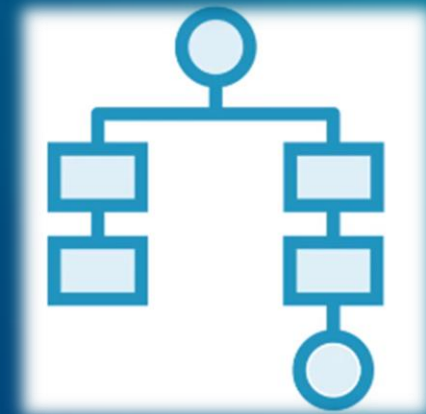


Université de BATNA 2
Département d'informatique

Cours : Les données semi-structurées

L3 ISIL



Chapitre 04 : Xml et les base de données

Djennane, A
Djennane.am@gmail.com

2023 -2024

Contenu

01

**Contexte et
problématique**

04

Xml et les base de données

02

XML *Core* : syntaxe de base,
espaces de noms.

05

XML avancé: XPATH, la
programmation XML (SAX,
DOM), ...etc

03

La validation d'un document
XML (DTD, XML Schema)

Contenu

01

Contexte et problématique

02

XML *Core* : syntaxe de base, espaces de noms.

03

La validation d'un document XML (DTD, XML Schema)

04

Xml et les base de données

05

XML avancé: XPATH, la programmation XML (SAX, DOM), ...etc

Contenu

01

Contexte et problématique

02

XML Core : syntaxe de base, espaces de noms.

03

La validation d'un document XML (DTD, XML Schema)

04

Xml et les base de données

05

XML avancé: XPATH, la programmation XML (SAX, DOM), ...etc

Contenu

01

Contexte et problématique

02

XML Core : syntaxe de base, espaces de noms.

03

La validation d'un document XML (DTD, XML Schema)

04

Xml et les base de données

05

XML avancé: XPATH, la programmation XML (SAX, DOM), ...etc

Contenu

01

Contexte et problématique

02

XML Core : syntaxe de base,
espaces de noms.

03

La validation d'un document
XML (DTD, XML Schema)

04

Xml et les base de données

05

**XML avancé: XPATH,
la programmation
XML (SAX, DOM),
...etc**

CHAPTER 4

On va voir dans ce chapitre :

- **Les besoins d'intégration du XML, avec les Bases de données,**

- **Les approches de gestion (import\export) des données XML.**

- **Le langage JSON (BONUS).**

Un document XML peut être considéré :

1. Orienté **Document** (*document-oriented*):

- Stocker dans un serveur WEB, similaire a une **page** WEB.
 - **L'ordre** (hiérarchie) est important
- Exemple: Content management systems**
(Wikipedia), Blog,

wikipedia.org/wiki/Progressive_web_app



Cet article ne s'appuie pas, ou pas assez, sur des sources **secondaires ou tertiaires** (juin 2019).

Pour améliorer la **vérifiabilité** de l'article ainsi que **son intérêt encyclopédique**, il est nécessaire, quand des **sources primaires** sont citées, de les associer à des analyses faites par des sources secondaires.

Une **progressive web app** (**PWA**, **application web progressive** en français¹) est une **application web** qui consiste en des **pages** ou des **sites web**, et qui peuvent apparaître à l'utilisateur de la même manière que les **applications natives** ou les **applications mobiles**. Ce type d'applications tente de combiner les fonctionnalités offertes par la plupart des **navigateurs** modernes avec les avantages de l'expérience offerte par les **appareils mobiles**^{2,3}.

Une **PWA** se consulte comme un site web classique, depuis une URL sécurisée mais permet une **expérience utilisateur** similaire à celle d'une application mobile, sans les contraintes de cette dernière (soumission aux App-Stores, utilisation importante de la **mémoire** de l'appareil...).

Elles proposent de conjuguer rapidité, fluidité et légèreté tout en permettant de limiter considérablement les coûts de développement⁴ : plus besoin de faire des développements spécifiques pour les applications en fonction de chacune des plateformes : **iOS**, **Android**...

