
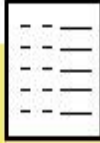




Oracle 12c

XML DB: XMLType

Rappel: Donnée et utilisation

								
	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				

Un mélange des deux

Données à forte structuration

Données à faible structuration

XML DB?

- Implémente les standards relatifs à XML (à partir de Oracle 9i Release 2)
- Permet l'accès:
 - SQL à des données XML
 - XML aux données SQL : **SQL/XML**
- Ajoute un Repository XML (comme SGF) à la BD Oracle avec HTTP, FTP et WebDAV(distributed authoring and versionning)
- Introduit le type XMLType avec support du XSD
- Réécriture des expressions Xpath

Plan

- Les possibilités pour stocker le XML
- Vue globale du XMLType
- Les options de stockage avec XMLType.
- Exemples:
 - Création d'une structure de rangement: en colonne ou en table
 - Gestion des options de stockage
 - XMLView
- Synthèse

Plan

- Les possibilités pour stocker le XML
- Vue globale du XMLType
- Les options de stockage avec XMLType.
- Exemples:
 - Création d'une structure de rangement: en colonne ou en table
 - Gestions options de stockage
 - XMLView
- Synthèse

Les différentes possibilités de stockage

- Annexe: Stockage CLOB
- Stockage dans XMLType

Stockage CLOB + OracleText Cartridge (1)

Aucune délégation au SGBD l'utilisateur est le seul à savoir qu'il gère du XML

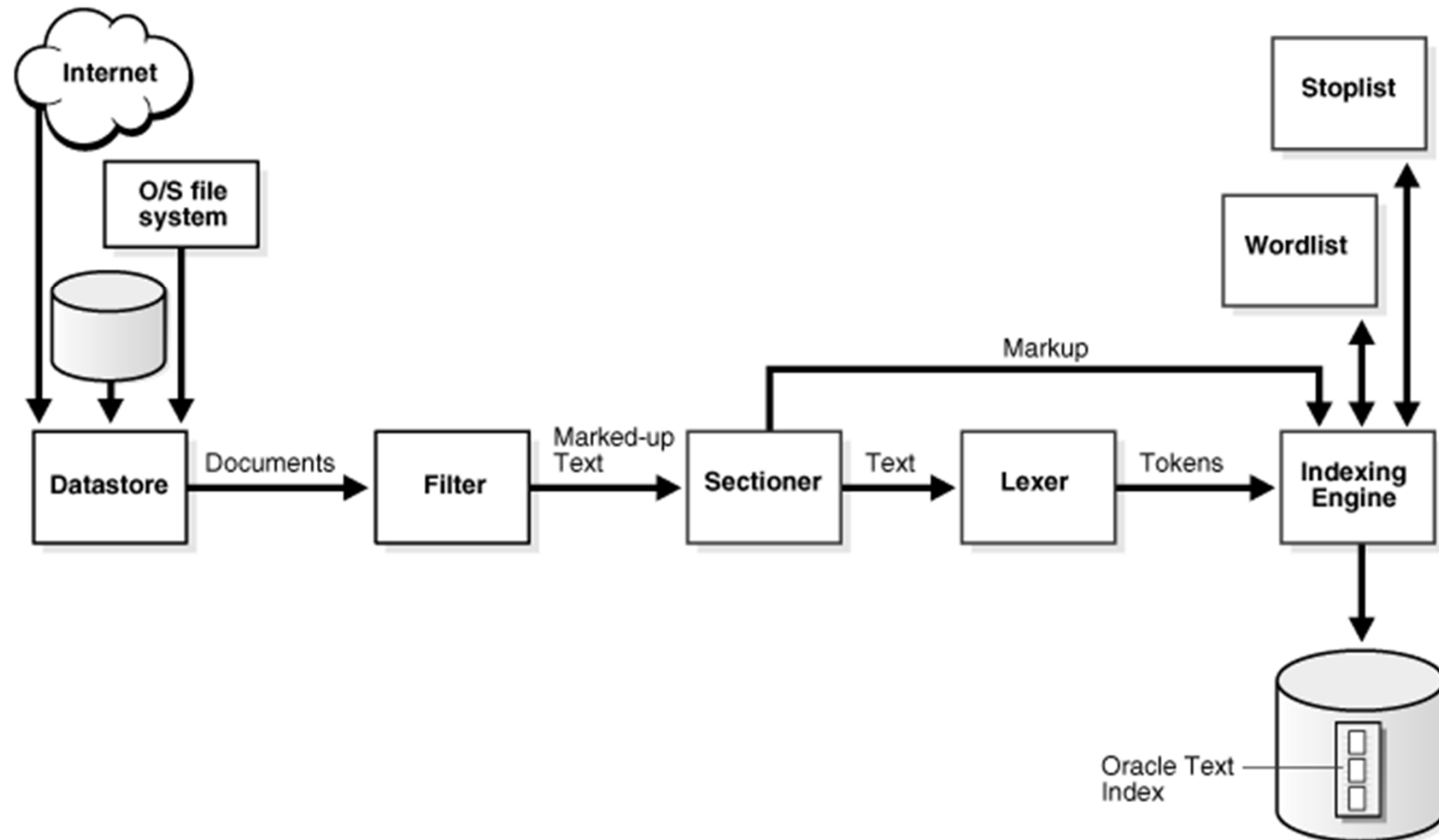
- Stocker un XML en tant que document unique

```
Create table Doc(id INTEGER, txt CLOB)
```

