

Module: XML et les bases de données

Identification des besoins XML- BDs

Houda Chabbi Drissi

houda.chabbi@hefr.ch

Trois types de besoins XML- BDs

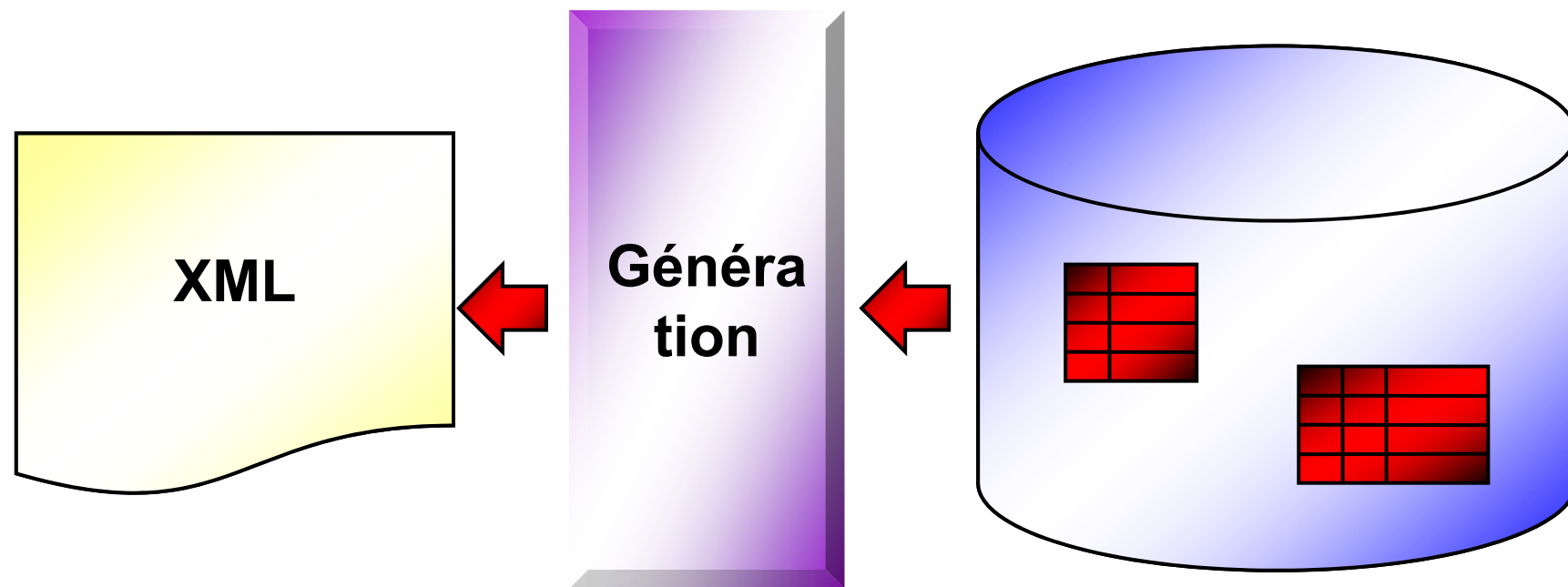
- Générer du XML d'une BDR

Données tabulaires → documents

- Stocker/extraire des documents XML
- Stocker/extraire/requeter des documents XML

