



République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université AMO de Bouira

Faculté des Sciences et des Sciences Appliquées

Département d'Informatique



Compte rendu de TP 3

Méthodes et Technologies d'Implémentation

Spécialité : Génie des Systèmes Informatiques

Schéma XML

Réalisé par

— AIFAOU SARA

— FEKAR ROUMAÏSSA

2019/2020

0.1 Séance TP

0.1.1 Introduction

XML Schema Définition, communément connu sous le nom XSD, est une façon de décrire précisément le langage XML.

XSD vérifie la validité de la structure et le vocabulaire d'un document XML contre les règles grammaticales de la langue XML appropriée.

Caractéristiques de schéma XML

- XSD peuvent être extensible pour des ajouts futurs .
- XSD est plus riche et plus puissant que DTD .
- XSD est écrit en XML .
- XSD prend en charge les types de données .
- XSD soutient namespaces .

0.1.2 Méthodologie du travail

un Schéma XML est constitué d'un ensemble de balises.

Syntaxe d'un type d'élément simple : `<xs:element name="element-name" type="element-type"/>`

Valeur par défaut : `<xs:element name="grade" type="xs:string" default="NA" />`

Valeur fixe : `<xs:element name="class" type="xs:string" fixed="1" />`

Attribut représente l'attribut d'un élément XML. XSD définit comme un type simple : `<xs:attribute name="attribute-name" type="attribute-type"/>`

En XML : `<student matricule="393" />`

Élément Complexe :

`<xsd:element name="mon_nom">`

`<xsd:complexType>`

`<xsd:complexType>`

`</xsd:element>`

0.1.3 Outils utilisés

- XML Copy Editor



XML Copy Editor est un éditeur XML léger et rapide. Mise à part la vérification de fermeture de tags, ce programme peut tout supporter (relax et strict), transformations XSL, Xpath, encodage, etc.).

0.1.4 Tests et résultats attendus

- 1eme Partie

Dans le premier partie en va :

- Tester la validité du fichier annuaire-v0.1.xml par rapport au schéma XSD.
- Corriger les éventuelles erreurs dans le fichier XML.(Voir Figure 1)



FIGURE 1 – Figure de XML valid

- Mettre à jour le XSD afin d'ajouter les attributs pour le téléphone. (Voir Figure 2)