- Utiliser *l'indexation oracle de texte* pour une recherche textuelle (extraction de sous-chaîne,...)

```
Create Index doc_ind on doc(txt)  
Indextype is CTXSYS.CONTEXT  
Parameters (...)
```

Indexation textuelle oracle



Stockage CLOB + OracleText Cartridge (2)

- Utiliser les fonctionnalités: *WITHIN, INPATH, HASPATH + XPATH*

```
Select id, txt from Doc  
where contains(txt, 'Zep inpath (/livres/BD/dessinateur)') > 0
```

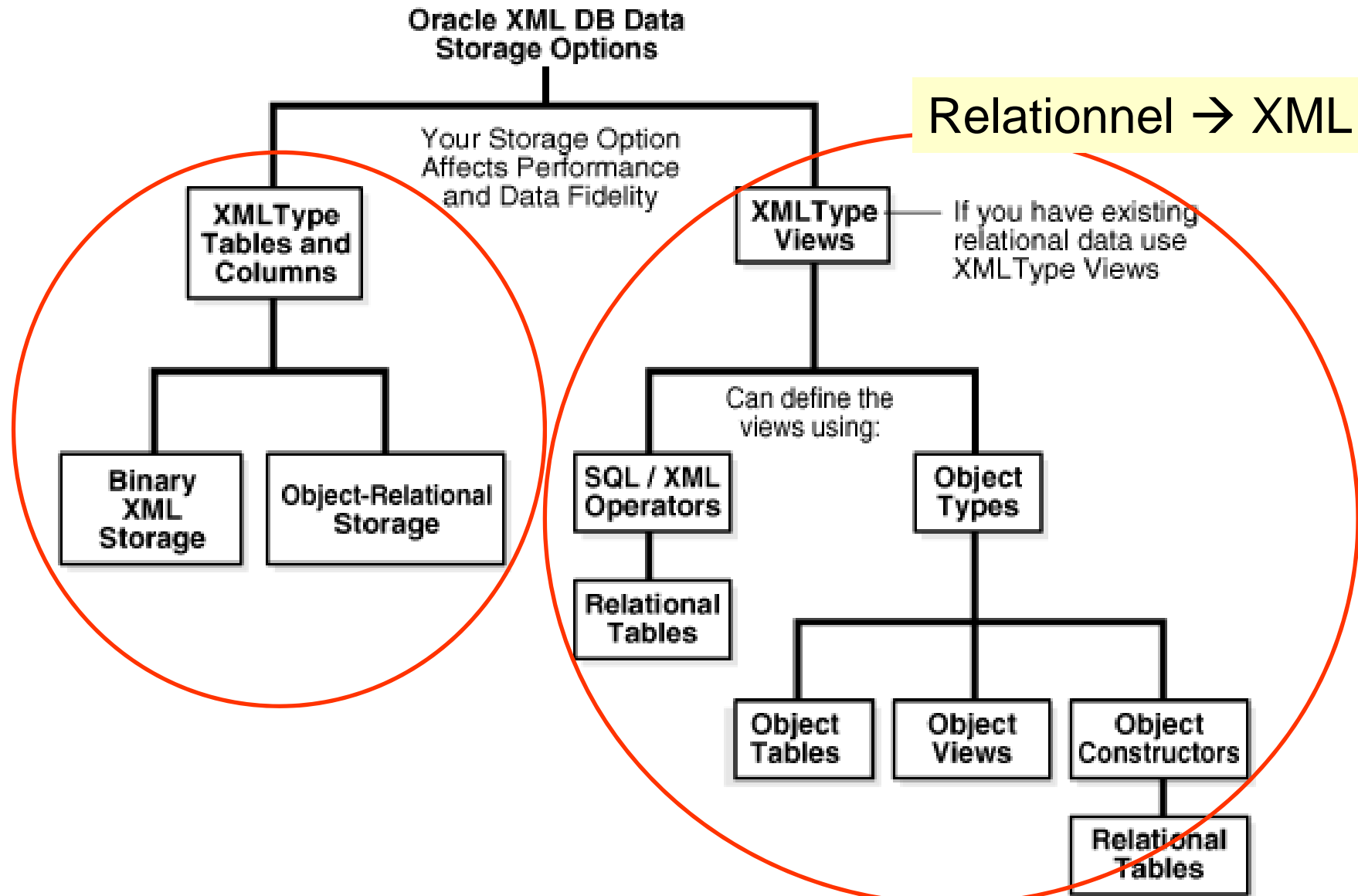
Plan

- Les possibilités pour stocker le XML
- Vue globale du XMLType
- Les options de stockage avec XMLType.
- Exemples:
 - Création d'une structure de rangement: en colonne ou en table
 - Gestions options de stockage
 - XMLView
- Synthèse

Type XML

- Un **type natif** de oracle utilisable pour les tables, les colonnes, les variables et les paramètres du PL/SQL. Il **peut être valide** à une XSD.
- Dispose des méthodes:
 - validation XML schema
 - transformations XSL
 - Requêtage XML
- APIs XML type :
 - Constructeurs, existsNode(), Extract(), schemaValidate(), Transform()

XMLType type de stockage



XMLType View

Relationnel → XML

- « Présente » des données relationnelles en format XML, pour être utilisée à la fois pour la publication Web et l'échange de données.

SQL/XML

- ☺ XMLType views peuvent être basé sur une XSD:
 1. Donc validation possible
 2. Optimisation des requêtes XPath.

