



# *Gestion de documents XML dans une base de données*

---

**François GOLDGEWICHT**

Consultant, animateur du Pôle « XML et Web Services »

**CCT CNES - 17 juin 2008**



# Agenda



**1 : Introduction**

**2 : XML et les bases de données relationnelles**

**3 : XML et les bases de données XML natives**

**4 : Techniques mixtes**

**5 : Conclusion**

# Agenda



## 1 : Introduction

2 : XML et les bases de données relationnelles

3 : XML et les bases de données XML natives

4 : Techniques mixtes

5 : Conclusion

# Introduction



## XML standardisé depuis 10 ans

- Langage universel de description de documents puis des données
- Puis : formats de données convertis en XML
- Puis : passage progressif du procédural au déclaratif
  - Ex : les Web Services comme traduction de CORBA

## XML largement démocratisé aujourd'hui

- Utilisation massive de XML et de ses outils dans les échanges
  - Ex : Web Services, SOA
- Multiplication des fichiers XML
  - Nécessité de gérer ces fichiers dans des bases de données

## XML et les bases de données

- Différentes approches parfois disparues du marché pour rester confinées dans les laboratoires de recherche
  - Bases de données objet, déductives, actives...
- Trois approches possibles
  - Base de données relationnelle
  - Base de données XML native
  - Techniques mixtes : passer par un médiateur qui masque l'implémentation

# De XML aux bases de données...



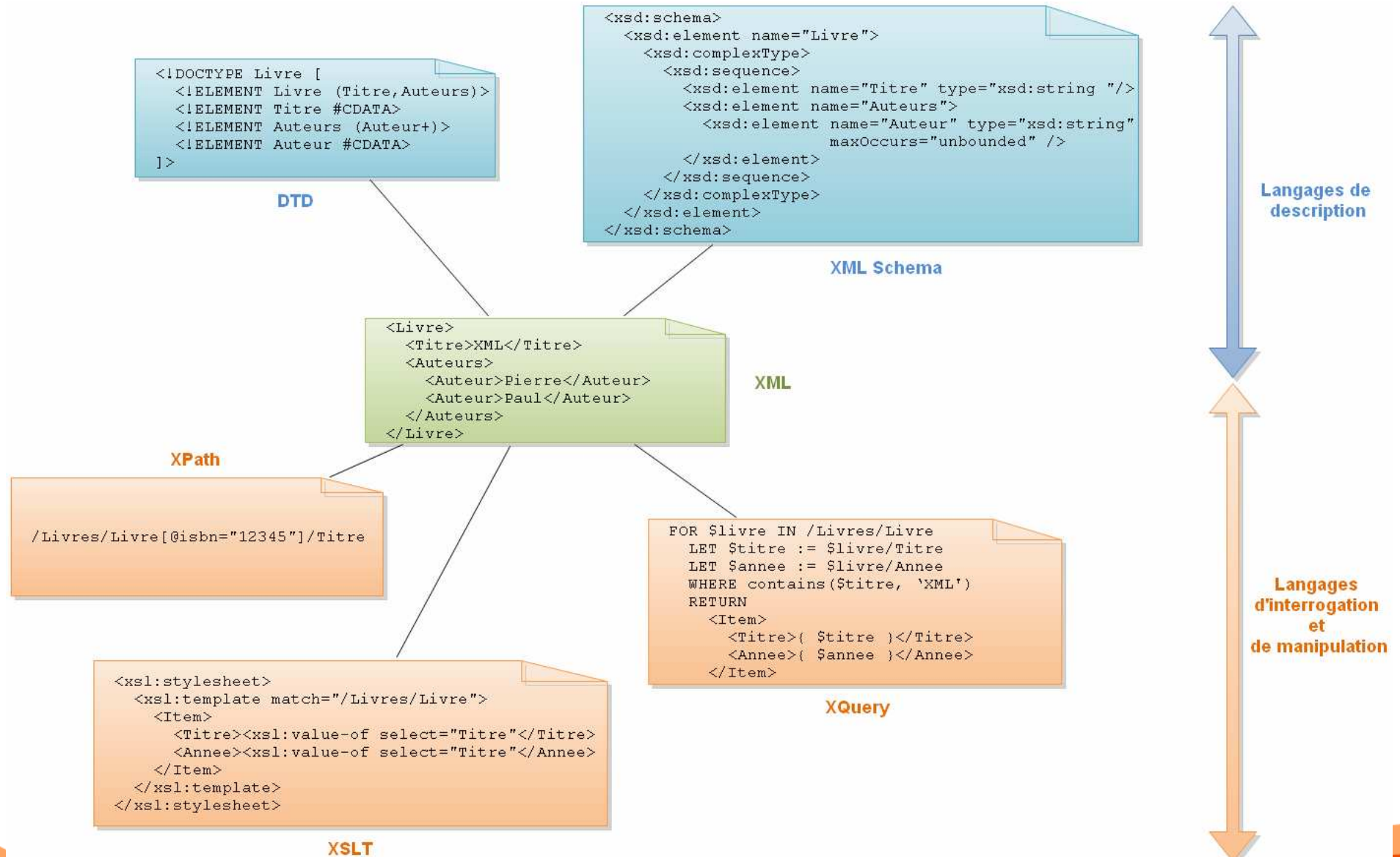
## **XML a des avantages**

- XML est structuré
- XML peut être décrit : DTD, XML Schema...
- XML peut être interrogé et manipulé : XPath, XSLT, XQuery...

## **Mais XML stocké dans des fichiers n'a pas les avantages des bases de données**

- Stockage efficace
- Indexation
- Sécurité
- Transactions
- Intégrité
- Accès multi-utilisateurs
- Triggers

# La boîte à outils XML



# Quelques mots sur XQuery...

## Définition



### Recommandation W3C

- 1998 à 2007
- Issue de nombreux travaux de spécialistes de SQL (notamment...)

