

## Bài 2. XML và XSD

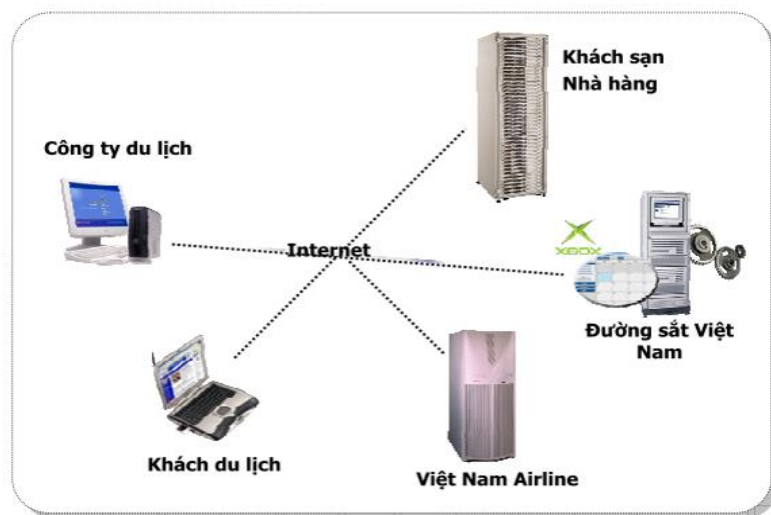
### I. XML

#### 1. XML là gì?

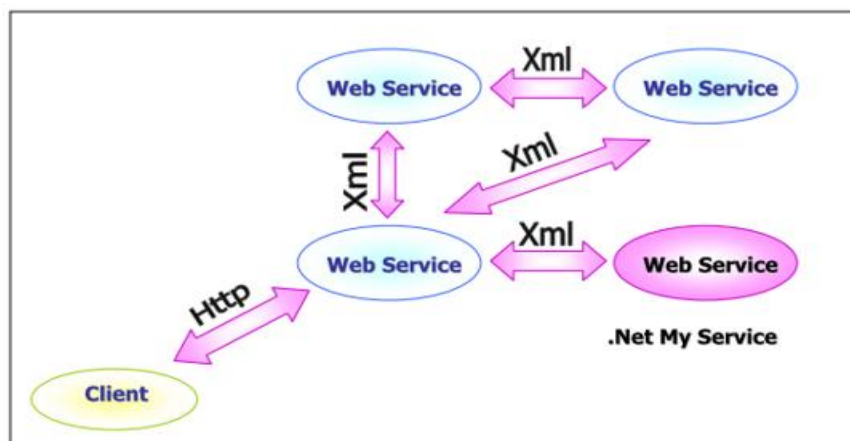
- XML, hoặc Extensible Markup Language (ngôn ngữ đánh dấu mở rộng), là một ngôn ngữ đánh dấu mà bạn có thể sử dụng để tạo ra **thẻ riêng** của mình.
- Nó được tạo nên bởi Liên minh mạng toàn cầu nhằm khắc phục những hạn chế của HTML - ngôn ngữ đánh dấu siêu văn bản, là cơ sở của mọi trang Web.
- Do tạo được các thẻ riêng nên XML mềm dẻo hơn HTML rất nhiều, vì XML là các thẻ cung cấp thông tin, cho nên nó truyền thông và tìm kiếm nhanh hơn khi truy cập vào database.
- XML là một chuẩn, được hỗ trợ trên các phần mềm và không phụ thuộc vào bất kỳ hệ điều hành nào.

#### 2. Ứng dụng XML

- Trong thực tế XML được sử dụng để đóng gói và trao đổi dữ liệu giữa các hệ thống.



