

Bases de données Oracle et XML

Solution de gestion de données non structurées

# Table des matières

## Table des figures

## Introduction

### Avant-propos

Depuis longtemps XML fut le langage recommandé par le consortium du web (W3C).

Ainsi unifie-t-il, comme moyen de représentation des données à la fois structurées et non structurées, deux mondes traditionnellement séparés : celui de la gestion de données et celui de la gestion de contenu. Par ailleurs le langage SQL est de son côté reconnu pour la gestion de données structurées tant en définition qu'en manipulation. Avec l'importance des volumes de données non structurées XML (documents internet et autres), souvent ces données sont stockées sous forme de fichiers non XML; dans des bases de données relationnelles.

Il faut donc mettre en place un processus de conversion des données du format actuel de la base en documents XML, puis du XML au format utilisé pour le traitement des données, cela se traduit par un plus long traitement des données. SQL s'est vu doter d'outils de manipulation de ces données afin de répondre à ce besoin (SQL 2003 et SQL 2005 : fonctionnalités de manipulation de données XML).

Oracle dans ce sens a été un des pionniers, avec l'organisme W3C, de la fusion de SQL et XML et de l'effort de normalisation de SQL/XML.

### Structure du document

Ce rapport a pour but de présenter, de manière non exhaustive, un aspect de la gestion de données XML dans une base de données relationnelles et les outils qu'offre oracle quant au stockage et la manipulation de ce type de données. La manipulation de documents XML sera abordée à travers la présentation des opérateurs SQL/XML et l'introduction de la base de données native XML (Oracle XMLDB). Ce rapport est divisé en trois chapitres; le premier **introduit le sujet de la gestion de données non structurées dans une base de données relationnelles, aborde les types et techniques de stockage de données XML dans une bases de données Oracle XMLDB d'une part et présente brièvement les opérateurs de manipulation de ces données à travers SQL/XML: création, extraction et modification de données XML d'autre part**, le deuxième évoque quelques avantages de l'utilisation d'Oracle XMLDB comme référentiel de gestion données structurées et non-structurées, donne un **exemple réel d'utilisation**. **Le dernier chapitre conclut sur l'utilisation et l'intérêt de gérer XML dans une base relationnelle et présente une bibliographie des sources utilisées et qui ont contribué à réaliser ce rapport.**

## Chapitre1

### Gestion de données semi ou non structurées

XML est en passe de devenir le standard utilisé systématiquement par les entreprises pour échanger des données dans des documents structurés, que ce soit en interne, avec des partenaires commerciaux ou dans des applications accessibles à tous sur internet. Toutefois, ces données sont souvent stockées dans un format de fichiers autre que le XML par exemple, dans une base de données relationnelle. Il faut donc mettre en place un processus de conversion des données du format actuel de la base en documents XML, puis du XML au format utilisé pour le traitement

des données, cela se traduit par un plus long traitement des données.

Des outils sont disponibles avec les bases de données relationnelles pour traiter ce genre de tâche, comme Oracle9i et IBM DB2. Ils peuvent traduire en XML les données, structurées ou non. Toutefois, si ces outils sont utilisés pour traduire des données en direct lors d'un échange XML, cela risque d'augmenter les temps de traitement des transactions XML et de ralentir également les autres applications qui accèdent à la base de données relationnelles.

Il existe cependant une autre approche : les bases de données XML natives. Oracle propose la solution Oracle XMLDB native