### Objectifs

- Langage de requêtage sur un ou plusieurs documents
- Sélection de documents entiers ou de sous-arbres
  - Par structure ou par contenu

### Langage fonctionnel nouveau... et méconnu

- Expressions de chemins XPath
  - Recommandations désormais très liées
- Expressions FLWR
  - For, Let, Where, Return
  - Comparable au Select-From-Where de SQL
- Expressions conditionnelles
- Fonctions

# Quelques mots sur XQuery...

## Exemples (1/4)



### Exemple : FLWR

- Livres dont le titre contient « XML » :

```
FOR $livre IN /Livres/Livre
LET $titre := $livre/Titre
LET $annee := $livre/Annee
WHERE contains($titre, 'XML')
RETURN
    <Livre>
        <Titre>{ $titre }</Titre>
        <Annee>{ $annee }</Annee>
    </Livre>
```

# Quelques mots sur XQuery...

## Exemples (2/4)



### Exemple : Quantificateurs

- Livres écrits *entre autres* par des « Pierre » :

```
FOR $livre IN /Livres/Livre
WHERE SOME $auteur IN $livre/Auteurs/Auteur
      SATISFIES contains($auteur, "Pierre")
RETURN $livre/Titre
```

- Livres écrits *uniquement* par des « Pierre » :

```
FOR $livre IN /Livres/Livre
WHERE EVERY $auteur IN $livre/Auteurs/Auteur
      SATISFIES contains($auteur, "Pierre")
RETURN $livre/Titre
```

# Quelques mots sur XQuery...

## Exemples (3/4)



### Exemple : Fonction

- Nombre de livres écrits *entre autres* par « Paul » :

```
define function countLivres($auteur as xs:string)
as xs:integer
{
    FOR $livre IN /Livres/Livre
    LET $count = count($livre)
    WHERE SOME $auteur IN $livre/Auteurs/Auteur
        SATISFIES $auteur = "Paul"
    RETURN $count
}
countLivres(document("livres.xml"))
```

# Quelques mots sur XQuery...

## Exemples (4/4)



### Exemple : Jointure

- Jointure entre les documents des livres et des éditeurs :

```
FOR $livre IN document("livres.xml")//Livre,  
  $editeur IN  
  document("editeurs.xml")//Editeur[nom=$livre/Editeur]  
RETURN  
  {$livre/Titre, $editeur/Nom, $editeur/Adresse}
```

# Agenda



1 : Introduction

**2 : XML et les bases de données relationnelles**

3 : XML et les bases de données XML natives

4 : Techniques mixtes

5 : Conclusion

# Bases de données relationnelles

## Introduction



Référence de stockage et d'interrogation de données depuis les années 1980

### Intégration de XML dès 2000

- Initialement : simples stockage/récupération de documents entiers ou de fragments
  - En tant que données XML : CLOB
  - En tant que données relationnelles : mise en lambeau sur plusieurs tables et colonnes
- Puis : définition d'un type natif XML et de fonctions associées
  - Transformation XML  $\leftrightarrow$  données relationnelles
  - Extensions d'interrogation : XPath, XQuery et SQL/XML
  - *XML-Enabled databases*

### Bonne solution mais conceptuellement inadaptée

- Modèle interne non XML !
- Document XML : arborescence pas toujours aussi structurée qu'une table relationnelle
- Mapping complexe en définition et requêtage, et coûteux en performance

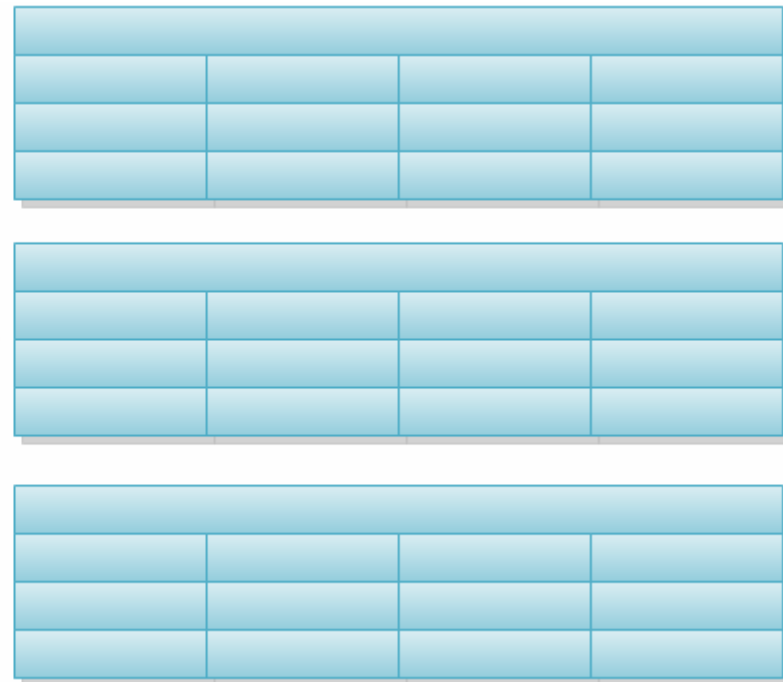
# Mapping XML / Relationnel

## Le problème

### Mapping tables / XML

- Comment passer du XML aux schéma relationnel et inversement ?

```
<Orders>
  <Order>
    <Number>123</Number>
    <Date>2008-06-17</Date>
    <Items>
      <Item>
        <Number>1</Number>
        <Quantity>10</Quantity>
        <Price>15.00</Price>
      </Item>
      <Item>
        <Number>2</Number>
        <Quantity>15</Quantity>
        <Price>7.50</Price>
      </Item>
    </Items>
  </Order>
</Orders>
```





