Cours XMLTYPE sur Oracle 10g XE

1. **Documentation XML DB Developper's Guide d'Oracle :**
   - 10 g : http://docs.oracle.com/cd/B19306_01/appdev.102/b14259/toc.htm
   - 11 g : http://docs.oracle.com/cd/B28359_01/appdev.111/b28369/toc.htm

2. **Introduction**
   Objectif : pouvoir stocker une donnée XML (arborescente) dans une BDD relationnelle en profitant de la puissance des 2 modèles (XML/XPATH/XSD/XSL et SQL)

3. **Utilisation**
   a. **Déclaration**
   Déclarer un champs contenant une donnée XML : type XMLTYPE

```
CREATE TABLE ma_table (
   champs_xml XMLTYPE
);
```

   b. **Insertion**
   Insérer une donnée de type XML : constructeur `XMLTYPE(s)` avec `s` de type VARCHAR2 ou CLOB

   I. donnée courte

```
INSERT INTO ma_table (champs_xml) VALUES (XMLTYPE(
'<?xml version="1.0" encoding="UTF-8" ?>
<racine>…</racine>'));
```

   II. donnée longue

```
DECLARE
   xml_string CLOB;
BEGIN
   xml_string := '<?xml version="1.0" encoding="UTF-8" ?>
            <racine>…</racine>';
   INSERT INTO ma_table (champs_xml)
            VALUES (XMLTYPE(xml_string));
END;
/
```

   c. **Sélection**
   I. accès au xml complet : select classique
   NB : pour sqlplus régler le nombre de caractères à visualiser pour un type CLOB/XMLTYPE

```
SET LONG 2000
```

   II. convertisseurs : méthodes "objets"
   – maDonneeXML.getClobVal() : XML extrait en tant que CLOB
   – maDonneeXML.getStringVal() : XML extrait en tant que VARCHAR2 (taille limite)
   – maDonneeXML.getNumberVal() : XML extrait en tant que nombre (élément simple ou attribut)
   RQ : si maDonneeXML est un attribut de table, il doit être préfixé par la table ou un alias :

```
Select x.champsXML.getClobVal() from ma_table x;
```

   III. fonctions d'accès
   – existsnode(maDonneeXML, XPATHExpression) : prédicat à utiliser dans un WHERE uniquement, renvoie 1 si vrai, 0 sinon
   – extract (maDonneeXML, XPATHExpression) : à utiliser dans les champs projetés (SELECT) ou dans un prédicat (WHERE) ; renvoie le nœud de type XMLTYPE correspondant à l'expression XPATH
   – extractvalue (maDonneeXML, XPATHExpression) équivaut à extract (maDonneeXML, XPATHExpression).getStringVal()
      - Pour un élément simple <elt>un texte</elt>, il n'est pas nécessaire de préciser le nœud fils, de type texte, dans l'expression XPATH (cf XSL et value-of) : elt/text() → elt suffit pour extraire 'un texte'

IV. méthodes objets d'accès :
- maDonneeXML .existsnode(XPATHExpression) : prédicat à utiliser dans un WHERE uniquement, renvoie 1 si vrai, 0 sinon
- maDonneeXML.extract (XPATHExpression) : à utiliser dans les champs projetés (SELECT) ou dans un prédicat (WHERE) ; renvoie le nœud de type XMLTYPE correspondant à l'expression XPATH
- pas d'extractvalue dans ce mode, utiliser extract + getStringVal ou getNumberVal()

d. Insertion

I. fonctions de modification partielles (sinon update classique) à combiner avec un UPDATE pour modification dans une table ou sans pour obtenir simplement un donnée modifiée :
- updateXML(donneeXML, XPATExpr, value) : renvoie donneeXML modifiee (si value=NULL => suppression du nœud)
- insertChildXML(donneeXML, XPATHExpr, child_name, child_value)
  - pour un élément : child_name est le nom de l'élement, child_value l'élément complet
  - pour un attribut : child_name est le nom de l'attribut, child_value sa valeur
- insertXMLbefore(donneeXML, XPATHExpr, value) : pour les éléments uniquements
- appendChildXML(donneeXML, XPATHExpr, value)
- deleteXML(donneeXML, XPATHExpr)

II. les mêmes en version objets

III. toutes ces fonctions peuvent avoir un paramètre complémentaire (en dernière position) définissant une liste d'espaces de noms

e. Transformation : application d'une feuille de style XSL sur une donnée XML :
- fonction SQL : XMLTransform(donneeXML, feuilleXSL) : renvoie la donnée transformée
- méthode objet : donneeXML.transform(feuilleXSL)
- Il peut être intéressant de stocker les feuilles de style dans une table

e. Validation par schema

I. DBMS_XMLSCHEMA. deleteSchema(uri du schéma)

II. DBMS_XMLSCHEMA. registerSchema(uri du schéma, arbre xsd)

III. associer un schéma à une colonne d'une table
  i. CREATE TABLE nomTable (…, nomCol XMLType) XMLType COLUMN nomCol XMLSCHEMA "uri du schéma" ELEMENT "nom de la balise racine";
  ii. => vérification auto à chaque modif (insert/update/delete)

IV. PL/SQL Functions
  i. XMLIsValid(donnéeXML [,schemaURL [, élément]])
  ii. donnéeXML.isSchemaValidated() : number (1 = true)
  iii. donnéeXML.isSchemaValid([schemaURL],[élément]) : NUMBER (1 = true)

V. PL/SQL procedure
  i. donnéeXML.schemaValidate
  ii. donnéeXML.setSchemaValidated(flag) (1 = true)

f. Indexation

Il est possible de définir un index sur nœud d'un document XML