## **XML et les bases de données relationnelles (Oracle XML DB)**

### **Oracle XML DB – Architecture**

#### **Modèles de stockage de XMLTYPE**

#### **Stockage des données XML**

#### **Opérateurs SQL/XML**

### **Chapitre2**

#### **Valeurs ajoutées de Oracle XML DB**

#### **Références Clients**

### **Chapitre3**

#### **Conclusion**

#### **Bibliographie**

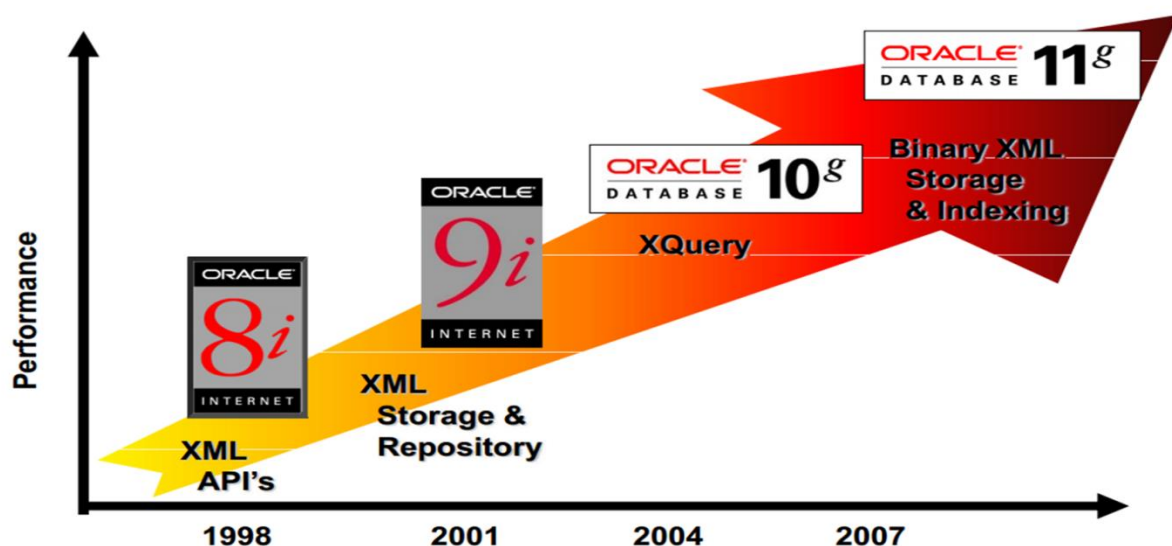


## C1 : Gestion de données semi ou non structurées

Le support de XML a commencé à partir de la version d'oracle 8i qui offrait des interfaces java (embarqués dans un utilitaire XSU: XML SQL Utility en plus des packages XMLPARSER, XMLDOM et XSLPROCESSOR) de manipulation de documents XML dans des tables relationnelles de la base oracle.

Avec l'arrivée de la version Oracle 9i, un nouveau type de données natif XMLTYPE est introduit pour stocker et manipuler XML. XMLTYPE offre une palette de fonction de manipulation de XML en utilisant les expressions XPATH comme XMLTYPE.extract(), XMLTYPE.extractvalue() et XMLTYPE.existsNode(). Des packages systèmes existent pour créer XMLTYPE: DBMS\_XMLGEN(), SYS\_XMLGEN() et SYS\_XMLAGG(). La référence au contenu basé sur l'URL est aussi stocké dans la base de données. Ainsi le traitement du XML est simplifié du fait qu'il n'est pas nécessaire d'initialiser et de maintenir des classes java dans la base oracle. Cependant ces fonctions XML ne performant pas si bien du fait que cela nécessite de construire les modèles DOM en mémoire avant de traiter le contenu XML. Enfin Oracle Search dans Oracle 9i prend en charge XPATH dans l'index de type CONTEXT. C'est ainsi qu'Oracle XML DB est arrivée avec la capacité de réécriture des requêtes XML, au moyen des expressions XPATH, dans les requêtes SQL pour accélérer l'accès aux données. Le référentiel XMLDB est apparu dans Oracle 9i release2 et est orienté XML document et améliore les performances de gestion de documents XML. Il permet plus d'interfaces d'accès aux données: FTP, HTTP et WEBDAV.

En Oracle 10g, les interfaces java citées ci-dessus (APIs basées sur le code java) ont été écartées au profit de leurs semblables mais écrites en PL/SQL basées sur du code en langage C pour des raisons de performances.



La version 11g d'Oracle, commercialisée depuis le 11 juillet 2007, est la plus récente. Historique de XML dans la base de données oracle

- 1998 Début de manipulation de données XML via des APIs
- 2001 intégration de SQL/XML dans la version oracle 9i release 2

- 2003 Support de database-native XQuery dans la version d'oracle 10g release 2.

## T2 : XML et les bases de données relationnelles (Oracle XML DB)

La base de données XML est un ensemble de techniques propres à XML.

- ✓ La plus part des données d'applications Web sont stockés dans des bases ou dans des fichiers.
- Problématique des volumes croissants de documents XML stockés.