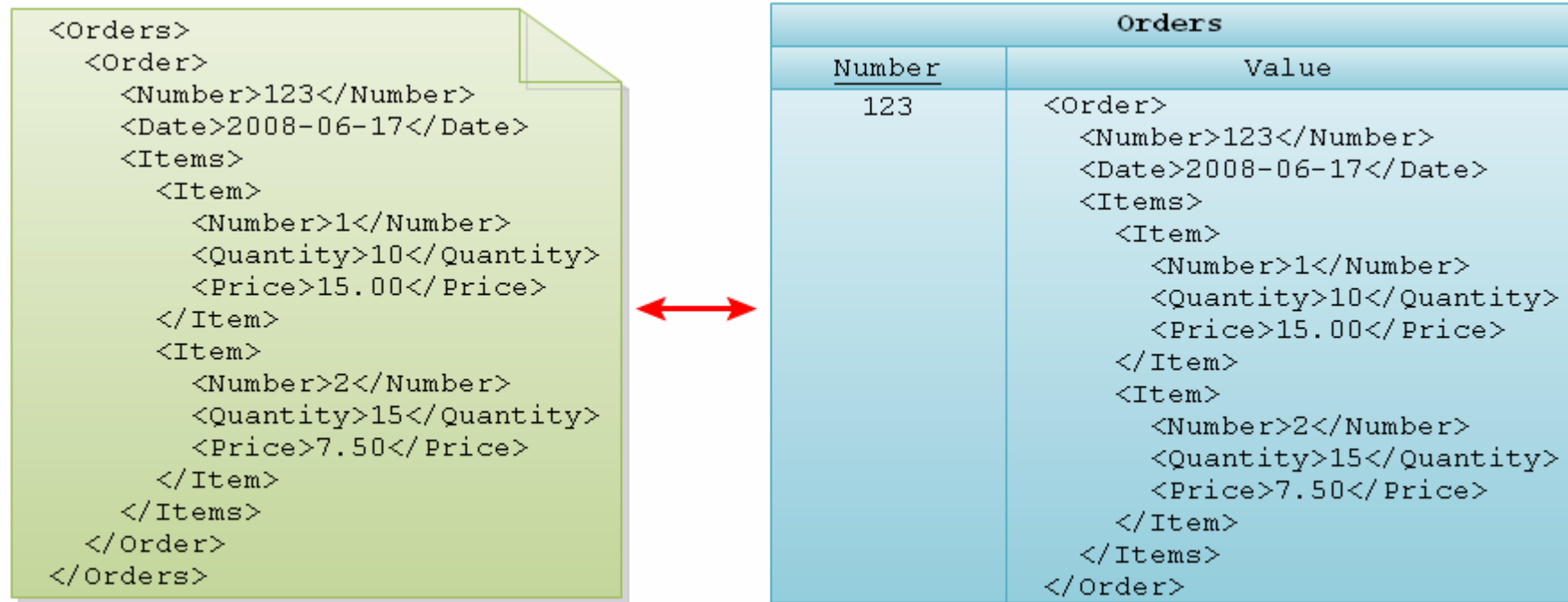

Plusieurs approches possibles

# Mapping XML / Relationnel

## Approche par documents

### Principe général

- Document ⇔ Colonne



### Bilan

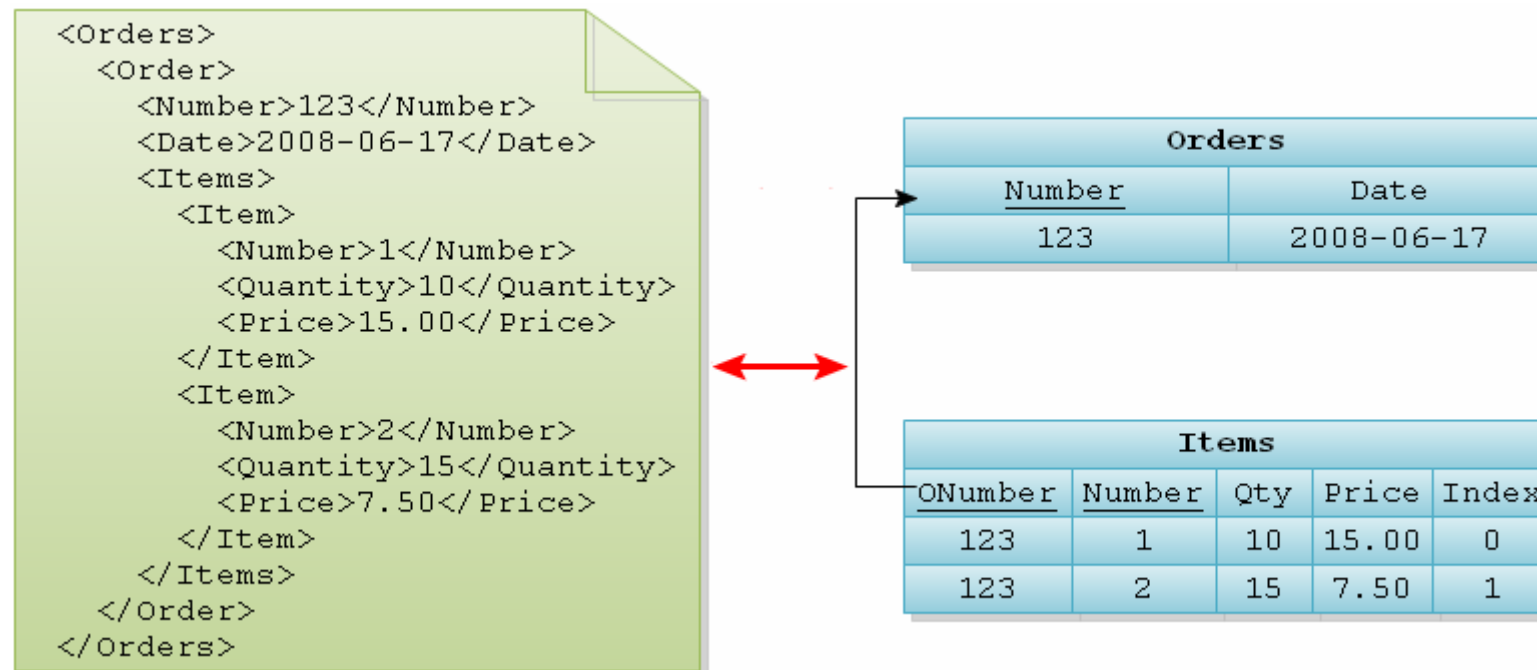
- Avantages : documents stockés sans distorsion, pas de traduction
- Inconvénient : requêtes peu exploitables et pas performantes

# Mapping XML / Relationnel

## Approche par éléments (1/2)

### Principe général

- Élément complexe  $\Leftrightarrow$  Table
- Élément simple ou attribut  $\Leftrightarrow$  Colonne





# Mapping XML / Relationnel

## Approche par éléments (2/2)

### Bilan

- Avantages
  - Document XML ramené à une structure connue
  - Principes proches de l'ORM : outils exploitables
- Inconvénients
  - Traduction schéma relationnel  $\Leftrightarrow$  DTD / Schéma XML
  - Document initial éclaté donc à reconstituer
  - Requêtes SQL complexes pour naviguer dans la hiérarchie

# Mapping XML / Relationnel

## Approche mixte

### Principe général

- Mélange des deux approches précédentes

```
<Orders>
  <Order>
    <Number>123</Number>
    <Date>2008-06-17</Date>
    <Items>
      <Item>
        <Number>1</Number>
        <Quantity>10</Quantity>
        <Price>15.00</Price>
      </Item>
      <Item>
        <Number>2</Number>
        <Quantity>15</Quantity>
        <Price>7.50</Price>
      </Item>
    </Items>
  </Order>
</Orders>
```



