

TD XML et DTD

Exercice 1 : Conformité DTD

1/ Parmi les documents suivants, lesquels sont valides (conformes à leur DTD), lesquels ne le sont pas. Pour chaque document non valide, expliquez pourquoi il ne l'est pas.

doc A <?xml version="1.0" standalone="yes"?> <!DOCTYPE publisher [<!ELEMENT document (#PCDATA title publisher)*> <!ELEMENT title (#PCDATA)> <!ELEMENT publisher (name)> <!ELEMENT name (#PCDATA)> > <document> <title> BD et sites Web </title> <publisher> <name>Alfred Pub</name> </publisher> Publishers are noted in report by name </document>	doc B <?xml version="1.0" standalone="yes"?> <!DOCTYPE document [<!ELEMENT document ((title publisher)*, name?) > <!ELEMENT title (#PCDATA)> <!ELEMENT publisher (name)> <!ELEMENT name (#PCDATA)> > <document> <title> BD </title> <publisher> Alfred Pub </publisher> </document>
doc C <?xml version="1.0" standalone="yes"?> <!DOCTYPE document [<!ELEMENT document (title?)> <!ELEMENT title (#PCDATA)> <!ATTLIST title num ID #REQUIRED info CDATA #IMPLIED> > <document> <title> Bases de données et sites Web </title> </document>	doc D <?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?> <!DOCTYPE document [<!ELEMENT document (title, publisher, #PCDATA)> <!ELEMENT title (#PCDATA)> <!ELEMENT publisher (#PCDATA)> > <document> <title> hello <title>The Publishers</title> <publisher>Alfred Pub</publisher> </title> Publishers are noted in report by name. </document>

2/ Modifier les DTD pour que les documents deviennent valides (sans utiliser de ANY)

Exercice 2 : DTD d'une famille

On veut écrire une DTD pour des documents décrivant des familles. Une famille porte un nom et est constituée d'une ou plusieurs personnes. Pour chaque personne de la famille, on a le prénom, l'âge, le poids en kilos (kg) ou le poids en livres (lb), et éventuellement la taille. Les liens de parenté (père et mère) sont gérés grâce à des attributs de type ID et IDREF.

1/ Ecrire la DTD correspondante famille.dtd. Celle-ci devra accepter par exemple le document suivant :

<?xml version="1.0" encoding="ISO-8859-1"?> <!DOCTYPE famille SYSTEM "famille.dtd"> <famille> <nom>Martin</nom> <personne pnumber="a1"> <prenom>Juliette</prenom> <age>30</age> <poids-kg>58</poids-kg> </personne> <personne pnumber="a2"> <prenom>Romeo</prenom> <age>31</age> <poids-lb>97</poids-lb> </personne> (suite colonne suivante...)	<personne pnumber="a3" mere="a1" pere ="a2"> <prenom>Max</prenom> <age>4</age> <poids-kg>12</poids-kg> <taille>1.25</taille> </personne> <personne pnumber="a4" mere="a1" pere ="a2"> <prenom>Marie</prenom> <age>3</age> <poids-lb>18</poids-lb> <taille>1.10</taille> </personne> </famille>
--	--

La DTD ne doit pas accepter le cas suivant :

```
<personne><prenom>Toto</prenom>...</personne>
```

2/ Pourquoi un document contenant une personne qui est son propre père est-il valide ?

3/ Ajouter dans famille un élément *animal* identifié par l'attribut *numero*.

```
<animal numero = " ..." > chat </animal> ,
```

puis modifier la DTD en conséquence.

4/ Un document avec une personne ayant le même numéro qu'un animal, est-il valide ?

5/ Un document avec une personne dont le père fait référence à un animal est-il valide ?

Exercice 3 : DTD d'une bibliothèque

Une bibliothèque désire réaliser une petite gestion de ses livres via des documents XML. Un livre peut être caractérisé par les informations suivantes :

- son auteur
- son titre
- son année de parution
- son éditeur
- son appartenance à une catégorie : Action / Policier / Aventure / Biographie
- son numéro ISBN (facultatif)

1/ Ecrire une DTD permettant le stockage de ces informations (fichier "dtdLivre.dtd"). Le numéro ISBN sera stocké dans un attribut "isbn"

2/ Ecrire un document contenant les livres suivants :

auteur : Verchu, titre : La mer, année : 2001, éditeur : Flammarion, catégorie : Action, numéro ISBN : 15555

auteur : Aragone, titre : Le père, année : 1946, éditeur : Hachette, catégorie : Aventure

3/ Enrichir la DTD ci-dessus afin de permettre de stocker les informations relatives à un éditeur. Un éditeur sera caractérisé par son nom et son adresse.

