

### Exercice 1 : Création d'une base de données XML

Soit une mini-base de données permettant de stocker les cours et les enseignants de M1 sous la forme de deux documents XML dont les DTD sont fournies ci-dessous.

```
<!-- cours.dtd -->
<!ELEMENT coursM1 (enseignement)*>
<!ELEMENT enseignement (titre, semestre, caractere, ects)>
<!ATTLIST enseignement id ID #REQUIRED enseignant CDATA "">
<!ELEMENT titre (#PCDATA)>
<!ELEMENT semestre (#PCDATA)>
<!ELEMENT caractere (#PCDATA)> <!-- Obligatoire ou optionnel -->
<!ELEMENT ects (#PCDATA)>

<!-- enseignants.dtd -->
<!ELEMENT enseignantsM1 (enseignant)*>
<!ELEMENT enseignant (nom, prenom, statut, faculte?)>
<!ATTLIST enseignant id ID #REQUIRED>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT prenom (#PCDATA)>
<!ELEMENT statut (#PCDATA)>
<!ELEMENT faculte (#PCDATA)>
```

1. Télécharger les documents XML suivants et les sauvegarder en local.
  - <http://eric.univ-lyon2.fr/~jdarmont/docs/cours.xml>
  - <http://eric.univ-lyon2.fr/~jdarmont/docs/enseignants.xml>
2. Lancer le logiciel client du SGBD natif XML eXist à partir de l'adresse <http://disora.univ-lyon2.fr:8088/exist/webstart/exist.jnlp> (« Ouvrir avec »).
3. Vous connecter avec votre login et votre mot de passe habituel.
4. Sélectionner votre base de données en double-cliquant sur votre login ou en utilisant la commande `cd "votre_login"`.
5. Insérer les documents *cours.xml* et *enseignants.xml* dans la collection (menu Fichier/Charger des fichiers). Visualiser le contenu des documents en double-cliquant sur leurs noms.

### Exercice 2 : Requêtes XPath

À l'aide de l'interface de requêtes du client eXist (bouton « jumelles »), formuler les requêtes suivantes à l'aide d'expressions de chemin.

1. Éléments *enseignement* du document *cours.xml* en indiquant explicitement le nom de ce dernier, puis en l'omettant.
2. Titres de tous les enseignements.
3. Identifiants de tous les enseignants (à partir du document *enseignants.xml*). Commentaire ?
4. Caractéristiques (titre, semestre, caractère, ECTS) du 10<sup>ème</sup> enseignement.
5. Titres des 5 premiers enseignements.

6. Titres des enseignements du semestre 2.
7. Caractéristiques des enseignements de l'enseignant JD au semestre 1.
8. Titres des enseignements associés à un enseignant (projets et stages ne le sont pas).
9. Noms des enseignants appartenant à une faculté.
10. Titres, caractères et crédits ECTS de tous les enseignements.

### Exercice 3 : Requêtes FLWOR

Toujours à l'aide de l'interface de requêtes du client eXist (bouton « jumelles »), formuler les requêtes suivantes à l'aide d'expressions FLWOR.

