



Votre numéro d'anonymat :

Bases de Données Réparties Avancées – M2

Examen du 26 janvier 2006

CORRIGÉ

Partie Bases de Données XML (B. Amann)

Documents autorisés

Les calculatrices, baladeurs et autres appareils électroniques sont interdits. Les téléphones mobiles doivent être éteints et rangés dans les sacs. Le barème sur 8 points (8 questions) n'a qu'une valeur indicative.

1 Grammaires d'arbres (3 pts)

Soit donné une DTD XML d'une agence de location de films (location.dtd) :

```
<!ELEMENT agence ( adresse , ( film ) * ) >
<!ELEMENT film ( titre , annee , format , ( artiste ) + ) >
<!ELEMENT titre (#PCDATA) >
<!ELEMENT annee (#PCDATA) >
<!ELEMENT format ( vhs | dvd ) , ( stereo | mono | dolby | dts ) >
<!ELEMENT artiste ( nom , role ? ) >
<!ELEMENT vhs (EMPTY) >
<!ELEMENT dvd (EMPTY) >
<!ELEMENT stereo (EMPTY) >
<!ELEMENT mono (EMPTY) >
<!ELEMENT dolby (EMPTY) >
<!ELEMENT dts (EMPTY) >
<!ELEMENT agence (#PCDATA) >
<!ELEMENT role (#PCDATA) >
```

Un document contient l'adresse de l'agence et les films à louer. Pour chaque film il contient le titre, l'année, le format et les artistes, dont le premier est le metteur en scène et les autres sont les acteurs. Un film est distribué en format VHS ou DVD avec différentes options pour le son (stéréo, mono, dolby ou dts).

Question 1 (1 point)

Traduisez cette DTD en grammaire d'arbre. On suppose l'existence de deux types #PCDATA (pour les chaînes de caractère) et EMPTY pour le type "vide".

Solution: Agence → agence (Adresse, Film+)
 Film → film (Titre, Annee, Format, (Artiste)+)
 Titre → titre #PCDATA
 Annee → annee #PCDATA
 Artiste → artiste (Nom, Role)
 Format → (VHS | DVD) (Stereo | Mono | Dolby | DTS)
 Artiste → artiste (Nom, Role ?)
 VHS → vhs EMPTY
 DVD → dvd EMPTY
 Adresse → adresse #PCDATA
 Nom → nom #PCDATA
 Role → role #PCDATA
 Stereo → stereo EMPTY
 Mono → mono EMPTY
 Dolby → dolby EMPTY
 DTS → dts EMPTY

Question 2 (1 point)

On veut maintenant valider uniquement les documents où

1. un film en VHS est en stéréo ou en mono,
2. un film en DVD propose un choix parmi les différents formats audio (mono, stéréo, dolby et dtd) et
3. seulement les acteurs ont un sous-élément `role`.

Indiquez les modifications nécessaires dans la grammaire d'arbre précédente.

Solution: Film → Film (Titre Annee Format Metteur (Acteur)*)
 Format → format (VHS (Stereo | Mono) | DVD (Stereo | Mono | Dolby | DTS)*)
 Metteur → artiste (Nom)
 Acteur → artiste (Nom, Role)
 ...

Question 3 (1/2 point)

Est-ce qu'il est possible d'exprimer les contraintes 1 et 2 sur les formats vidéo/audio dans une DTD. Si oui, donnez les modifications nécessaires dans la DTD d'origine. Sinon, justifiez votre réponse.

Solution: Oui, il est possible d'exprimer les contraintes 1 et 2 en DTD :
 <!ELEMENT format ((vhs (stereo | mono)) | (dvd (stereo | mono | dolby | dts)*)) >

Question 4 (1/2 point)

Est-ce qu'il est possible d'exprimer la contrainte 3 sur la différence entre les metteurs en scène et les acteurs dans une DTD. Si oui, donnez les modifications nécessaires dans la DTD d'origine. Sinon, justifiez votre réponse.

Solution: Non, la troisième contrainte ne peut pas être exprimée en DTD (Metteur et Acteur sont deux types concurrents)

2 Stockage et interrogation de données XML (3 pts)

Voici un schéma de stockage relationnel pour stocker des documents validés par la DTD `location.dtd` de l'exercice 1 :

Agence(idAg, adresse)

Film(idFil, pos, idAg, titre, annee, video, audio)

Artiste(idArt, pos, idFil, nom, role)

L'attribut `pos` permet de maintenir l'ordre d'un document en indiquant pour chaque noeud sa position parmi ses frères. Les formats vidéo et audio sont représentés comme valeurs de deux attributs `video` et `audio`.

Question 5 (1 point)

Donnez les requêtes XPath et SQL qui retournent les titres de tous les films de Clint Eastwood comme metteur en scène (premier artiste). Attention, Clint Eastwood peut aussi être seulement acteur dans un film.

Solution:

XPath : `//film[artiste[position()=1 and nom = "Eastwood"]]`

SQL :

```
select titre
from Film , Artiste
where Film.idFil = Artiste.idFil
and Artiste.pos = 1
and Artiste.nom = 'Eastwood'
```

Question 6 (2 points)

On suppose que la plupart des clients cherchent les films à partir de leur metteur en scène et le format vidéo. Expliquez comment on pourrait modifier le schéma relationnel pour d'améliorer le temps de réponse pour ce type de requêtes fréquentes ("workload").

Solution:

1. On peut stocker le nom du metteur en scène avec le film et les acteurs séparément.
2. On crée deux tables séparées pour les films en DVD et les films en VHS.

Agence(idAg, adresse)

FilmDVD(idFil,pos,idAg, titre, annee, metteur, audio)

FilmVHS(idFil,pos,idAg, titre, annee, metteur, audio)

Artiste(idArt,pos,idFil, nom, role)

3 Évaluation et optimisation de requêtes (2 pts)

Soit donnée une relation `Noeud(pre, post, prof, lab)` qui permet de stocker des arbres étiquetés et ordonnés (arbres XML simplifiés). Chaque noeud est représenté sous forme d'un n-uplet qui contient son identifiant structurel (`pre, post, prof`) et son étiquette (`lab`). L'attribut `prof` indique la profondeur d'un élément dans l'arbre (valeur 0 pour l'élément racine). Traduisez les requêtes XPath suivantes en requêtes SQL sur cette table (on suppose que chaque requête SQL retourne le couple (`pre, post`) des noeuds identifiés par la requête XPath) :

Question 7 (1 point)

`/child::A/child::B`

Solution:

```
select B.pre , B.post
from Noeud A, Noeud B
where A.lab = 'A'
and B.pre > A.pre
and B.post < A.post
and B.prof = 1
```

Question 8 (1 point)

`/descendant::A[not(preceding::B)]`

Solution:

```
select B.pre , B.post
from Noeud A
where A.lab = 'A'
and not exists (select *
                from Noeud B
                where B.pre < A.pre)
```