```

Phần II - Bài tập tự làm

Bài 1: tạo file Schema (**Orders.xsd**) để Valid tài liệu XML với cấu trúc được mô tả như sau:

Tài liệu cần lưu trữ một danh sách các đơn đặt hàng (**Orders**). Mỗi đơn đặt hàng (**Order**) có các thông tin (**OrderNo, OrderDate, Customer, Detail**) trong đó OrderNo là thuộc tính số đặt hàng của đơn hàng đó (Order), Detail là thông tin chi tiết về các mặt hàng bao gồm các thuộc tính **ItemNo, ItemName, Quantity, Price, Customer** có các thuộc tính **Name, Address, Email**

Constraints : Các thông tin của đơn đặt hàng bắt buộc phải nhập, **OrderNo** phải là duy nhất, trong danh sách đơn đặt hàng có thể có 0 hoặc nhiều đơn đặt hàng, trong 1 đơn đặt hàng (**OrderNo, OrderDate, CustomerName**) chỉ xuất hiện 1 lần nhưng có thể có 1 hoặc nhiều chi tiết đặt hàng, Các thuộc tính Quantity mặc định là 1 và phải là nằm trong khoảng 1-100, Price mặc định là 0 và phải là số dương . Thuộc tính Email phải đúng định dạng.

Tạo file XML (**Orders.xml**) và link **Orders.xsd** để valid dữ liệu của nó. Nhập 5 đơn đặt hàng demo.