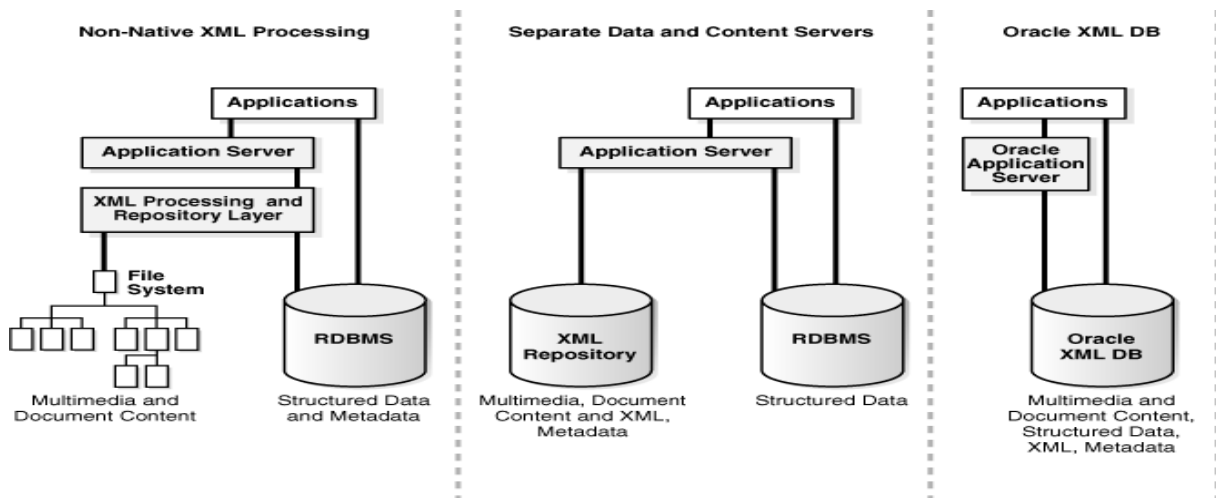


Figure 1-1 : Données et contenus unis

### Alternative à XML?

Stocker les documents sur un système de fichiers

- Lourd à gérer : structure de répertoire, convention de nommage, etc
- Besoin d'avoir une partie des données dans la base et un processeur XML de requête de documents XML
- Extraction partielle du document XML est moins facile
- Problème d'indexation des documents XML et de sa maintenance
- Besoin de deux sauvegardes : bases de données et systèmes de fichiers

Avoir une bases de données qui gère les données relationnelles et les documents XML résoud le problème

Deux approches pour stocker les documents:

- DB + XML Capabilities: adaptée pour les schémas avec peu de documents XML en attachement

- XML DB native, conçues pour gérer efficacement les documents XML (non ou peu relationnels) : pour les bases de gestion de contenu où le nombre de documents XML accédés et manipulés est très important
  - Un ensemble de techniques de gestion de recherche et de stockage de données XML plus performant que dans une base relationnelle
  - Oracle XML DB est plus adapté aux applications JAVA ou PL/SQL dont la données traitées sont représentés en XML
  - Type natif de données XMLTYPE représentant un document XML accessible par SQL
  - Prend en charge XML Schema, XPATH, XQUERY, XSLT et DOM
  - XMLIndex pour indexer les objets XMLTYPE

## **T2: ORACLE XML DB**

- XML est utilisé pour gérer des données sensibles (avec l'extérieur et via des Web services)
- Besoin de mieux gérer XML (efficacité, productivité, volumétrie, disponibilité, passage à l'échelle, sécurité, compatibilité)
- Recherche et accès rapide aux information
- XML DB est une base de données XML natif, hybride (accès SQL au contenu XML et accès XML au contenu relationnel)
- Multiples options de stockage XML pour optimiser les performances des applications
- Stockage et indexation optimisés pour le contenu structuré ou non structuré

### ***T3 : Les principales fonctionnalités de Oracle XML DB:***

XMLType Data Type

XML Schema Support

XMLType Storage Models

XML/SQL Duality

SQL/XML Standard Functions

Automatic Rewriting of XQuery and XPath Expressions

Oracle XML DB Repository

D1 : Les fonctions XMLType pour requêter les données XML avec le type XMLTYPE :

Extract(Xpath-expression),

ExistsNode(Xpath-expression),

GetNamespace(),

GetRootElement(),

GetStringVal(),

IsFragment(),

Transform(XSL-Stylesheet-Script)

## T1: Oracle XML DB ARCHITECTURE

- Stockage de XMLType tables et vues
- Le référentiel Oracle XML DB
- Support de XML data messaging