Générer du XML d'une BDR

XML est extrait à partir des données relationnelles



- Echanges inter-applications
- Publication

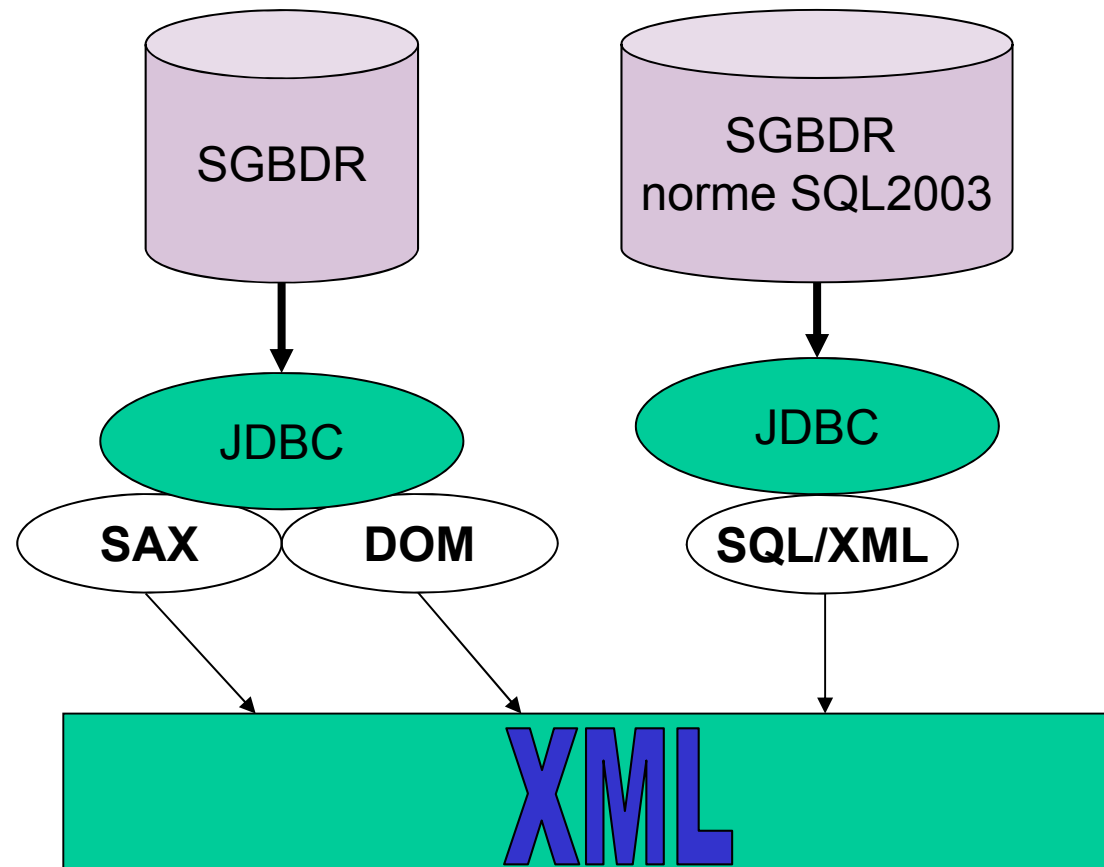
Génération de XML d'une BDR: approches

- Dans un **applicatif**:

- {**JDBC ou ADO**} + **SAX**
(cf. annexe SAX)

- Dans le **SGBD**:

- **SQL/XML** (SQL2003): Créer des vues XML (cf. suite)

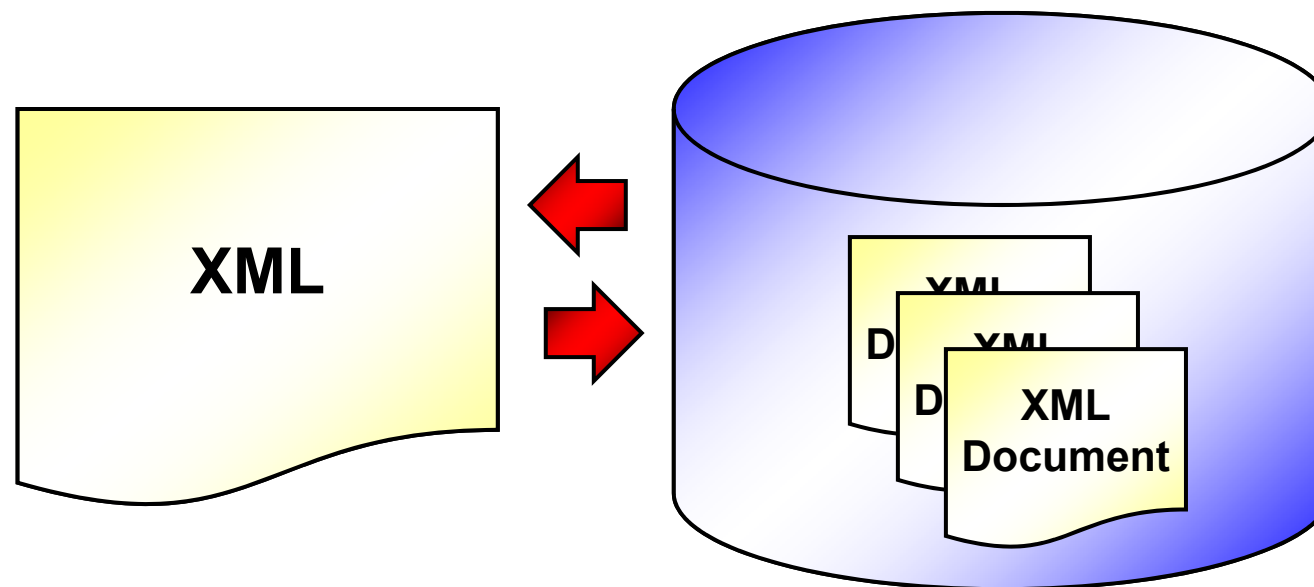


Trois types de besoins XML-BDs

- Générer du XML d'une BDR
- Stocker/extraire des documents XML:
granularité → document
- Stocker/extraire/requêter des documents XML

Stocker/extraire des documents XML

On doit stocker des documents (ou fragments) dans la base pour les travailler dans leur globalité.



