

TME : XSCHEMA

Exercice 1 : Schéma du guide touristique

Question 1 : Définir le schéma (*base.xsd*) du guide touristique (*base3.xml*).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<base xmlns="Espace_TPXML"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="Espace_TPXML
                          base3.xsd">

  <restaurant nom="la tour d'argent" etoile="3" ville="Paris">
    <fermeture>dimanche et lundi</fermeture>
    <menu nom="buffet" prix="200" />
    <menu nom="gourmet" prix="300"/>
  </restaurant>

  <ville nom="Paris" departement="75">
    <plusBeauMonument nom="tour Eiffel" tarif="30"/>
  </ville>

  <restaurant nom="chez Bocuse" etoile="2" ville="Lyon">
    <fermeture>dimanche</fermeture>
    <menu nom="specialites lyonnaises" prix="200"/>
    <menu nom="vegetarien" prix="299"/>
  </restaurant>

  <restaurant nom="MacDo" etoile="0" ville="Avignon">
    <menu nom="standard" prix="35"/>
    <menu nom="enfant" prix="20"/>
    <menu nom="big" prix="49"/>
  </restaurant>

  <ville nom="Lyon" departement="69">
  </ville>
  <ville nom="Avignon" departement="84">
    <plusBeauMonument nom="le pont" tarif="0"/>
  </ville>

  <restaurant nom="Les 4 saisons" etoile="1" ville="Paris">
    <fermeture>mardi</fermeture>
    <menu nom="assiette printaniere" prix="50" />
    <menu nom="salade d'ete" prix="30"/>
    <menu nom="menu d'automne" prix="150"/>
    <menu nom="menu d'hiver" prix="99"/>
  </restaurant>
```

```

<ville nom="Brest" departement="29">
  <plusBeauMonument nom="oceanopolis" tarif="95"/>
</ville>

</base>

```

Question 2 : Définir le schéma (*base-contrainte.xsd*) du guide touristique (*base-contrainte.xml*) pour vérifier les contraintes suivantes :

c1 : un restaurant a toujours au moins 2 menus

c2 : une ville peut avoir un plus beau monument, certaines villes n'en ont pas.

c3 : un monument a toujours un nom et un tarif

c4 : une ville a toujours un nom et un département

c5 : un restaurant a entre 0 et 3 étoiles.

c6 : la ville d'un restaurant doit exister dans la base (*i.e.*, il doit exister un élément *ville* pour chaque nom de ville référencé dans un élément *restaurant*). Pour définir cette contrainte, utiliser les éléments *key* et *keyref*

base-contrainte.xml

```

<?xml version="1.0" encoding="ISO-8859-1"?>

<base xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="base-contrainte.xsd">