Hình 1: Mối liên kết giữa các hệ thống



Hình 2: Dữ liệu trao đổi giữa các hệ thống

- Khi có sự trao đổi dữ liệu giữa các hệ thống khác nhau thì dữ liệu đó được tổ chức dưới dạng XML. Hệ thống quản lý của **Nhà hàng** muốn lấy thông tin của khách du lịch từ hệ thống của **Công ty du lịch** thì giữa các hệ thống cần phải thực hiện các bước sau:
  - o Bước 1: Giữa các hệ thống phải thống nhất cấu trúc của tài liệu XML
  - o Bước 2: Công ty du lịch sẽ trích xuất dữ liệu từ hệ thống của mình, sau đó đóng gói dữ liệu dưới dạng XML theo cấu trúc đã thỏa thuận ở bước 1.
  - o Bước 3: Hệ thống phần mềm của nhà hàng sẽ tiến hành phân tích và trích xuất dữ liệu từ tài liệu XML nhận được từ hệ thống của công ty du lịch.

### 3. Tạo 1 tài liệu XML

- Tạo phần tử gốc – mỗi tài liệu XML phải có 1 phần tử gốc  
`<?xml version="1.0" encoding="utf-8" ?>`  
`<SACH>`  
`....`  
`</SACH>`
- Tạo mã XML – Mã XML bao gồm dữ liệu nằm giữa 2 thẻ/phần tử Mở và Đóng:
  - Thẻ Mở và đóng cùng 1 khuôn dạng
  - Phân biệt chữ hoa và thường
  - Thẻ đóng phải chính xác với thẻ mở
  - Các thẻ phải lồng nhau đúng cấp
  - Thẻ không có dấu space
  - Giá trị của thuộc tính của thẻ phải được đặt trong cặp dấu nháy kép (“”) hoặc cặp dấu nháy đơn (’).

**Ví dụ:** Tài liệu XML lưu trữ thông tin về sinh viên của một lớp.

**Cách 1:** chỉ sử dụng phần tử để lưu trữ DL

```

<LOP>
  <MaLop> A01  </MaLop>
  <TenLop> ĐHKHMT1K4 </TenLop>
  <SinhVien>
    <MaSV> 001 </MaSV>
    <HoTen> Lê Văn An </HoTen>
  </SinhVien>
  <SinhVien>
    <MaSV> 002 </MaSV>
    <HoTen> Trần Thu Hà </HoTen>
  </SinhVien>
  ...
</LOP>

```

**Cây XML lưu trữ:**

```

<>Lop
  <>MaLop
  <>TenLop
  <>SinhVien
    <>MaSV
    <>HoTen
  ...

```

## Cách 2: kết hợp thuộc tính của phần tử để lưu trữ DL

```
<LOP MaLop='T1' TenLop='Tin1-K18'>
  <SinhVien>
    <MaSV> 001 </MaSV>
    <HoTen> Lê Văn An </HoTen>
  </SinhVien>
  <SinhVien>
    <MaSV> 002 </MaSV>
    <HoTen> Trần Thu Hà </HoTen>
  </SinhVien>
  ...
</LOP>
```

### Cây XML lưu trữ:

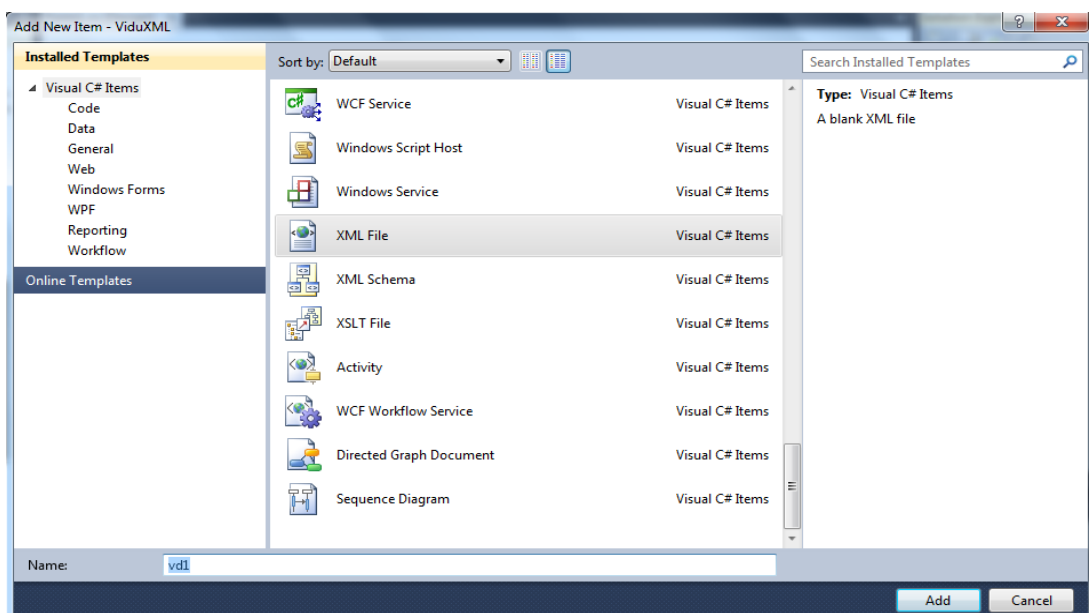
```
<>Lop
  @MaLop
  @TenLop
  <>SinhVien
    <>MaSV
    <>HoTen
  ...
```

