

Examen de BDA, M1 MIAGE INITIAL ET INFORMATIQUE, 2013

14 mai 2013

Les documents sont permis. Le barème est donné à titre indicatif et peut être modifié. ECRIVEZ LE NOM DE VOTRE PARCOURS DANS LA PARTIE CACHEE DE LA COPIE, A COTE DE VOTRE NOM.

Exercice 1 : Modélisation Objet : ODL, 7 points

On considère une base de données pour une bibliothèque universitaire. La base contient des informations sur les étudiants, les livres, et les emprunts des livres par les étudiants. Un étudiant a un nom, un prénom, un numéro de carte d'étudiant, une adresse e-mail, parfois un numéro de téléphone (pas plus qu'un, si il en a un), et un ou plusieurs départements (un étudiant peut faire, en même temps, un L3 Math et un M1 Informatique, par exemple). Pour ce qui concerne les départements, on connaît juste leur sigles (INFO, BIO, MATH,...), qui agissent comme des identifiants, et le numéro (unique) de téléphone du secrétariat. Un livre a un code, un titre, un ou plusieurs auteurs (dont on connaît juste le numéro de sécurité sociale -le NSS-, le nom et le prénom) et une maison d'édition (dont on connaît juste le nom). On supposera que la bibliothèque peut avoir plusieurs exemplaires de chaque livre, et, pour chaque livre, on connaît son nombre d'exemplaires. Un exemplaire d'un livre est identifié par le code du livre suivi par un nombre. Un exemplaire d'un livre peut être emprunté par un étudiant au maximum, pour une période donnée, mais un étudiant peut emprunter plusieurs livres au même temps.

1. Donner un schéma ODL pour cette base. On indiquera de façon claire les classes définies, et, pour chaque classe, ses attributs et ses *relationships* éventuelles. On indiquera aussi la clé de la classe.
2. Si jamais des contraintes qui se trouvent dans la description informelle de la base (de façon explicite ou implicite) ne sont pas exprimées de façon adéquate par votre schéma ODL, on en fera la remarque, et on expliquera pourquoi on ne les a pas exprimées.

CORRECTION : Juste une remarque, sur l'Exercice 1

Il existe plusieurs modélisations possibles, et je les ai toutes, plus ou moins, acceptées. Ce que, évidemment, je n'ai pas pu accepter : des modélisations où on ne pouvait pas retrouver qui avait emprunté quoi! Et j'ai vu des choses de ce type...

Exercice 2 : Requêtes Xquery, 13 points AVEC CORRECTIONS

NB : je n'ai pas testé les solutions sur machine, donc quelques erreurs de syntaxe peuvent être là. Mais la "structure logique" des solutions est OK.

On considère à nouveau la base de données mondial.xml vue en TD, dont le DTD est donné en annexe. Ecrire les requêtes Xquery suivantes :

1. Donner le nombre des langues parlées en France.
`count(doc("mondial.xml")/country[name="France"]/languages)`

2. Donner les noms des religions qui sont pratiquées en Grèce (*Greece*), mais pas en France.

```
(doc("mondial.xml")/country[name="Greece"]/religions)
except
(doc("mondial.xml")/country[name="France"]/religions)
```

Ce qui ne marchera pas : balayer avec un `for $r1` les religions de Grèce, balayer avec un `for $r2` les religions de France, et garder une valeur de `$r1` si elle est différente d'une valeur de `$r2`. Par exemple, supposons qu'en Grèce on pratique A et B, et en France B et C. On voudrait juste A comme réponse. Mais quand on compare la valeur B de `$r1` avec la valeur C de `$r2`, on garde B aussi, comme réponse!

Une autre façon correcte de faire est utiliser un "not (some... satisfies...)" (construction vue en cours pour s'assurer que la valeur courante de `$r1` n'appartient pas aux religions de France.

3. Donner les noms des religions qui sont pratiquées en France et en Grèce, et, pour chacune d'elles, l'information sur le pourcentages respectifs dans les deux pays.

Oublier le pourcentage (j'avais mal lu le DTD) ; j'ai corrigé juste la partie : Donner les noms des religions qui sont pratiquées en France et en Grèce.

```
(doc("mondial.xml")/country[name="Greece"]/religions)
intersect
(doc("mondial.xml")/country[name="France"]/religions)
```

4. Donner les noms des religions pratiquées dans les pays frontaliers à la France, en évitant les doublons.

```
let $rel := for $c in doc('mondial.xml')/country
where $c/ border[@country='FR']/religions
return $c/religions
return distinct-values($rel)
```

Idée : la variable `$rel` reçoit la liste de toutes les religions des pays qui ont comme pays frontalier la France. Puis on élimine les doublons.

5. Donner les noms des langues éventuelles qui sont parlées seulement en Belgique (*Belgium*).

Une façon possible de faire :

```
for $l in doc('mondial.xml')/country[name='Belgium']/languages where not
(some $c in for $c in doc('mondial.xml')/country satisfies
($c/name != 'Belgium' and $c/religions = $l)
)
return
$l
```

Attention à ne pas faire des erreurs semblables à celles indiquées pour la requête 3.

6. Donner les noms des pays éventuels où la religion musulmane (*Muslim*) n'est pas pratiquée.

Le type de difficulté est semblable à celui de la requête ci-dessus.

7. Afficher les noms des 3 derniers pays (selon l'ordre dans le document) avec ce format :

```
Nom_du_dernier_pays
Nom_de_l'avant_dernier_pays
Nom_de_l'avant_avant_dernier_pays
```

```
let $n = count( doc('mondial.xml')/country)
return
(doc('mondial.xml')/country[$n]/name,
doc('mondial.xml')/country[$n-1]/name,
doc('mondial.xml')/country[$n-2]/name
)
```

8. Pour chaque pays, afficher le nom de sa capitale, selon le format de résultat suggéré ci-dessous (ce n'est pas important de respecter les retours à la ligne de l'exemple).

*

La capitale du pays nommé Albania est Tirane

*

La capitale du pays nommé Greece est Athens

Je saute cette correction, qui est facile.

9. Pour chaque pays, s'il a au maximum deux pays frontaliers, afficher le nom du pays, suivi par ":", puis par les noms de ce deux pays, tandis que s'il en a plus, on affichera le nom du pays, suivi par ":", puis par les noms des deux premiers pays frontaliers (selon l'ordre avec lequel on les trouve), puis par "," et, enfin, par l'expression "et d'autres encore".

Je saute cette correction aussi; il est clair que il faudra une construction de la forme : return (if... then ... else ...).

10. Poser la requête booléenne suivante :

Est-il vrai que le pays ayant la surface (*area*) la plus grande est aussi celui où on parle le plus de langues ?

```
let $a1:= for $a in doc("mondial.xml")//country order by ($a/@area)
return $a/name
let $a2:= for $a in doc("mondial.xml")//country order
by (count($a/languages))
return $a/name
return $a1[last()] is $a2[last()]
```