

Bases de Données et

Didier DONSEZ

Université Joseph Fourier (Grenoble 1)

IMA – IMAG/LSR/ADELE

`Didier.Donsez@imag.fr`

XML et Bases de Données

■ Plusieurs aspects

- Extraction XML d'une base de données
- Fédération des Bases de Données Hétérogènes
- Bases de Données Semi-Structurés

Extraction XML d'une base de données

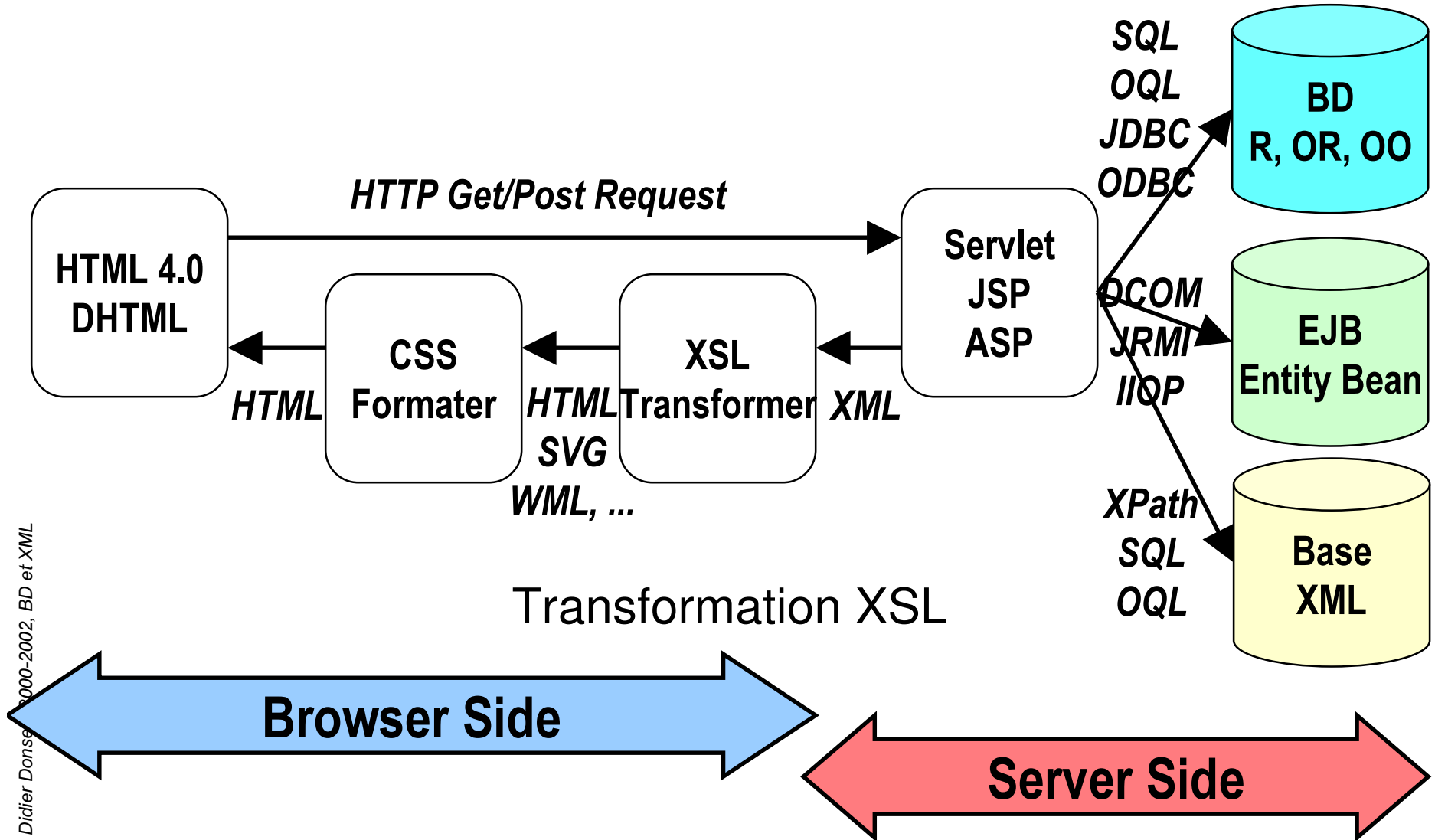
■ La BD vue comme une source XML

- Document « Plat »
 - Requête sur une BD Relationnelle
- Document « plus profond »
 - Requête sur une BD Objet-Relationnelle
 - SQL-99: colonne de type ADT ou Multiset d'ADT
 - ODMG' OQL
 - Association Many-To-Many
- Exemple : XSQLServlet