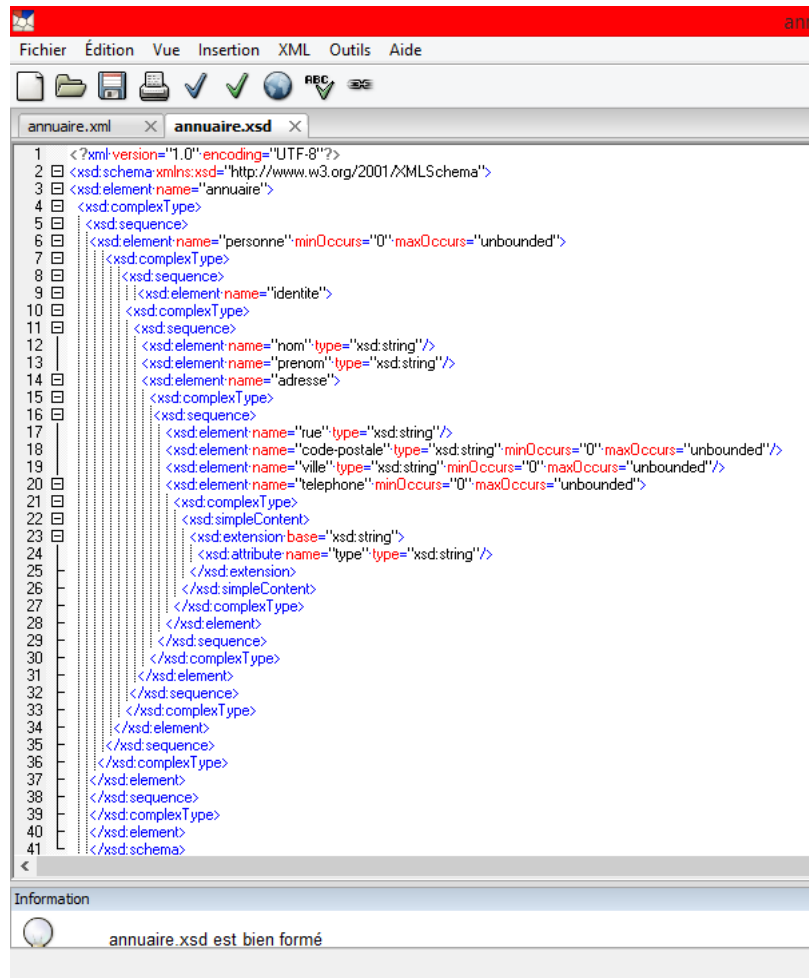


FIGURE 2 – Figure de XSD valid

● 2eme Partie

Dans la deuxième partie on va :

(Voi Figur 3 4 5)

- Définir la DTD et le XML Schema pour le fichier bank.xml afin de satisfaire les exigences suivantes :
 - Il y a deux type de comptes : checking and savings accounts (compte courant et compte d'épargne) .
 - l'Id de compte est uniques .
 - l'Id client est unique .
 - 'c_id' réfère le client et 'ac_id' réfère le compte .
 - Le crédit doit être supérieur à -5000 .

- Utiliser l'héritage pour dériver les compte courants (checking) et les comptes d'épargne (saving) à partir d'un type de compte basique.