**\*Lưu ý:** Thuộc tính chỉ đặt trong phần tử mở. Trong một phần tử, mỗi thuộc tính chỉ xuất hiện một lần.

**Bài tập 1:** Đưa bảng sau về tài liệu XML:

Họ đệm	Tên	TBKT	Điểm thi	Kết quả	
				Bảng số	Bảng chữ
Phạm Tuấn	Đạt	6	4	2	Hai
Trần Văn	Bình	6	5	3	Ba

- Tạo Project trên Visual Studio (C#)
- Trên solution – click chuột phải – add new items



- Gỡ các bài tập trên file xml sinh ra – Nếu báo lỗi có dấu ‘~~~’ bên dưới dòng lệnh – ngược lại thì không

```

<DSSV>
  <sinhvien>
    <hodem>Phạm Tuấn</hodem>
    <ten>Đạt</ten>
    <TBKT>6</TBKT>
    <diemthi>4</diemthi>
    <ketqua>
      <bangso>2</bangso>
      <bangchu>Hai</bangchu>
    </ketqua>
  </sinhvien>
  <sinhvien>
    <hodem>Trần Văn</hodem>
    <ten>Bình</ten>
    <TBKT>6</TBKT>
    <diemthi>5</diemthi>
    <ketqua>
      <bangso>3</bangso>
      <bangchu>Ba</bangchu>
    </ketqua>
  </sinhvien>
</DSSV>

```

**Bài tập 2:** Đưa bảng sau về tài liệu XML:

<b>BẢNG KẾT QUẢ HỌC TẬP</b>				
(Năm học 2009)				
Họ tên: Nguyễn Việt Đức				
Ngày sinh: 5/3/1997				
Lớp: KTPM5-K9				
STT	Môn học	Lần 1	Lần 2	Cả năm
1	Toán	5	6	8
2	Văn	7	8	4
3	Ngoại ngữ	3	5	5
4	Sử	2	5	3
5	Sinh	8	8	4
6	<b>Tổng kết</b>	6	7	7
Xếp loại học lực: Trung bình				
Xếp loại hạnh kiểm: Khá				
<b>Giáo viên chủ nhiệm</b>				
Trần Mai Linh				

```

<kqht>
  <namhoc nam="2009">
    <lop malop="ktpm5-k9" gvcn="Linh">
      <sv hoten="Nguyễn Việt Đức" ngaysinh="5-3-1997">
        <monhoc>
          <mon stt="1" tenmon="toan">
            <Diem>
              <lan1>5</lan1>
              <lan2>6</lan2>
              <canam>8</canam>
              <tk>6</tk>
            </Diem>
          </mon>
          <mon tenmon="van" stt="2">
            <Diem>
              <lan1>7</lan1>
              <lan2>8</lan2>
              <canam>4</canam>
              <tk>6</tk>
            </Diem>
          </mon>
          .....
        </monhoc>
        <xeploai>
          <hocluc>Tb</hocluc>
          <hanhkiem>Kha</hanhkiem>
        </xeploai>
      </sv>
      .....
    </lop>
    .....
  </namhoc>
  .....
</kqht>