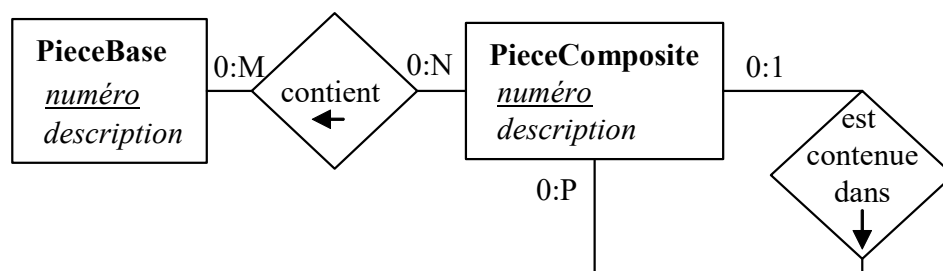
... idem base3.xml

</base>

```

Exercice 2 : Xschéma et l'intégrité référentielle

Soit le schéma entité association suivant :



Légende: numéro est un identifiant unique

Association pièce de base – pièce composite :

une pièce composite contient (0 à N) pièce de base,

une pièce de base est contenue dans (0 à M) pièce composites.

Association récursive :

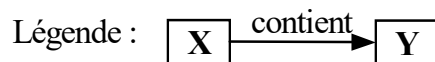
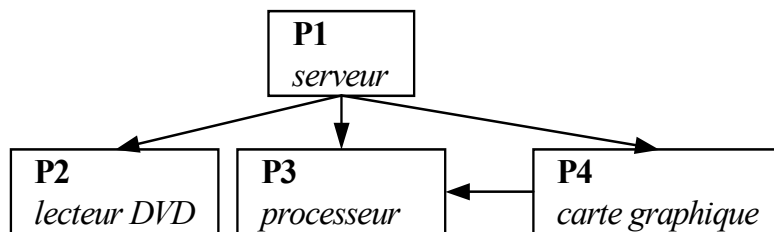
une pièce composite contient (0 à P) pièce composite,

une pièce composite est contenue dans (0 à 1) pièce composite.

a) Transformer le diagramme E/A ci-dessus en un schéma XML tel que :

- toutes les pièces sont contenues dans l'élément racine <stock>,
- les entités sont exprimées par des éléments XML,
- le numéro est un nombre entier,
- une association 1-N est représentée par un attribut,
- une association N-M est représentée par une séquence d'éléments de type simple (*i.e.*, de type entier),
- l'intégrité référentielle est représentée par des éléments **key** et **keyref**.

b) Ecrire le document XML conforme au schéma de la question a) et contenant les pièces suivantes :



Annexe pour la partie b

Stock.dtd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!ELEMENT Stock (PC|PB) *>
```

```
<!ELEMENT PC EMPTY>
```

```
<!ATTLIST PC num ID #REQUIRED
  description CDATA #IMPLIED
  contientPC IDREFS #IMPLIED
  contientPB IDREFS #IMPLIED
  contenue_dans IDREF #IMPLIED>
```

```
<!ELEMENT PB EMPTY>
```

```
<!ATTLIST PB num ID #REQUIRED
  description CDATA #IMPLIED
  contenue_dans IDREFS #IMPLIED>
```

Exercice 3 : Schéma pour une librairie

1) Proposer un schéma (*BookStore.xsd*) pour le catalogue (*BookStore.xml*). Un élément *xs:element* doit contenir la définition de son type (attribut *type* ou sous élément *xs:complexType*).

BookStore.xml

```
<?xml version="1.0"?>

<BookStore xmlns="http://www.books.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.books.org/BookStore.xsd">

  <Book>
    <Title>My Life and Times</Title>
    <Author>Paul McCartney</Author>
    <Date>1998</Date>
    <ISBN>1-56592-235-2</ISBN>
    <Publisher>McMillin Publishing</Publisher>
  </Book>
  <Book>
    <Title>Illusions The Adventures of a Reluctant
Messiah</Title>
    <Author>Richard Bach</Author>
    <Date>1977</Date>
    <ISBN>0-440-34319-4</ISBN>
    <Publisher>Dell Publishing Co.</Publisher>
  </Book>
  <Book>
    <Title>The First and Last Freedom</Title>
    <Author>J. Krishnamurti</Author>
    <Date>1954</Date>
    <ISBN>0-06-064831-7</ISBN>
    <Publisher>Harper & Row</Publisher>
  </Book>
</BookStore>
```

2) Ecrire le schéma *BookStore2.xsd* en imbriquant au maximum la définition des éléments. La racine du schéma (élément *xs:schema*) ne doit posséder qu'un seul sous élément direct *xs:element*.

3) Ecrire le schéma *BookStore3.xsd* en séparant la définition des types et des éléments. Définir d'abord tous les types complexes (*xs:complexType*), puis définir ensuite tous les éléments.

4) Ecrire le schéma *BookStore4.xsd* en définissant un type *Publication*, ainsi qu'un sous-type *BookPublication* qui en hérite.

5) Ecrire le schéma *BookStore5.xsd* en définissant un type, *ISBN-type*, basé sur le type *String* et permettant d'imposer un format à l'élément *ISBN*.