Historique [[modifier](#) | [modifier le code](#)]

Durant les années 2000, les technologies du web sont progressivement passées de documents statiques (HTML) à des solutions dynamiques exécutées côté serveur avec des langages tels que **PHP**. Lors de l'arrivée des **smartphones** en 2007, les technologies du web ne sont plus adaptées face à l'augmentation de l'utilisation des applications mobiles. Ces dernières offrent une bien meilleure expérience utilisateur. Les ressources packagées et l'accès direct aux fonctions du téléphone ont permis aux applications

Progressive web app

PWA

Informations

Développé par	Google
Première version	~2015
Écrit en	JavaScript, JSON
Environnement	Android, iOS, Windows

[modifier](#) - [modifier le code](#) - [voir Wikidata \(aide\)](#)

2. Orienté données (*data-oriented*) :

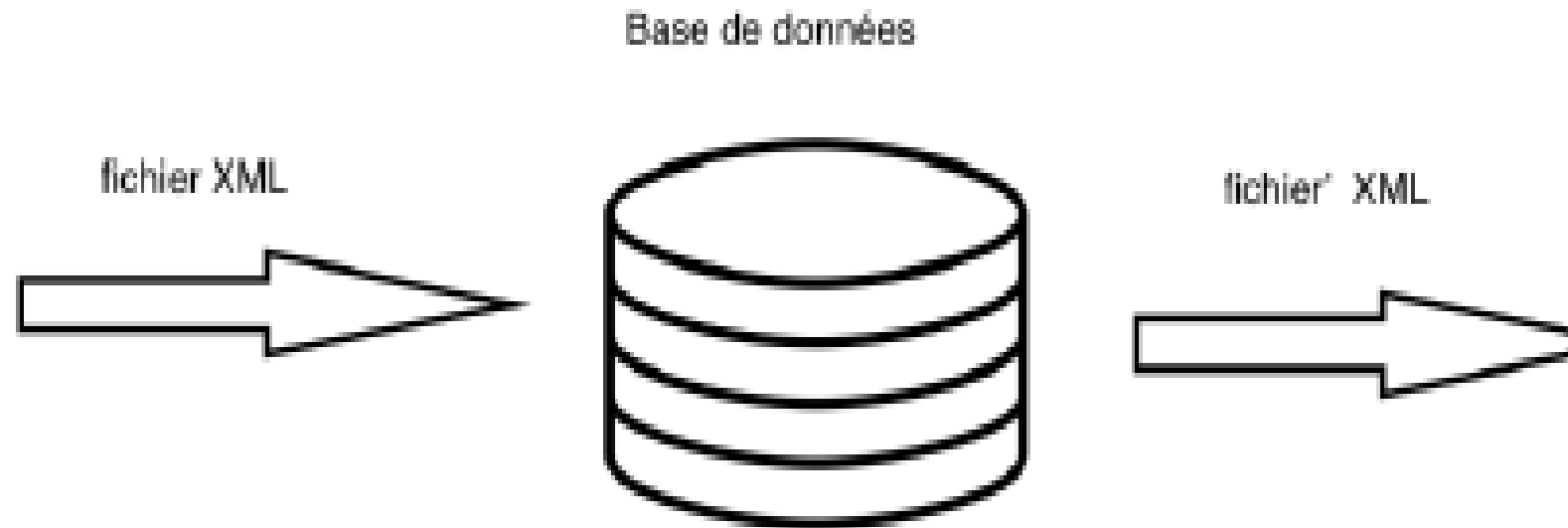
Le document XML est conçu pour stocker et transmettre des données dans un **structuré**,

tels que :

- Les enregistrements de **base de données**,
- les **paramètres de configuration**,

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <employees>
3   <employee>
4     <id>1</id>
5     <name>John Doe</name>
6     <department>IT</department>
7     <salary>50000</salary>
8   </employee>
9   <employee>
10    <id>2</id>
11    <name>Jane Smith</name>
12    <department>HR</department>
13    <salary>45000</salary>
14  </employee>
15  <!-- More employee records -->
16 </employees>
```

- Besoin de **stocker** des documents XML ("données XML") dans une BD, les **interroger** et les **mettre à jours** : *intégration à un SI existant*.
- Besoins de **convertir en forme XML d'autres données**, e.g. celles retrouvées dans des BD habituelles, *en vue de les transmettre (autre SI, autre format, ...)*.



- Stocker XML sous forme de **fichier binaire (BLOB), ou chaîne de caractère (Varchar, ...)**.
- **XML Column**, tout un document XML est mis comme valeur d'un attribut dans un tuple (**type XML**).
- **Shred & publish**, "découper" le document et mettre les différents morceaux comme valeurs régulières dans différents attributs dans différentes tables. Le recomposer à la demande.
- **XML DataBase**, tout les documents dans une "BD" à part, dédiée et qui ne contient que du XML, et propose XPath et XQuery pour les interroger.

Approche traditionnelle

Bases de données étendues

BD XML native

1. Approche traditionnelle

04. XML et BD

- **Premières solutions très simples** : garder du XML dans une chaîne de caractères, exemple : VARCHAR2, ou BLOB (Binary Large Object)

facture (BLOB)	ID
<pre><facture id='245' > <client>Ali Ahmed</client> <produit id ='11'>...</produit> <produit id ='21'>... </produit> </facture></pre>	1

Un Binary Large Object (BLOB) est un type de données MySQL qui peut stocker des données binaires telles que des images, du multimédia et des fichiers PDF.

- **La sérialisation XML (XML serialization) : conversion XML -> BD**

- **étendre les fonctionnalités** des BD du marché (Oracle, DB2 d'IBM, PostgreSQL et SQLServer de Microsoft) afin de supporter XML comme **type natif**.
- **Offre un support XML.**

- **Solution actuelle** : type de données à part pour XML, et sur ce type des fonctions spécifiques XML sont implémentées.

Le type XML permet de stocker des éléments XML dans une table SQL :