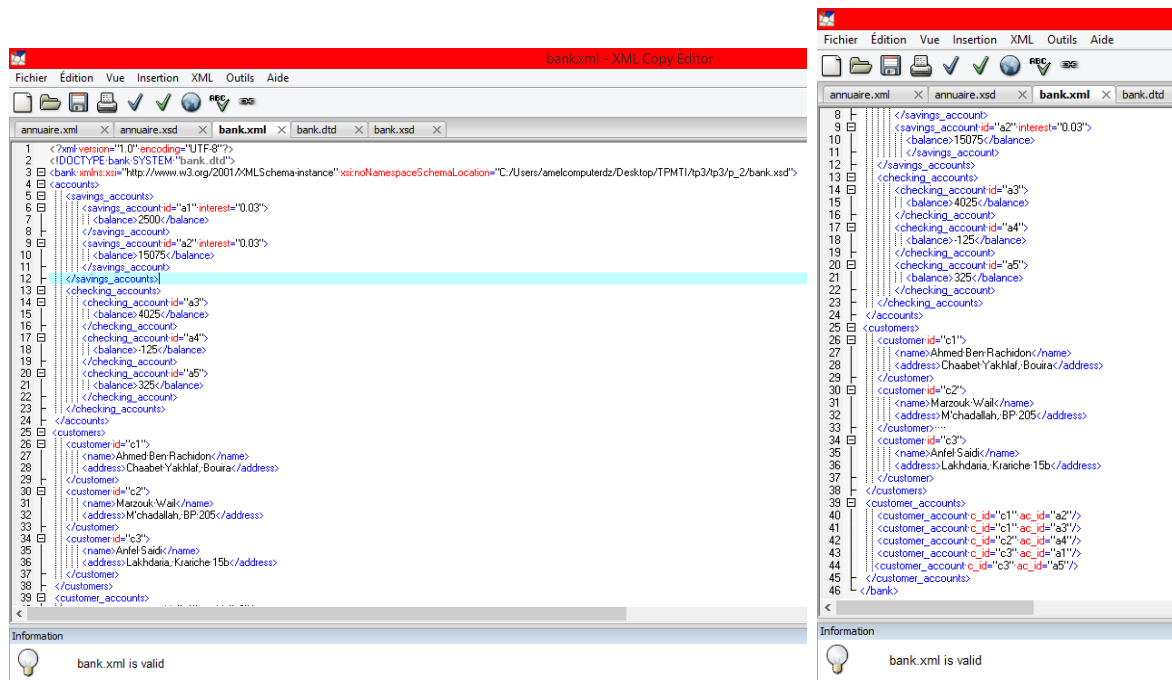


FIGURE 3 – Figure de XML valid

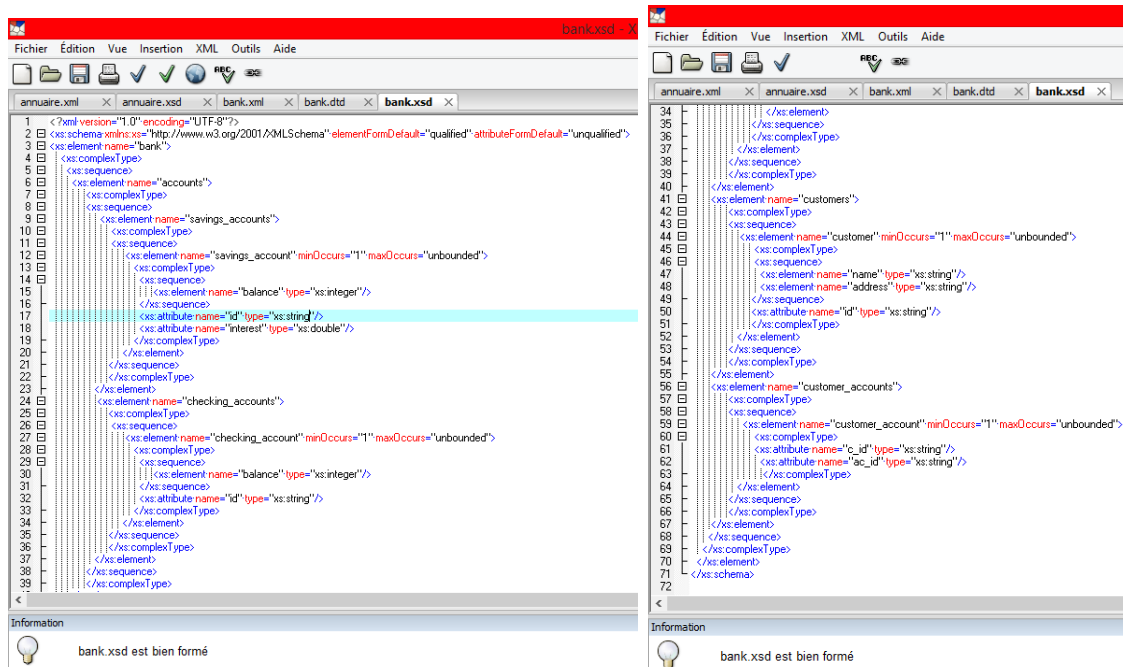


FIGURE 4 – Figure de XSD valid

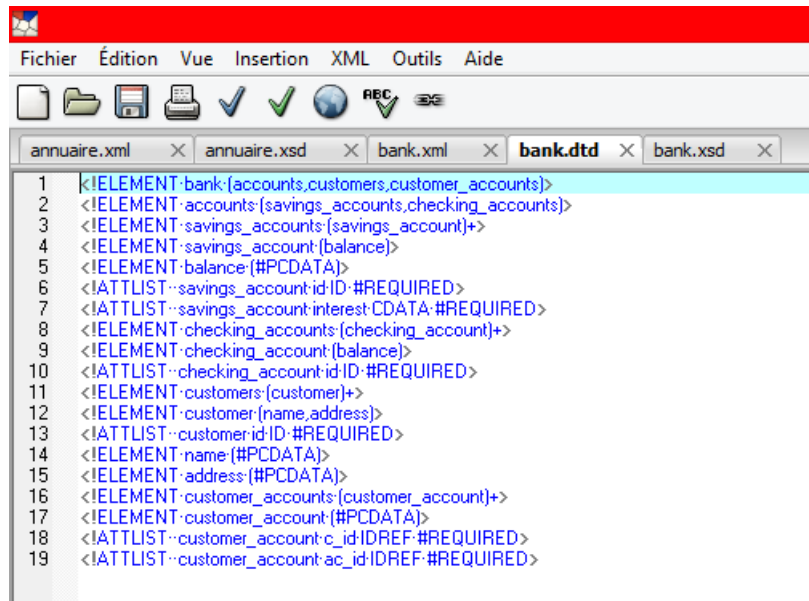


FIGURE 5 – Figure de dtd valid

0.2 Travail à domicile

0.2.1 Introduction

Dans cette partie on va Créer un Schéma XML pour le document Book.xml

- A. l'élément books doit avoir un ou plusieurs livres (un au moins).
- B. Chaque livre doit contenir les éléments suivants en séquence : title, edition, author (un ou plusieurs), publishedDate, type, et price
- C. Chaque livre doit contient un ISBN comme un attribut clé obligatoire.
- D. « title » contient n'importe quel caractères (type string).
- E. « title » contient un attribut « type » qui prend que les valeurs "P" et "H".
- F. « edition » est un élément optionel, et doit contenir un nombre qui ne dépasse pas un octet.
- G. Il peut y avoir un ou plusieurs auteurs (au moins un auteur) pour chaque livre. Chaque auteur doit avoir les éléments enfants suivants : First, middle, Last. Les éléments doivent être dans l'ordre First, middle, et Last, mais le Middle est un élément facultatif. Les éléments First, Middle et Last sont tous de types de chaînes, mais le Middle doit avoir au plus un caractère.
- H. L'élément « publishedDate month » est défini comme un type de unsignedByte avec une valeur inclusive minimale de 1 et une valeur maximale de 12.

L'élément « publishedDate day » est défini comme un type de unsignedByte avec une valeur inclusive minimale de 1 et une valeur maximale de 31.

L'élément « publishedDate year » est défini comme un type de gYear.

I. L'élément « type » est une chaîne de caractères et doit être soit « fiction » ou « non-fiction » pour être valide.

J. L'élément de prix est défini comme un type de décimal avec seulement trois caractères à gauche de la virgule décimale et deux à droite. Le montant doit être supérieur à zéro.

Annexe **A**

Code en annexe

A.1 partie tp

A.1.1 1eme Partie

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="annuaire">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="personne" minOccurs="0" maxOccurs="unbounded">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="identite">
                <xsd:complexType>
                  <xsd:sequence>
                    <xsd:element name="nom" type="xsd:string"/>
                    <xsd:element name="prenom" type="xsd:string"/>
                    <xsd:element name="adresse">
                      <xsd:complexType>
                        <xsd:sequence>
                          <xsd:element name="rue" type="xsd:string"/>
                          <xsd:element name="code-postale" type="xsd:string"/>

```