```

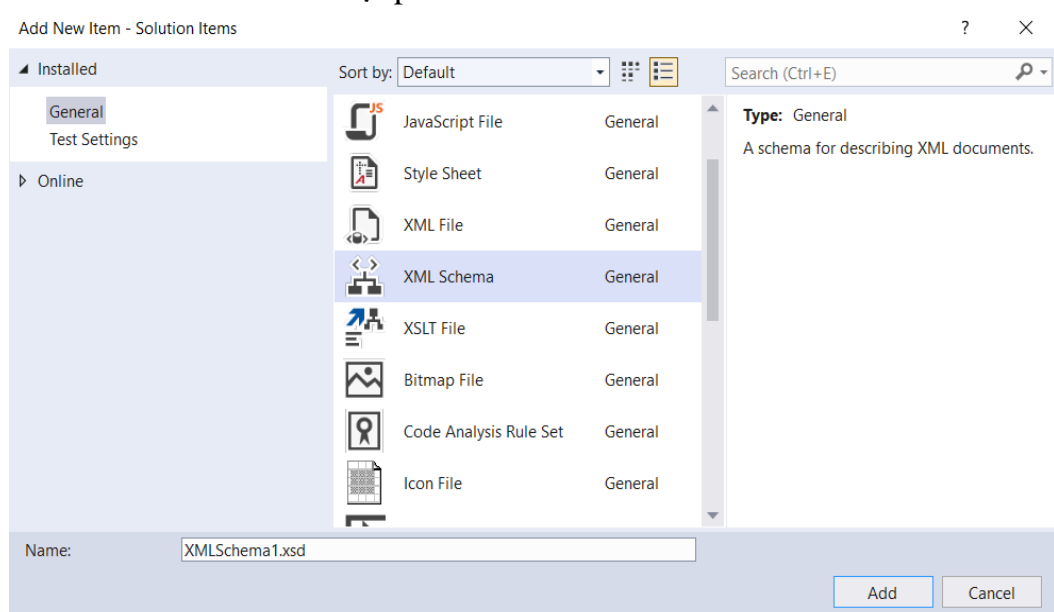
## II. Lược đồ XML - XSD

### 1. Lược đồ XML?

- Một lược đồ là 1 tập các luật để ràng buộc cấu trúc và truyền tải tập thông tin của các tài liệu XML.
- Một lược đồ mô tả một mô hình cho toàn bộ các tài liệu, mô tả cách đánh dấu dữ liệu và chỉ rõ sự sắp xếp có thể của các thẻ và văn bản trong 1 tài liệu hợp lệ.
- Một lược đồ có thể xem như 1 bộ từ vựng chung để trao đổi tài liệu giữa những tổ chức khác nhau.

### 2. Tạo lược đồ trong VS

- Tạo Project trên Visual Studio (C#)
- Trên solution – click chuột phải – add new items



### 3. Cấu trúc chung của lược đồ(Schema) XML

```
<?xml version="1.0"?>
<xs:schema id="định_danh_lược_đồ"
  xmlns:xs=http://www.w3.org/2001/XMLSchema >

  // Khai báo các phần tử

</xs:schema>
```

Phần tử **Schema** là phần tử gốc

### 3. Khai báo phần tử kiểu đơn giản

#### 3.1. Định nghĩa:

Phần tử kiểu đơn giản (Simple element): Là một phần tử XML không có thuộc tính và không chứa các phần tử con khác.

Kiểu đơn giản có 2 loại: Kiểu đơn giản xây dựng sẵn và kiểu đơn giản tự định nghĩa.

#### 3.2. Khai báo kiểu đơn giản xây dựng sẵn

```
<xs:element name="x" type="y"/>
```

- x: tên của phần tử
- y: kiểu dữ liệu của phần tử (hoặc tên 1 phần tử khác)
- Các kiểu dữ liệu thông dụng:
  - xs:ID – kiểu định danh – thường sử dụng cho mã duy nhất
  - xs:boolean – kiểu logic – true/false
  - xs:binary – kiểu nhị phân
  - xs:date – kiểu ngày tháng - ví dụ 2017 – 02- 19
  - xs:number – kiểu số nguyên/thực
  - xs:string – kiểu chuỗi ký tự
  - xs:integer – số nguyên
  - xs:decimal – số thực
  - xs:real – số thực có mũ vd: 3.4E4
  - xs:time – kiểu thời gian

