

Modèles et langages bases de données avancées - S1

[Tableau de bord](#) / [Mes cours](#) / [MU4IN801 - S1](#) / [Projet SQL3/XML](#) / [Description du projet](#)

[↑ Retour à « Projet SQL3/XML »](#)

Description du projet

- Ce mini-projet sera noté sur **20 points et représente 20% de la note finale**.
- Date limite pour le dépôt sur Moodle : **18/12/2020 à minuit**
- Format de dépôt:
 - Un **fichier PDF** nommé *projetMLBDA-NOM-Prenom.pdf* contenant un résumé de ce que vous avez réussi à faire. Préciser également si chaque doc XML produit est conforme ou non à sa DTD.
 - **3 fichiers sql** : *projetMLBDA-ex1-NOM-Prenom.sql* (idem pour ex2 ex3 et ex4.sql (optionnel). Ajouter des phrases détaillées de commentaire pour expliquer les étapes de votre solution.

Explications

L'objectif de ce projet est de générer des documents XML à partir des données stockées dans une base de données relationnelle en utilisant SQL3 et le type abstrait XMLType implanté dans Oracle.

L'évaluation prendra en compte l'originalité de la solution et les pistes d'ouvertures explorées (par exemple dans l'utilisation du type XMLType).

Description de la base Mondial:

- <http://www-bd.lip6.fr/wiki/site/enseignement/master/mlbda/tmes/tpmondial>
- site web: <https://www.dbis.informatik.uni-goettingen.de/Mondial/>

Installation :

- Vous devez charger et exécuter le fichier [mondial_synonym.sql](#)

Exemple

Pour vous aider, vous pouvez télécharger et exécuter le fichier [exemple.sql](#) qui génère un document XML respectant la DTD suivante à partir des tables de la base Mondial:

DTD Exemple:

```
<!ELEMENT pays (nom, code, montagne*)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT code (#PCDATA)>
<!ELEMENT montagne (nom)>
```

L'exemple utilise le type XMLType implémenté dans Oracle pour générer et modifier des fragments XML.

Documentation XMLType dans Oracle 11: https://docs.oracle.com/cd/B28359_01/appdev.111/b28369/toc.htm

Exercice 1

Pour chaque DTD donnée, écrire en SQL3:

- Créer les types SQL3 et les tables objet pour stocker les données de la base Mondial nécessaires pour générer les documents XML demandés.
- Définir les méthodes pour générer les documents XML contenant les données de la base Mondial.
- Validation: vérifier que votre résultat XML est conforme à la DTD à traiter (*cf.* la validation vue en TME).

Remarque: Les types des objets et les méthodes peuvent varier selon la DTD à traiter.

Indication : s'inspirer de l'exemple ci-dessus.

DTD 1: La base mondial contient plusieurs pays. Un pays est associé à un ou plusieurs continents, il contient des provinces et éventuellement des

aéroports. Une province peut contenir des montagnes et des déserts, puis des îles.

Attention: la DTD a été modifiée:

- la définition des attributs de l'élément <country> a été modifiée (voir aussi le forum du projet):

```
<!ELEMENT mondial (country+) >

<!ELEMENT country (continent+, province+, airport*) >
<!ATTLIST country idcountry ID #REQUIRED
                  nom CDATA #REQUIRED>

<!ELEMENT province ( (mountain|desert)*, island* ) >
<!ATTLIST province name CDATA #REQUIRED
                  capital CDATA #REQUIRED >

<!ELEMENT mountain EMPTY >
<!ATTLIST mountain name CDATA #REQUIRED
                  height CDATA #REQUIRED >

<!ELEMENT desert EMPTY >
<!ATTLIST desert name CDATA #REQUIRED
                  area CDATA #IMPLIED >

<!ELEMENT island (coordinates?) >
<!ATTLIST island name CDATA #REQUIRED >

<!ELEMENT coordinates EMPTY >
<!ATTLIST coordinates latitude CDATA #REQUIRED
                  longitude CDATA #REQUIRED>

<!ELEMENT continent EMPTY >
<!ATTLIST continent name CDATA #REQUIRED
                  percent CDATA #REQUIRED >

<!ELEMENT airport EMPTY>
<!ATTLIST airport name CDATA #REQUIRED
                  nearCity CDATA #IMPLIED >
```

DTD 2 :

Attention: la DTD a été modifiée:

- l'identifiant code est optionnel dans l'élément <country>
- on peut générer des <country> sans sous-élément <language>