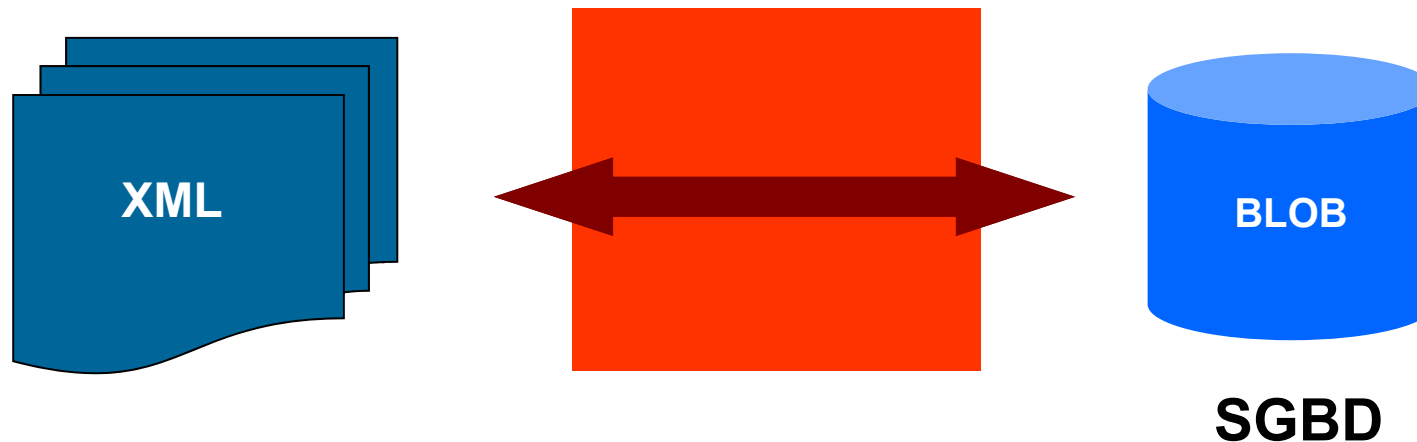
Publication



Stockage Files system + URL dans BD

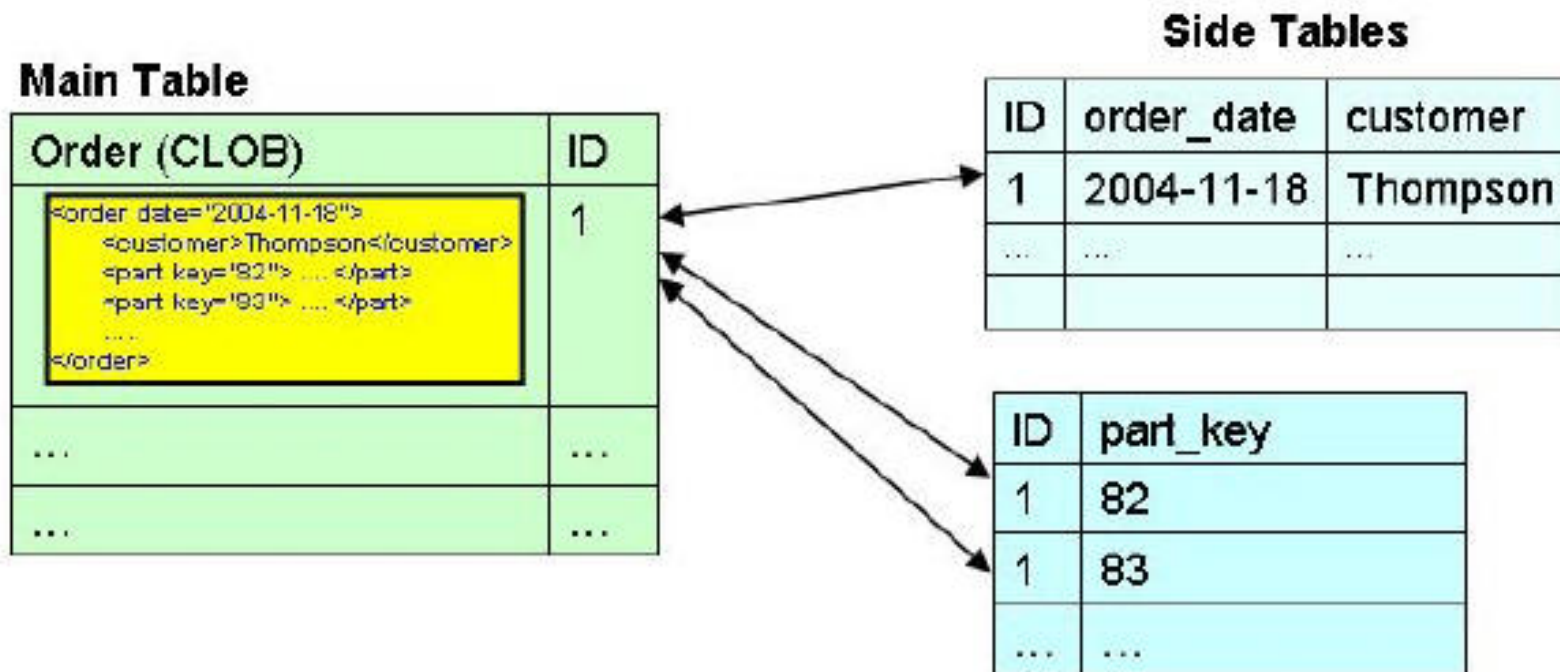
- Très simple
- Outils externe comme *grep* pour accéder XML
- ☹ Attention à la persistance!

Stocker XML dans un CLOB (1)



- **Mapper** les documents XML dans un champ binaire : BLOB (CLOB) ou Varchar
- Utilisation des technologies **full-text**

Stocker XML dans un CLOB (2)



Utilisation des CLOBs

- Avantages ☺
 - transactionnel
 - Sécurité
 - multiutilisateur
 - Administration
- Quelques produits sont XML-aware
 - Elimine le problème des tags
 - Compatible avec le langage XQUERY



Oracle 9i+
Db2

Utilisation des CLOBs

■ Désavantages ☹

- XML monolithique
- Pas d'accès à la structure
- Certaines recherches plein-texte peuvent-être inefficace:

Le rouge & le noir
œufs
Mr. <!-- --> Bean

"Le rouge & le noir" not found
"œufs" not found
" Mr. Bean" not found

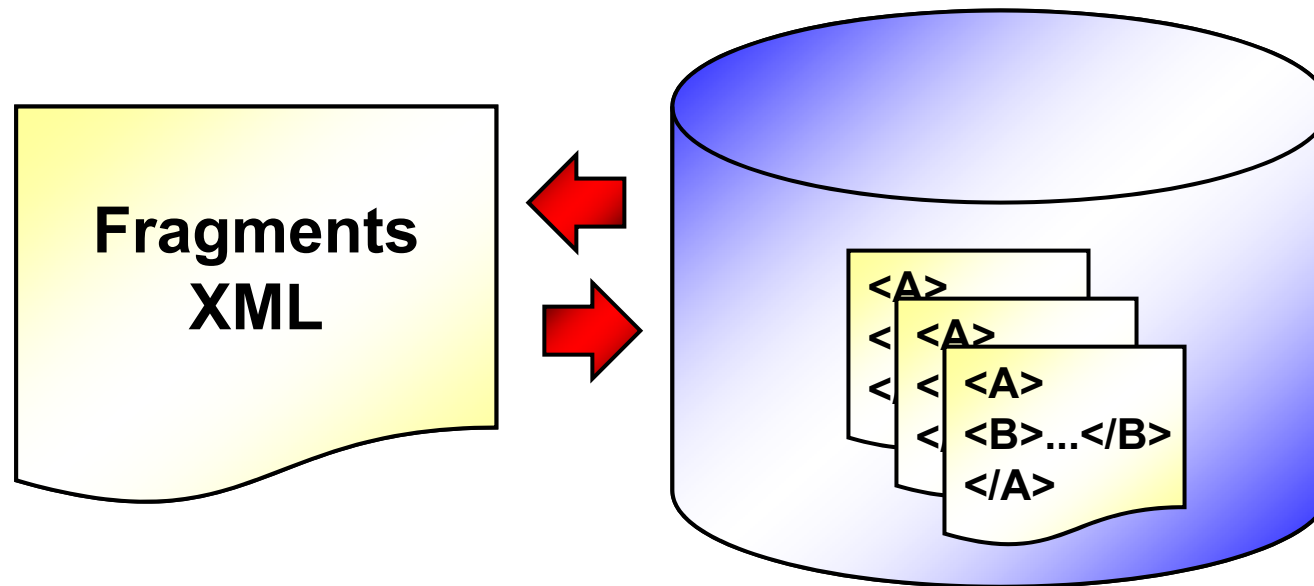
Trois types de besoins XML-BDs

- Générer du XML
- Stocker/extraire des documents XML
- Stocker/extraire/requêter des documents XML

granularité → éléments/attributs

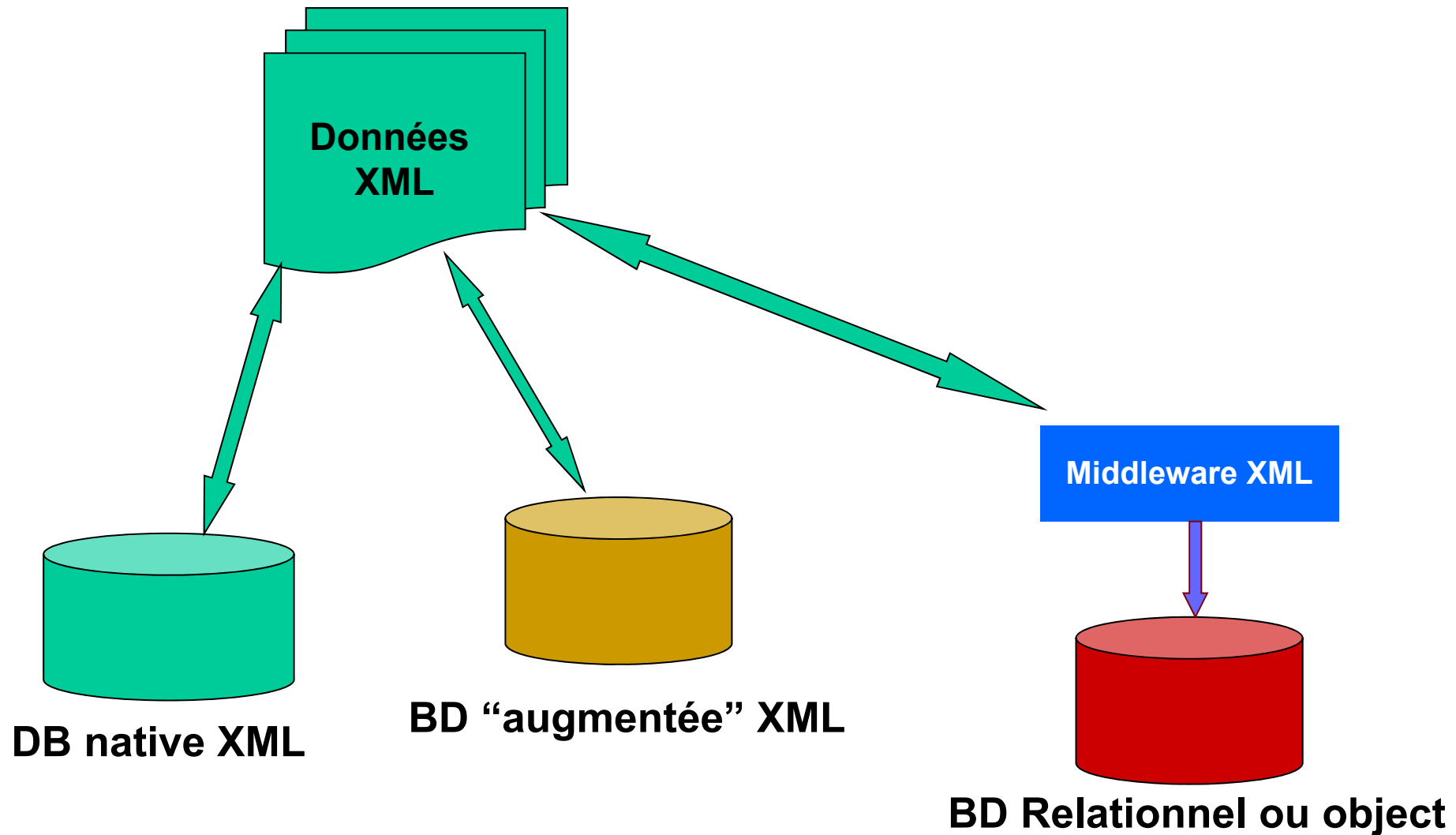
Stocker/extraire/requêter des documents XML