Orders		
Number	Date	Items
123	2008-06-17	<Items> <Item> <Number>1</Number> <Quantity>10</Quantity> <Price>15.00</Price> </Item> <Item> <Number>2</Number> <Quantity>15</Quantity> <Price>7.50</Price> </Item> </Items>

### Bilan

- Compromis entre les deux approches précédentes

# Mapping XML / Relationnel

## Approche générique

### Principe général

- Modèle de données générique

```
<Orders>
  <Order>
    <Number>123</Number>
    <Date>2008-06-17</Date>
    <Items>
      <Item>
        <Number>1</Number>
        <Quantity>10</Quantity>
        <Price>15.00</Price>
      </Item>
      <Item>
        <Number>2</Number>
        <Quantity>15</Quantity>
        <Price>7.50</Price>
      </Item>
    </Items>
  </Order>
</Orders>
```



Orders				
<u>Id</u>	ParentId	Element	Value	Index
1		Order		
2	1	Number	123	0
3	1	Date	2008-06-17	1
4	1	Items		2
5	4	Item		0
6	5	Number	1	0
7	5	Quantity	10	1
8	5	Price	15	2
9	4	Item		1
10	9	Number	2	0
11	9	Quantity	15	1
12	9	Price	7.50	2

### Bilan

- Complexe à exploiter

# Mapping XML / Relationnel

## Bilan



### Approches très différentes

- Points communs
  - Traduction coûteuse en développement... et en performance !
  - Exploitation du schéma relationnel pas évidente
- Comment choisir ?

### Plusieurs critères de sélection

- Nature de la structure des documents
  - Simple ou complexe ?
  - Documents orientés Données ou Présentation ?
- Stabilité de la structure des documents
  - Présence d'une DTD ou d'un Schéma XML ?
- Besoins du système
  - Lecture et/ou écriture ?

# Bases de données relationnelles SQL/XML



## Extensions au SQL pour manipuler des données XML stockées en base de données relationnelles

- Définition de fonctions de création de contenu XML à partir de données en base

```
select xmlelement(name "Customer",
  xmlelement(name "CustId", c.CustId),
  xmlelement(name "CustName", c.Name)
  xmlelement(name "City", c.City))
from Customers c
```

```
<Customer>
  <CustId>1</CustId>
  <CustName>Woodworks</CustName>
  <City>Baltimore</City>
</Customer>
```

- Définition d'un type de données XML : XMLType
  - Représente les données construites par les fonctions précédentes

# Bases de données relationnelles

## Etat de l'art



### IBM DB2

- Fonctions SQL/XML et outils de mapping programmatiques
  - XML Extender
- Implémentation XQuery

### Oracle

- Fonctions SQL/XML
  - XML DB
- Outils de mapping programmatiques
  - XML-SQL Utility
- Implémentation XQuery

### Microsoft SQL Server

- Fonctions SQL/XML
- Kit de développement SQLXML
- Implémentation XQuery

# Agenda



1 : Introduction

2 : XML et les bases de données relationnelles

**3 : XML et les bases de données XML natives**

4 : Techniques mixtes

5 : Conclusion

# Bases de données XML natives

## Définition



### Origine

- 2001 : Software AG pour son système Tamino
- Terme apparu suite à une campagne de publicité : pas de consensus sur la définition

### Définition pratique

- Bases de données
  - Fonctionnalités similaires aux autres bases de données
  - Langages d'interrogation et de manipulation, transactions...
- Créées spécifiquement pour gérer des données XML
  - Modèle logique interne basé sur les standards XML
  - Modèle physique sous-jacent non imposé

# Bases de données XML natives

## Structure générale

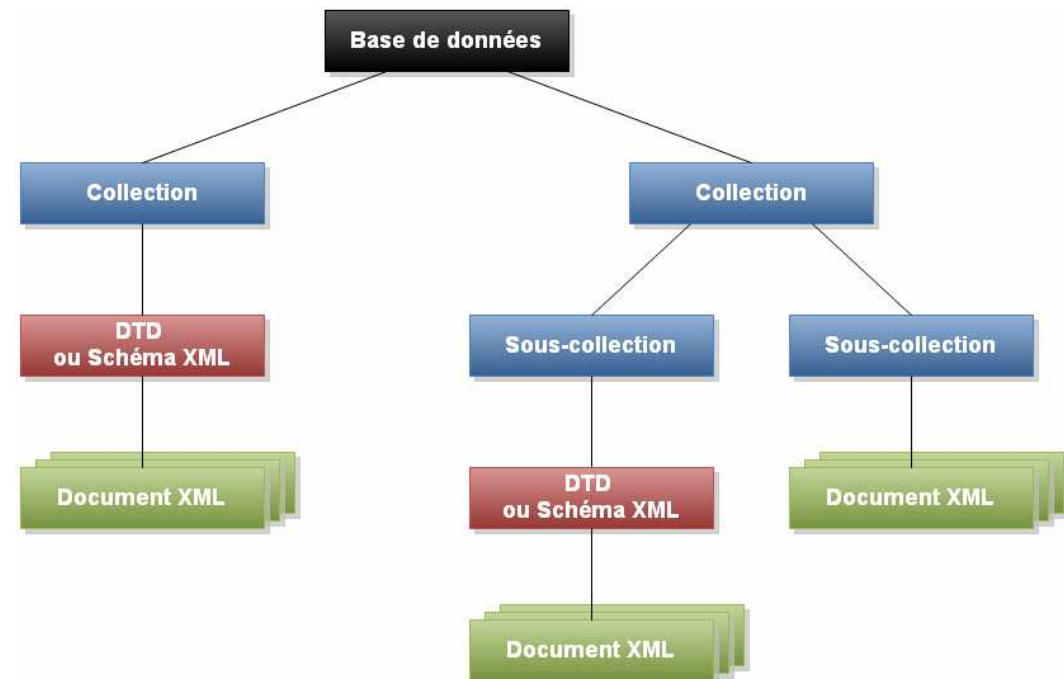


### Unité fondamentale de stockage : document XML

- « Equivalent » à 1 tuple dans une table d'un SGBDR
- Stockage sans distorsion (*round-tripping*)