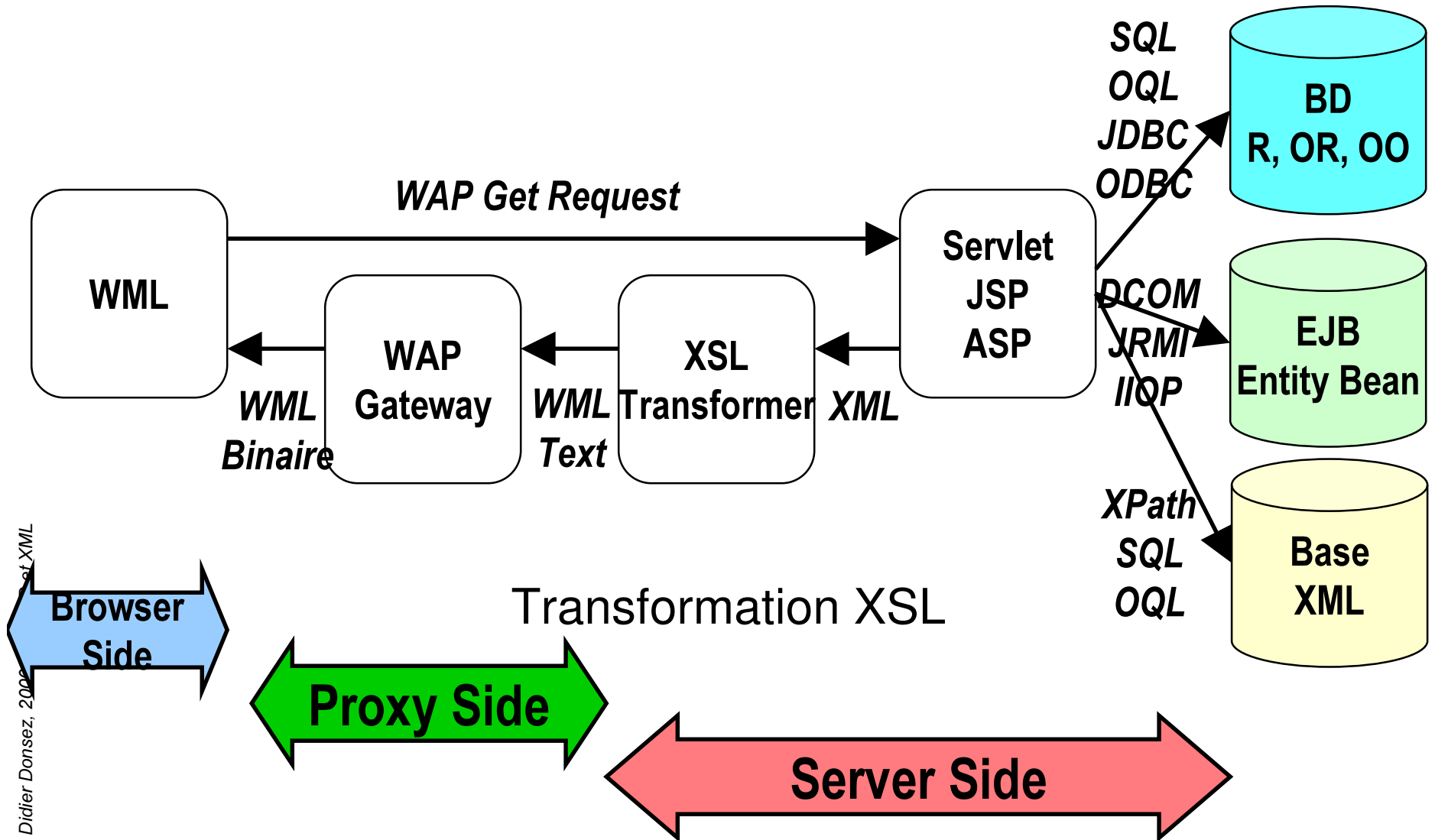
■ Mise à jour

- Génération automatique de Xform
à partir du XML Schema (extrait de la métabase MetaData)

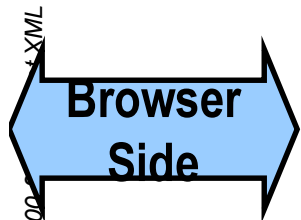
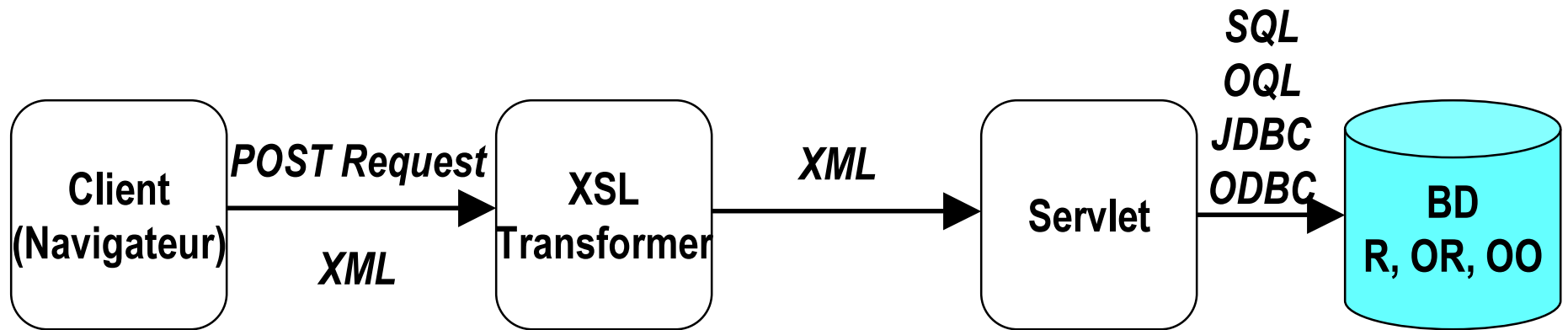
XML et Bases de Données



XML et Bases de Données



XML et Bases de Données



XML et Bases de Données

- Insertion et Extraction de documents XML d'une base de données

- JDBC
 - classe `sun.jdbc.rowset.WebRowSet`
- Outils
 - Oracle XSQL Servlet et OracleJSPTagLib
 - Expresso XML
 - <http://freshmeat.com/project/expressoframework/>
 - dbXML
 - www.dbxml.org
 - Cocoon
 - xml.apache.org

Oracle XSQLServlet

■ Servlet

- parse des documents .xsql (XML) contenant des requêtes nichés dans des éléments `xsql:query`, `xsql:insert`, ...
- basé sur JDBC (indépendance au SGBD sauf O8i SQL99)
- transformation XSL (FOP) coté Serveur