1. Afficher tous les chiffres de 0 à 9, en colonne dans 10 éléments XML `<chiffre> </chiffre>`.
2. Afficher tous les chiffres de 0 à 9, en ligne dans 1 élément XML `<chiffres> </chiffres>`.
3. Titres de tous les enseignements.
4. Nom et position dans le document *enseignants.xml* de chaque enseignant, au format XML `<enseignant pos=" " > </enseignant>`.
5. Titres des enseignements du semestre 2.
6. Caractéristiques des enseignements de l'enseignant JD au semestre 1.
7. Titres des enseignements associés à un enseignant.
8. Noms et prénoms des enseignants appartenant à une faculté.
9. Noms des enseignants, par ordre alphabétique inverse.
10. Enseignements, par ordre de coefficient (ECTS) décroissant et par ordre alphabétique.
11. Titres de tous les enseignements, par ordre alphabétique, dans une liste à puces XHTML.
12. Idem, mais en rajoutant un lien (`<a href=" " > </a>`) autour de chaque titre et pointant vers la page *ID.html*, où ID est l'identifiant de l'enseignement concerné.
13. Caractéristiques (id, nom, prénom, statut) des enseignants sous forme de tableau XHTML. Trier le résultat par identifiants croissants.
14. Afficher les titres des enseignements dans des balises XML `<s1> </s1>` pour ceux du semestre 1 et `<s2> </s2>` pour ceux du semestre 2.
15. Idem en rajoutant en attributs des balises `<s1>` et `<s2>` l'identifiant et les crédits ECTS de chaque cours.
16. Liste des enseignants au format suivant.

```
<enseignant>
  <nom>NOM EN MAJUSCULES</nom>
  <prenom>PRENOM EN MAJUSCULES</prenom>
</enseignant>
```
17. Identifiants de tous les enseignants (à partir du document *cours.xml*), sans doublon.
18. Nombre de cours.
19. Minimum, maximum et moyenne des crédits ECTS.
20. Titre du cours placé en dernière position dans le document *cours.xml*.

#### Exercice 4 : Requêtes complexes

Toujours à l'aide de l'interface de requêtes du client eXist (bouton « jumelles »), formuler les requêtes XQuery suivantes.

1. Titres des enseignements, ainsi que le nom et le prénom de l'enseignant correspondant au format XML suivant (jointure). Proposer deux solutions équivalentes.

```
<enseignement>
  <titre>Titre du cours</titre>
  <enseignant>Nom, prénom</enseignant>
</enseignement>
```

2. Nombre total d'ECTS par semestre (regroupement). Indiquer le numéro de semestre en attribut du résultat exprimé en XML.

3. Nombre de cours par enseignant. Indiquer le nom et le prénom des enseignants (regroupement + jointure). Trouver un format XML approprié pour l'affichage. Commentaire sur la fonction `distinct-values()` ?

4. Nombre de cours par enseignant et par semestre. Indiquer l'identifiant des enseignants (regroupement multiple + jointure). Commentaire sur l'utilisation de `distinct-values()` ?

5. Total des crédits ECTS par faculté et par semestre. Commentaire ?

#### Correction Exercice 2

```
<!-- 1 -->
doc("cours.xml")/coursM1/enseignement
/coursM1/enseignement
//enseignement

<!-- 2 -->
//titre

<!-- 3 -->
doc("enseignants.xml")//data(@id)

<!-- 4 -->
//enseignement[10]/*

<!-- 5 -->
//enseignement[position()<6]/titre

<!-- 6 -->
//enseignement[semestre="2"]/titre

<!-- 7 -->
//enseignement[@enseignant="JD" and semestre="1"]/*

<!-- 8 -->
//enseignement[@enseignant]/titre

<!-- 9 -->
//enseignant[faculte]/nom

<!-- 10 -->
//titre | //caractere | //ects
```