### Documents regroupés dans des collections et des sous-collections

- « Equivalent » à 1 table dans un SGBDR
- Indépendantes de la structure des documents :  
**pas d'obligation d'y associer une DTD ou un schéma XML !**



# Bases de données XML natives

## Mise en œuvre



### Simplicité de mise en œuvre

- **Aucun mapping nécessaire**



- Il suffit de charger des répertoires de fichiers XML dans la base
- XQuery permet de spécifier la requête et la structure du résultat

### Techniques d'indexation

- But : optimiser les requêtes XPath
- Plusieurs types
  - Structure : porte sur les relations entre les éléments
  - Contenu : porte sur les valeurs des attributs et des nœuds de type texte
  - Full-text : porte sur tout le contenu des documents

# Indexation structurelle

## Principes



### Mise en place d'un plan de numérotation des nœuds pour :

- Décision : deux nœuds donnés ont-ils une relation ?
  - Ancêtre/descendant, parent/enfant, frère/frère
- Reconstruction : quels sont les voisins d'un nœud donné ?
  - Ancêtres, parent, frères, enfants, descendants

### Création d'un modèle reposant sur cette numérotation

```
<Livre>
  <Titre>XML</Titre>
  <Auteurs>
    <Auteur>Pierre</Auteur>
    <Auteur>Paul</Auteur>
  </Auteurs>
  <Editeur>Jacques</Editeur>
  <Annee>2008</Annee>
</Livre>
```



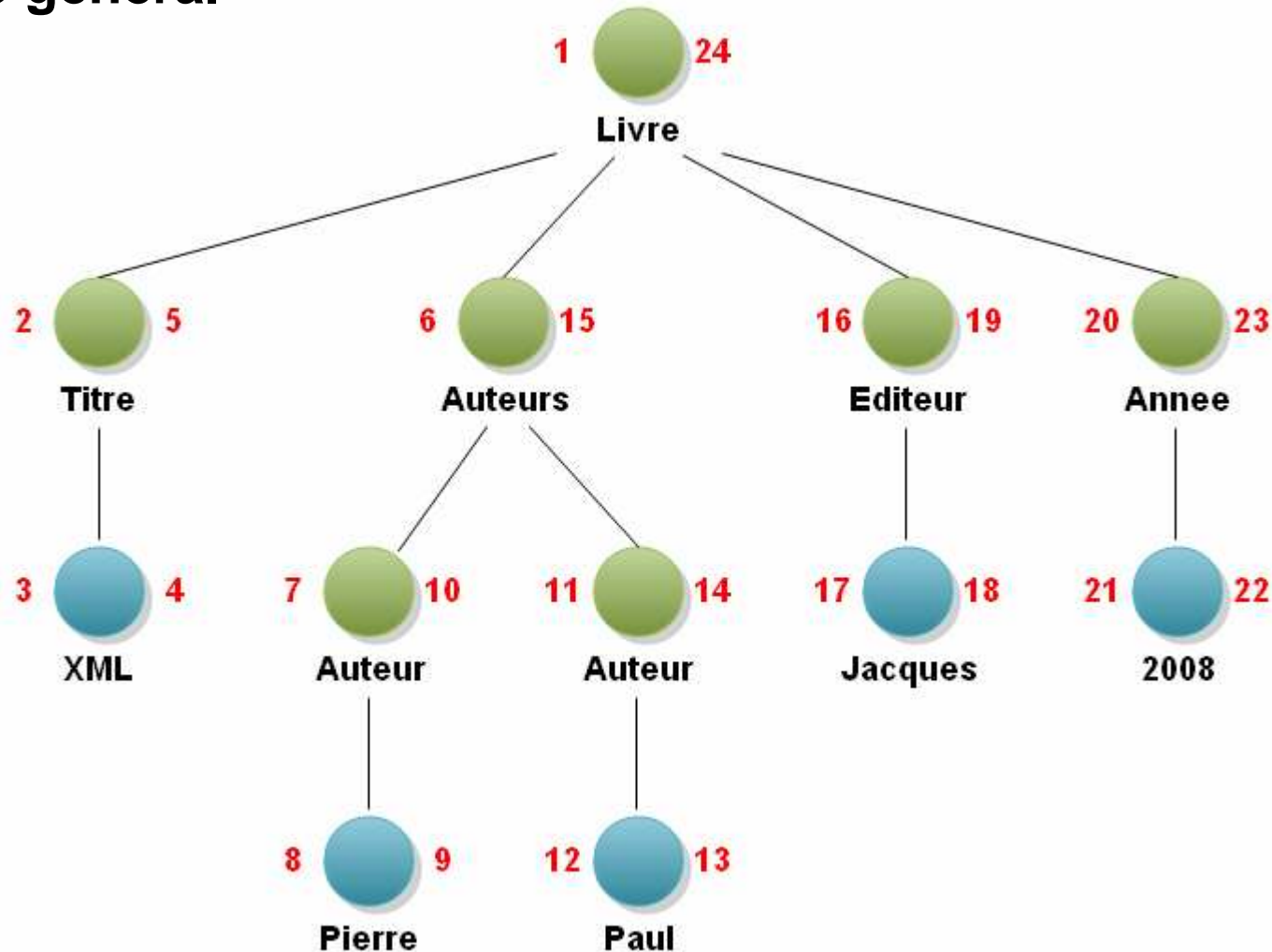
NodeName	DocumentId	NodeIds
Livre	1	?
Titre	1	?
Auteurs	1	?
Auteur	1	?, ?
Editeur	1	?
Annee	1	?

# Indexation structurelle

## Numérotation par intervalles (1/2)



### Principe général





# Indexation structurelle

## Numérotation par intervalles (2/2)

### **Avantage : relations entre les nœuds faciles à déterminer**

- Temps constant grâce aux intervalles
- Plusieurs variantes
  - Basées également sur des parcours en profondeur
  - Ex : numérotation de Dietz, numérotation de Zhang...