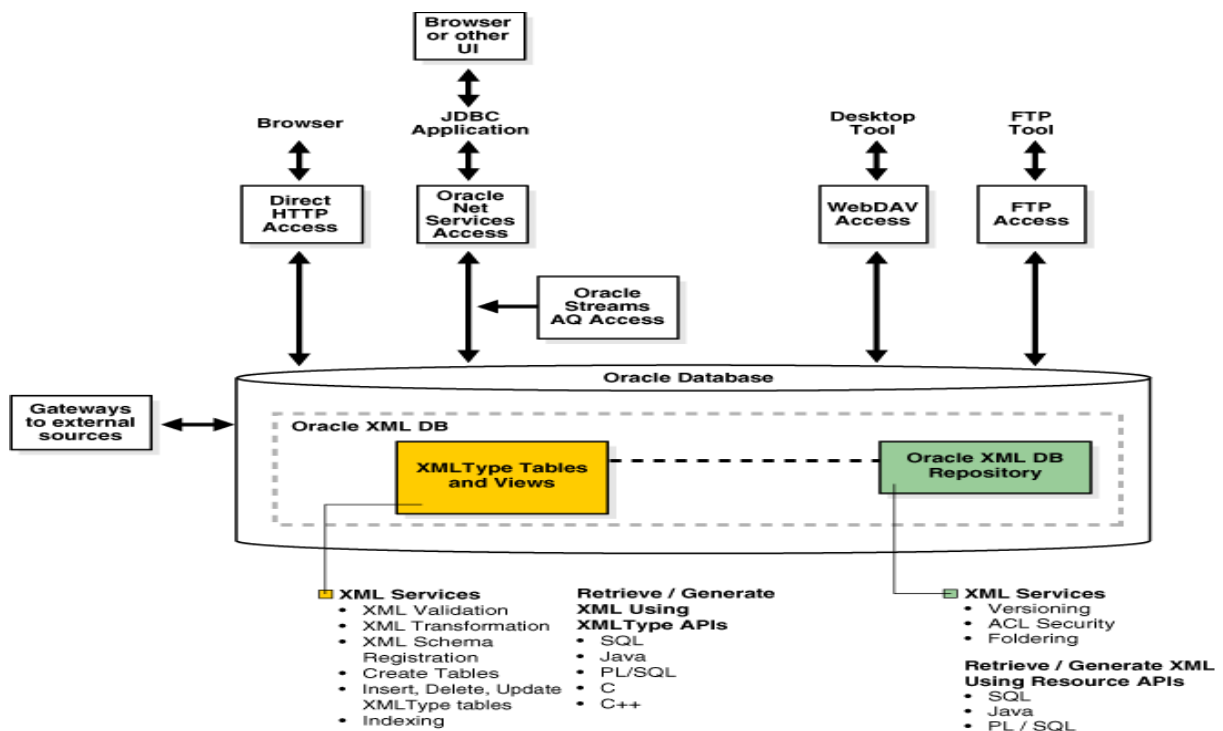


Figure 1-2 XMLType et le référentiel Oracle XML DB

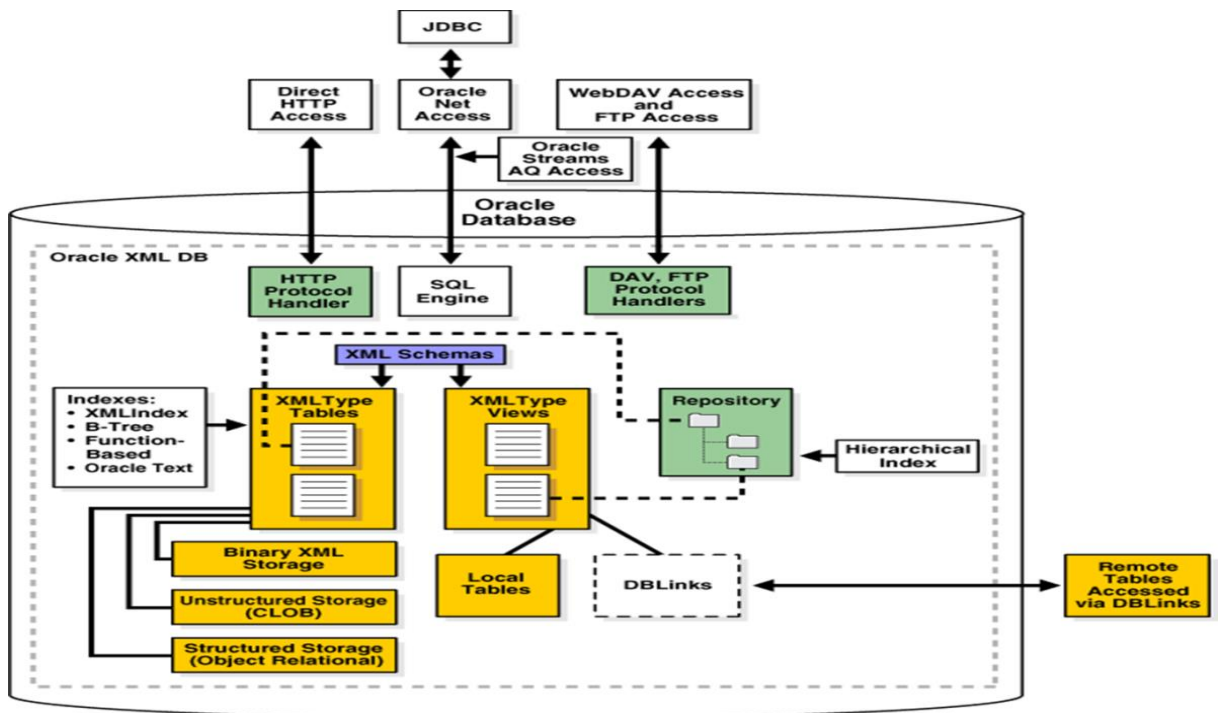
D1: Architecture de Oracle XMLDB:

- Stockage de XMLType tables et vues

Les tables et vues XMLType peuvent être indexés en utilisant les type d'indexes XMLIndex, B-tree et Oracle Text indexes.