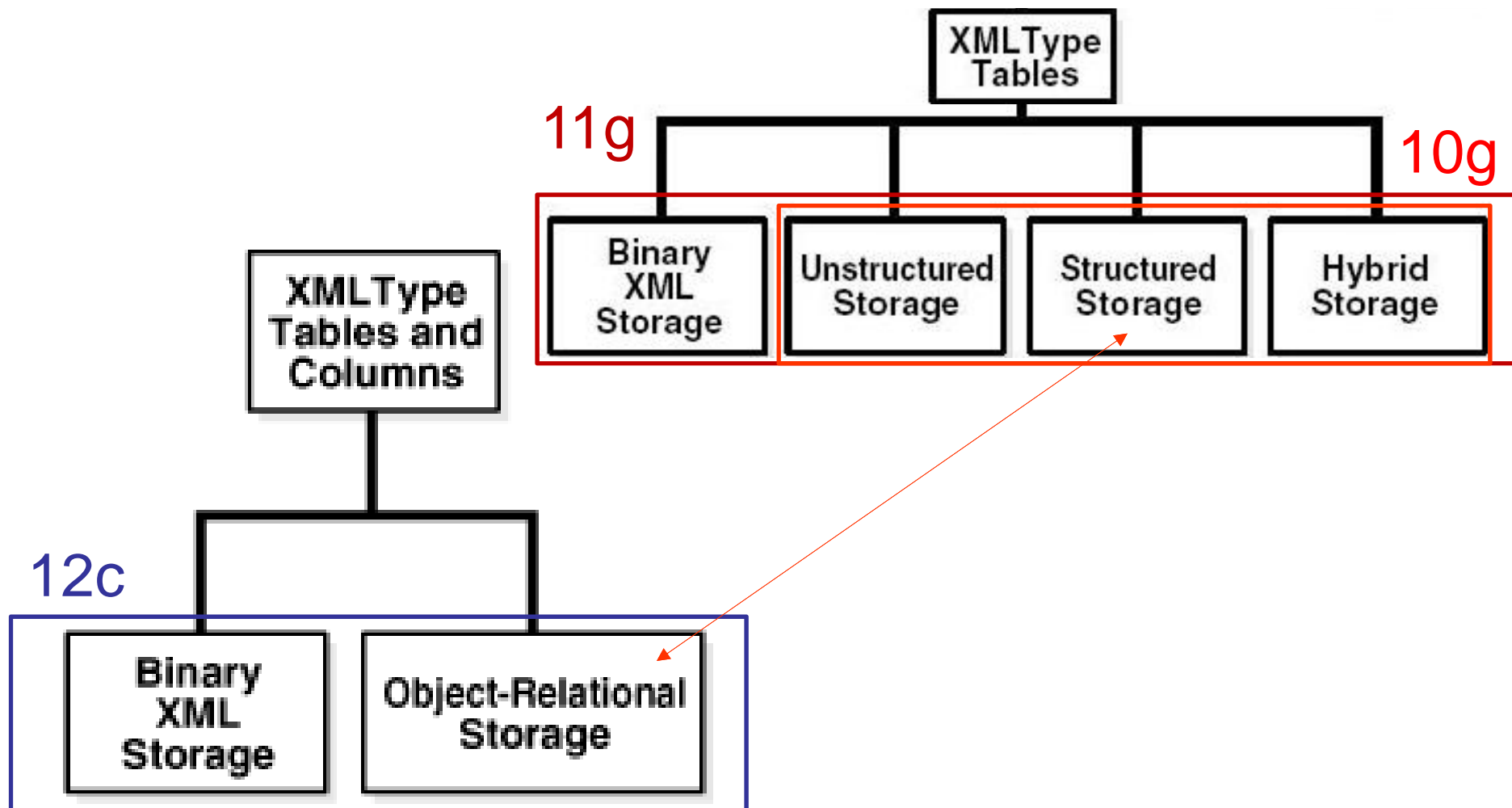
XMLType View: remarques

- 😊 Très rapide pour une utilisation pure SQL
- 😞 Les vues avec beaucoup d'imbrications sont lentes
- 😞 Mise à jour / insertion nécessite des triggers
- 😞 Pas de garantie pour l'ordre
- 😊 Prise en charge de plusieurs XSD pour une même table relationnelle

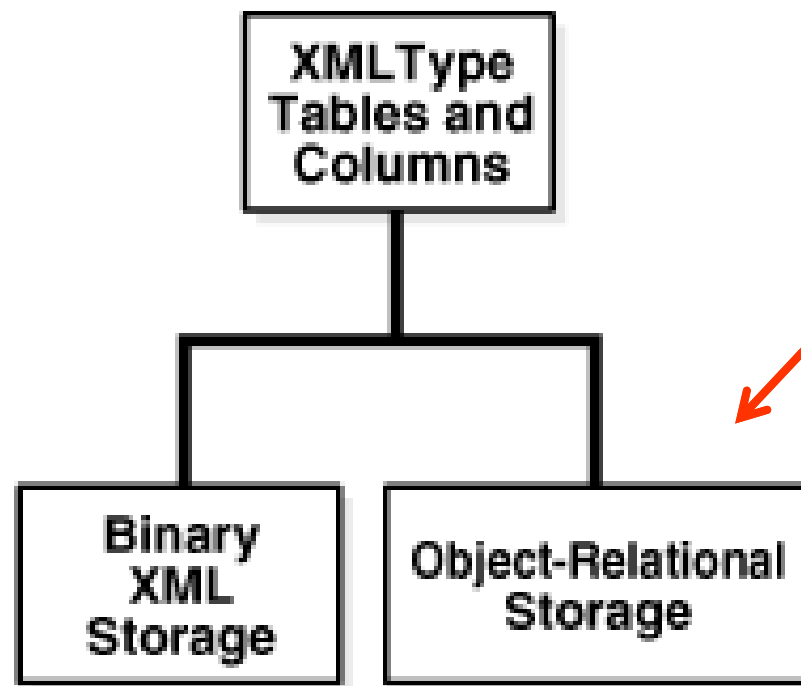
Plan

- Les possibilités pour stocker le XML
- Vue globale du XMLType
- Les options de stockage avec XMLType.
- Exemples:
 - Création d'une structure de rangement: en colonne ou en table
 - Gestions options de stockage
 - XMLView
- Synthèse

Evolution des options du stockage XMLType dans oracle



Object-Relationnal Storage



XMLType: structuré

Appli. Content centric

- **Catégorie:** Stockage structuré via un découpage en tables (via XSD, utilisation d'un mapping générique)
- **Particularité:** Intégrité DOM du XML (pas de conservation des espaces)

☺ Optimisation SGBD maximale (index,mémoire,...):
requêtes XPATH et update locaux.