```
<xsd:element name="ville" type="xsd:string"/>
<xsd:element name="telephone" minOccurs="0" maxOccurs="unbounded">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:attribute name="type" type="xsd:string"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
```

A.1.2 2eme Partie

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault=
    "unqualified">
  <xs:element name="bank">
  <xs:complexType>
  <xs:sequence>
  <xs:element name="accounts">
  <xs:complexType>
  <xs:sequence>
  <xs:element name="savings_accounts">
  <xs:complexType>
  <xs:sequence>
  <xs:element name="savings_account" minOccurs="1" maxOccurs=
    "unbounded">
  <xs:complexType>
  <xs:sequence>
  <xs:element name="balance" type="xs:integer"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:ID" use="required"/>
  <xs:attribute name="interest" type="xs:double" use="required"/>
  </xs:complexType>
  </xs:element>
  </xs:sequence>
  </xs:complexType>
  </xs:element>
  <xs:element name="checking_accounts">
  <xs:complexType>
  <xs:sequence>
  <xs:element name="checking_account" minOccurs="1" maxOccurs=
    "unbounded">
```

```
<xs:complexType>
<xs:sequence>
<xs:element name="balance" type="xs:integer"/>
</xs:sequence>
<xs:attribute name="id" type="xs:ID" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="customers">
<xs:complexType>
<xs:sequence>
<xs:element name="customer" minOccurs="1" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="name" type="xs:string"/>
<xs:element name="address" type="xs:string"/>
</xs:sequence>
<xs:attribute name="id" type="xs:ID" use="required"/>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="customer_accounts">
<xs:complexType>
<xs:sequence>
<xs:element name="customer_account" minOccurs="1"
```

```
    maxOccurs="unbounded">
<xs:complexType>
<xs:attribute name="c_id" type="xs:IDREF" use="required"/>
<xs:attribute name="ac_id" type="xs:IDREF" use="required"/>
</xs:complexType>
<xs:key name="idaccount">
<xs:selector xpath="./account"/>
<xs:field xpath="@id"/>
</xs:key>
<xs:key name="idcustomer">
<xs:selector xpath="./customer"/>
<xs:field xpath="@id"/>
</xs:key>
<xs:keyref name="idRefaccount" refer="idaccount" >
<xs:selector xpath="./customer_account"/>
<xs:field xpath="@ac_id"/>
</xs:keyref>
<xs:keyref name="idRefcustomer" refer="idcustomer" >
<xs:selector xpath="./customer_account"/>
<xs:field xpath="@c_id"/>
</xs:keyref>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

```
<!ELEMENT bank (accounts ,customers ,customer_accounts)>
<!ELEMENT accounts (savings_accounts ,checking_accounts)>
<!ELEMENT savings_accounts (savings_account)+>
<!ELEMENT savings_account (balance)>
<!ELEMENT balance (#PCDATA)>
<!ATTLIST savings_account id ID #REQUIRED>
<!ATTLIST savings_account interest CDATA #REQUIRED>
<!ELEMENT checking_accounts (checking_account)+>
<!ELEMENT checking_account (balance)>
<!ATTLIST checking_account id ID #REQUIRED>
<!ELEMENT customers (customer)+>
<!ELEMENT customer (name,address)>
<!ATTLIST customer id ID #REQUIRED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT customer_accounts (customer_account)+>
<!ELEMENT customer_account (#PCDATA)>
<!ATTLIST customer_account c_id IDREF #REQUIRED>
<!ATTLIST customer_account ac_id IDREF #REQUIRED>
```

A.2 Travail à domicile

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="books">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="book" minOccurs="1" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="title">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute name="type" use="required">
                        <xs:simpleType>
                          <xs:restriction base="xs:string">
                            <xs:enumeration value="P"/>
                            <xs:enumeration value="H"/>
                          </xs:restriction>
                        </xs:simpleType>
                      </xs:attribute>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="edition" type="xs:integer"/>
              <xs:element name="author" minOccurs="1" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="first" type="xs:string"/>
```

```
<xs:element name="middle" type="xs:string" minOccurs="0" maxOccurs="1" />
<xs:element name="last" type="xs:string" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="publishedDate">
<xs:complexType>
<xs:sequence>
<xs:element name="month" maxOccurs="1">
<xs:simpleType>
<xs:restriction base="xs:unsignedByte">
<xs:minInclusive value="1" />
<xs:maxInclusive value="12" />
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="day" maxOccurs="1">
<xs:simpleType>
<xs:restriction base="xs:unsignedByte">
<xs:minInclusive value="1" />
<xs:maxInclusive value="31" />
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="year" type="xs:gYear" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="type">
<xs:simpleType>
<xs:restriction base="xs:string">
<xs:enumeration value="fiction" />
```

```
<xs:enumeration value="nonfiction" />
</xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="price" type="xs:decimal" />
</xs:sequence>
<xs:attribute name="isbn" type="xs:string" use="required" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```