

Master 2 Humanités numériques – Bases de données semi-structurées TD 3 : Documents XML valides (XML Schema)

J. Darmont - <https://eric.univ-lyon2.fr/jdarmont/>

Exercice 1

1. Créer un document XML Schema (nouveau fichier nommé villes.xsd) à même de représenter le schéma de données du document villes.xml du TD 1.
 - La version de XML pour XML Schema doit être la 1.0 (le validateur ne supporte pas la 1.1).
 - L'élément racine *villes* comprend une séquence (type complexe) d'au moins un élément *ville* sans borne supérieure.
 - La définition de l'élément *ville* (xsd:string) passe par un type « simple ». L'attribut *id* (xsd:ID) est requis.
2. Utiliser le validateur en ligne <https://www.xmlvalidation.com> .
 - a) Copier/coller ou téléverser le fichier villes.xml.
 - b) Cocher l'option « Validate against external XML schema », puis valider.
 - c) Copier/coller ou téléverser le fichier villes.xsd puis valider.
3. En cas d'erreur, corriger le XML Schema jusqu'à ce que le document XML soit valide.

Exercice 2

1. Reprendre le document du TD 2 trains+dtd.xml. Le copier/coller sous le nom trains+villes.xml. Y supprimer la DTD.
2. Créer un document XML Schema nommé trains.xsd à même de représenter le schéma de données du document trains+villes.xml.
 - L'élément racine *trains* comprend une séquence d'au moins un élément *ville* sans borne supérieure et d'au moins un élément *train* sans borne supérieure.
 - L'élément *ville* est le même qu'à l'exercice 1.
 - L'élément *train* est une séquence d'éléments *wagon*, d'une part, et des attributs *id* (xsd:ID, requis), *depart* (xsd:IDREF, requis), *arrivee* (xsd:IDREF, requis) et *autresgares* (xsd:IDREFS, optionnel), d'autre part.
 - L'élément *wagon* est une séquence d'éléments *container* sans borne et de l'attribut *id* (xsd:ID, requis).

- L'élément *container* (xsd:string) est un type simple et l'attribut *volume* (xsd:float).

3. Réutiliser le validateur comme à l'exercice 1. En cas d'erreur, corriger le XML Schema jusqu'à ce que le document XML soit valide.

Exercice 3

1. Créer un document XML Schema nommé *menus.xsd* à même de représenter le schéma de données du document *menus.xml* du TD 1.

- L'élément racine *menus* comprend une séquence de 1 à 5 éléments *menu*. Utiliser une référence afin de déporter la spécification de *menu* plus loin dans le document, à des fins de clarté du code XML Schema.
- L'élément *menu* comprend, d'une part, une séquence avec 1 *entree*, 1 *plat* et 1 *dessert* et, d'autre part, un attribut *jour* requis. Utiliser de nouveau des références pour alléger le code.
- L'élément *entree* (xsd:string) est un type simple.
- L'élément *plat* est un choix entre un *plat_unique* (xsd:string) et une séquence composée d'une *viande* (xsd:string) et d'un *accompagnement* (xsd:string).
- L'élément *dessert* (xsd:string) est un type simple.
- L'attribut *jour* (xsd:ID) est une restriction des valeurs possibles de *jour*, soit l'énumération Lundi, Mardi, Mercredi, Jeudi, Vendredi.

2. Réutiliser le validateur comme aux exercices 1 et 2. En cas d'erreur, corriger le XML Schema jusqu'à ce que le document XML soit valide.

Correction Exercice 1

```
<?xml version="1.0" encoding="utf-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="villes">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ville"
          minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="ville">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:string">
          <xsd:attribute name="id" type="xsd:ID" use="required" />
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Correction Exercice 2

```
<?xml version="1.0" encoding="utf-8" ?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="trains">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="ville"
          minOccurs="1" maxOccurs="unbounded" />
        <xsd:element ref="train"
          minOccurs="1" maxOccurs="unbounded" />
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="ville">
    <xsd:complexType>
      <xsd:simpleContent>
        <xsd:extension base="xsd:string">
          <xsd:attribute name="id" type="xsd:ID" use="required" />
        </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>
  </xsd:element>
```

```

<xsd:element name="train">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="wagon"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID" use="required" />
    <xsd:attribute name="depart" type="xsd:IDREF" use="required" />
    <xsd:attribute name="arrivee" type="xsd:IDREF" use="required" />
    <xsd:attribute name="autresgares" type="xsd:IDREFS"
      use="optional" />
    </xsd:complexType>
  </xsd:element>

<xsd:element name="wagon">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="container"
        minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:ID" use="required" />
  </xsd:complexType>
</xsd:element>

<xsd:element name="container">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="volume" type="xsd:float" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

</xsd:schema>

```

Correction Exercise 3

```

<?xml version="1.0" encoding="utf-8" ?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="menus">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="menu" minOccurs="1" maxOccurs="5"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

```

```
<xsd:element name="menu">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="entree" minOccurs="1" maxOccurs="1" />
      <xsd:element ref="plat" minOccurs="1" maxOccurs="1" />
      <xsd:element ref="dessert" minOccurs="1" maxOccurs="1" />
    </xsd:sequence>
    <xsd:attribute ref="jour" use="required" />
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="entree">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string" />
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="plat">
  <xsd:complexType>
    <xsd:choice>
      <xsd:element name="plat_unique" type="xsd:string" />
      <xsd:sequence>
        <xsd:element name="viande" type="xsd:string" />
        <xsd:element name="accompagnement"
          type="xsd:string" />
      </xsd:sequence>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:element name="dessert">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string" />
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```

```
<xsd:attribute name="jour">
  <xsd:simpleType>
    <xsd:restriction base="xsd:ID">
      <xsd:enumeration value="Lundi"/>
      <xsd:enumeration value="Mardi"/>
      <xsd:enumeration value="Mercredi"/>
      <xsd:enumeration value="Jeudi"/>
      <xsd:enumeration value="Vendredi"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
```

</xsd:schema>