Remarque: dans cette DTD, un pays même peut apparaitre plusieurs fois dans différentes organisations. Vous pouvez donner votre avis sur les avantages et inconvénients d'une telle redondance.

```
<!ELEMENT mondial (organization+) >
<!ELEMENT organization (country+, headquarter) >

<!ELEMENT country (language*, borders) >
<!ATTLIST country code CDATA #IMPLIED
                  name CDATA #REQUIRED
                  population CDATA #REQUIRED >

<!ELEMENT language EMPTY >
<!ATTLIST language language CDATA #REQUIRED
                  percent CDATA #REQUIRED >

<!ELEMENT borders (border*) >

<!ELEMENT border EMPTY>
<!ATTLIST border countryCode CDATA #REQUIRED
                  length CDATA #REQUIRED >

<!ELEMENT headquarter EMPTY>
<!ATTLIST headquarter name CDATA #REQUIRED>
```

Exercice 2

Dans cet exercice vous devez définir des DTD dont certains attributs sont calculés : la valeur de l'attribut est le résultat d'un appel de méthode. On suppose que la **racine** contient une sequence d'elements country :

```
<!ELEMENT ex2 (country+) >
```

- L'élément country a un élément geo qui contient la liste distincte de toutes les montagnes, déserts et iles du pays. Écrire la méthode qui

retourne cette liste qui sera constituée à partir de la liste de montagnes, déserts et îles des provinces du pays. Les éléments mountain, desert et island sont ceux de la DTD1.

```
<!ELEMENT country (geo) >
<!--ATTLIST country name CDATA #REQUIRED -->
<!ELEMENT geo ( (mountain|desert)*, island* ) >
```

2. Ajouter à l'élément country un élément peak qui contiendra la hauteur de sa plus haute montagne (attribut **height**). La méthode que vous devez écrire retourne 0 si le pays n'a pas de montagnes.

```
<!ELEMENT country (geo, peak?) >
<!ELEMENT peak EMPTY >
<!--ATTLIST peak height CDATA #REQUIRED -->
```

3. L'élément country a son continent principal (un seul continent principal par pays) sous forme d'attribut. Ajouter aussi un élément contCountries qui contient la liste de tous les pays qui se trouvent sur le même continent que ce pays. L'élément border est celui de la DTD2. Vous pouvez supposer que contCountries contient les pays **frontaliers** qui se trouvent (au moins en partie) sur le même continent que le pays courant.

Bonus pour ceux qui s'intéressent à la *réursion* : vous pouvez considérer que contCountries contient les pays directement **ou indirectement** frontaliers qui se trouvent (au moins en partie) sur le même continent que le pays courant. Par exemple pour la France, contCountries contient l'Autriche et le Portugal mais pas l'Angleterre qui est pourtant en Europe.

```
<!ELEMENT country (contCountries) >
<!--ATTLIST country name CDATA #REQUIRED continent CDATA #REQUIRED -->

<!ELEMENT contCountries (border*) >
```

4. L'élément country a la longueur totale de sa frontière sous forme d'attribut blength. La longueur totale de sa frontière est calculée comme étant la somme des longueurs des frontières avec chacun de ses voisins.

```
<!ELEMENT country (contCountries) >
<!--ATTLIST country name CDATA #REQUIRED blength CDATA #REQUIRED -->
```

Exercice 3

Dans cet exercice vous devez définir une nouvelle DTD et exporter les données de telle manière qu'il est possible de répondre avec des expressions XPath aux requêtes suivantes. Votre rapport PDF doit contenir les **requêtes Xpath** accompagnées de quelques phrases d'explication.

- 1- La séquence des pays qui détiennent la première place en terme de total de population par rapport aux pays du même continent.
- 2- La séquence des noms d'organisations auxquelles chaque pays est affilié triés par la date de création de ces organisations.
- 3- Pour une province donnée la description (nom, altitude, latitude, longitude) de la plus haute montagne qui s'y trouve, si c'est le cas ou l'information qu'aucune montagne ne s'y trouve le cas échéant.
- 4- Pour un pays donné la séquence des rivières qui prennent source dans ce pays (sans ordre particulier).
- 5- Le pays avec la plus longue frontière.

Exercice 4 (ouverture)

On vous propose plusieurs pistes d'exploration optionnelles :

- utilisation de XMLType pour la validation des documents XML
- transformation des DTDs en XML Schema et validation
- utilisation de XMLType pour interroger les fragments avec SQL+Xpath (méthode extractValue)
- définir et évaluer des nouvelles requêtes en XQuery sur des documents exportés

Modifié le: mercredi 16 décembre 2020, 08:34