Les tables et vues XMLType peuvent être stockées localement ou sur une base éloignée accessible via un dblink.

- Le référentiel Oracle XML DB.
  - On peut stocker tout type de documents dans le référentiel, y compris les documents XML associés à un schéma XML enregistré dans Oracle XML DB :
    - Les documents dans le référentiel sont accessibles de plusieurs façons:
    - HTTP(S), WebDav au moyen des "handlers" des protocoles respectifs
    - WebDAV et FTP au moyen des "handlers" des protocoles associés
    - SQL à travers Oracle Net Services (TNS) et JDBC
- Support de XML data messaging en utilisant Oracle Streams Advanced Queuing (AQ) et les Web Services.



**Figure 1-3 Stockage de XMLType**

D1 : Restrictions liées à Oracle XMLDB :

[http://docs.oracle.com/cd/B28359\\_01/appdev.111/b28369/appjspec.htm#CIHEHIAJ](http://docs.oracle.com/cd/B28359_01/appdev.111/b28369/appjspec.htm#CIHEHIAJ)

Les fonctions XMLTYPE ne supportent pas le driver JDBC thin, plutôt le driver OCI ou JDBC thick.

Les types de données locaux NCHAR, NVARCHAR, NCLOB non supportés

Limite de la longueur d'in identifiant XML (4000 caractères)

Limite de la taille d'un fichier dans le référentiel XML (4 Go)

## T1 : Modèles de stockage de XMLTYPE

|               |    |                                                                                                                     |  |                                            |  |  |
|---------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|               | Data-Centric                                                                        |                                                                                                                     | Document-Centric                                                                  |                                            |                                                                                     |                                                                                     |
| Use Case      | XML schema-based data, with little variation and little structural change over time | XML schema-based data, with some embedded variable data                                                             | Variable, free-form data, with some fixed embedded structures                     | Variable, free-form data                   |                                                                                     |                                                                                     |
| Typical Data  | Employee record                                                                     | Employee record that includes a free-form resume                                                                    | Technical article, with author, date, and title fields                            | Web document or book chapter               |                                                                                     |                                                                                     |
| Storage Model | Object-Relational (Structured)                                                      | Hybrid                                                                                                              | CLOB (Unstructured) or Binary XML                                                 |                                            |                                                                                     |                                                                                     |
| Indexing      | B-tree index                                                                        | <ul style="list-style-type: none"><li>• B-tree index</li><li>• XMLIndex index with unstructured component</li></ul> | XMLIndex index with structured and unstructured components                        | XMLIndex index with unstructured component |                                                                                     |                                                                                     |

## T1 : Stockage des données XML

### T2 : Aperçu des LOBs (Large Object)

- \* Les LOBs permettent de stocker les gros blocks de données non structurées au format ASCII (CLOB) ou binaire (BLOB) .
- \* Oracle fournit deux options de stockage des LOBs:
  - \* BasicFiles (par défaut, mais obsolete en 11gR2)
  - \* SecureFiles (exige des segments ASSM et DB\_SECUREFILE)
- \* Les tables et colonnes XMLTYPE ont un stockage:
  - \* Structuré: Objets relationnels (persistance basée objet)
  - \* Non-structuré : CLOB (persistance basée text)
  - \* Binaire: BLOB (XMLTYPE post-parsé et schema-aware)

## T1 : Opérateurs SQL/XML

Maintenant qu'on a compris l'intérêt de mixer des documents XML avec une base de données, on a besoin d'outils pour accomplir certaines tâches:

- Accès → XPATH (ex: /order[total > 1000] recherche de factures d'un montant de plus de 1000€.)
- Efficacité de l'accès dépend du nombre de documents → Indexation pour un accès rapide
- Extraction partielle du XML (ex: adresse client, articles avec discount, etc), transformation → XQuery
- Mises-à-jour des documents XML → Bases de données Natives XML

### D1 : Le standard SQL/XML :

XMLELEMENT, XMLATTRIBUTES, XMLCONCAT, XMLAGG, XMLCOLATTVAL, XMLFOREST, XMLTRANSFORM (applique du XSL au document)

#### Exemples:

SELECT

XMLELEMENT("region", XMLATTRIBUTES(REGION AS "name"))

,XMLELEMENT("population", POPULATION))

FROM REGION;