- Utilisation dédiée XML



Applications orientées XML

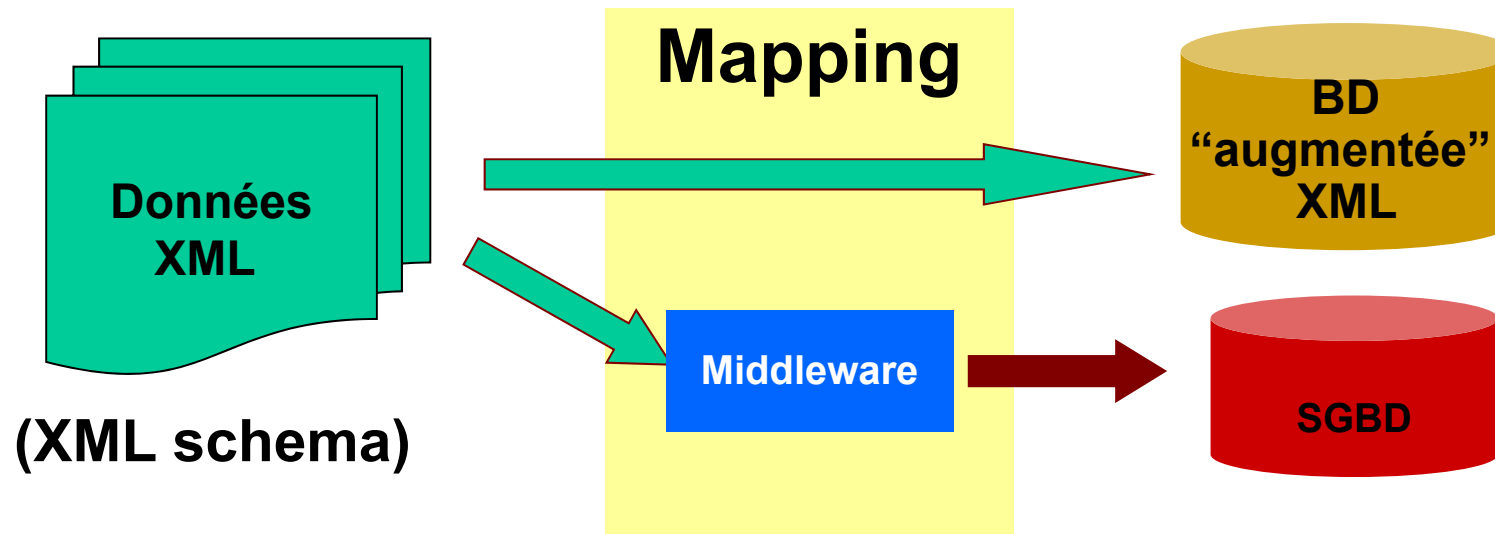
Les solutions pour gérer du XML



Les solutions dédiés XML

- Solutions avec SGBDs existants:
 - Middleware (Mécanisme de mapping)
- Solutions avec SGBDs augmentés XML
 - (SQL/XML)⁺ & Mécanisme de mapping
- Solutions avec XBDs

Mécanisme de mapping (1)



➤ “Mapper” = transformer des documents XML en tables et de tables en documents XML.

➤ Permet l'utilisation d'un langage XML 😊

Mécanisme de mapping (2)