```
CREATE TABLE TestXML (id INTEGER PRIMARY KEY, data XML);  
INSERT INTO TestXML VALUES (1, '<test>ok</test>');  
SELECT * FROM TestXML;
```

En fait, le type XML est quasiment identique à VARCHAR ou TEXT. Le seul avantage est la vérification du XML :

```
INSERT INTO TestXML VALUES (2, '<test>mauvais</verif>');
```

```
line 1: Opening and ending tag mismatch: test line 1 and ve  
line 1: chunk is not well balanced
```

Solution MySQL

Le stockage de données depuis XML dans une base MySQL en utilisant la déclaration **LOAD XML** (implémentée dans MySQL 6.0) ;

Solution Microsoft ACCESS

Importation de données

2. Les BD étendues : Exportation

04. XML et BD

Solution MySQL

Le stockage de données depuis XML dans une base MySQL en utilisant la déclaration

LOAD FILE, LOAD XML
(implémentée dans MySQL 6.0.3)

Solution Microsoft ACCESS

Importation de données

Solution Microsoft SQL Server

Les procédures stockées.

Commande MySQL

```
mysql> USE xmltest;
mysql> CREATE TABLE xmldocs (
  ->   id INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  ->   doc_content BLOB NOT NULL,
  ->   comment VARCHAR(100) NOT NULL DEFAULT ''
  -> );
Query OK, 0 rows affected (0.04 sec)

mysql> INSERT INTO xmldocs VALUES
  ->   (NULL, LOAD_FILE('/tmp/2cities.xml'), '2 cities file');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM xmldocs\G
```

Sortie XML générée

```
id: 1
doc_content: <?xml version="1.0"?>

<resultset statement="SELECT name,country FROM cities LIMIT 2
" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <row>
    <field name="name">Mumbai (Bombay)</field>
    <field name="country">India</field>
  </row>

  <row>
    <field name="name">Seoul</field>
    <field name="country">South Korea</field>
  </row>
</resultset>