SELECT XMLAGG(

```

XMLELEMENT("demographics"
,POPULATION||' '||REGION||' '||AREA) ORDER BY POPULATION DESC)
FROM REGION;

<demographics>2100636517 Far East 15357441</demographics><demographics>
1499157105 N ear East 4721322</demographics><demographics>789548670 Africa
26780325</demographic s> ...

SELECT
XMLELEMENT(
"region", XMLATTRIBUTES(R.REGION "name"),
XMLFOREST(
    R.POPULATION AS "population",
    R.AREA AS "area",
    SUM(CO.POPULATION) AS "populationSum",
    SUM(CO.AREA) AS "areaSum")
).EXTRACT('/*')
FROM REGION R JOIN COUNTRY CO ON (CO.REGION_ID=R.REGION_ID)
WHERE R.REGION_ID IN (3,9)
GROUP BY R.REGION, R.POPULATION, R.AREA ORDER BY R.REGION;

<region name="Australasia">
<population>24340222</population>
<area>7886602</area>
<populationSum>24340222</populationSum>
<areaSum>7886602</areaSum>
</region>

<region name="Near East">
<population>1499157105</population>
<area>4721322</area>

```

<populationSum>1499157105</populationSum>

<areaSum>4721322</areaSum>

</region>

## T2 : Les principales opérations liées à XML :

- \* Former des documents XML à partir de données relationnelles
- \* Créer un nouveau document XML dans la base oracle
- \* Extraire des données à partir de document XML
- \* Modifier le contenu de document XML

### D1 : Exemples :

#### Créer un nouveau document XML dans la base oracle:

```
CREATE TABLE XML(  
ID NUMBER PRIMARY KEY NOT NULL,  
XML XMLType);  
INSERT INTO XML(ID, XML) SELECT 2, XMLType(OBJ) FROM TMP_CLOB;  
COMMIT;  
SELECT XML FROM XML WHERE ID=2  
<?xml version="1.0"?><demographics><region id="1"><name>Africa</name><population  
>789548670</population><area>26780325</area><country id="1" code="AG"><name>Alge  
ria</name><population><year year="1950" population_id="12009" population="889271  
8"><births_per_1000>0</births_per_1000><deaths_per_1000>0</deaths_per_1000><migr  
ants_per_1000>0</migrants_per_1000><natural_increase_percent>0</natural_increase  
_percent><growth_rate>0</growth_rate></year><year year="1951" population_id="120  
10" population="9073304"><births_per_1000>0</births_per_1000><deaths_per_1000>0<  
/deaths_per_1000><migrants_per_1000>0</migrants_per_1000><natural_increase_perce  
nt>0</natural_increase_percent><growth_rate>0</growth_rate></year><year year="19  
52" population_id="12011" population="9279525"><births_per_1000>0</births_per_10  
00><deaths_per_1000>0</deaths_per_1000><migrants_per_1000>0</migrants_per_1000><...  
SELECT X.XML.EXTRACT('/') AS test FROM XML X WHERE X.ID = 2;  
<demographics>  
<region id="1">  
<name>Africa</name>  
<population>789548670</population>  
<area>26780325</area>  
<country id="1" code="AG">  
<name>Algeria</name>  
<population>  
<year year="1950" population_id="12009" population="8892718">  
<births_per_1000>0</births_per_1000>  
<deaths_per_1000>0</deaths_per_1000>...
```

#### Modifier le contenu de document XML:

```

UPDATE XML SET XML = UPDATEXML(XML, '/demographics/region/country[name="Canada"]
/state[name="British Columbia"]/name/@code', 'BU')
WHERE ID = 2;
UPDATE XML SET XML = UPDATEXML(XML,
'/demographics/region/country[name="Canada"]/state[name="British Columbia"]', NULL)
WHERE ID = 2;

```

D1:

XQuery\_string is a complete XQuery expression, possibly including a prolog, as a literal string. It can contain XQuery variables that you bind using the XQuery PASSING clause (XML\_passing\_clause in the syntax diagram). The predefined namespace prefixes recognized for SQL/XML function XMLQuery are also recognized in XQuery\_string — see "Predefined Namespaces and Prefixes".

The XML\_passing\_clause is the keyword PASSING followed by one or more SQL expressions (expr) that each return an XMLType instance or an instance of a SQL scalar data type. All but possibly one of the expressions must each be followed by the keyword AS and an XQuery identifier. The result of evaluating each expr is bound to the corresponding identifier for the evaluation of XQuery\_string. If there is an expr that is not followed by an AS clause, then the result of evaluating that expr is used as the context item for evaluating XQuery\_string. Oracle XML DB supports only passing BY VALUE, not passing BY REFERENCE, so the clause BY VALUE is implicit and can be omitted.

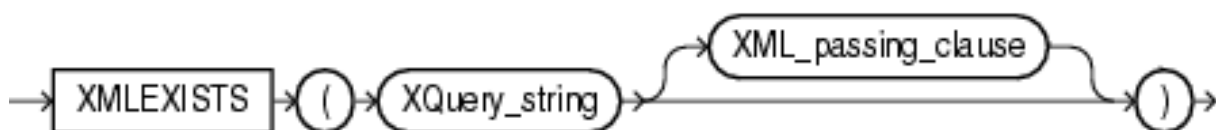
If an XQuery expression such as /PurchaseOrder/Reference or /PurchaseOrder/Reference/text() targets a single node, then XMLEExists returns true for that expression. If XMLEExists is called with an XQuery expression that locates no nodes, then XMLEExists returns false.

