Khai báo thực thể:

&**lt;** Thay bằng dấu **<**

&**gt;** Thay bằng dấu **>**

&**amp;** Thay bằng dấu &

&**quot;** Thay bằng dấu **“**

&**apos;** Thay bằng dấu ‘

**DTD:**

**1. ELEMENT**

<!ELEMENT name content\_model>

Name: Tên Thẻ

Content\_model: Kiểu của thẻ( EMPTY, ANY, #PCDATA)

* **EMPTY: Định nghĩa thẻ rỗng**
* **ANY: Định nghĩa thẻ có nhiều kiểu dữ liệu**
* **#PCDATA: Định nghĩa thẻ kiểu văn bản**

Định nghĩa thẻ con

<!ELEMENT name (child\_name1)>

* Child\_name\*: không hoặc nhiều thẻ con child\_name
* Child\_name+: một hoặc nhiều thẻ con child\_name
* Child\_name?: không hoặc một thẻ con child\_name
* (Child\_1|child\_2): có 1 thẻ con hoặc là child\_1 hoặc là child\_2
* (Child\_1, child\_2): có 2 thẻ con child\_1 và child\_2

**2. ATTLIST**

<!ATTLIST element\_name attribute\_name attribute\_type default\_value>

* Element\_name: Tên thẻ cần định nghĩa thuộc tính
* Attribute\_name: Tên thuộc tính cần định nghĩa
* Attribute\_type: Kiểu Thuộc tính
* Default\_value: Giá trị mặc định của thuộc tính

Các kiểu giá trị thuộc tính:

* CDATA: Cho biết thuộc tính này chỉ có thể chứa kiểu dữ liệu ký tự
* ID: Cho biết thuộc tính này là một ID, tức là các giá trị của thuộc tính này không được trùng nhau và phải bắt đầu bởi một chữ cái
* IDREF: Cho biết giá trị của thuộc tính này phải là một trong các giá trị của thuộc tính ID của các phần tử khác
* IDREFS: Cho biết giá trị của thuộc tính này phải là các giá trị của các thuộc tính có kiểu ID
* NMTOKEN: Cho biết giá trị của thuộc tính là các giá trị hợp với quy tắc đặt tên của phần tử của tài liệu XML
* NMTOKENS: Cũng giống như NMTOKEN nhưng nó cho phép chứa nhiều NMTOKEN
* ENTITY: Cho biết thuộc tính này nhận giá trị là một tên tham chiếu của thực thể
* ENTITIES: Cho biết thuộc tính này nhận giá trị là các tên tham chiếu của thực thể và cách nhau bởi khoản trắng

Giá trị default:

* value: là 1 giá trị mặc định nào đó
* #REQUIRED: Thuộc tính đang xét là thuộc tính bắt buộc phải có. Đây là cách sử dụng phổ biến nhất.
* #IMPLIED: Thuộc tính đang xét là tùy chọn và không bắt buộc phải có
* #FIXED value: thuộc tính này chỉ mang duy nhất giá trị value này

Ex:

<!ATTLIST BBB month (1|2|3|4|5|6|7|8|9|10|11|12) #IMPLIED>

**3. Entity**

a. thực thể tổng quát nội

<!ENTITY enity-name “entity-value”>

Ex: <!ENTITY StoreName “chú zịt căn”>

b. thực thể tổng quát ngoại

<!ENTITY enity-name SYSTEM “URI/URL”>

Ex: <!ENTITY logo SYSTEM “logo.img”>

Ex :

<!DOCTYPE example [

<!ENTITY greeting "Hello, world!">

]>

<example>

<message>&greeting;</message>

</example>

**XSD**

**Định nghĩa**

1. Thẻ đơn giản(simple element) là thẻ không có thuộc tính và không chứa thẻ khác

<xs:element name=”color” type=”xs:string”/>

2. Thẻ phức( complex element) là thẻ chứa các thẻ khác và/hoặc có thuộc tính

**Phân loại:** Có 3 loại kiểu dữ liệu chính

**1. Kiểu cơ sở( BultinType):**

-string, integer, float, double,date, boolean, ID, month, binary.