■ `xsql:query`

- Interroge la base de données (select)
- Retourne un document résultat

■ `xsql:insert`, `xsql:update`, `xsql:delete`

- Mise à jour de la base de données à partir des paramètres ou d'un document XML posté

Oracle XSQLServlet

Fonctionnement de xsql:query

- replace les éléments xsql:query, ...
par une « table » XML générée par l'exécution de la requête
- Applique ensuite les transformations xml-stylesheet
 - transformation en HTML, XML, WML avec XSL
 - transformation en PDF avec FOP
- Sources de données décrites
dans le fichier de configuration de la servlet

Oracle XSQLServlet Exemple

```
<?xml version="1.0"?>
<sales-by-year xmlns:xsql="urn:oracle-xsql" connection="salesdb">
  <period id="H1" year="CY99">
    <xsql:query>
      SELECT salesperson, SUM(sales) AS Total FROM sales
      WHERE sale_date between '01-JAN-99' and '30-JUN-99' GROUP BY salesperson
    </xsql:query>
  </period>
  <period id="H2" year="CY99">
    <xsql:query>
      SELECT salesperson, SUM(sales) AS Total FROM sales
      WHERE sale_date between '01-JUL-99' and '31-DEC-99' GROUP BY salesperson
    </xsql:query>
  </period>
</sales-by-year>
```

Oracle XSQLServlet

Exemple : résultat

```

<?xml version="1.0"?>
<sales-by-year connection="salesdb">
  <period id="H1" year="CY99">
    <ROWSET>
      <ROW id="1">
        <SALESPERSON>Steve</SALESPERSON>
        <TOTAL>23465500</TOTAL>
      </ROW>
      <ROW id="2">
        <SALESPERSON>Mark</SALESPERSON>
        <TOTAL>39983400</TOTAL>
      </ROW>
    </ROWSET>
  </period>

```

...

```

...
<period id="H2" year="CY99">
  <ROWSET>
    <ROW id="1">
      <SALESPERSON>Steve</SALESPERSON>
      <TOTAL>67788400</TOTAL>
    </ROW>
    <ROW id="2">
      <SALESPERSON>Mark</SALESPERSON>
      <TOTAL>55786990</TOTAL>
    </ROW>
  </ROWSET>
</period>
</sales-by-year>

```

Oracle XSQLServlet

Requêtes paramétrées

- Les paramètres des requêtes sont de la forme {@param}
- Les paramètres sont pris dans l'ordre dans
 - Paramètre HTTP
 - Variable de page `xsql:set-page-param`
 - Variable de session `HttpSession.getValue()` et `xsql:set-session-param`
 - Valeur de Cookie `xsql:set-cookie`
 - Paramètre par défaut

■ Exemple

```
<?xml version="1.0"?>
  <!-- http://localhost/xsql/emp.xsql?sort=NUMEMP -->
  <!-- http://localhost/xsql/emp.xsql?sort=SAL -->
  <!-- http://localhost/xsql/emp.xsql peut produit un erreur -->
  <xsql:query connection="employee" xmlns:xsql="urn:oracle-xsql">
    SELECT * FROM EMPLOYEE ORDER BY {@sort}
  </xsql:query>
```

Oracle XSQLServlet

Requêtes paramétrées

- La valeur par défaut d'un paramètre est spécifié sous la forme d'un attribut de `xsql:query`, ...

- Exemple

```
<?xml version="1.0"?>
```

```
<!-- http://localhost/xsql/emp.xsql?sort=NUMEMP -->
```

```
<!-- http://localhost/xsql/emp.xsql?sort=SAL -->
```

```
<!-- http://localhost/xsql/emp.xsql est équivalent à
```

```
http://localhost/xsql/emp.xsql?sort=ENAME -->
```

```
<xsql:query connection="employee" xmlns:xsql="urn:oracle-xsql"
```

```
sort="ENAME">
```

```
SELECT * FROM EMPLOYEE ORDER BY {@sort}
```

```
</xsql:query>
```

Oracle XSQLServlet

Liaison de paramètres

```
<?xml version="1.0"?>
<page connection="employee" xmlns:xsql="urn:oracle-xsql">
  <xsql:query bind-params="id name">
    SELECT empno, ename FROM employee
    WHERE empno = ? /* BindVar 1 */ or ename = ? /* BindVar 2 */
    UNION ALL
    SELECT empno, ename FROM former_employee
    WHERE empno = :1 /* BindVar 1 */ or ename = :2 /* BindVar 2 */
  </xsql:query>
</page>
```

Oracle XSQLServlet

Attributs supplémentaires

```

<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="emp.xsl"?>
  <!-- http://localhost/xsql/emp.xsql?find=DU&sort=SAL -->
  <!-- http://localhost/xsql/emp.xsql?sort=SAL -->
<xsql:query connection="employee" xmlns:xsql="urn:oracle-xsql"
  sort="ENAME" null-indicator="yes"
  rowset-element="employees" row-element="employee"
  tag-case="lower" include-schema="yes"
  skip-rows="10" max-rows="20">
  SELECT * FROM EMPLOYEE
  WHERE UPPER(ENAME) LIKE UPPER('%{@find}%')
  ORDER BY {@sort}
</xsql:query>