Function XMLEExists can be used in queries, and it can be used to create function-based indexes to speed up evaluation of queries.

Note:

Oracle XML DB limits the use of XMLEExists to a SQL WHERE clause or CASE expression. If you need to use XMLEExists in a SELECT list, then wrap it in a CASE expression: CASE WHEN XMLEExists(...) THEN 'TRUE' ELSE 'FALSE' END

**Figure 2-1 XMLEExists Syntax :**



**XML\_passing\_clause ::=**



**Exemple:**

**SELECT OBJECT\_VALUE**

**FROM purchaseorder**

**WHERE XMlexists('/PurchaseOrder[SpecialInstructions="Expedite"]'**

**PASSING OBJECT\_VALUE);**

**OBJECT\_VALUE**

-----  
**<PurchaseOrder xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"**

- **XMLCast()**

- Convert XML scalars into SQL scalars

- **XMLTransform()**

- XSL based transformation

- **XMLNamespaces()**

- Namespace management

- **SchemaValidate()**

- XMLType method for validating document against an XML Schema

**D1:**

SQL/XML standard function XMLCast casts its first argument to the scalar SQL data type specified by its second argument. The first argument is a SQL expression that is evaluated. Any of the following SQL data types can be used as the second argument:

NUMBER

VARCHAR2

CHAR

CLOB

BLOB

REF XMLTYPE

any SQL date or time data type

Note:

Unlike the SQL/XML standard, Oracle XML DB limits the use of XMLCast to cast XML to a SQL scalar data type. Oracle XML DB does not support casting XML to XML or from a scalar SQL type to XML.

Figure 2-2 XMLCast Syntax :



Exemple:

#### Example 1-1 Accessing a Text Node Value using XMLCAST and XMLQuery

The following query returns the value of the text node associated with the Reference element that matches the target XPath expression. The value is returned as a VARCHAR2 value.

```

SELECT XMLCast(XMLQuery('$p/PurchaseOrder/Reference/text()'
                        PASSING OBJECT_VALUE AS "p" RETURNING CONTENT)
        AS VARCHAR2(30))
FROM purchaseorder;
  
```

```

XMLCAST(XMLQUERY('$P/PURCHASEO
  
```

-----

```

SBELL-2002100912333601PDT
  
```

1 row selected.

#### Example 1-2 Searching XML Content using XMExists, XMLCast, and XMLQuery

The following query uses XMExists to check if the XML document contains an element named Reference that is a child of the root element PurchaseOrder:

```

SELECT count(*) FROM purchaseorder
WHERE XMExists('$p/PurchaseOrder/Reference' PASSING OBJECT_VALUE AS "p");
COUNT(*)
  
```

-----

```

132
  
```

1 row selected.

```
SELECT XMLCast(XMLQuery('/PurchaseOrder/Reference' PASSING OBJECT_VALUE
                        RETURNING CONTENT)
              AS VARCHAR2(100)) "REFERENCE"
FROM purchaseorder
WHERE XMlexists('/PurchaseOrder[SpecialInstructions="Expedite"]'
              PASSING OBJECT_VALUE);
```