☹ On ne peut pas changer de XSD sans tout recharger
les données

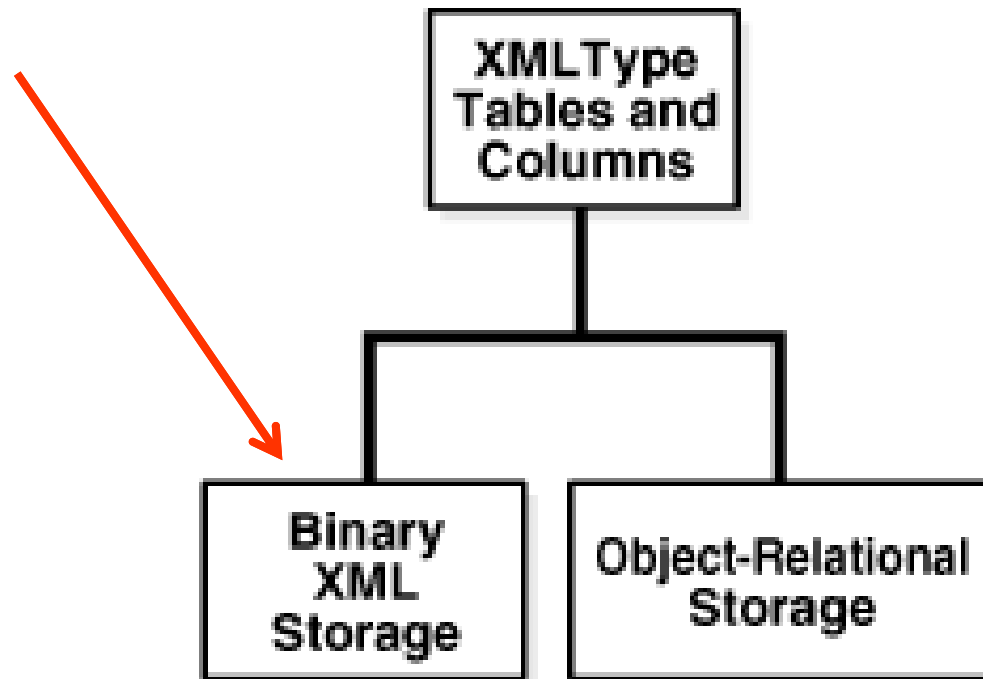
☹ Petit overhead pour la reconstitution entière du
document

XMLType structuré: démarche

- Si avec XSD alors: **Enregistrer une XSD** puis l'associer à la colonne ou à la table de type XMLType via **XMLSHEMA**.
- Créer des tables XMLType en spécifiant **DEFAULT TABLE** annotation en **enregistrant une XSD**. Permet à Oracle lors de l'insertion des documents dans ORACLE XML DB repository d'être automatiquement stockés dans ces tables par défaut (donc utilisation possible de SQL + ftp + http/Webdav).

☞ Toutes ces approches génèrent un ensemble de tables/objets via les techniques de mapping.

Binary XML storage



XMLType: Stockage binaire

Appli. Document centric

- **Catégorie:** Stockage optimisé du XML (**sans XSD obligatoire**)
- **Particularité:** préserve infoset et fidélité DOM

☺ Accepte tout document.
☺ Efficace pour la recherche/modif/insertion
☺ Index XML dédiés *(that is a new way to improve access performance of your XML. In the past, your options were Xpath function based indexes (using extract or extractvalue) or Oracle Text. Both of those have some limitations that XML Index tries to address.)*

XML instance document: employees.xml

```
...
<employee>
  <first_name>Shelli</last_name>
  <last_name>Baida</last_name>
  <email>sbaida</email>
  ...
  <hire_date>24-DEC-97</hire_date>
  ...
  <dept_id>30</dept_id>
</employee>
...
```

XML schema: employees.xsd

```
...
<sequence>
  <element name="first_name" type="string"/>
  <element name="last_name" type="string"/>
  <element name="email" type="string"/>
  ...
  <element name="hire_date" type="date"/>
  ...
  <element name="dept_id" type="integer"/>
</sequence>
...
```

**Create
XMLType
Table**

Binary XML Storage

**Object-Relational
Storage**

employees Tables			
...	XMLType Column
...	Binary XML
...	Binary XML
...	Binary XML
...	Binary XML
...	Binary XML

XML data stored as
binary XML

employees Tables			
first_name	last_name	email	dept_id
shelli	baida	sbaida	30
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

XML data stored in
object-relational columns and tables

Plan

- Les possibilités pour stocker le XML
- Vue globale du XMLType
- Les options de stockage avec XMLType.
- Exemples:
 - Création d'une structure de rangement: en colonne ou en table
 - Gestions options de stockage
 - XMLView
- Synthèse