#### Ví dụ:

- Ta có các phần tử XML:

```
<HoTen>Nguyễn Phú Trọng</HoTen>  
<NgaySinh>1944-04-27</NgaySinh>  
<HSLuong>8.8</HSLuong>
```

- Định nghĩa lược đồ cho các phần tử trên:

```
<xs:element name="HoTen" type="xs:string"/>  
<xs:element name="NgaySinh" type="xs:date"/>  
<xs:element name="HSLuong" type="xs:double" />
```

❖ **Chỉ định giá trị mặc định và cố định cho phần tử**

- Thuộc tính **fixed**: dùng để gán giá trị bất buộc.
- Thuộc tính **default**: dùng để gán giá trị mặc định.

Ví dụ:

```
<xs:element name="SoLuong" fixed="100" />
```

=> Phần tử SoLuong có giá trị bất buộc là 100

```
<xs:element name="SoLuong" default="100" />
```

=> Phần tử SoLuong có giá trị mặc định là 100

❖ **Chỉ định số lần xuất hiện của các phần tử**

- Thuộc tính **minOccurs** xác định số lần xuất hiện ít nhất.
- Thuộc tính **maxOccurs** xác định số lần xuất hiện tối đa (maxOccurs="unbounded" là không giới hạn số lần)
- Mặc định của **minOccurs** và **maxOccurs** là 1

Ví dụ:

```
<xs:element name="sinhvien" type="svtype" minOccurs="10" maxOccurs="50"/>
```

### 3.3. Khai báo kiểu đơn giản tự định nghĩa

```
<xs:element name="x">
  <xs:simpleType >
    <xs:restriction base="y">
      // định nghĩa ràng buộc
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- x: tên kiểu cần định nghĩa.
- y: kiểu dữ liệu cơ sở có sẵn.

Các **ràng buộc** dùng để giới hạn nội dung hoặc yêu cầu dữ liệu thỏa mãn một mẫu nào đó. Có 4 loại ràng buộc:

❖ **Ràng buộc kiểu miền giá trị:**

Cú pháp:

```
<xs:minExclusive value="giá trị min"/>
<xs:maxExclusive value="giá trị max"/>
```



**Ví dụ:** Tạo kiểu đơn giản tự định nghĩa mang tên **dayofmonth**, kiểu này chỉ mang giá trị từ 1 – 31, nghĩa là  $1 \leq \text{dayofmonth} \leq 31$

```
<xs:element name="dayofmonth">
  <xs:simpleType >
    <xs:restriction base="xs:integer">
      <xs:minExclusive value="1"/>
      <xs:maxExclusive value="31"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

❖ **Ràng buộc kiểu liệt kê:**

Cú pháp:

```
<xs:enumeration value="giá trị 1"/>
<xs:enumeration value="giá trị 2"/>
...
<xs:enumeration value="giá trị k"/>
```

**Ví dụ:** Tạo kiểu đơn giản tự định nghĩa weekday, chỉ có thể chứa các giá trị “Sunday”, “Monday”, ..., “Saturday”

```
<xs:element name="weekday">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Sunday"/>
      <xs:enumeration value="Monday"/>
      ...
      <xs:enumeration value="Saturday"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

❖ **Ràng buộc kiểu so mẫu:**

Cú pháp:

```
<xs:pattern value="mẫu dữ liệu"/>
```

Bảng các ký hiệu thường dùng trong “mẫu dữ liệu”

Ký hiệu	Mô tả
[0-9]	một ký tự số từ 0-9
[a..z]	một ký tự từ a đến z
	chọn mẫu này hoặc mẫu khác
\w	ký tự thay thế phải là 1 chữ cái
\d	ký tự thay thế phải là 1 chữ số
?	quy định số lần xuất hiện 0 hoặc 1 lần
*	quy định số lần xuất hiện 0 hoặc nhiều lần
{n}	quy định số lần xuất hiện chính xác n lần

**Ví dụ:** Tạo kiểu đơn giản tự định nghĩa password, chỉ chứa 4 ký tự: đầu tiên là 1 ký tự a hoặc b, tiếp theo là 3 ký tự số.

