

# Travail pratique Oracle & XML

Houda CHABBI DRISSI

## 1 Introduction

L'utilisation de données XML dans les applications devient de plus en plus important. Par conséquent, les aspects de stockage de ce nouveau type de données sont également très importants. Depuis sa version 9i, Oracle a intégré de nombreuses fonctionnalités permettant de travailler avec des données XML. Ces fonctionnalités ont par la suite été complétées et améliorées jusqu'à la version actuelle. Nous allons donc effectuer un tour de ces fonctionnalités offerte par ces deux versions d'Oracle.

## 2 Objectifs

Les objectifs de ce travail concernant le stockage et l'interrogation de données XML dans Oracle sont les suivants :

- Comprendre et utiliser les mécanismes offerts par Oracle afin d'interroger des données relationnelles et de les transformer en XML
- Comprendre les différentes manières de stocker un document XML dans une base de données Oracle
- Se familiariser avec l'écriture de requête XQuery pour Oracle
- Savoir interroger des données XML dans Oracle directement depuis une application Java avec JDBC.

## 3 Travail à réaliser

Les exercices de ce travail pratique seront réalisés sur les versions suivantes d'Oracle :

Exercice	Version
3.1 Relationnel vers XML	Oracle 11gR2
3.2 XML → XML	Oracle 11gR2 (partie travail dirigé)
3.3 XQuery dans Oracle avec Java et JDBC	Oracle 11gR2

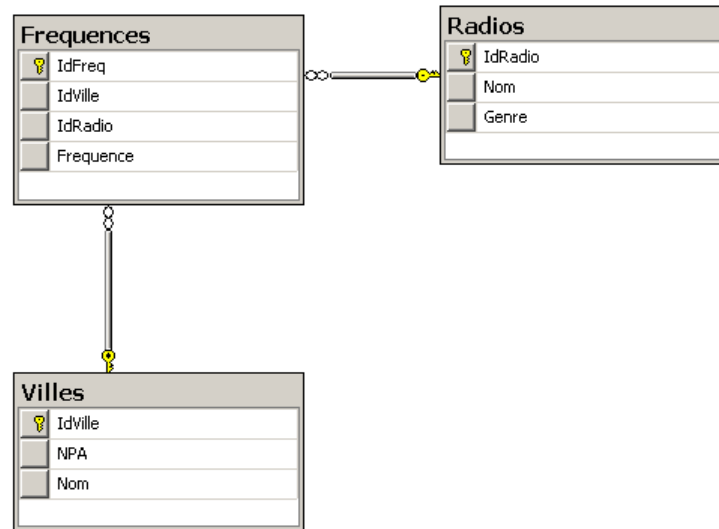
Une base de données Oracle est installée au sein de l'EIA-FR. L'outil SQL Developer sera donc utilisé pour se connecter et effectuer les exercices concernés.

Pour vous aider lors de la réalisation des exercices ci-après, vous pouvez consulter les documentations en ligne de Oracle (voir réf. S3 et S5).

### 3.1 Du relationnel vers XML

Pour ce premier exercice, le but est de convertir au format XML une série de données contenues dans une structure relationnelle simple afin de les stocker dans une base de données XML.

Les données à convertir représentent les fréquences de plusieurs stations radios par rapport aux villes dans lesquelles on se trouve. La structure relationnelle permettant de stocker ces données est illustrée ci-dessous :



Voici maintenant un aperçu des données stockés dans cette structure :

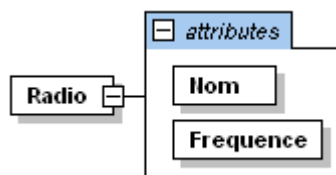
IdFreq	IdVille	IdRadio	Frequence
8	1	1	102.60
9	1	2	88.60
10	1	3	89.40
11	2	1	102.60
12	2	2	98.50
13	3	1	94.90
14	3	2	104.40

IdRadio	Nom	Genre
1	La Première	Radio principale ...
2	Couleur 3	Programme de la...
3	Radio Fribourg	Radio de la régio...