```

Oracle XSQLServlet

Transformation XSLT

- Une feuille XSLT est appliquée sur le résultat `<rowset>` en fonction du type de client

```
<?xml version="1.0"?>
```

```
<?xml-stylesheet type="text/xsl" media="wap" href="emp2wml.xsl" ?>
```

```
<?xml-stylesheet type="text/xsl" media="msie 5.0" client="yes" href="emp-ie5.xsl" ?>
```

```
<!-- La transformation n'est pas réalisé par le serveur -->
```

```
<?xml-stylesheet type="text/xsl" media="msie" href="emp2dhtml.xsl" ?>
```

```
<?xml-stylesheet type="text/xsl" href="emp2html40.xsl" ?> <!-- par défaut -->
```

```
<page connection="employee" xmlns:xsql="urn:oracle-xsql">
```

```
  <xsql:query>SELECT * FROM EMPLOYEE</xsql:query>
```

```
</page>
```

Remarque : la paire `xml-stylesheet=none` inhibe la transformation

<http://yourserver/yourdatapage.xsql?param1=value&xml-stylesheet=none>

Oracle XSQLServlet

Transformation XSLT et Sérialisateur

- Une feuille XSLT est appliquée sur le résultat `<rowset>`
- Puis elle est passée à un « sérialisateur » qui transforme la transformation vers un format non-XML
 - RTF, PDF, PS, GIF, ...
- Exemple avec FOP (PDF)

```
<?xml version="1.0"?>
```

```
<?xml-stylesheet type="text/xsl" href="emp2pdf.xsl" serializer="FOP" ?>
```

```
<page connection="employee" xmlns:xsql="urn:oracle-xsql">
```

```
  <xsql:query>SELECT * FROM EMPLOYEE</xsql:query>
```

```
</page>
```

Remarque : Les sérialiseurs sont des classes qui doivent être déclarées (par défaut : FOP pour PDF, ...)

Oracle XSQLServlet

UDT et Collection

- SQL3/99 (et les SGBDs Objet-Relationnels) définissent des colonnes
 - de types structurés (UDT : User Data Type)
 - des collections de valeurs ou d'UDT

- XSQLServlet retourne une hiérarchie d'éléments

- Voir Cours BD : « Les Objets dans SQL3 »

Oracle XSQLServlet

Exemple avec un UDT

```
CREATE TYPE POINT AS OBJECT (X NUMBER, Y NUMBER);  
CREATE TABLE LOCATION ( NAME VARCHAR2(80), ORIGIN POINT );  
INSERT INTO LOCATION VALUES ( 'Someplace', POINT(11,17) );
```

```
<!-- http://yourmachine.com/xsql/demo/point.xsql?x-coord=11 -->  
<xsql:query connection="demo" xmlns:xsql="urn:oracle-xsql">  
  SELECT name, origin FROM location loc WHERE loc.origin.x = {@x-coord}  
</xsql:query>
```

```
<ROWSET>  
  <ROW num="1">  
    <NAME>Someplace</NAME>  
    <ORIGIN>  
      <X>11</X>  
      <Y>17</Y>  
    </ORIGIN>  
  </ROW>  
</ROWSET>
```

Oracle XSQLServlet

Exemple avec une Collection

```
<xsql:query connection="demo" xmlns:xsql="urn:oracle-xsql">
  SELECT dname, CURSOR( SELECT ename,sal FROM emp
                        WHERE emp.deptno = dept.deptno) as employees
  FROM dept
</xsql:query>
```

```
<ROWSET>
  <ROW num="1">
    <DNAME>ACCOUNTING</DNAME>
    <EMPLOYEES>
      <EMPLOYEES_ROW num="1">
        <ENAME>CLARK</ENAME><SAL>2450</SAL>
      </EMPLOYEES_ROW>
      <EMPLOYEES_ROW num="2">
        <ENAME>KING</ENAME><SAL>5000</SAL>
      </EMPLOYEES_ROW>
    </EMPLOYEES>
  </ROW>
  ...
```

```
...
  <ROW num="2">
    <DNAME>RESEARCH</DNAME>
    <EMPLOYEES>
      <EMPLOYEES_ROW num="1">
        <ENAME>MILLER</ENAME><SAL>1300</SAL>
      </EMPLOYEES_ROW>
    </EMPLOYEES>
  </ROW>
</ROWSET>
```

Oracle XSQLServlet

Exemple avec une Collection

```
<xsql:query connection="demo" xmlns:xsql="urn:oracle-xsql" tags-case="lower">  
  SELECT ename, CURSOR( SELECT * FROM TABLE(employee.phones)) as tels  
  FROM Employee WHERE UPPER(ename) LIKE UPPER('%{@find}%')  
</xsql:query>
```

```
<rowset>  
  <row num="1">  
    <ename>DUPONT</ename>  
    <tels>  
      <tels_row num="1"><num>01234567</num><ext>2450</ext></tels_row>  
      <tels_row num="2"><num>0654321</num></tels_row>  
    </tels>  
  </row>  
  <row num="2">  
    <ename>DURANT</ename>  
    <tels>  
      <tels_row num="1"><num>03691215</num><ext>2450</ext></tels_row>  
    </tels>  
  </row>  
</rowset>
```

Oracle XSQLServlet

Autres éléments xsql

■ xsql:set-session-param

```
<xsql:set-session-param name="current_user" value="{@userid}"/>
<xsql:set-session-param name="current_user">
  SELECT member_id FROM member_session_info
  WHERE session_id = {@sessionCookie}
</xsql:set-session-param>
```

■ xsql:set-cookie

```
<xsql:set-cookie name="last_selection" value="{@choice}"/>
<xsql:set-cookie name="shopping_cart_id">
  SELECT cartmgr.new_cart_id(UPPER('{@current_user}')) FROM dual
</xsql:set-cookie>
```