```
<xs:element name="password">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[ab]\d{3}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

❖ **Ràng buộc kiểu về độ dài chuỗi:**

Cú pháp:

```
<xs:minLength value="độ dài min"/>
<xs:maxLength value="độ dài max"/>
<xs:length value="độ dài cố định"/>
```

**Ví dụ:** Tạo kiểu đơn giản tự định nghĩa maso, chỉ chứa từ 4 đến 8 kí tự:

```
<xs:element name="maso">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="4"/>
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

## 4. Phân tử kiểu phức

### 4.1. Định nghĩa

Phân tử phức hợp là một phân tử XML chứa các phân tử khác hoặc có thuộc tính.

Tài liệu XML :

```
<congtv>
  <donvi madv="a01">
    <tendv> phòng tổ chức </tendv>
    <dienthoai>0437856864</dienthoai>
    <nhanvien>
      <manv>tc01</manv>
      <hoten>Lê Thu Hà</hoten>
      <ngaysinh>1988-12-23</ngaysinh>
      <gioitinh>nữ</gioitinh>
      <hsluong>2.34</hsluong>
    </nhanvien>
  </donvi>
</congtv>
```

## 4.2. Khai báo

```
<xs:element name="X">
  <xs:complexType>
    <xs:sequence>
      <!-- khai báo phần tử con khác -->
    </xs:sequence>
    <!-- khai báo thuộc tính -->
  </xs:complexType>
</xs:element>
```

- x: tên của phần tử phức hợp.

**Chú ý:** Thuộc tính phải có kiểu đơn giản, không được chứa kiểu phức hợp.

+ Khai báo thuộc tính kiểu đơn giản có sẵn:

```
<xs:attribute name="y" type="z"/>
```

+ Khai báo thuộc tính kiểu đơn giản tự định nghĩa:

```
<xs:attribute name="y">
  <xs:simpleType >
    <xs:restriction base="z">
      <!-- định nghĩa ràng buộc -->
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

**Ví dụ 1:** Tạo XSD cho tài liệu XML để lưu trữ thông tin cho một nhân viên, gồm các thông tin: mã nhân viên, họ tên, ngày sinh, hệ số lương.

+ Tài liệu XML (dạng 1)

```
<nhanvien manv='a001'>
  <hoten>Trần Minh Đại </hoten>
  <ngaysinh>1991-09-23 </ngaysinh>
  <hesoluong> 3.33 </hesoluong>
</nhanvien>
```

+ Tài liệu XML (dạng 2)

```
<nhanvien >
  <manv> a001</manv>
  <hoten>Trần Minh Đại </hoten>
```

```

    <ngaysinh>1991-09-23 </ngaysinh>
    <hesoluong> 3.33 </hesoluong>
</nhanvien>

```

Lược đồ XML cho tài liệu XML trên:

```

<xs:element name="nhanvien">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="hoten" type="xs:string"/>
      <xs:element name="ngaysinh" type="xs:date" />
      <xs:element name="hesoluong" type="xs:double" />
    </xs:sequence>
    <xs:attribute name="manv" type="xs:string" />
  </xs:complexType>
</xs:element>

```

#### 4.3. Chỉ định ràng buộc và gán giá trị cho thuộc tính

Ràng buộc/Giá trị	Mô tả
use="required"	thuộc tính phải có giá trị
use="optional "	Thuộc tính có thể có hoặc không có giá trị
fixed ="value"	Giá trị gán cho thuộc tính là cố định
default ="value"	Giá trị gán cho thuộc tính là mặc định

**Ví dụ:**

```

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

```

**Ví dụ 2:** Tạo XSD cho tài liệu XML lưu danh sách các địa chỉ liên hệ(contact), thỏa mãn các yêu cầu sau:

- + Mỗi mục địa chỉ (mỗi phần tử contact) chứa các phần tử con: tên người liên hệ (name), số điện thoại di động (tel), địa chỉ (address), thư điện tử (email), tuổi (age), giới tính (gender).
- + Phần tử contact có một thuộc tính id, và thuộc tính này phải duy nhất.
- + Phần tử name có chứa 2 phần tử con: firstname, lastname.
- + Phần tử tel chỉ nhận giá trị là 10 kí tự số.
- + Phần tử age chỉ nhận giá trị trong đoạn từ 1 – 150
- + Phần tử gender nhận 1 trong 2 giá trị male và female