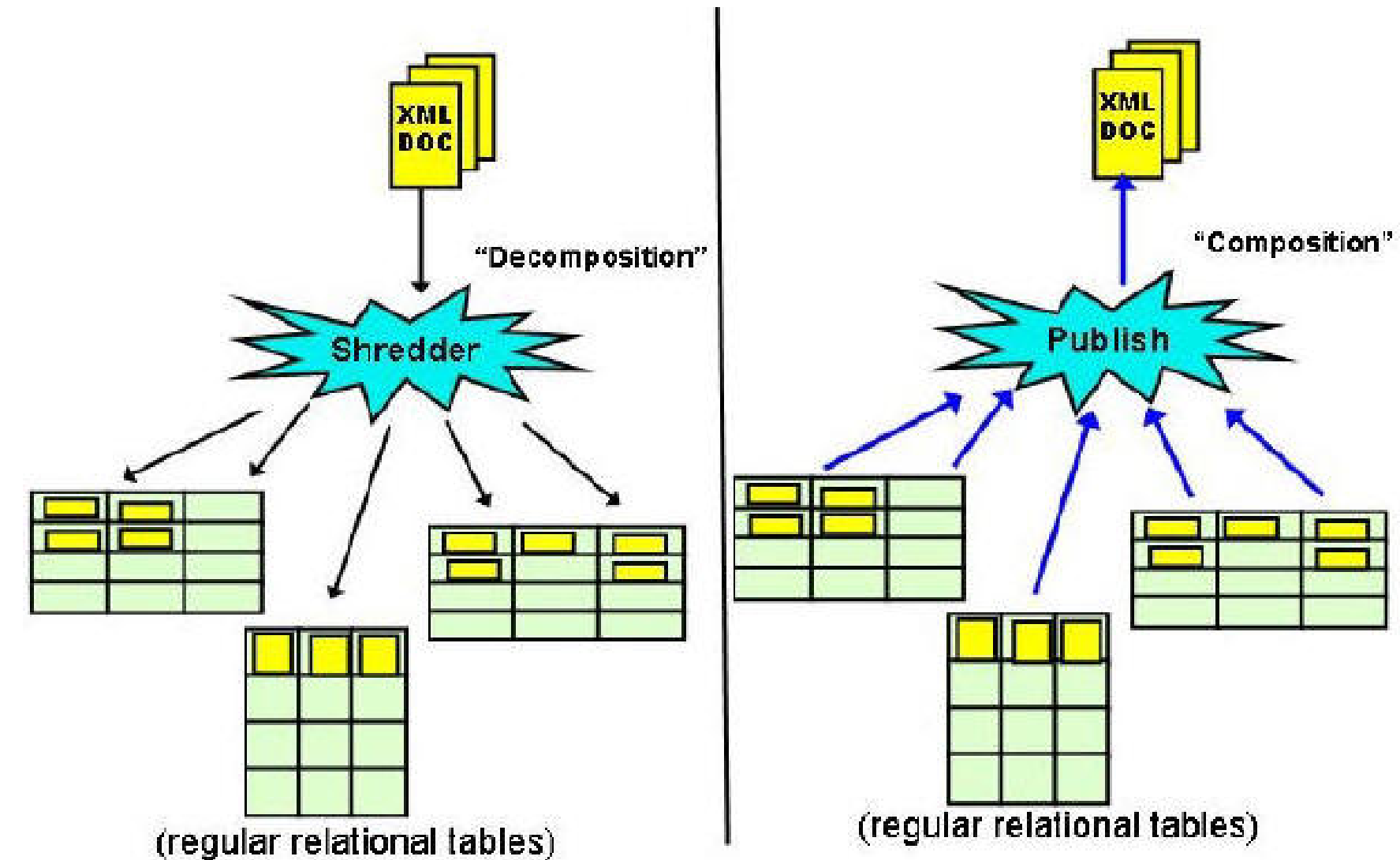
- Le mapping concerne (cf. suite)
 - Les éléments, attributs et textes
 - Ignore la structure physique (CDATA sections, encoding,)
 - Ignore les structures logiques (processing instructions, comments, order,)

- 2 approches du mapping
 - Table-Based mapping
 - Object-relational mapping



Un "import et un export" ne crée pas le même fichier (mais les mêmes données)

Le mapping: illustré



Les mécanismes pour les solutions dédiés XML

- Solutions avec SGBDs existants:
 - **Middleware (Mécanisme de mapping)**
- Solutions avec SGBDs augmentés XML
 - (SQL/XML) & Mécanisme de mapping
- Solutions avec XBDs

Middleware (1)

Logiciel en sur-couche qu'on utilise à travers une application pour importer ou exporter du XML d'un SGBD.

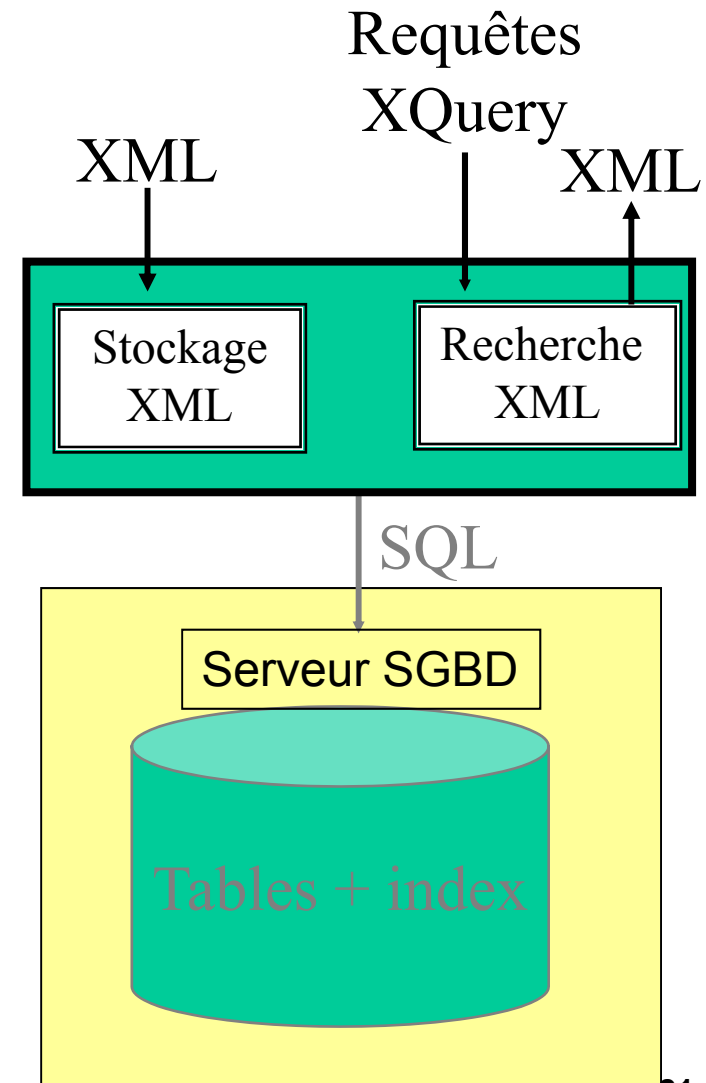
Exemples:

- **Castor:** (Open Source) <http://castor.exolab.org/index.html>
- **XMLizer (E-XMLMedia)**
<http://www.e-xmlmedia.com/prod/xmlizer.htm>

Middleware (2)

Permet:

- Stockage et interrogation XML
- Importé du XML dans tables
- Exporté du XML



Utilisation des techniques
de mapping



Les solutions dédiés XML

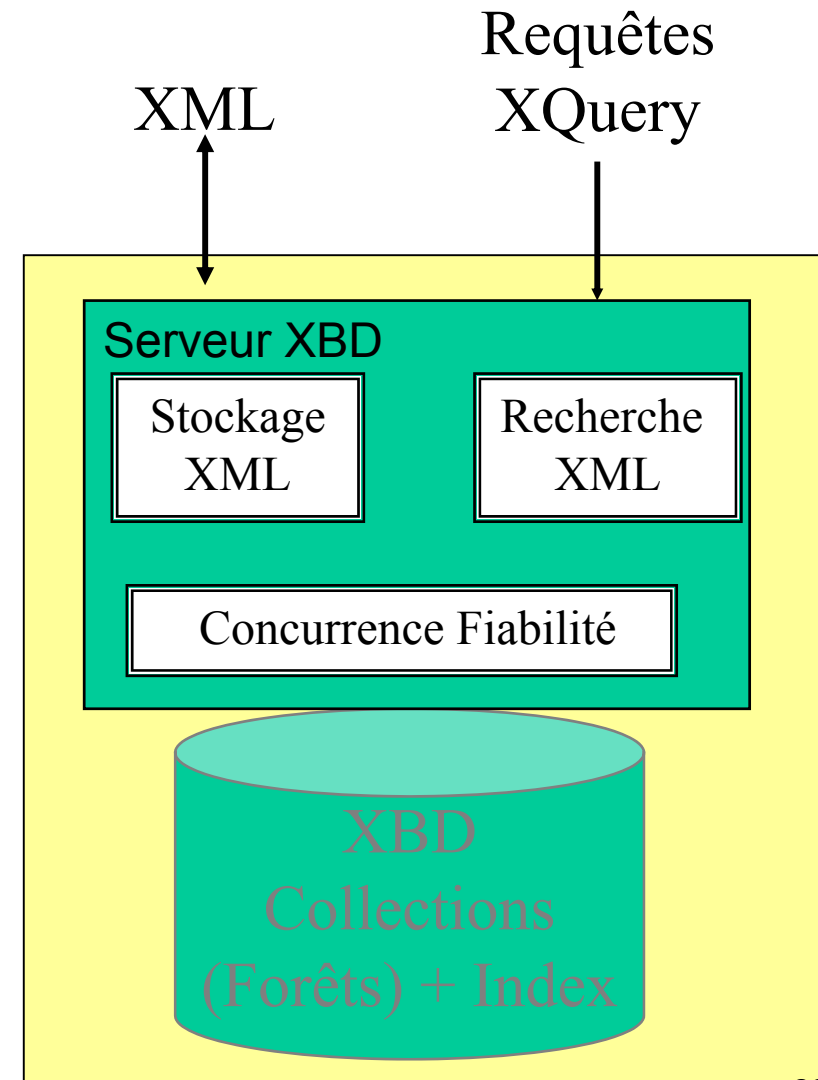
- Solutions avec SGBDs existants:
 - Mapping
 - Middleware

- Solutions avec SGBDs augmentés XML

- Solutions avec XBDs

Solution XDB

- SGBD conçu pour XML,
- Stocke:
 - ✓ Les documents en entier «sans les décomposer en éléments»,
 - ✓ Des fragments XML
 - ✓ Nœuds: DOM, ou autres
- Utilise des techniques d'indexation d'arbres spécifiques.