■ xsql:include-xml

```
<xsql:include-xml href="http://stock.com/quotes?symbol={@symbols}"/>
<xsql:include-xml href="list-of-states.xml"/>
<xsql:include-xml href="{@pagename}"/>
```

Oracle XSQLServlet

Autres éléments xsql

■ xsql:set-page-param

```
<xsql:set-page-param name="max-rows-pref">  
  SELECT max_rows FROM user_profile WHERE userid = {@userid}  
</xsql:set-page-param>  
<xsql:query max-rows="{@max-rows-pref}">  
  SELECT title, url FROM newsstory ORDER BY date_entered DESC  
</xsql:query>
```

Oracle XSQLServlet

Les modifications

■ xsql:insert, xsql:update, xsql:delete

- Réalisent des Update/Delete/Insert à partir de document posté ou des paramètres
- Peut utiliser la transformation XSLT avant de traiter la modification

```
<!-- Update columns in dept table based on 'deptno' key. -->
```

```
<xsql:update-request table="dept" transform="doc-to-dept.xml"  
key-columns="deptno"/>
```

```
<!-- Update only DNAME column in dept table based on 'deptno' key. -->
```

```
<xsql:update-request table="dept" transform="doc-to-dept.xml"  
key-columns="deptno columns="dname"/>
```

```
<!-- Delete rows in dept table based on 'deptno' key. -->
```

```
<xsql:delete-request table="dept" transform="doc-to-dept.xml" key-columns="deptno"/>
```

```
<!-- Parse/transform contents of posted XML document or HTML Form for insert -->
```

```
<xsql:insert-request table="purchase_order" transform="purchseorder-to-rowset.xml"/>
```


Oracle XSQLServlet

Les modifications

- Insertion à partir des paramètres de la requête HTTP
 - Méthode POST et ContentType application/x-www-form-urlencoded

- Forme

```
<request>
  <parameters>
    <firstparamname>firstparamvalue</firstparamname>
    <lastparamname>lastparamvalue</lastparamname>
  </parameters>
  <session>
    <firstparamname>firstsessionparamvalue</firstparamname> :
    <lastparamname>lastsessionparamvalue</lastparamname>
  </session>
  <cookies>
    <firstcookie>firstcookievalue</firstcookiename>
    <lastcookie>firstcookievalue</lastcookiename>
  </cookies>
</request>
```

Oracle XSQLServlet

Les modifications

■ Remarque

- Quand une URL comporte des ensembles de paramètres
- `majemp.xsql?id=101&name=Alice&id=102&name=Bob&operation=update`
- La requête est « linéarisée » (row-ified)

```
<request>
```

```
<parameters>
```

```
<row><id>101</id> <name>Alice</name></row>
```

```
<row><id>102</id> <name>Bob</name></row>
```

```
<operation>update</operation>
```

```
</parameters>
```

```
</request>
```

Oracle XSQLServlet

Les modifications

- Insertion à partir d'un document `<rowset>` Posté
 - Méthode POST et ContentType text/xml
- Remarque
 - Le document posté peut être transformé en `<rowset>` par le serveur avec une feuille de style

Oracle XSQLServlet Invocations

- Par XSQLServlet URL .xsql
- Depuis une servlet
 - `XSQLRequest.process()`
- Depuis une JSP
 - `<jsp:include>` `<jsp:forward>`
- En ligne de commande
 - `oracle.xml.xsql.XSQLCommandLine`

Oracle XSQLServlet

Retour d'erreur

■ Retour d'un élément xsql-error

- Peut être transformée par le client (xml-stylesheet=none) ou le serveur

■ Exemple d'erreur retournée

```
<xsql-error action="xsql:query">
  <statement>SELECT * FROM EMP ORDER BY SALARY</statement>
  <message>ORDER BY column, SALARY, must be in the column selection</message>
</xsql-error>
```

■ Exemple de transformation de l'erreur

```
<xsl:if test="//xsql-error">
  <table style="background:yellow">
    <xsl:for-each select="//xsql-error">
      <tr><td><b>Action</b></td><td><xsl:value-of select="@action"/></td></tr>
      <tr valign="top"><td><b>Message</b></td><td><xsl:value-of select="message"/></td></tr>
      <tr valign="top"><td><b>Statement</b></td><td><xsl:value-of select="statement"/></td></tr>
    </xsl:for-each>
  </table>
</xsl:if>
```

JSP Tags and Beans

Important programming requirement:

Easy to write, maintain, and enhance

■ Many vendors provide JSP tags, JavaBeans

- Easy syntax
- Packaged robust solutions
- Modular programming components

■ Examples (based on **Oracle9iAS**)

- Oracle data access beans
- OracleJSP SQL tags

Oracle Data Access Beans (1)

- JavaBeans to access database and execute SQL
- Can be used in both Servlets and JSPs
- “Beanified” JDBC
 - Setter/Getter methods for attributes
 - HTML / XML output of query results
- Use in different scopes
 - page, request, session, application
- Robust: frees resource when scope expires

Oracle Data Access Beans (2)

- **DBBean**: To open a connection and execute queries
- **ConnBean**: Represents a JDBC Connection
- **QueryBean**: General-purpose bean for all JDBC statements
- **ConnCacheBean**: Connection Caching Bean
 - Different Schemes
 - Configurable

OracleJSP SQL Tags (1)

- **dbOpen**: Opens a JDBC connection, can use a connection pool
- **dbClose**: Closes a connection
- **dbExecute**: Executes DML & DDL
- **dbQuery**: Creates a cursor

```
<sql:dbQuery [queryId="<query-id>"]  
              [connId="<connection-id>"]  
              [output=" {HTML | XML | JDBC}"] >  
    ... SQL Query ...  
</sql:dbQuery>
```