### **Inconvénient : réindexation nécessaire en cas de mise à jour**

- Pas idéal pour une base de données !
- Parade : laisser des « trous » dans la numérotation
  - Ex : numérotation XISS
  - Ne fait « que » retarder les réindexations, de toute façon nécessaires quand il n'y a plus d'identifiant libre

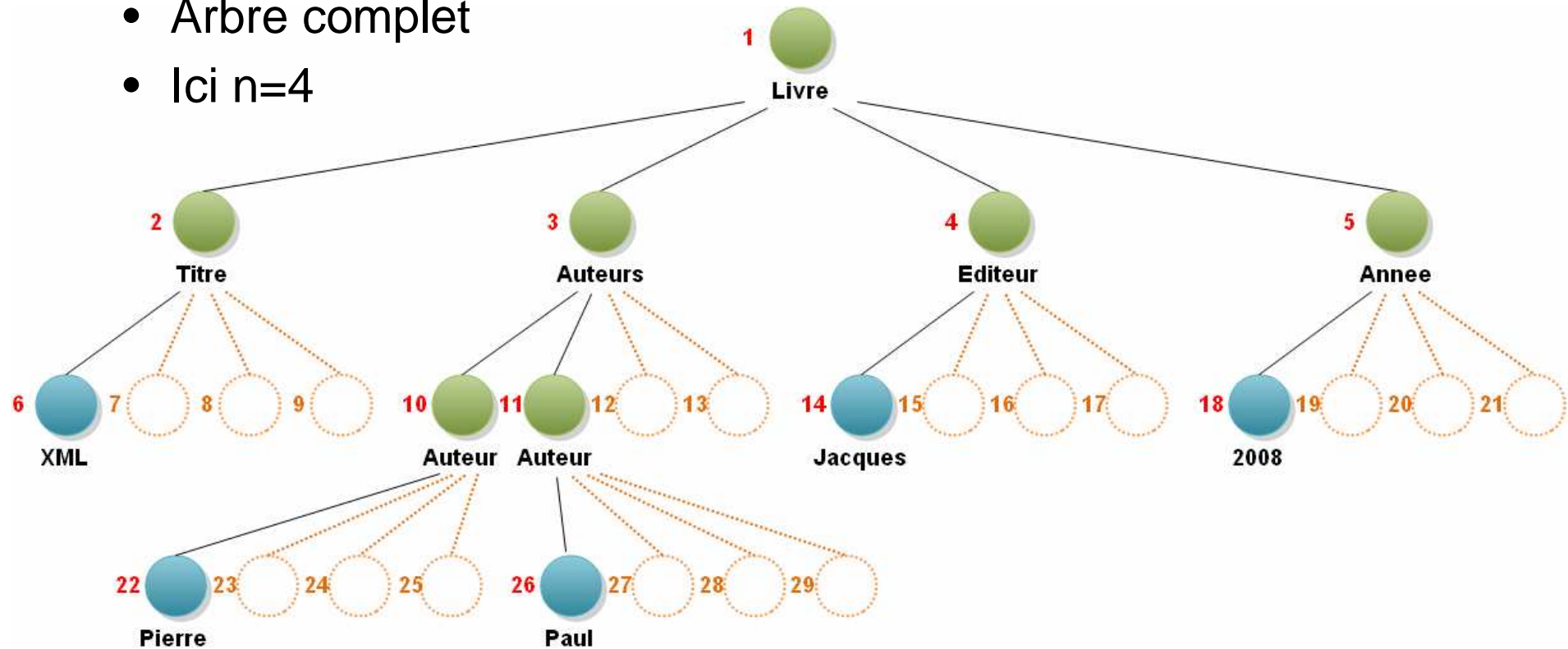
# Indexation structurelle

## Numérotation en arbre n-aire (1/2)



### Principe général

- Arbre complet
- Ici  $n=4$





# Indexation structurelle

## Numérotation en arbre n-aire (2/2)

**Avantage : relations entre les nœuds faciles à déterminer**

- $\text{Parent}(i) = (i-2)/n + 1$
- $\text{Fils}(i,j) = n(i-1) + j + 1$

**Inconvénient : Identificateurs trop clairsemés**

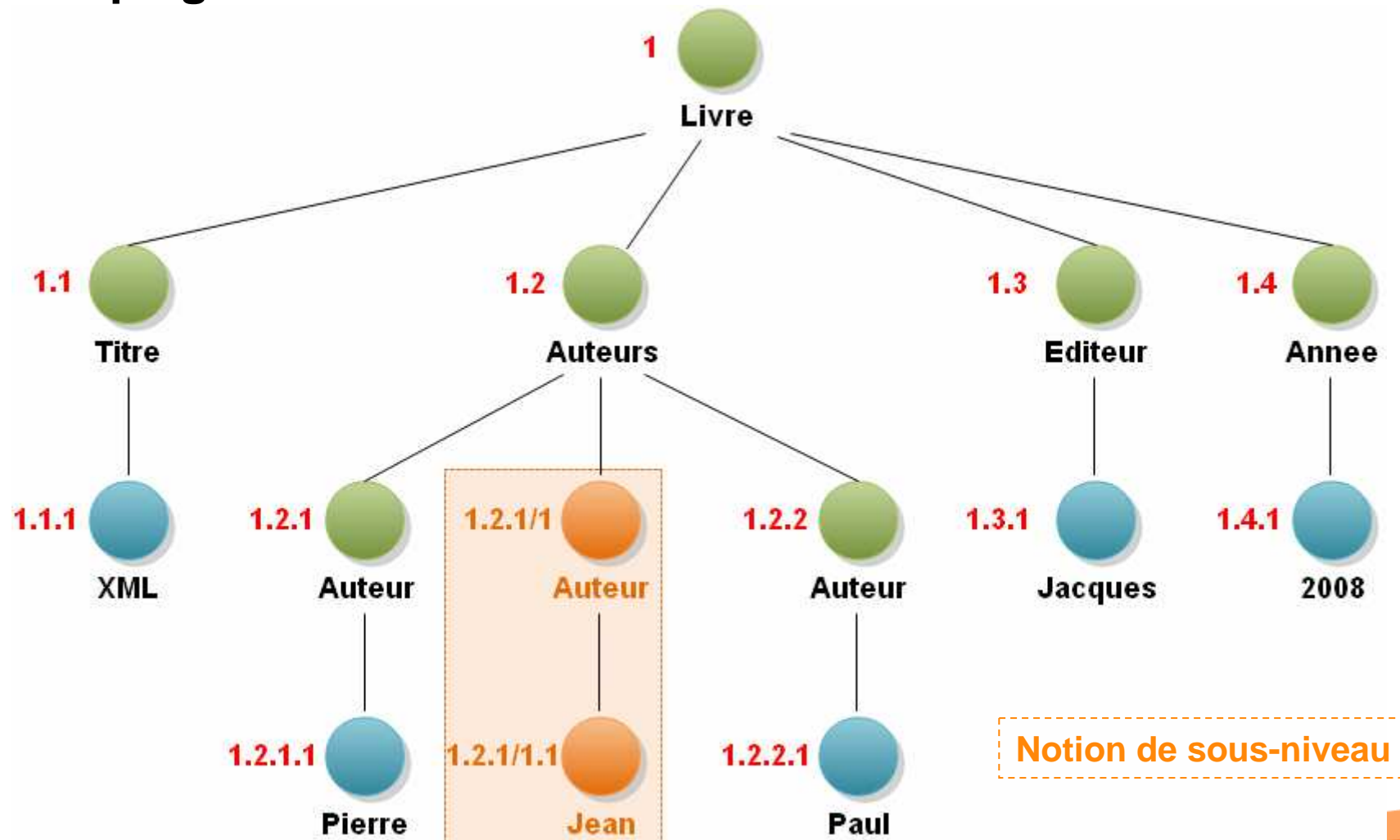
- Dû à la contrainte de complétude
- Documents rarement aussi équilibrés donc perte de place : limite la taille des documents indexables
- Parade : complétude spécifique à chaque niveau et non plus globale à l'arbre (un « n » par niveau)
  - Ex : numérotation virtuelle par niveau
  - Moins de nœuds virtuels mais réindexation toujours nécessaire en cas de mise à jour