IdVille	NPA	Nom
1	1700	Fribourg
2	1000	Lausanne
3	1200	Genève

Utilisez les fonctions « XMLElement() », « XMLAttribute() », « XMLAgg() » et « XMLForest() » de Oracle (voir réf. S1) afin de structurer les résultats des requêtes suivantes au format XML.

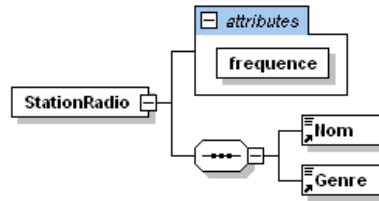
1. Afficher toutes les radios disponibles à Fribourg sous la forme d'une liste d'éléments « Radio » contenant les informations suivantes sous la forme d'attributs :



Exemple :

```
<Radio Nom="La Première" Frequence="102.60" />
```

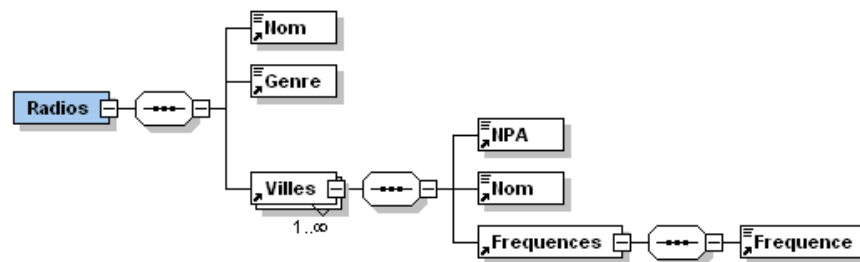
2. Afficher toutes les stations de radio disponible à Genève sous la forme suivante :



Exemple :

```
<StationRadio frequence="...">
  <Nom>...</Nom>
  <Genre>...</Genre>
</StationRadio>
```

3. Afficher les fréquences des radios en les regroupant par radios, puis par ville. Le résultat ne doit contenir que des éléments (pas d'attributs). Sa structure doit être semblable à celle décrite ci-dessous :

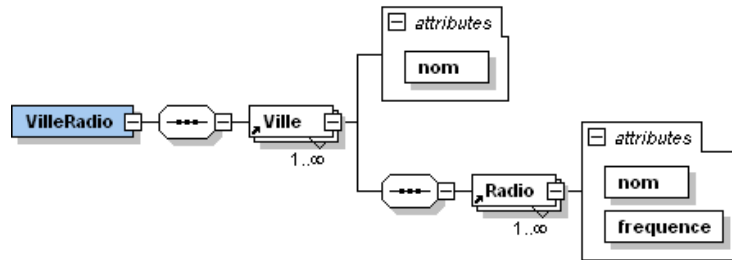


Exemple :

```
<Radios>
  <Nom>Couleur 3</Nom>
  <Genre>Programme de la RSR</Genre>
  <Villes>
    <NPA>1700</NPA>
    <Nom>Fribourg</Nom>
    <Frequences>
      <Frequence>88.60</Frequence>
    </Frequences>
  </Villes>
</Radios>
```

Conseil : Suivez les étapes décrites ci-dessous pour écrire votre requête...

- 1) Afficher toutes les radios avec leur nom et genre
  - 2) Afficher les radios avec les villes dans lesquelles elles peuvent être captées
  - 3) Compléter avec l'affichage des fréquences des radios.
4. Afficher la liste des radios disponible dans chaque ville avec leurs fréquences respectives sous la forme suivante :



Voici un exemple de solution pour les villes de Fribourg et Genève :

```

<VilleRadio>
  <Ville nom="Fribourg">
    <Radio nom="Couleur 3" frequence="88.60" />
    <Radio nom="La Première" frequence="102.60" />
    <Radio nom="Radio Fribourg" frequence="89.40" />
  </Ville>
  <Ville nom="Genève">
    <Radio nom="Couleur 3" frequence="104.40" />
    <Radio nom="La Première" frequence="94.90" />
  </Ville>
</VilleRadio>
  
```

Conseil : Suivez les étapes décrites ci-dessous pour écrire votre requête...

- 1) Afficher toutes les villes
- 2) Afficher les radios disponibles dans chaque ville
- 3) Encapsuler les résultats dans une racine unique nommée « VilleRadio ».