- **dbCloseQuery**: Closes a cursor
- **dbNextRow**: Fetch and process a row

OracleJSP SQL Tags (2)

- **dbCreatePool**: Tag to create a connection pool

```
<sql:dbCreatePool config="app1config.xml"  
    [poolName="<pool-name>"]  
    [ validationMechanism="ANONYMOUS"]  
>
```

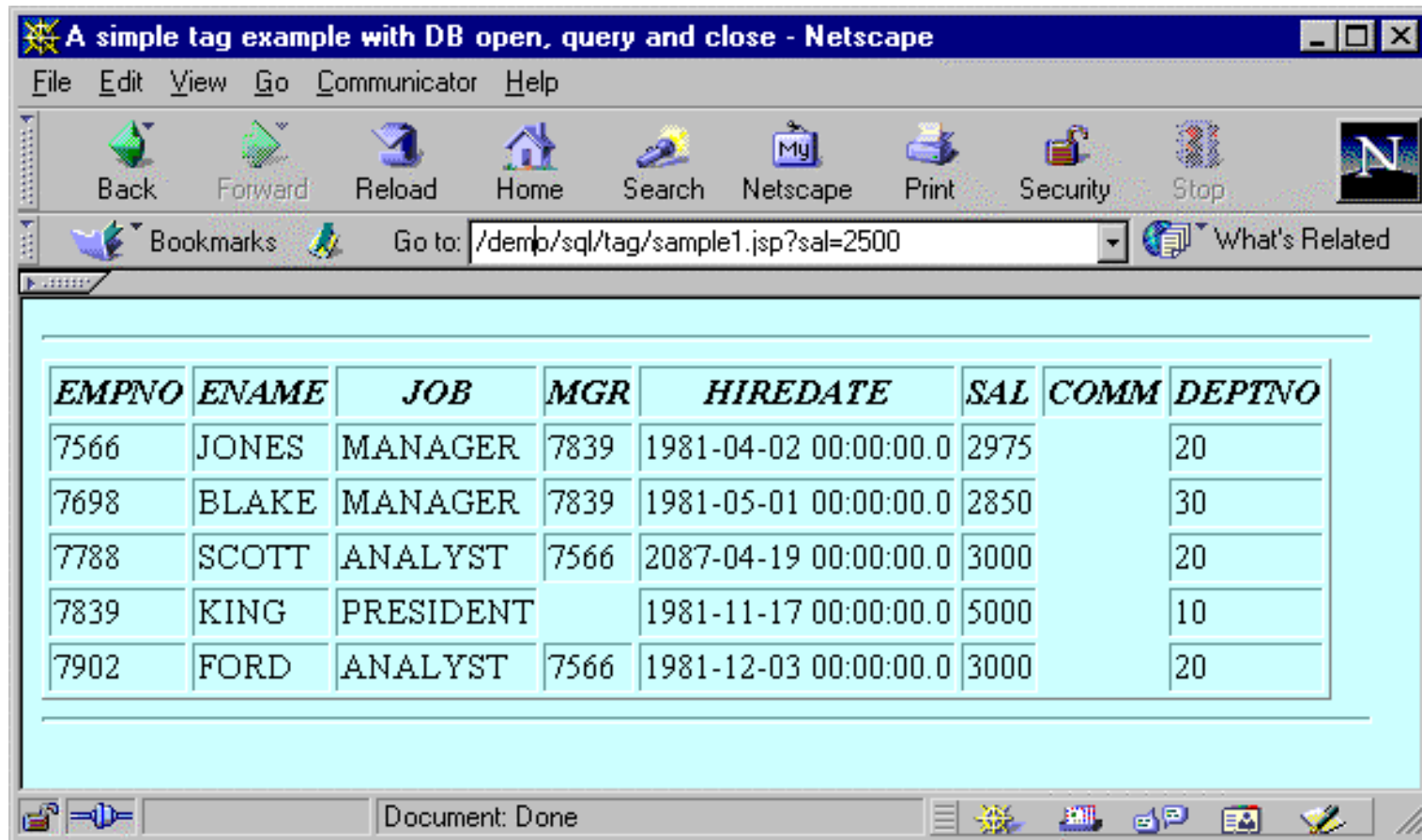
- **dbMonitorPool**: Tag to initiate monitoring and statistics

```
<sql:dbMonitorPool poolName="app1Pool" />  
  
....  
<sql:dbEndMonitor poolName="app1Pool" />
```

OracleJSP SQL Tags Example

```
<sql:dbOpen>
  URL="jdbc:oracle:oci8:@"   usePool="true"
  user="scott" password="tiger" connId="con1">
</sql:dbOpen>
<sql:dbQuery connId="con1" >
  select * from EMP
  where sal > '<%= request.getParameter("sal")
  %>'
</sql:dbQuery>
<sql:dbClose connId="con1" />
```

OracleJSP SQL Tags Output



A screenshot of a Netscape browser window titled "A simple tag example with DB open, query and close - Netscape". The address bar shows the URL "/demp/sql/tag/sample1.jsp?sal=2500". The main content area displays a table with the following data:

| <i>EMPNO</i> | <i>ENAME</i> | <i>JOB</i> | <i>MGR</i> | <i>HIREDATE</i> | <i>SAL</i> | <i>COMM</i> | <i>DEPTNO</i> |
|--------------|--------------|------------|------------|-----------------------|------------|-------------|---------------|
| 7566 | JONES | MANAGER | 7839 | 1981-04-02 00:00:00.0 | 2975 | | 20 |
| 7698 | BLAKE | MANAGER | 7839 | 1981-05-01 00:00:00.0 | 2850 | | 30 |
| 7788 | SCOTT | ANALYST | 7566 | 2087-04-19 00:00:00.0 | 3000 | | 20 |
| 7839 | KING | PRESIDENT | | 1981-11-17 00:00:00.0 | 5000 | | 10 |
| 7902 | FORD | ANALYST | 7566 | 1981-12-03 00:00:00.0 | 3000 | | 20 |

The browser interface includes a menu bar (File, Edit, View, Go, Communicator, Help), a toolbar with icons for Back, Forward, Reload, Home, Search, Netscape, Print, Security, and Stop, and a status bar at the bottom showing "Document: Done".