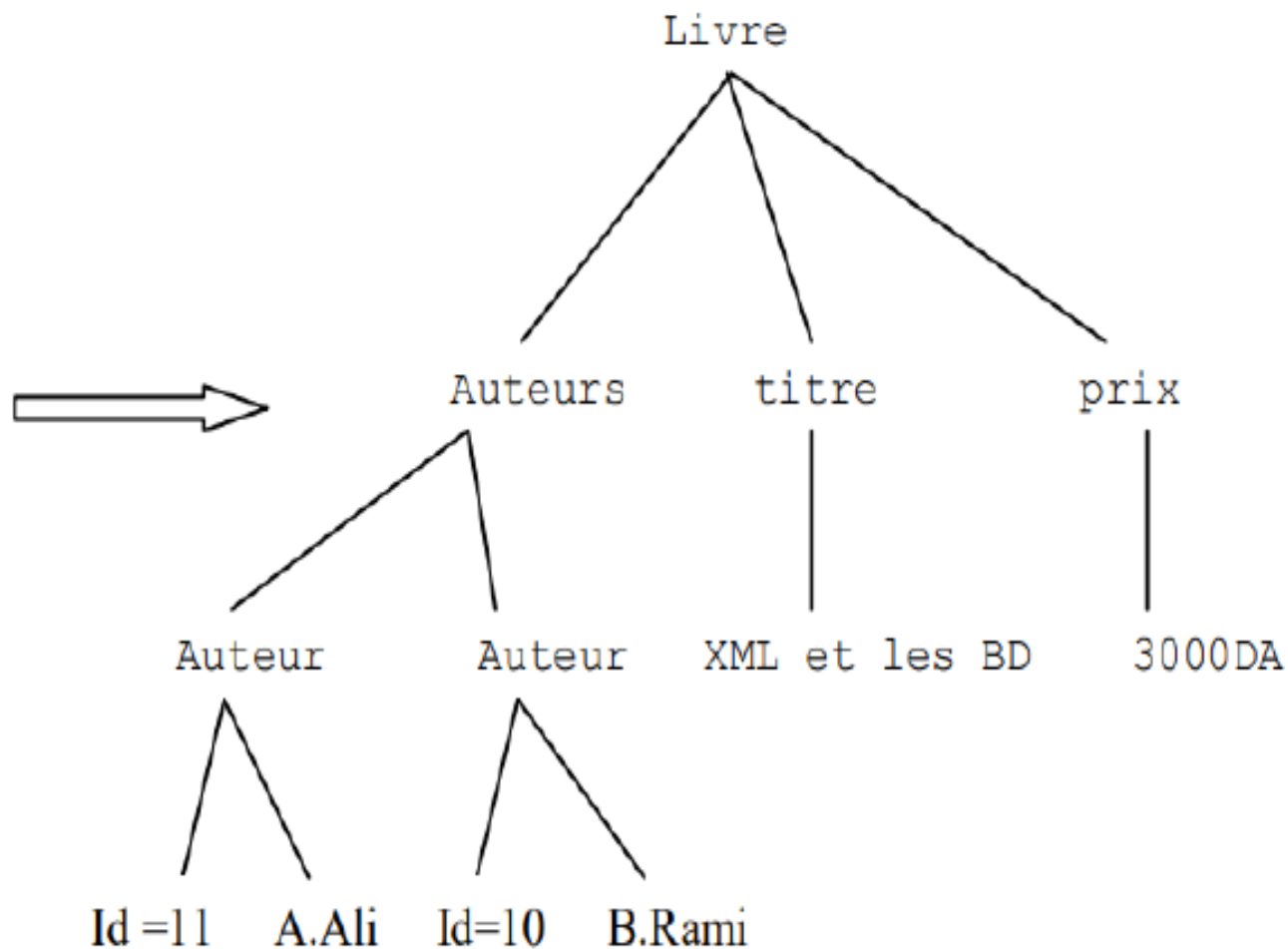
comment: 2 cities file
1 row in set (0.00 sec)
```

Avantages des BD étendues :

- Bénéficier de l'utilisation du **langage SQL** au niveau relationnel pour l'interrogation et la mise à jour des données XML => requêtes
- On tire profit des notions très fortes de **normalisation, indexation**, qui sont déjà implémentées dans les SGBDR => performances
- On profite également de toutes **les fonctions SGBD** accessibles et fiables : accès multiples, gestion complexe d'utilisateurs, techniques de sauvegarde.

- Les bases de données XML natives sont plus utiles pour le stockage des contenus **orientés document**, car elles préservent l'ordre interne du document, les instructions de traitement, les commentaires, les sections CDATA. Etc.
- Les bases XML natives supportent les **langages de requêtes XML (XQuery)**, qui permettent d'effectuer des recherches difficiles à effectuer dans un langage tel que SQL.

```
<livre>
  <Auteurs>
    <auteur id=11>
      A.Ali
    </auteur>
    <auteur id=10>
      B.Rami
    </auteur>
  <titre>
    XML et les BD
  </titre>
  <prix> 3000 DA </prix>
</livre>
```



Les critères à prendre en considération dans le choix du type de base de données à utiliser pour le stockage des données XML sont :

- **Les requêtes** : si les **opérations** sont **nombreuses** le choix d'un SGBD relationnel s'impose, s'il s'agit de **requêtes complexes** sur les contenus, alors le choix d'une base de données XML native est plus adapté.
- **La taille de flux XML à traiter en entrée et en sortie du système** : plus ils sont nombreux et les formats de données sont variés, plus il faudra prévoir dans le cas d'un choix du SGBDR.
- **la complexité du schémas XML utilisé** : les schémas très arborescents seront mieux traités par une base de données XML native.
- **Sécurité.**

4. Langage de requêtes

04. XML et BD

5. Exemple WEB avec PHP

04. XML et BD

Research

- **Etudiez le support de XML pour MySQL : Load XML**
- **Etudiez quelques SGBD XML natives , existant dans le marche
avec une petite comparaison**