REFERENCE

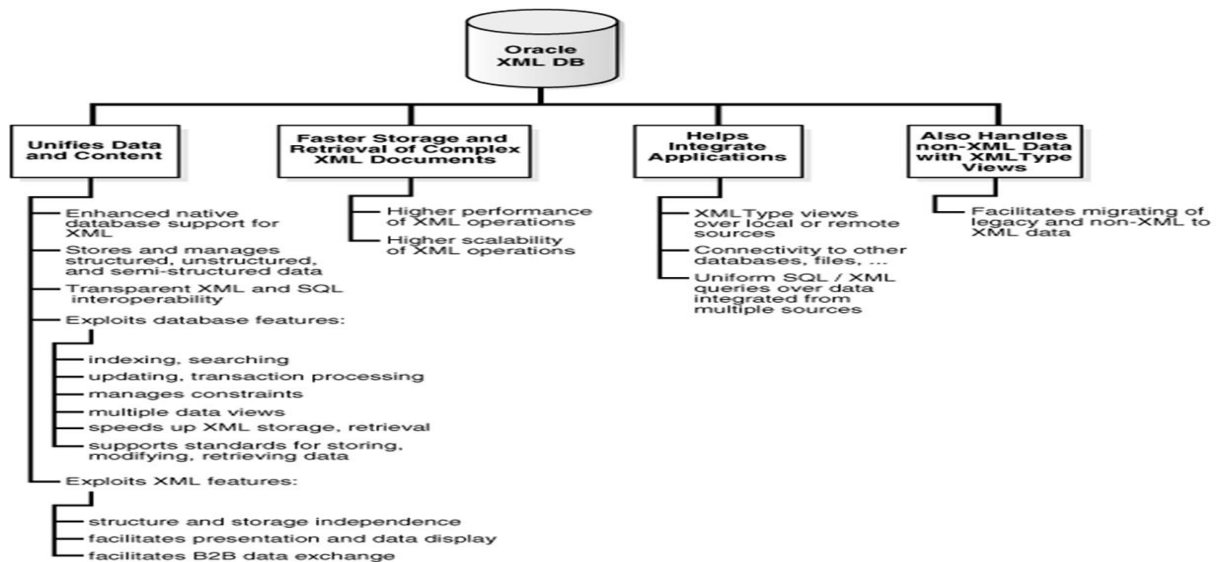
-----

AMCEWEN-20021009123336271PDT

SKING-20021009123336321PDT

## **T1: Valeurs ajoutées de Oracle XML DB**

- Développement rapide et aisé d'application XML natif
- Base de données hybride
  - Accès SQL au contenu XML
  - Accès XML aux données relationnelles
- Options de stockage XML multiple (tuning optimal des performances des applications)
  - Indépendance du code du modèle de stockage
  - Stockage et indexation optimisés pour XML



## T1 : Références Clients

| Oracle Customer Snapshot                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  <p><b>UPS</b><br/>Atlanta, GA<br/><a href="http://www.UPS.com">www.UPS.com</a></p> <p><b>Industry:</b><br/>Transportation Services</p> <p><b>Annual Revenue:</b><br/>Over US\$1B</p> <p><b>Employees:</b><br/>Over 10,000</p> <p><b>Oracle Products &amp; Services:</b><br/>Oracle Database 10g<br/>Oracle XML DB<br/>Oracle Siebel Analytics</p> <p><i>"The implementation of Oracle XML DB allows us to seamlessly manage thousands of daily transactions generated by our revolutionary DIAD technology. All this is accomplished without expensive, customized code." — Jim Schuetz, Project Manager, UPS</i></p> | <p><b>Oracle XML DB Enables UPS to Daily Streamline Management of its Tens of Thousands DIAD Transactions</b></p> <p>Founded in 1907 as a messenger company in the United States, UPS has grown into a \$42.6 billion corporation by clearly focusing on the goal of enabling commerce around the globe. Today UPS is a global company with one of the most recognized and admired brands in the world and has become the world's largest package delivery company and a leading global provider of specialized transportation and logistics services. Every day, UPS manages the flow of goods, information and funds in more than 200 countries and territories worldwide. UPS revolutionized the package delivery business when it developed and deployed the first Delivery Information Acquisition Device, known as the DIAD.</p> <p><b>Challenges</b></p> <ul style="list-style-type: none"> <li>Collect XML-based DIAD data sent from various field locations to a central Oracle XML DB parsing engine</li> <li>Enable functionality that transforms unstructured content from the DIADs into structured data for further processing.</li> <li>Manage up to 66 weeks of DIAD detail data</li> </ul> <p><b>Solution</b></p> <ul style="list-style-type: none"> <li>With Oracle XML DB, UPS manages 80,000 to 115,000 DIAD XML transactions (100KB each) per day</li> <li>Using Oracle XML DB to manage the DIAD versions of differing XML schemas, UPS is able to process up to 66 weeks of transactions of varying schema types</li> <li>Utilized the industry-leading Oracle Database and Oracle XML DB solution to support DIAD XML formats without incurring the time and expense involved in programming and custom code.</li> </ul> |

Copyright © 2006 Oracle. All rights reserved. Printed in the U.S.A. June 2006

**ORACLE®**

## T1 : Références

- \* [http://docs.oracle.com/cd/E11882\\_01/appdev.112/e23094/xdb04cre.htm#i1024803](http://docs.oracle.com/cd/E11882_01/appdev.112/e23094/xdb04cre.htm#i1024803)
- \* [http://www.oracle.com/pls/db112/portal.portal\\_db?selected=7&frame=#oracle\\_secure\\_files\\_and\\_large\\_objects](http://www.oracle.com/pls/db112/portal.portal_db?selected=7&frame=#oracle_secure_files_and_large_objects)
- \* <http://www.javaworld.com/javaworld/jw-03-2001/jw-0302-xmlmessaging.html>
- \* [http://books.google.fr/books/about/Beginning\\_XML.html?id=E\\_5CaOqSuAC&redir\\_esc=y](http://books.google.fr/books/about/Beginning_XML.html?id=E_5CaOqSuAC&redir_esc=y)