EQSL dans Cocoon (xml.apache.org)

■ ESQL logicsheet

- XSP logicsheet
 - Produisant des documents XML a partir de requêtes SQL
 - Support les PreparedStatement
 - Peut se mixer avec d'autres logicsheets

Exemple d'EQSL dans Cocoon (i)

```
<datasources>
  <jdbc name="employee_connection">
    <pool-controller min="5" max="10"/>
    <dburl>jdbc:oracle:thin:@dbserver:1521:employee</dburl>
    <user>scott</user>
    <password>tiger</password>
  </jdbc>
  <j2ee name="MyJ2eeConnection">
    <dbname>cocoonDB</dbname>
  </j2ee>
</datasources>
```

Exemple d'EQSL dans Cocoon (ii)

```
<xsp:page language="java" xmlns:xsp="http://apache.org/xsp"
  xmlns:esql="http://apache.org/cocoon/SQL/v2">
<esql:connection>
<esql:pool>employee_connection</esql:pool>
<esql:execute-query>
  <esql:query>SELECT name, salary FROM emp</esql:query>
  <esql:results>
    <table>
      <esql:row-results>
        <tr>
          <td><esql:get-string column="name"/></td>
          <td><esql:get-float column="salary"/></td>
        </tr>
      </esql:row-results>
    </table>
  </esql:results>
  <esql:no-results>
    <p>Sorry, no results!</p>
  </esql:no-results>
</esql:execute-query>
</esql:connection>
</xsp:page>
```

Bibliographie et Sites

■ Livres

- Steve Muench , « Building Oracle XML Applications », September 2000, O'Reilly, 1-56592-691-9, <http://www.oreilly.com/catalog/orxmlapp>
- Kevin Williams, Michael Brundage, Patrick Dengler, Jeff Gabriel, "XML et les bases de données", Eyrolles - 04/2001, ISBN: 2-212-09282-2

■ Sites

- <http://technet.oracle.com>

Bases de Données Semi-Structurées et XML

Didier DONSEZ

Université Joseph Fourier (Grenoble 1)

IMA – LSR/ADELE

`Didier.Donsez@imag.fr`

Motivations

■ Limites de BD-R

- Modèle « plat »
- Difficulté à représenter des informations dont la structure est variable
 - par exemple : document XML, (HTML, SMIL, ...)
- Difficulté d 'exprimer des requêtes sur la structure et le contenu
 - langage, calcul, indexation

■ Limites de BD-OO

- Modèle hiérarchique
 - Document = Graphe de nœuds DOM
 - Faible granularité des objets
- Problème de performances

■ Limite des indexeurs plein-texte

- Indexation des fichiers ASCII
- Recherche sur le contenu pas sur la structure

La solution

- Bases de données semi-structurées

Stockage de Document XML

■ Stockage d'un document XML sur une BD relationnel

- dans une colonne (CLOB, ...)
- « éclaté » sur plusieurs tables
 - Binding DOM/Relational DOM/Object-Relationnal

