University of M'hamed Bouguerra Boumerdès
Faculty of Sciences - Department of Computer ScienceModule: DSS

Sector: L3 (ISIL)

Year: 2023-2024

Lab No. 2: XML Schema

Objective: learn DTDs and XML schema.
Noticed: Use the OXYGEN XML editor.

Exercise 01(*)

1. Write a DTD for a bibliography. This bibliography contains authors and documents.

- The documents can be books, articles and dissertations;
- For each author, we want to have elements of their marital status (last name, first name, date of birth) and their website
- the information needed for a book is:
  - its general title;
  - author codes;
  - its volumes and for each volume, their number of pages;
  - general information about its edition such as the name of the publisher, the place of publication, its ISBN number, and the year of publication;
- the information needed for an article is:
  - his title ;
  - author codes;
  - its publication references: name of the journal, page number, year of publication and journal number
- the information necessary for a final dissertation is:
  - his title ;
  - author codes;
  - the name of the university;
  - the year of defense
- Add an optional subtitle attribute to the title element;
- Make the volume element an empty element and add a required nb_pages attribute and an optional subtitle attribute;
- Add attribute to log element`log_name`and assign the unknown value as default.
- Add an optional attribute for memories`memo_type`, having both values: national or international.

2. Test this DTD with an XML file that we write from scratch and validate.

3. Write an XML schema for this bibliography with user namespace=”http://si.dz/serie2/exercice1”.

We will now define our own data types, deriving them from the built-in types.

4. Define a simple type named codeAuthor, based on a restriction of the xsd:string type, limiting itself to a character string that begins with “a_”. Example: “a_Ben01”
5. Define a simple type named typeISBN, based on a restriction of the type xsd:string, limited to a character string composed of 10 digits. Use it in the ISBN element declaration.
6. Declare a typePages type, based on a restriction of the type xsd:string, limited to a number, then the character string "to", then another number.
7. Create a complex typeBiblio type, taking the declaration of the root element, and assign it to the root element.
8. Check the validity of the XML schema file with your XML file.

Exercise 02

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name='magasin'>
      <xs:complexType>
         <xs:sequence>
            <xs:element name='clients'>
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name='client' type='ClientType' minOccurs='0' maxOccurs='unbounded' />
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
            <xs:element name='commandes'>
               <xs:complexType>
                  <xs:sequence>
                     <xs:element name='commande' type='CommandeType' minOccurs='0' maxOccurs='unbounded' />
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
   <xs:complexType name='ClientType'>
      <xs:sequence>
         <xs:element name='nom' type='xs:string'/>
         <xs:element name='prenom' type='xs:string'/>
         <xs:element name='dateNaissance' type='xs:string'/>
         <xs:choice>
            <xs:element name='telephone' type='xs:string'/>
            <xs:element name='email' type='xs:string'/>
         </xs:choice>
      </xs:sequence>
      <xs:attribute name='clientID' type='xs:integer'/>
   </xs:complexType>
   <xs:complexType name='CommandeType'>
      <xs:sequence>
         <xs:element name='clientID' type='xs:integer'/>
         <xs:element name='dateCommande' type='xs:date'/>
         <xs:element name='dateLivraison' type='xs:date'/>
         <xs:element name='article' type='xs:string'/>
      </xs:sequence>
   </xs:complexType>
</xs:schema>
```

1. Produce a DTD equivalent to this XML schema.
2. Produce the smallest valid XML document possible.
3. Produce the smallest valid XML document containing all elements.