4/ Compléter les données XML avec le nom et l'adresse des éditeurs Flammarion et Hachette

Exercice 4 : DTD générique du modèle relationnel

Question 1: Ecrire une DTD *schema-relationnel.dtd* pour représenter un schéma relationnel quelconque. Pour cela, pensez aux définitions de schémas en SQL (avec l'instruction *create table*) et adaptez sa syntaxe à la syntaxe XML. Par exemple :

- Un **schéma** relationnel a un **nom** et plusieurs **relations**. Une relation a un nom, des **attributs** et des contraintes.
- Un **attribut** a un **nom** et un **type**. Il peut être *null* ou *not null* et avoir une valeur par **défaut**.
- Un **type** est un nombre **entier** ou **décimal**, une **date** ou une **chaîne** de caractères.
 - La taille d'une chaîne de caractères est son nombre de caractères.
 - La taille d'un nombre entier est son nombre de chiffres. Pour un nombre décimal on connaît aussi la taille de la partie décimale.
- Une **contrainte** porte sur un ou plusieurs attributs d'une relation. Représenter seulement les contraintes suivantes : contrainte d'**unicité**, **clé primaire** et **clé étrangère** qui fait référence à la clé primaire d'une relation.

Question 2 : Soit le schéma *Cinema* suivant :

REALISATEUR (nom, âge)

PRODUCTEUR (nom, budget)

FILM (numFilm, titre, réalisateur, producteur)

PROGRAMME (salle, jour, numFilm, nbEntrées)

titre est une clé minimale de FILM ; l'attribut *numFilm* de PROGRAMME est une clé étrangère qui référence l'attribut de même nom de FILM ; il y a deux étages 1 et 2 dans un cinéma, chacun comprenant deux salles, A et B (par défaut, la salle est 1A); l'âge et le budget ne sont pas obligatoires ; le nombre d'entrées *nbEntrees* est obligatoire.

Décrire le schéma *Cinema* : pour cela créer le fichier *schema-cinema.xml* conforme à la DTD *schema-relationnel.dtd* définie précédemment.

Question 3 : Une fois le schéma défini, il faut introduire les données dans la base. Nous voulons que les données de chaque schéma relationnel soient introduites dans un document XML qui suivra une DTD dédiée à ce schéma relationnel. Cette DTD est générée à partir de la définition du schéma (par exemple *cinema.xml*) au travers d'un ensemble de règles. Déterminez cet ensemble pour que la DTD puisse prendre en compte les contraintes d'intégrité suivantes : attribut non nul avec valeur par défaut, unicité, clé primaire composée de plusieurs attributs, intégrité référentielle, contrainte de domaine, contrainte de tuple, contrainte globale.

Créer la DTD pour le schéma *Cinéma*.

Question 4 : Finalement, écrire l'instance de *Cinema.dtd* qui permet d'introduire les tuples («A», 32), («B», 5000) et (4, «Le fabuleux destin d'Amelie Poulain», «A», «B») dans les relations REALISATEUR, PRODUCTEUR et FILM, respectivement.

Exercice 5 (facultatif): XML → modèle relationnel

Soit le document XML suivant:

```
<?xml version="1.0" standalone="yes" ?>
<!DOCTYPE R [
  <!ELEMENT R (A)*>
  <!ELEMENT A (B?, C+)>
  <!ELEMENT B (#PCDATA)>
  <!ELEMENT C ( (D,E) | (E,D) )>
  <!ELEMENT D (#PCDATA)>
  <!ELEMENT E (#PCDATA)>
]>
<R>
  <A>
    <B>v1</B>
    <C> <D>v2</D> <E>v3</E> </C>
    <C> <E>v4</E> <D>v5</D> </C>
  </A>
  <A>
    <C> <E>v6</E> <D>v7</D> </C>
    <C> <D>v2</D> <E>v3</E> </C>
    <C> <E>v4</E> <D>v5</D> </C>
  </A>
</R>
```

a) Traduire la DTD en un schéma relationnel en éliminant les redondances au maximum, en indiquant les diverses contraintes (clé primaire, valeur non nulle et intégrité référentielle) et en préservant l'ordre d'occurrence des éléments de type C.

b) Donner les tuples du schéma relationnel qui correspondent au document XML.