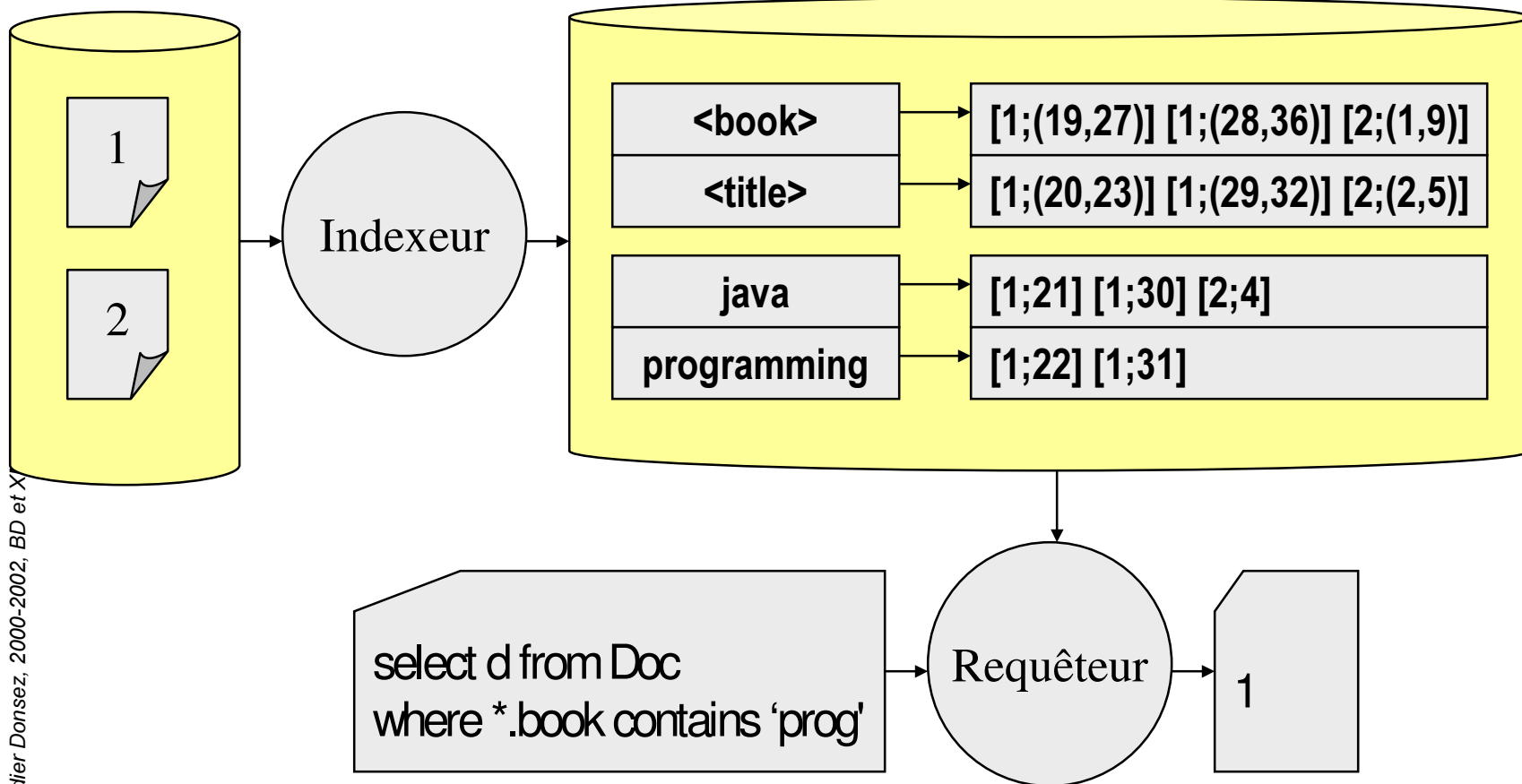
■ Recherche

- Plein texte (Full Text)
 - Recherche sur index (Oracle ConText Cartridge, DB2 Text Extender)
- Structurelle
 - Find books with title "AltaVista Search Revolution".
 - Find the section with the last subsection of which containing the keyword "multimedia".
 - Find the first author of the book titled "Modern Information Retrieval".
 - In the speech spoken by "Antonio", find the line that contains "merchandise".
 - Find authors containing "donald" followed by "knuth" within 2 words.
 - Find all citations in the article titled "Web Databases".
- Remarque : XPath désigne les éléments XML vérifiant un critère structurel

Recherche

Méthodes d'indexation

■ Inverted Files



Langage de requête XML QL

SGBD XML

- SGBDs Relationnels étendus aux documents semi-structurés
 - Oracle8i
 - IBM DB2 XML Extender
 - MS SQL Server
 - Niagara
 - Inverted Files contruits au dessus d'un SGBD-R
 - www.cs.wisc.edu
- SGBDs Objets étendus aux documents semi-structurés
 - Taminos,
 - ObjectStore
 - Ardent
- Les natifs
 - dbXML
 - XIndice (Apache), ...

dbXML

<http://www.dbxml.org>

■ Serveur de Documents XML

- Documents (taille moyenne) regroupés en collections
- Indexation/Compression/Caching
- Interrogation avec Xpath (W3C)
- Mise à jour avec Xupdate (XML:DB Initiative)

■ Interfaces

- Commande en ligne (scripting)
- API Java
- Service CORBA (JacORB) pour les autres langages
- HTTP/SOAP
- Intégration a Cocoon

- Serveur écrit en Java JDK1.3
- Version CORE en OpenSource
- Version Entreprise Commerciale

dbXML

Exemple d'application en Java

```
import org.xmldb.api.base.*; import org.xmldb.api.modules.*; import org.xmldb.api.*;
// Simple XML:DB API example to query the database.
public class Example1 {
    public static void main(String[] args) throws Exception {
        Collection col = null;
        try {
            String driver = "org.vendor.xmldb.DatabasImpl";
            Class c = Class.forName(driver);
            Database database = (Database) c.newInstance();
            DatabaseManager.registerDatabase(database);
            col = DatabaseManager.getCollection("xmldb:vendorx://db.xmlmovies.com:2030/movies");
            String xpath = "//movie[@title='Music Man']";
            XPathQueryService service = (XPathQueryService) col.getService("XPathQueryService", "1.0");
            ResourceIterator results = service.query(xpath);
            while (results.hasMoreResources()) {
                Resource res = results.nextResource();
                System.out.println((String) res.getContent());
            }
        }
        catch (XMLDBException e) {
            System.err.println("XML:DB Exception occured " + e.errorCode);
        }
        finally { if (col != null) { col.close(); } }
    }
}
```

Xindice

<http://xml.apache.org/xindice>

■ BD native XML

- Organiser en Collection (de Collection ...) de Document
 - Valides (DTD et XML Schéma)
- Requêtage Xpath
- Import/Export de documents XML (format texte/fichier)
- Implémenté en Java
- Non transactionnel

■ API

- Java
- GUI
 - http://www.schatten.info/software/xindice_browser/xindice_browser.html
- Intégration à Cocoon

Bibliographie

■ Livres

- Kevin Williams, Michael Brundage, Patrick Dengler, Jeff Gabriel, "XML et les bases de données", Eyrolles - 04/2001, ISBN: 2-212-09282-2

■ Sites

- Etat de l'art
 - <http://www.rpbouret.com/xml/>
- Niagara
 - <http://www.cs.wisc.edu>
- Oracle XML SQL Utility – XSU
 - http://technet.oracle.com/tech/xml/oracle_xsu/content.html