Stockage d'arbres XML et Interrogation XPATH et SQL

Information plus ou moins structurée

- une table : Stockage statique assez plat
- un arbre XML : Stockage dynamique en profondeur
- Lien entre les deux modes : les clés étrangères

Information plus ou moins structurée

- le type VARCHAR2
- le type CLOB
- le type XMLTYPE

Exemple d'arbres XML

```
<commande id="1">
    <lc id="1">c1 : ma première ligne</lc>
    <lc id="2">c1 :ma deuxième ligne</lc>
</commande>

<commande id="2">
    <lc id="1">c2 : ma première ligne</lc>
    <lc id="2">c2 : ma deuxième ligne</lc>
</commande>
```

Stockage de l'arbre dans Oracle

```
CREATE TABLE Commandes
(
   idCom number,
   comm XMLTYPE,
   CONSTRAINT pk_commande
   PRIMARY KEY(idCom)
);
```


Insertion d'un arbre dans la base

- Oracle a créé la fonction XMLTYPE pour gérer les arbres XML.

```
INSERT INTO commandes VALUES
  (1,XMLTYPE ('<commande id="1">...</commande>'));

INSERT INTO commandes VALUES
  (2,XMLTYPE ('<commande id="2">...</commande>'));
```

- La fonction XMLTYPE convertit les balises en un contenu de type XMLTYPE

- Elle génère une erreur si l'information de la colonne comm n'est pas un arbre XML

La fonction UPDATEXML

- Une requête de mise à jour de lignes

```
UPDATE commandes SET comm =
    UPDATEXML(comm,
  '/commande/lc[@id="1"]/text()', 'two')
```

- Résultat
  Tous les arbres XML de la colonne Comm auront le texte des nœuds lc, dont l'attribut id vaut 1, remplacé par le texte 'two'. Dans notre cas, les deux lignes seront changées

La fonction UPDATEXML

- UPDATE(xmltype,xpath,value)
- UPDATE commandes SET comm =
  UPDATEXML(comm, '/commande/lc[@id="1"]/text()', 'two')
  WHERE idComm = 1

- Résultat
  L'arbre XML de la colonne Comm de la ligne dont la clé vaut 1 aura le texte des nœuds lc, dont l'attribut id vaut 1, remplacé par le texte 'two'.

- Si le dernier paramètre vaut null alors les nœud selectionnés sont supprimés

Autres mises à jour

- APPENDCHILD(XMLType,XPATH,XMLType fils)

```
UPDATE warehouses SET warehouse_spec =
APPENDCHILDXML(warehouse_spec, 'Warehouse/Building',
XMLType('<Owner>Grandco</Owner>')) WHERE
EXTRACTVALUE(warehouse_spec, '/Warehouse/Building') =
'Rented';
```

- APPENDCHILD(XMLType,XPATH,XMLType frère au dessus)

```
UPDATE warehouses SET warehouse_spec =
        INSERTXMLBEFORE(warehouse_spec,
        '/Warehouse/Building/Owner[2]',
        XMLType('<Owner>ThirdOwner</Owner>')) WHERE
        warehouse_id = 3;
```

- DELETEXML(XMLType,XPATH) détruit tous les noeuds sélectionnés

```
UPDATE warehouses SET warehouse_spec =
DELETEXML(warehouse_spec, '/Warehouse/Building/Owner')
WHERE warehouse_id = 2;
```

Sélection d’un arbre dans la base

- La commande suivante permet d'obtenir le contenu entier d'un arbre XML

```
SELECT EXTRACT (comm,'/')
FROM commandes
```

Sélection d’un sous arbre dans la base

- La commande suivante permet d'obtenir les contenus de sous arbres d'une colonne XMLTYPE

```
SELECT EXTRACT
(comm,'commande/lc') FROM
commandes
```

Sélection d’un sous arbre dans la base

- Les commandes suivantes permettent d'obtenir une partie des arbres d’une colonne XMLTYPE

```
SELECT EXTRACT(comm, 'commande[lc/@id =
"1"]') FROM commandes
```

- Résultat

  Toutes les lignes de la table sont sélectionnées. Pour chacune, on affiche un arbre vide ou tout l'arbre s'il existe un noeud lc dont l'attribut id vaut 1

Sélection d’un sous arbre dans la base

- Les commandes suivantes permettent d'obtenir une partie des arbres d’une colonne XMLTYPE

```
SELECT EXTRACT(comm, 'commande[lc/@id =
"1"]') FROM commandes     where idComm=1
```

- Résultat

  On sélectionne la ligne dont la clé vaut 1.

  On affiche un arbre vide ou tout l'arbre s'il existe un noeud lc dont l'attribut id vaut 1

Sélection d'un sous arbre dans la base

- SELECT EXTRACT(…).getStringVal() …

  renvoie le résultat sous forme de VARCHAR2

- SELECT EXTRACT(…).getNumberVal() …

  renvoie le résultat sous forme de NUMBER

- SELECT EXTRACT(…).getCLOBVal() …

  renvoie le résultat sous forme de CLOB

- SELECT EXTRACTVALUE(XMLType,XPATH) …

  renvoie le résultat sous forme de VARCHAR2. Il y a erreur si le noeud sélectionné n'est pas un noeud texte.

Sélection d'un sous arbre dans la base

- ExistsNode(XMLType,XPATH, [NS]) renvoie 1 ou 0 suivant que le noeud sélectionné existe ou pas.
- ora:contains est une fonction qui renvoie vrai ou faux suivant que la première chaîne contient ou ne contient pas la deuxième chaîne
- SELECT id FROM purchase_orders_xmltype WHERE existsNode(doc, '/purchaseOrder/comment [ora:contains(text(), "is", "MY_NOSTOPWORDS_POLICY") > 0]', 'xmlns:ora="http://xmlns.oracle.com/xdb"') = 1;

Sélection d'un sous arbre dans la base

- ExistsNode(XMLType,XPATH, [NS]) renvoie 1 ou 0 suivant que le noeud sélectionné existe ou pas.
- ora:contains est une fonction qui renvoie vrai ou faux suivant que la première chaîne contient ou ne contient pas la deuxième chaîne
- SELECT id FROM purchase_orders_xmltype WHERE existsNode(doc, '/purchaseOrder/comment [ora:contains(text(), "is", "MY_NOSTOPWORDS_POLICY") > 0]', 'xmlns:ora="http://xmlns.oracle.com/xdb"') = 1;

Transformation des données

Transformation des données avec XSL : XMLTransform ou transform

- XMLTransform(xml as XMLTYPE, xsl as XMLTYPE) -> XMLTYPE
- data XMLTYPE : data.transform(xsl as XMLTYPE) -> XMLTYPE

Validation des données avec un schéma

- Ajout/Suppression de schéma dans une BDD
  - DBMS_XMLSCHEMA.registerSchema(XMLschema, schemaURL) pour stocker un schéma dans la base
  - DBMS_XMLSCHEMA.deleteSchema(XMLschema, schemaURL) pour retirer un schéma de la base.