#### Correction Exercice 3

```
<!-- 1 -->
for $i in (0 to 9)
return <chiffre>{$i}</chiffre>

<!-- 2 -->
let $i := (0 to 9)
return <chiffres>{$i}</chiffres>

<!-- 3 -->
for $e in //enseignement
return $e/titre

<!-- 4 -->
for $e at $p in //enseignant/nom
return <enseignant pos="{ $p }">{data($e)}</enseignant>

<!-- 5 -->
for $e in //enseignement
where $e/semestre="2"
return $e/titre

<!-- 6 -->
for $e in //enseignement
where $e/@enseignant="JD" and $e/semestre="1"
return $e/*
```

```

<!-- 7 -->
for $e in //enseignement
where $e/@enseignant
return $e/titre

<!-- 8 -->
for $e in //enseignement
where $e/faculte
return $e/nom | $e/prenom

<!-- 9 -->
for $e in //enseignant
order by $e/nom descending
return $e/nom

<!-- 10 -->
for $e in //enseignement
order by $e/ects descending, $e/titre
return $e

<!-- 11 -->
<ul> {
for $e in //enseignement/titre
order by $e
return <li>{data($e)}</li>
} </ul>

<!-- 12 -->
<ul> {
for $e in //enseignement
order by $e/titre
return <li><a href="{data($e/@id)}.html">{data($e/titre)}</a></li>
} </ul>

<!-- 13 -->
<table>
<tr> <th>ID</th> <th>Nom</th> <th>Prénom</th> <th>Statut</th> </tr> {
for $e in //enseignant
order by $e/@id
return <tr> <td>{data($e/@id)}</td> <td>{data($e/nom)}</td>
<td>{data($e/prenom)}</td> <td>{data($e/statut)}</td> </tr>
} </table>

<!-- 14 -->
for $e in //enseignement
return if ($e/semestre="1")
then <s1>{data($e/titre)}</s1>
else <s2>{data($e/titre)}</s2>

<!-- 15 -->
for $e in //enseignement
return if ($e/semestre="1")
then <s1 id="{data($e/@id)}"
ects="{data($e/ects)}">{data($e/titre)}</s1>
else <s2 id="{data($e/@id)}"
ects="{data($e/ects)}">{data($e/titre)}</s2>

<!-- 16 -->
for $e in //enseignant
let $nmaj := upper-case($e/nom), $pmaj := upper-case($e/prenom)
return <enseignant> <nom>{$nmaj}</nom> <prenom>{$pmaj}</prenom> </enseignant>

```

```

<!-- 17 -->
for $e in distinct-values//@enseignant)
return <id_enseignant>{data($e)}</id_enseignant>

<!-- 18 -->
let $n := count(//enseignement)
return <resultat>{$n}</resultat>

<!-- 19 -->
let $minects := min(//ects), $maxects := max(//ects), $moyects := avg(//ects)
return <resultat>
<minects>{$minects}</minects>
<maxects>{$maxects}</maxects>
<moyects>{$moyects}</moyects>
</resultat>

<!-- 20 -->
let $e := //enseignement[last()]/titre
return $e

```

#### Correction Exercise 4

```

<!-- 1 -->
for $e1 in //enseignement,
    $e2 in //enseignant
where $e1/@enseignant = $e2/@id
return <enseignement>
<titre>{data($e1/titre)}</titre>
<enseignant>{data($e2/nom)}, {data($e2/prenom)}</enseignant>
</enseignement>

<!--lbis -->
for $e1 in //enseignement,
    $e2 in //enseignant[@id = $e1/@enseignant]
return <enseignement>
<titre>{data($e1/titre)}</titre>
<enseignant>{data($e2/nom)}, {data($e2/prenom)}</enseignant>
</enseignement>

<!-- 2 -->
for $s in distinct-values(//semestre)
let $totects := sum(//enseignement[semestre = $s]/ects)
return <totects semestre="{s}">{$totects}</totects>

<!-- 3 -->
for $e in //enseignant
let $n := count(//enseignement[@enseignant = $e/@id])
order by $e/nom, $e/prenom
return <enseignant>
<nom>{data($e/nom)}</nom>
<prenom>{data($e/prenom)}</prenom>
<nb_cours>{$n}</nb_cours>
</enseignant>

```

```

<!-- 4 -->
For $e in distinct-values(//enseignant/@id),
  $s in distinct-values(//enseignement/semestre)
let $n := count(//enseignement[@enseignant = $e and semestre = $s])
order by $e, $s
return <enseignant>
      <id>{data($e)}</id>
      <semestre>{data($s)}</semestre>
      <nb_cours>{$n}</nb_cours>
    </enseignant>

<!-- 5 -->
For $f in distinct-values(//faculte),
  $s in distinct-values(//semestre)
let $t := sum(//enseignement[semestre = $s and @enseignant =
//enseignant[faculte = $f]/@id]/ects)
order by $f, $s
return <groupe>
      <faculte>{data($f)}</faculte>
      <semestre>{data($s)}</semestre>
      <tot_ects>{$t}</tot_ects>
    </groupe>

```