# Indexation structurelle

## Numérotation dynamique par niveaux (1/2)



### Principe général





# Indexation structurelle

## Numérotation dynamique par niveaux (2/2)

### Avantages :

- Relations entre les nœuds facile à déterminer
  - Opération triviale
- Plus de nœud virtuel et de « trou » dans la numérotation
  - Gain de place
- Réindexation plus nécessaire en cas de mise à jour
  - Grâce à la notion de sous-niveau
  - Réindexation effectuée de temps en temps pour optimiser les calculs
  - Ex : 1.1/1 puis 1.1/0/1 puis 1.1/0/0/1 puis 1.1/0/0/0/1...

### Inconvénient :

- Problèmes liés à la structure des identifiants
  - Taille : encodage nécessaire pour les raccourcir
  - Type : travail sur les chaînes de caractères coûteux

# Bases de données XML natives

## Autres avantages



### **Gestion native de l'historique des modifications des documents**

- Attention aux changements de structure !
- Données de l'historique modifiables mais attention à l'intégrité

### **Gestion native des versions des documents**

- Travail collaboratif
- Verrous

# Bases de données XML natives

## Retour sur XQuery...



### Recherche full-text

- Aujourd'hui : implémentée de manière propriétaire par certaines bases de données
- Standardisée depuis peu : mai 2008
  - XQuery & XPath Full Text  
<http://www.w3.org/TR/2008/CR-xpath-full-text-10-20080516/>

### Mise à jour des données

- Aujourd'hui : extensions à XQuery implémentées de manière propriétaire par certaines bases de données
  - Ex : XUpdate
- Standardisée depuis peu : mars 2008
  - XQuery Update Facility  
<http://www.w3.org/TR/xquery-update-10/>

# Bases de données XML natives

## Etat de l'art



### Software AG Tamino

- <http://www.softwareag.com/Corporate/products/wm/tamino/default.asp>
- Mature mais abandonné

### Mark Logic

- <http://www.marklogic.com/>
- « Oracle des bases de données XML natives »
  - Volumétrie supportée, fiabilité...
- Commercial
  - Licence coûteuse

### eXist

- <http://exist.sourceforge.net/>
- Référence des bases de données XML natives Open Source
- En constante amélioration
  - Comparé à MySQL

### Berkeley DB XML

- <http://www.oracle.com/database/berkeley-db/xml/index.html>
- Projet Open Source développé par Sleepycat
- Repris par Oracle
  - Fiable et mature
- Conception différente des autres
  - Library embarquée à utiliser en C, Java, PHP...

# Agenda



1 : Introduction

2 : XML et les bases de données relationnelles

3 : XML et les bases de données XML natives

**4 : Techniques mixtes**

5 : Conclusion

# Techniques mixtes

## Introduction

### Principe

- Choix entre SGBD et XND guidé par l'existant
- Comment intégrer XQuery et les bases XML natives dans le parc rempli de bases relationnelles ?

### XQuery comme langage universel d'interrogation de données hétérogènes

- Outils de médiation
  - Libres : Pathfinder (traducteur XQuery -> SQL)
  - Commerciaux : DataDirect (traducteur XQuery -> \*)
- Attention, mise à jour problématique
  - XUpdate bien géré que sur les bases relationnelles
- Cas particulier d'architecture d'intégration orientée service
  - Entreprise Service Bus (ESB)

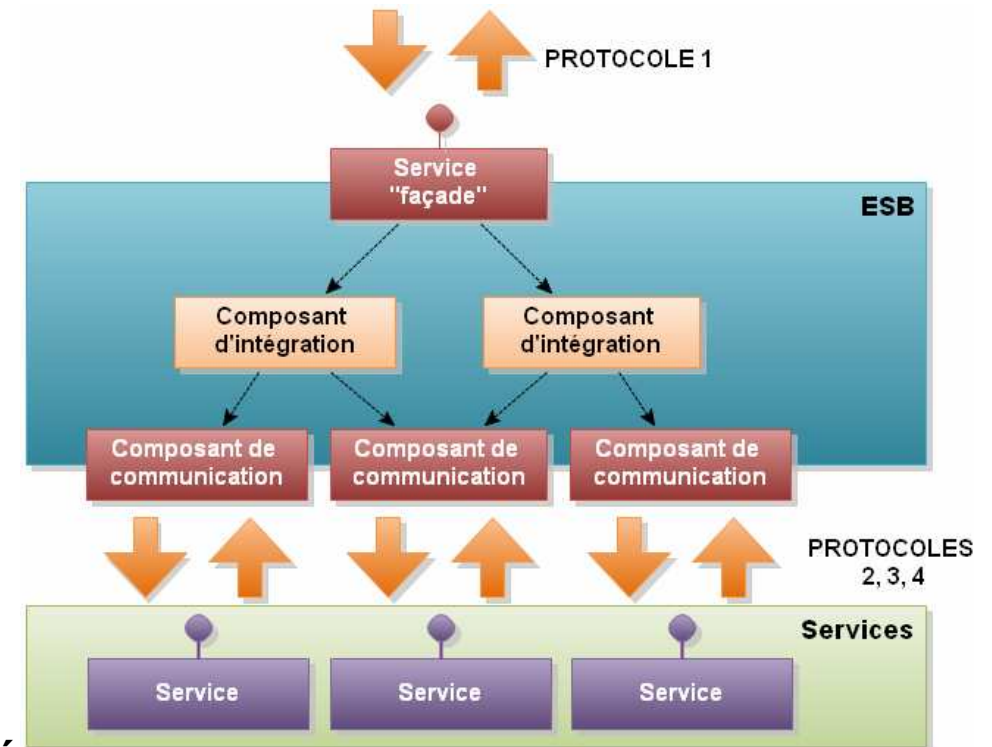
# Un (tout petit) peu de SOA...