Tài liệu XML:

```
<contact id="C01">
  <name>
    <firstname>Tran Van</firstname>
    <lastname>Hung</lastname>
  </name>
  <tel>0912184155</tel>
  <address>12A Bà Triệu Hà Nội</address>
  <email>abc@yahoo.com</email>
  <age>70</age>
  <gender>male</gender>
</contact>
```

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="son"
....
>
  <xs:element name="contact">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="firstname" type="xs:string"/>
              <xs:element name="lastname" type="xs:string"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
```

```

<xs:element name="tel">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="\d{10}" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="address" type="xs:string" />
<xs:element name="email" type="xs:string" />
<xs:element name="age">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1" />
      <xs:maxInclusive value="150" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>

```

```

<xs:element name="gender">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="male" />
      <xs:enumeration value="female" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
  <xs:attribute name="ID" type="xs:ID" />
</xs:complexType>
</xs:element>
</xs:schema>

```

**Bài tập 1:** Cần lưu trữ thông tin về các nhân viên trong từng đơn vị của một công ty.

- Thông tin về nhân viên gồm: mã nhân viên, họ tên, ngày sinh, giới tính, hệ số lương.
- Các nhân viên được bố trí vào các đơn vị, mỗi đơn vị gồm: mã đơn vị, tên đơn vị, điện thoại và danh sách các nhân viên.

Tạo lược đồ XML mô tả cấu trúc của tài liệu XML để lưu trữ các thông tin trên với các yêu cầu về ràng buộc như sau:

- Mã nhân viên và mã đơn vị phải duy nhất, độ dài tối đa=4.
- Giới tính phải là: “nam” hoặc “nữ”.
- Hệ số lương (từ 2.34 đến 8.8).
- Điện thoại phải là 10 kí tự số.
- Cho phép nhập tối đa 10 đơn vị và không hạn chế số nhân viên.

**Tài liệu XML :**

```
<?xml version="1.0" encoding="utf-8"? >
<congty>
  <donvi madv="a01">
    <tendv> phòng tổ chức </tendv>
    <dienthoai>0437856864</dienthoai>
    <nhanvien>
      <manv>tc01</manv>
      <hoten>Lê Thu Hà</hoten>
      <ngaysinh>1988-12-23</ngaysinh>
      <gioitinh>nữ</gioitinh>
      <hsluong>2.34</hsluong>
    </nhanvien>
    ...
  </donvi>
  ...
</congty>
```

**Bài tập 2:** Cho file HoaDonHang.xml lưu trữ thông tin về các mặt hàng của từng hóa đơn.

```
<DS>
  <HoaDon>
    <MaHD> HD1 </MaHD>
    <NgayBan> 2015-11-09 </NgayBan>
    <LoaiHang MaLoai="A" TenLoai="Điện tử">
      <Hang MaHang="H1">
        <TenHang>Ti Vi </TenHang>
        <SoLuong> 60 </SoLuong>
        <DonViTinh>Chiếc</DonViTinh>
        <DonGia>6000000</DonGia>
      </Hang>
      ...
    </LoaiHang>
    ...
  </HoaDon>
  ...
</DS>
```

Các thông tin phải đảm bảo các ràng buộc sau:



- Mã hàng phải duy nhất (tối đa 6 kí tự)
  - Đơn giá phải lớn hơn 0.
  - Đơn vị tính chỉ có thể là một trong các giá trị sau: gói, hộp, chiếc.
  - Cho phép nhập tối đa 4 hóa đơn, 10 loại hàng và không hạn chế các mặt hàng.
1. Xây dựng lược đồ XSD mô tả cấu trúc của file XML để lưu trữ các thông tin trên.
  2. Xây dựng file XML hợp lệ với XSD đã xây dựng để lưu trữ thông tin trên (nhập vào 2 hóa đơn, mỗi hóa đơn 2 loại hàng và mỗi loại hàng nhập 3 mặt hàng).