Création/insertion: table XMLType

```
create table poDoc_tbl of XMLtype;
```

```
Insert into poDoc_tbl values  
(XMLType(bfilename('XMLDIR', 'invoicexml.xml'),  
          nls_charset_id('AL32UTF8')));
```

Pointeur créer sous oracle vers le répertoire (créer par ailleurs sur la machine) où sont entreposés les doc XML

```
CREATE OR REPLACE DIRECTORY XMLDIR AS  
'C:/myXMLdir';
```


Création: XMLType column

```
SQL> CREATE TABLE po_xml_tab(  
    poid number,  
    poDoc SYS.XMLTYPE  
);
```

```
SQL> ALTER TABLE po_xml_tab ADD (custDoc sys.XMLType);
```

```
SQL> DESC po_xml_tab;
```

```
-----  
POID          NUMBER  
PODOC         SYS.XMLTYPE  
CUSTDOC       SYS.XMLTYPE
```

```
SQL> ALTER TABLE po_xml_tab DROP (custDoc );
```

Insertion: XMLType.createXML

```
SQL> INSERT INTO po_xml_tab VALUES  
      (100, sys.XMLType.createXML(''  
        <PO>  
          <PONO>1</PONO>  
          <PNAME>Po_1</PNAME>  
          <CUSTNAME>John</CUSTNAME>  
          <SHIPADDR>  
            <STREET>1033, Main Street</STREET>  
            <CITY>Sunnyvale</CITY>  
            <STATE>CA</STATE>  
          </SHIPADDR>  
        </PO>''));
```

```
-- SQLPlus Display  
SQL> SET LONG 10000
```

```
SQL> SELECT podoc FROM po_xml_tab;
```

```
PODOC
```


```
-----  
<PO>  
  <PONO>1</PONO>  
  <PNAME>Po_1</PNAME>  
  <CUSTNAME>John</CUSTNAME>  
  <SHIPADDR>  
    <STREET>1033, Main Street</STREET>  
    <CITY>Sunnyvale</CITY>  
    <STATE>CA</STATE>  
  </SHIPADDR>  
</PO>
```

Création d'une table ou colonne binary XML

```
CREATE TABLE XMLBIN of XMLType XMLTYPE  
store AS BINARY XML;
```

```
SQL> insert into XMLBIN values (XMLType(BFILENAME  
( 'XML_DIR', 'test_document.xml'), nls_charset_id('AL32UTF8')));
```

```
CREATE TABLE product(  
  id VARCHAR(10),  
  name VARCHAR2(100),  
  description XMLType  
XMLType COLUMN description  
STORE AS BINARY XML;
```



Définit la colonne description utilisant le type binaire

Plan

- Les possibilités pour stocker le XML
- Vue globale du XMLType
- Les options de stockage avec XMLType.
- **Exemples:**
 - Création d'une structure de rangement: en colonne ou en table
 - **Gestions options de stockage**
 - XMLView
- Synthèse

Enregistrement d'une XSD: utilité

- Permet le mapping XML-SQL: Stockage structuré
- Permet la validation

☞ On peut demander un stockage en CLOB avec une XSD.

DBMS_XMLSCHEMA.registerSchema

XSD dans un fichier du répertoire

Permet d'enregistrer un schéma:

- En spécifiant une URL unique.
- La source du XSD
- CSID encoding character-set ID si XSD est un BFILE ou BLOB

```
BEGIN
DBMS_XMLSCHEMA.registerSchema(
  SCHEMAURL =>
    'http://xmlns.oracle.com/xdb/documentation/purchaseOrder.xsd',
  SCHEMADOC => bfilename('XMLDIR','purchaseOrder.xsd'),
  CSID => nls_charset_id('AL32UTF8'));
END;
```


DBMS_XMLSCHEMA.registerSchema

BEGIN

XSD directement créer

```
DBMS_XMLSCHEMA.registerSchema(  
  SCHEMAURL=>'http://xmlns.oracle.com/xml/content.xsd',  
  SCHEMADOC=>'<?xml version="1.0" encoding="UTF-8"?>  
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"  
  elementFormDefault="qualified">  
  <xs:element name="DESCRIPTION">  
    <xs:complexType mixed="true">  
      <xs:choice minOccurs="0" maxOccurs="unbounded">  
        <xs:element name="KEYWORD" type="xs:string"  
          maxOccurs="unbounded"/>  
      </xs:choice>  
    </xs:complexType>  
  </xs:element>  
</xs:schema>'  
END;
```