**2. Kiểu đơn giản( SimpleType)**

- Ràng buộc kiểu dữ liệu:

Ex: xát định cận dưới và cận trên

<xs:simpleType name = “diem\_so”>

<xs:restriction base=”xs:int”>

<xs:minInclusive value = “0”>

<xs:maxInclusive value = “10”>

</xs:restriction>

</xs:simpleType>

1. minInclusive và maxInclusive:

Ví dụ: <xs:element name="age" type="xs:integer" minInclusive="0" maxInclusive="120"/>

Giải thích: Phần tử "age" phải là một số nguyên (integer) và phải nằm trong phạm vi từ 0 đến 120 (bao gồm cả hai giá trị đó).

2. minLength và maxLength:

Ví dụ: <xs:element name="username" type="xs:string" minLength="3" maxLength="20"/>

Giải thích: Phần tử "username" phải là một chuỗi (string) và độ dài của nó phải từ 3 đến 20 ký tự.

3. minOccurs và maxOccurs:

Ví dụ: <xs:element name="address" type="xs:string" minOccurs="1" maxOccurs="1"/>

Giải thích: Phần tử "address" phải xuất hiện ít nhất một lần (minOccurs="1") và nhiều nhất là một lần (maxOccurs="1").

4. pattern:

Ví dụ: <xs:element name="zipcode" type="xs:string" pattern="[0-9]{5}"/>

Giải thích: Phần tử "zipcode" phải là một chuỗi (string) và phải tuân theo mẫu được chỉ định, trong ví dụ này là một chuỗi gồm 5 chữ số.

5. enumeration:

Ví dụ:

<xs:element name="day">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="Monday"/>

<xs:enumeration value="Tuesday"/>

<xs:enumeration value="Wednesday"/>

<xs:enumeration value="Thursday"/>

<xs:enumeration value="Friday"/>

</xs:restriction>

</xs:simpleType>

</xs:element>

Giải thích: Phần tử "day" phải là một chuỗi (string) và chỉ có thể có giá trị là "Monday", "Tuesday", "Wednesday", "Thursday", hoặc "Friday".

6. totalDigits và fractionDigits:

Ví dụ: <xs:element name="price" type="xs:decimal" totalDigits="10" fractionDigits="2"/>

Giải thích: Phần tử "price" phải là một số thập phân (decimal) và có tổng số chữ số không quá 10, trong đó không quá 2 chữ số ở phần thập phân.

**3. Kiểu phức( ComplexType)**

-Khai báo tuần tự <xs:sequence>//element</xs:sepuence>

**Xpath**

**Biểu thức xpath theo đường dẫn:**

1. Nodename

Ex: Bookstore, chọn tất cả các nút có tên Bookstore

2. /

Ex: /Persion, chọn nút gốc bắt đầu từ Persion đầu tiên

3. //

Ex: //book, chọn tất cả nút book mà không quan tâm nút đó đang ở đâu

4. .

Ex: .Book, chọn nút hiện thời

5. ..

Ex: ..Book, chọn nút cha của nút hiện thời

6. @

Ex: //@conves, chọn ra những giá trị thuộc tính conves

Ví dụ:

-bookstore/book, chọn tất cả các nút con book bắt đầu từ nút cha bookstore

-bookstore//book, chọn tất cả các nút book và cháu chắc của book bắt đầu từ nút cha bookstore

-bookstore/book/@type, chọn ra tất cả các giá trị của thuộc tính type

**Biểu thức điều kiện:**

1. [@attributeName] - Chọn các nút có thuộc tính cụ thể.

Ex: //book[@title], chọn tất cả các nút có thuộc tính title

2. [@attributeName='value'] - Chọn các nút có thuộc tính cụ thể và giá trị cụ thể.

Ex: //book[@lang=’vietnam’]

3. [position()] - Chọn nút ở vị trí cụ thể trong danh sách.

Ex: //book[position()=3], chọn nút book ở vị trí số 2

Ex: //book[position()<10], chọn 9 nút đầu tiên

4. last()] - Chọn phần tử cuối cùng trong danh sách.

**Biểu thức xpath theo đường dẫn chưa biết nút:**

1. \* Tất cả các nút

Ex /bookstore/\* tất cả các nút con của bookstore

2. @\* tất cả cá thuộc tính

Ex /bookstore/book/title[@\*] tất cả các nút title con của book và cháu của bookstore có thuộc tính