Stockage entier du document

- *Avantages:*
 - OK si manipulation par document entier.
 - OK si on veut requêter et que les documents sont petits

- *Désavantages:*
 - Pour les gros documents
 - Pour les requêtes et les updates

Stockage de nœuds XML (1)

- *Avantages:*
 - Pour les requêtes et les updates partiels.
 - L'indexation

- *Désavantages:*
 - A l'insertion
 - La sérialisation
 - Inflation dans la place occupée peut être

Stockage de nœuds XML (2)

■ Défis:

- Choix du modèle de stockage: quels nœuds? Avec ou sans le type?
- La gestion des updates
- La numérotation des nœuds:
 - ✓ Utile pour l'indexation
 - ✓ Utile pour établir l'ordre (précédence ..)
 - ✓ Utile pour faciliter les insertions / delete

Indexation

- Critique pour les performances.
- Inter-documents ou Intra-document.
- Peuvent être:
 - Orienté structure
 - Orienté valeur

Index structurel

- **Path**: Permet de répondre aux requêtes de type
 - Trouve tous les éléments: /a/b/c
 - D'existence: /a/b[c]
- **Sous-chemin**: Plus général que le **path** avec plus d'espace puisque indexe tous les chemins même les partiels:
 - Trouve tous les éléments: /a/b/c ou b/c

Indexation implicitement utilisée sur plusieurs systèmes

Index de valeur

- Permet de répondre aux requêtes:
 - **XPATH2** et **Xquery**
 - Supporte la comparaison "//nom[.='Toto']"
 - Supporte les intervals "//age[.<'25']"
- **Sous-chemin**: Plus général que le **path** avec plus d'espace puisque indexe tous les chemins même les partiels:
 - Trouve tous les éléments: /a/b/c ou b/c

Indexation implicitement utilisée sur plusieurs systèmes

XBD vs SGBDR

Une base de données XML ou relationnelle est une application qui masque la façon dont sont stockées les données : elle n'offre qu'une vue logique et un langage de requête **adaptée** à cette vue logique

	SGBDR	XBD
Vue logique	Tables = ensemble de nuplets	Collections = Forets
Langages de requête	SQL	XPath XQuery
Services	Transactions ACID	?Transactions ACID?

Synthèse sur les différentes techniques (1)

- Le mapping ou shredding

Cette technique consiste à répartir les données contenues dans un document XML au sein de tables relationnelles

- Le blobing ou clobing

Cette méthode consiste à insérer des fichiers entiers

- Le natif

Cette méthode n'implique aucune transformation explicite du document XML

Synthèse sur les différentes techniques (2)

	Avantages	Inconvénients
Mapping	Bénéficie des mécanismes des SGBDR (ACID, ...)	Peu extensible, Risque de perte de l'intégrité du document original, Nécessité de découper et reconstituer le document (jointure)
Clobing	Plus flexible que le mapping, Respect de l'intégrité, Permet d'effectuer des recherches plein texte	Granularité = Document Impossibilité d'effectuer des requêtes de type XPATH
Natif	Flexibilité, Respect de l'intégrité	Point critique: l'optimisation des requêtes.

Conclusion

Ce qu'apporte XML aux bases de données :

- **Un modèle complet** (XML Schema) :
hiérarchique+relations+attributs+héritage+typage
- **Un format d'échange standard** inter et intra-entreprise (XML) ainsi qu'un schéma inclus dans les données
- Un standard attendu permettant de **mixer documents et données**
- **Un langage de requêtes** puissant (XQuery)
- Une simplicité de **génération depuis des formats divers**:
SQL, fichiers légataires, documents (HTML, RTF)

Annexes

- Produits
- Bibliographie

The market for XML database products

- Middleware : Software you call from your application to transfer data between XML documents and databases
- XML-enabled database
- Native XML database
- Wrappers : Software that treats XML documents as a source of relational data. These products typically query XML documents using SQL.
- XML Query engines : Standalone engines that can query XML documents
- XML Data bindings : Products that can bind XML documents to objects. Some of these can also store/retrieve objects from the database.

Products : XML-Enabled database

- Main databases are XML-Enabled
- if you buy recent versions
- with important differences in the implementation of XML features

ACCESS (2000+)	Microsoft	Relational
DB2	IBM	OR
Informix	IBM	Relational
Oracle 9i 12c	Oracle	OR
SQL Server (2000+)	Microsoft	Relational
Sybase ASE	Sybase	Relational

Products : Native XML database

- Lot of opensource implementations

<u>BaseX</u>	OpenSource	Proprietary model
<u>MarkLogic Server</u>	<u>Commercial</u>	Object-oriented
eXist	OpenSource	Text model
sedna	OpenSource	Object-oriented
XIndice (Apache)	OpenSource	Proprietary (Model-based)

Comparison relational, XML-enabled, Native

yours needs	Relational DB	XML-Enabled DB	Native XML DB
Database to store very structured data. Complex and different queries / analysis, Lot of data	The best solution Allow to answer all queries with the good average performance	Why not ? To allow to exchange with XML formats	?
Database to store XML documents. Always the equivalent queries	XML documents = a binary information Not optimized	A good solution allowing to use the same database	The best solution.
Database to store structured data. Lot of import / export in XML exchanges.	A “classical” solution. Need import / export interface	A good solution	Why not ? A good solution. Allow to have quickly a database
Database to store structured data and to publish information (HTML, PDF, WebServices,)	A “classical” solution but need to develop tools to publish information	The best solution. Store in relational database but allows the push of XML Documents and transformation	Why not if you succeed in organizing your data

Links about XML Databases

- XML family : www.w3c.org
- Introduction to native XML database :
<http://www.xml.com/pub/a/2001/10/31/nativexmlldb.html>
- XQuery : <http://www.w3.org/TR/xquery-30>
- Oracle : <http://www.oracle.com/technetwork/database/database-technologies/xmlldb/overview/index.html>
- XIndice : <http://xml.apache.org/xindice/>
- eXist : <http://exist-db>
- BaseX: basex.org