## Notion d'ESB



ESB : bus logiciel de médiation

- **Met en relation les services du SI et leurs consommateurs**
  - « Orienté service »
  - Indépendant des implémentations, protocoles et localisations des services
  - Par assemblage de composants d'intégration et de communication
- **Basé sur les standards du Web**
  - HTTP, SOAP, SMTP...
- **Un des deux piliers des SOA**
  - Autre pilier : Business Process Modeling
- **Applications variées**
  - Transformation de protocoles
  - Aggrégation de services
  - Encapsulation de services
  - Gestion de versions de services



ESB = Tendence XML-isée de l'EAI

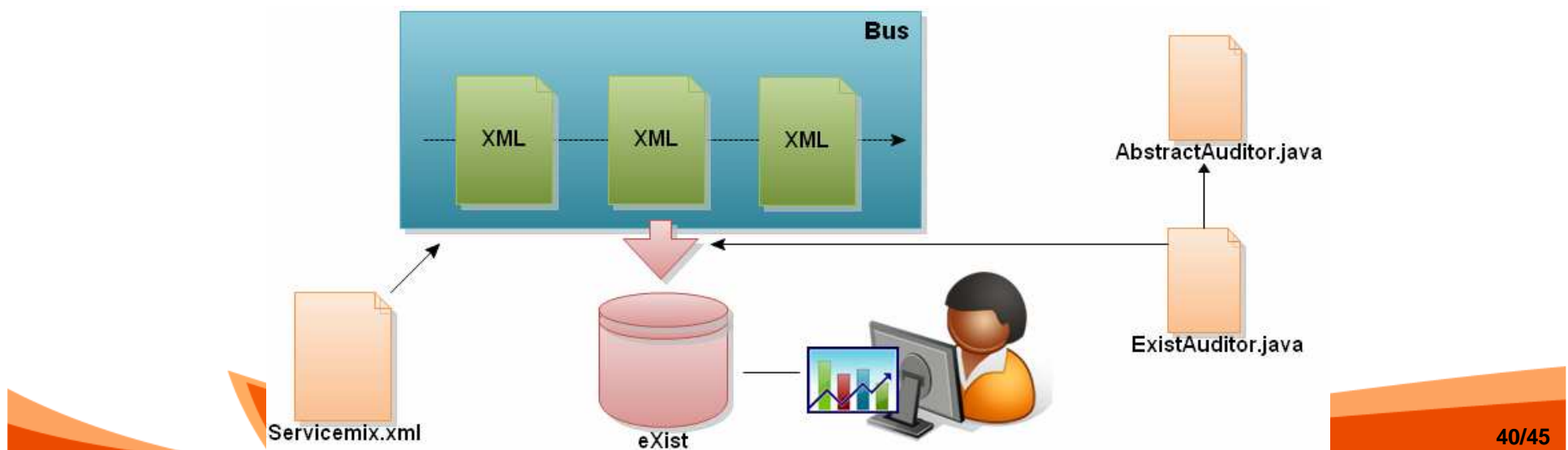
- **Construits sur XML et Web Services**
- **XQuery devient un candidat privilégié**
  - Pour spécifier les interfaces des services et interroger des données hétérogènes masquées
  - Ex : Web Service qui prend du XQuery en entrée et réalisé par du SQL

# Techniques mixtes

## Cas d'utilisation n°1

### Supervision des flux circulant dans un ESB JBI

- Monitoring = point essentiel
- JBI : documents qui circulent sont tous au format XML
- Documents sans structure commune : dépendent des différents services du bus
- Implémentation
  - Monitoring d'un bus JBI Apache Servicemix avec la base eXist



# Techniques mixtes

## Cas d'utilisation n°2

### Moteur de recherche d'un site Web

- Contexte : plusieurs centaines de pages HTML statiques et de fichiers PDF
- Implémentation
  - Stockage des documents dans une base eXist
- Trois étapes
  - Transformation des documents HTML et PDF en XML
  - Structuration des documents en fonction de la ponctuation : éléments `<phrase>`
  - Chargement dans la base et indexation

# Techniques mixtes

## Autres cas d'utilisation

### Annuaire de services

- Standard UDDI très complexe
  - Peu adapté à la description des caractéristiques du service, souvent complexes
- Format XML plus adapté
  - Solution naturelle : intégration d'une base de données XML native

### Gestionnaire de cache

- Optimisation d'un Web Service au format SOAP (XML)
- Attention au réel intérêt
  - Utile si appels redondants
  - Prendre en compte le temps de recherche en base !
- Une base embarquée permet d'éviter les appels réseaux

Voir <http://www.ibm.com/developerworks/xml/library/x-accsoa/index.html>

# Agenda



1 : Introduction

2 : XML et les bases de données relationnelles

3 : XML et les bases de données XML natives

4 : Techniques mixtes

**5 : Conclusion**

# Conclusion



## Bases de données relationnelles

- Structure des données régulière
- Données utilisées par des applications non XML
- XML comme format de transport
- Documents orientés données (*data-centric*)
  - Ordres de ventes, dossiers de patients, données scientifiques...

## Bases de données XML natives

- Structure des données moins régulière : semi-structurée (peu ou pas de schéma)
- Données utilisables par des humains
- XML comme format de représentation
- Documents orientés présentation (*document-centric*)
  - Pages Web statiques, manuels utilisateurs, brochures...

## Techniques mixtes

- Dans la lignée des approches SOA
- Bon compromis entre les deux approches vis-à-vis de l'existant



# *Gestion de documents XML dans une base de données*

---

**Questions**

