

TP N° XML et les BDs

Exercice I :

L'objectif de cet exercice est de vous familiariser avec les fonctions élémentaires de

JDeveloper pour créer et ouvrir des documents XML et avoir un aperçu de l'apparence par défaut d'un document XML dans un navigateur.

1. Démarrez Oracle JDeveloper.
2. Ouvrez un espace de travail appelé XMLFundamentalsWorkspace.jws, dans le répertoire « labs ». Cet espace de travail contient un projet nommé XMLLabsProject.
3. Créez un document XML appelé employees.xml.
4. Ajoutez l'élément XML employees suivant après la déclaration XML :

```
<?xml version='1.0' encoding='windows-1252'?>  
<employees>  
  Company Employee Data  
</employees>
```

5. Enregistrez le fichier.
6. Ouvrez le document XML appelé regions.xml.
7. Visualisez les documents XML dans les navigateurs Web que vous disposez.

Exercice 2 :

L'objectif de cet exercice est de créer un document XML syntaxiquement correct qui contient des informations sur les employés. Vous allez utiliser SQL*Plus pour exécuter une interrogation et obtenir le contenu de votre document. Une fois le document créé, on vous demandera de provoquer certaines erreurs et d'expliquer en quoi ces modifications enfreignent les règles qui définissent un document syntaxiquement correct.

1. Générez les données nécessaires à un document XML à partir de la base de données.

a. Connectez-vous à la base de données à l'aide de l'utilisateur (hr/hr).

b. Exécutez la requête suivante pour générer les données à inclure dans le document XML :

```
select employee_id, first_name, last_name, salary  
from employees  
where department_id = 60
```

2. Modifiez le document employees.xml dans le projet XMLLabsProject afin qu'il contienne les éléments correspondant aux employés du département 60.

a. Convertissez le texte « Company Employee Data » de l'élément <employees> en commentaire XML.

b. Pour chaque ligne d'employé de la requête précédente, ajoutez un élément employee à l'aide de la structure suivante :

```
<employee>  
  
  <employee_id><!-- id --></employee_id>  
  
  <first_name><!-- first name --></first_name>  
  
  <last_name><!-- last name --></last_name>  
  
  <salary><!-- salary --></salary>  
  
</employee>
```

c. Enregistrez les modifications que vous avez apporté au document, puis vérifiez que le document employees.xml est syntaxiquement correct. Corrigez les erreurs éventuelles et enregistrez les modifications.

d. Affichez le fichier employees.xml dans le navigateur.

3. Afficher le document XML qui contient des données de département dans le projet XMLLabsProject et corrigez les erreurs de syntaxe éventuelles.

Remarque : Après avoir recherché les erreurs de syntaxe et avant de les corriger dans JDeveloper, vous pouvez visualiser le document XML dans le navigateur pour observer comment ce dernier signale les erreurs.

4. Ajoutez un attribut appelé num, auquel vous affecterez la valeur 1, dans la balise de début du premier élément <department>. Ne mettez pas de guillemets ni d'apostrophes autour de la valeur de l'attribut. Vérifiez la syntaxe du document XML et corrigez l'erreur.

5. Répondez aux questions suivantes à propos des documents XML :

a. XML est un langage avec balises permettant de formater et d'afficher du texte et des images. Vrai/Faux

b. XML permet d'effectuer de simples échanges de données d'une application à l'autre. Vrai/Faux

c. Les balises d'éléments qui figurent dans un document XML permettent à d'autres applications d'analyser le document en vue de rechercher des informations spécifiques. Vrai/Faux

d. Le document XML suivant est syntaxiquement correct :

```
<?XML version="1.0"?>
  <customer>
    <name>Mary Lewis>
    <state>VA</state>
  </customer>
```

Vrai/Faux

e. Un élément racine ____.

- doit obligatoirement figurer en première ligne d'un document XML
- contient tous les autres éléments
- n'est pas obligatoire

f. Les éléments qui contiennent des données doivent comporter ____.

- des balises de début et de fin
- une déclaration XML
- des attributs

g. Si les besoins s'accroissent, les éléments _____

- sont employés pour des métadonnées et s'avèrent utiles
- ne peuvent pas être ajoutés facilement
- peuvent être ajoutés facilement

h. Bien que ce ne soit pas obligatoire, il est préférable et fréquent d'inclure _____

- une déclaration XML
- un élément racine
- des balises de début et de fin

Exercice 3 :

L'objectif de cet exercice est de créer un document XML valide basé sur des déclarations définies dans une DTD.

1. Ajoutez le nouvel employé suivant comme dernier élément <employee> dans le fichier employees.xml, juste avant la balise de fin </employees>.

```
<employee>

  <emp_id>007</emp_id>

  <last_name>Bond</last_name>

  <first_name>James</first_name>

</employee>
```

Vérifiez la syntaxe du document employees.xml et enregistrez les modifications.

2. Examinez les déclarations définies dans la DTD « labs\lab\employees.dtd ».

3. Associez le fichier employees.dtd à employees.xml en insérant une déclaration de type de document; utilisez pour cela une entrée

```
<!DOCTYPE employees SYSTEM "employees.dtd">
```

Vérifiez la syntaxe XML, et enregistrez les modifications.

4. En fonction des informations affichées dans la fenêtre Structure, apportez les corrections nécessaires au document employees.xml, puis enregistrez les modifications.

5. Ajoutez une référence DOCTYPE dans departments.xml afin d'utiliser la DTD « labs\lab\departments.dtd ».

6. Validez le document departments.xml

Indice : L'attribut num est utilisé comme numéro de séquence.