DBMS_XMLSCHEMA.registerSchema

Contrôle du stockage

```
BEGIN
DBMS_XMLSCHEMA.registerSchema(
  SCHEMAURL=>'http://xmlns.oracle.com/xml/content.xsd',
  SCHEMADOC=>'<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
  <xs:element name="DESCRIPTION">
    ....
  </xs:element>
</xs:schema>',
  LOCAL=>TRUE,
  GENTYPES=>TRUE, -- default TRUE
  GENTABLES=>FALSE );
END;
```

DBMS_XMLSCHEMA.register(): paramètres

- **Parameter LOCAL** controls whether the XML schema is registered as a local (TRUE) or global (FALSE) schema:
 - If the XML schema is registered as a global XML schema in the XML DB, it can be shared across different database users. Otherwise, XML schema sharing is not allowed.

Pour partager

- **Parameter GENTYPES** A boolean value determining if the schema compiler should generate object types. The default value is TRUE.

Pour préparer pour le stockage ou que validation

- **Parameter GENTABLES** boolean value determining if the schema compiler should generate default tables. The default value is TRUE.

Pour permettre le stockage dans les tables.

Sinon création manuelle des tables en vue d'optimiser la place

Si GENTYPES => TRUE

```
SQL> SELECT object_name, object_type FROM user_objects;
```

Voici les objets créés:

OBJECT_NAME	OBJECT_TYPE
-------------	-------------

DESCRIPTION163_T	TYPE
------------------	------

KEYWORD164_COLL	TYPE
-----------------	------

```
<xs:element name="DESCRIPTION">
  <xs:complexType mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="KEYWORD" type="xs:string"
        maxOccurs="unbounded"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

```
SQL> desc DESCRIPTION163_T
```

DESCRIPTION163_T is NOT FINAL

Name	Null?	Type
------	-------	------

SYS_XDBPD\$		XDB.XDB\$RAW_LIST_T
KEYWORD		KEYWORD164_COLL

```
SQL> desc KEYWORD164_COLL
```

KEYWORD164_COLL VARRAY(2147483647) OF VARCHAR2(4000)

To preserve
the DOM
fidelity. is
Intended for
Oracle
XML DB
internal use
only.

unbounded.

Création d'une colonne XML structuré

```
CREATE TABLE product(  
  id VARCHAR(10),  
  name VARCHAR2(100),  
  description XMLType)  
XMLType COLUMN description  
XMLSCHEMA "http://xmlns.oracle.com/xml/content.xsd"  
ELEMENT "DESCRIPTION";
```

Définit la colonne description comme basée sur une XSD + Mapping

Insertion dans un XMLType lié à une XSD

```
INSERT INTO product(id, name, description)
VALUES('xdk', 'XML Developer's Kit',
XMLTYPE('<DESCRIPTION>
<KEYWORD>xdk </KEYWORD> is a set of
standards-based utilities that helps to
build<KEYWORD>XML</KEYWORD> applications.
It contains XDK Java, C/C++
Components.</DESCRIPTION>').CreateSchemaB
asedXML(
'http://xmlns.oracle.com/xml/content.xsd'));
```

If not specified you will get the error - ORA-19007: Schema and element do not match.

Interroger un XMLType

```
SQL> SELECT p.description.extract('//KEYWORD')  
FROM product p;
```

```
P.DESCRPTION.EXTRACT('//KEYWORD')
```

```
<KEYWORD>xdk</KEYWORD>
```

```
<KEYWORD>XML</KEYWORD>
```


Quels sont les schémas enregistrés?

La partie du dictionnaire à interroger:

```
DESCRIBE DBA_XML_SCHEMAS;
```

Requête à envoyer:

```
SELECT OWNER, LOCAL, SCHEMA_URL  
FROM DBA_XML_SCHEMAS;
```

Plan

- Les possibilités pour stocker le XML
- Vue globale du XMLType
- Les options de stockage avec XMLType.
- **Exemples:**
 - Création d'une structure de rangement: en colonne ou en table
 - Gestions options de stockage
 - XMLView
- Synthèse

Les différentes type de “vues” avec du XML Type

1. A view where the XMLType **is a single column**: using a SQL/XML statement;
2. An **object XMLType view**, where each row object is an XMLType instance and is associated with a unique *OBJECT ID* (OID).

XMLType View - Example