## 3.2 XML → XML

### 3.2.1 Jeux de données

Pour cette partie du TP, les requêtes devront être exécutées sur un jeu de données représentant une collection de bandes dessinées (table « BD ») avec leurs auteurs et illustrateurs (table « Personnes ») ainsi que les points de vente dans lesquels elles sont vendues (tables « Distributeurs »).

«Table»	
<b>Distributeurs</b>	
<b>PK</b> DIS_PK: int Not Null	
DIS_List: XMLType	

«Table»	
<b>Personnes</b>	
<b>PK</b> PER_PK: int Not Null	
PER_List: XMLType	

«Table»	
<b>BD</b>	
<b>PK</b> BD_PK: int Not Null	
BD_List: XMLType	

Chacune des tables ci-dessus ne comporte qu'un seul enregistrement. Le champ XMLType de chacun des ces enregistrement contient l'instance du document XML correspondant. La structure de ces instances de documents est décrite dans les différents fichiers PDF fournis avec ce TP.

### 3.2.2 Requêtes avec les outils clients

#### 3.2.2.1 Requêtes de lecture

Ecrivez les requêtes correspondantes aux questions ci-dessus. Pour exécuter ces requêtes, il faut utiliser les méthodes « XMLQuery() » ou « XMLTable() » implémentées par Oracle (voir réf. S1 et S2).

1. Tous les titres des DB triés par ordre alphabétique (utiliser les méthodes « XMLQuery() » et « XMLCast() »)
2. Les titres des DB sorties après 2000 avec leur date de sortie, sous la forme de données relationnelles. (utiliser la méthode « XMLTable() »)
3. Combien de DB compte la série « Le retour a la terre » (utiliser la méthode « XMLQuery() »)

#### 3.2.2.2 Requêtes d'écriture

Ecrivez maintenant les requêtes permettant d'effectuer les modifications suivantes sur notre base de données. Pour exécuter les requêtes suivantes, il faut utiliser les méthodes « appendChildXML () », « updateXML() » et « deleteXML() » fournies par Oracle (voir réf. S1).

1. Ajouter la DB suivante à la série nommée « Le retour à la terre » :

```
<bd id="123" numero="4">
  <titre>Le Deluge</titre>
  <auteur ref="JYFerri"/>
  <illustrateur ref="Larcenet"/>
  <sortie>2005</sortie>
</bd>
```

2. Modifier la date de sortie de cette BD en « 2006 »
3. Supprimer cette bande dessinée de la base de données

### 3.3 XQuery dans Oracle avec Java et JDBC

Le but de cet exercice est de comprendre les mécanismes liés à l'interrogation de données au format XML dans Oracle depuis une application Java. La librairie permettant de réaliser ces interrogations est JDBC (voir réf. S4).

Le but de cet exercice est de compléter la petite application fournie avec ce TP (dans le répertoire « OracleJava ») afin de lui ajouter le code nécessaire pour l'exécution d'une requête permettant d'afficher tous les titres des bandes dessinées de la série « Titeuf ».

Les différentes librairies (JAR) nécessaires pour cet exercice se trouvent dans le sous-répertoire « lib » se trouvant dans « OracleJava ».

La table BD ainsi que son contenu devra être mise en place à l'aide de l'outil SQL\*Plus de Oracle. Le script permettant la création de cette table se trouve dans le fichier « DB\_Setup.plsql » du répertoire « DB ».

La requête à exécuter est la suivante :

```
select b.BD_LIST.EXTRACT('//serie[titreserie="Titeuf"]/bd/titre') from BD b;
```

Pour cette requête nous avons utilisé la fonction « Extract() » qui permet d'exécuter des expressions XPath sur des documents stockés dans une colonne XMLType. Une requête XPath sera suffisante pour retrouver les informations demandées pour cet exercice.

La chaîne de connexion à utiliser afin de se connecter au serveur de test est la suivante :

```
jdbc:oracle:thin:@160.98.2.105:9203:dbinf
```

Le login et le mot de passe sont identiques à ceux utilisés pour l'exercice 1.