```
CREATE VIEW po_view of XMLTYPE
XMLSCHEMA "po.xsd" ELEMENT "PurchaseOrder" AS
SELECT
    XMLElement("PurchaseOrder",
        XMLForest(p.ponum "PONum",
            p.company "Company"),
        (SELECT XMLAGG(
            XMLElement("Item",
                XMLForest(i.part "Part",
                    i.price "Price"))
            FROM items_rel_tab i WHERE i.po_id = p.id))
    FROM po_rel_tab p;
```


Les différentes type de “vues” avec du XML Type

1. A view where the XMLType is a single column: using a SQL/XML statement;
2. An **object XMLType view**, where each row object is an XMLType instance and is associated with a unique *OBJECT ID* (OID).

Either view, can be queried with SQL as a relational table with XMLType columns or an XMLType object table.

An object XMLType view

Creating XMLType object views, needs to specify the OBJECT ID via the OF XMLType WITH OBJECT ID syntax and assign the OBJECT ID by extracting a scalar value out of the XMLType.



```
CREATE VIEW clients_v OF XMLType
WITH OBJECT ID (extract(SYS_NC_ROWINFO$,
                        '/client/_id/text()').getnumberval())
AS
SELECT XMLELEMENT("client",
    XMLATTRIBUTES(C.C_id AS "_id"),
    XMLFOREST(C.cnom AS "nom")
) AS client
FROM Clients_rel C
```

Associate an XML schema with a view

XML Type

1- Enregistrer le schéma

```
BEGIN
  DBMS_XMLSCHEMA.registerSchema('http://www.oracle.com/emp_simple.xsd',
    <xs:schema xmlns=http://www.w3.org/2001/XMLSchema
      targetNamespace=http://www.oracle.com/emp_simple.xsd
      xmlns:xdb=http://xmlns.oracle.com/xdb
      elementFormDefault="qualified">
        <xs:element name="client" type="TClient"/>
        <xs:complexType name="TClient">
          <xs:sequence>
            <xs:element name="nom" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="_id" type="xs:integer"/>
        </xs:complexType>
      </xs:schema> ,
    TRUE, TRUE, FALSE);
END;
```

Associate an XML schema with a view

XML Type

2. Créer la vue en la rattachant au XSD

```
CREATE VIEW clients_v OF XMLType
XMLSCHEMA http://www.servidor.com/client.xsd
ELEMENT "client"
WITH OBJECT ID (extract(OBJECT_VALUE,
                        '/client/_id/text()').getnumberval())
AS
SELECT XMLELEMENT("client",
    XMLATTRIBUTES(
        'http://www.servidor.com/client.xsd' AS "xmlns",
        'http://www.w3.org/2001/XMLSchema-instance' AS "xmlns:xsi",
        'http://www.servidor.com/client.xsd' AS "xsi:schemaLocation",
        C.C_id AS "_id"),
    XMLATTRIBUTES(C.C_id AS "_id"),
    XMLFOREST(C.cnom AS "nom")
) AS client
FROM Clients_rel C
```


Plan

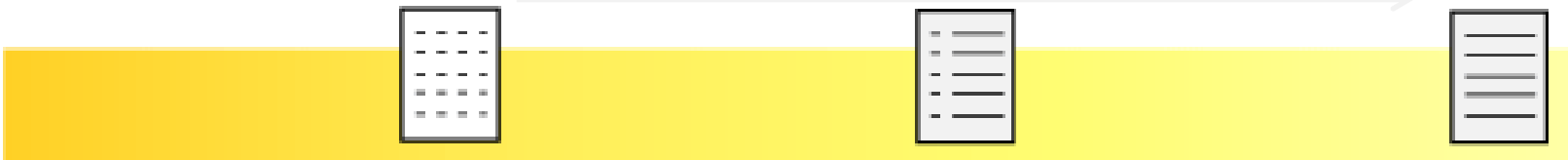
- Les possibilités pour stocker le XML
- Vue globale du XMLType
- Les options de stockage avec XMLType.
- Exemples:
 - Création d'une structure de rangement: en colonne ou en table
 - Gestions options de stockage
 - XMLView
- Synthèse

Synthèse

- **Stockage /Extraction en XML** : dispose d'un type de données natif XML
 - Création d'index relatifs au contenu des documents ou de leur structure,
 - gestion de pointeurs de type URL à l'intérieur de la base,
 - la reconnaissance du langage XPath/Xquery.
- **Manipulation XML**: API pour Java, C, C++ et PL/SQL. Exploitation de DOM et de SAX dans les analyseurs.

Index et stockage

From highly structured data to highly unstructured data.



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