

**Université de Jijel**  
**Faculté des Sciences Exactes et d'Informatique**  
**Département d'Informatique**  
**L3 – Systèmes Informatiques**

# **Données Semi-Structurées**

## **Chapitre 05**

### **XML et les BDD**

**Tarek Boutefara**  
[t\\_boutefara@univ-jijel.dz](mailto:t_boutefara@univ-jijel.dz)  
**2020/2021**

# Plan

- Utilisations du XML
- XML et BDD
- Modèles
- Architectures
- Exemple : BaseX
- XQuery

# Plan

- **Utilisations du XML**
- XML et BDD
- Modèles
- Architectures
- Exemple : BaseX
- XQuery

# Utilisation du XML

- XML fait partie des langages les plus utilisés pour stocker des données semi-structurées.
- Il est utilisé pour :
  - Stocker des données,
  - Sauvegarder des configurations (maven, ant, ...)
  - Modèles de base pour d'autres langages de programmation (RDF, OWL, ...)
  - Support de modélisation des données (XSD, ...)
  - Sites Web.

# Utilisation du XML

- Dans le contexte des données, XML peut être utilisé (principalement) :
  - L'échange des données
    - AJAX,
    - SOAP,
    - ...
  - Stocker des données
    - Base de Données XML

# Utilisation du XML

- Échange de données :
  - Plusieurs environnements :
    - Échange Client/Serveur (front-end/back-end),
    - Échange entre Systèmes d'Information.
  - Les données des deux parties ne sont pas forcément stockées dans des fichiers XML :
    - Des transformations de schémas sont nécessaires,
    - XML comme support d'échange seulement (pas d'interrogation, d'indexation, ou autres fonctionnalités des SGBD).

# Utilisation du XML

- Stocker des données :
  - Les données du Systèmes d'Information sont stockées, au niveau physique, dans des fichiers XML,
  - On parle de Base de Données XML,
  - Toutes les fonctionnalités d'un SGBD doivent être assurées :
    - Indexation,
    - Interrogation,
    - Contraintes,
    - Accès concurrent,
    - ...

# Plan

- Utilisations du XML
- **XML et BDD**
- Modèles
- Architectures
- Exemple : BaseX
- XQuery



# XML et BDD

- Niveaux d'abstraction
  - Niveau Conceptuel,
  - Niveau Logique,
  - Niveau Physique.

# XML et BDD

- Niveaux d'abstraction
  - Niveau Conceptuel → MCD (Entité/Association)
  - Niveau Logique (??),
  - Niveau Physique → XML.

# XML et BDD

- Niveau Logique
  - Le modèle conceptuel (MCD) doit être traduit en un modèle logique adéquat au mode de stockage choisi,
  - Le modèle relationnel est le modèle logique adéquat lorsqu'on envisage l'utilisation d'une base de données relationnelle,
  - Inadéquat pour un stockage basé sur les fichiers XML.

# XML et BDD

- Niveau Logique
  - Modèle Hiérarchique
    - Le modèle logique le plus adéquat pour un fichier XML,
    - Repose aussi sur une représentation sous forme d'un arbre (arborescence).

# XML et BDD

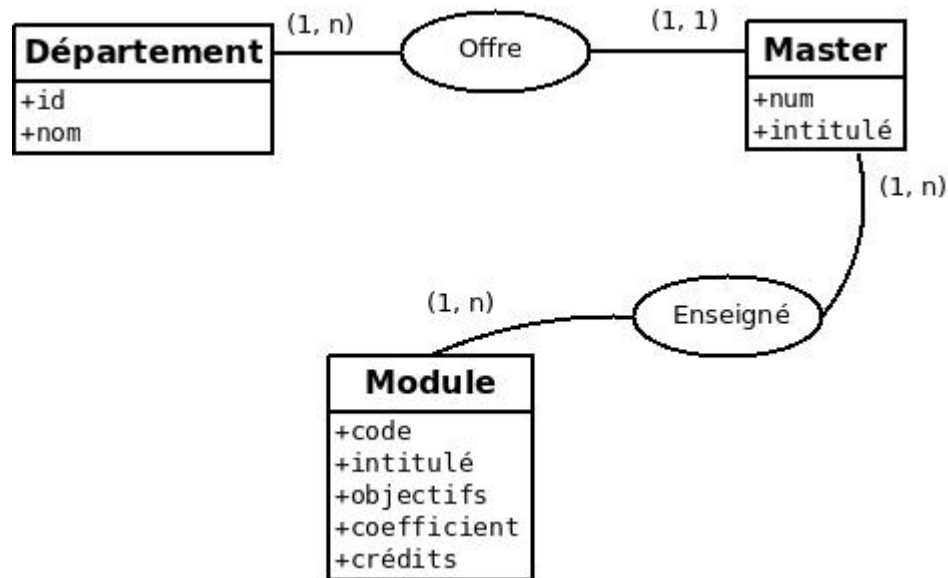
- Niveau Logique
  - Modèle Hiérarchique
    - Toutes les associations sont des association de type 1 :N,
    - Les associations de type N:M doivent être converties à une association de type 1:N
      - On choisit un sens de l'association,
    - Les association de dimension supérieur à deux ( $\geq 3$ ) doivent être décomposées.

# XML et BDD

- Niveau Logique
  - Modèle Hiérarchique au niveau physique :
    - Chaque entité devient un élément,
    - Chaque propriété peut être traduit comme :
      - Un élément (nœud feuille) avec un contenu,
      - Un attribut.
    - Les liens (qui étaient des associations au niveau conceptuel) peuvent être traduits :
      - Sous-éléments (solution la plus utilisée)
      - Liens Xpointer (moins supportée)

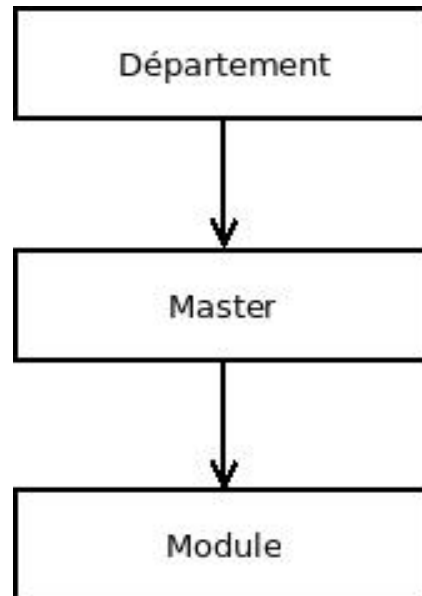
# XML et BDD

- Niveau Logique
  - Exemple : soit le MCD :



# XML et BDD

- Niveau Logique
  - Exemple : une possibilité du modèle logique (selon le choix du sens de l'association) :





# XML et BDD

- Niveau Logique
  - Exemple : Le fichier XML (entité module simplifié) (partie 1) :
    - <departements>
      - <departement id="1">
        - <nom>Informatique</nom>
        - <masters>
          - <master num="1" intitule="SIAD">
            - <module>ED</module>
            - <module>BDA</module>
          - </master>
          - <master num="2" intitule="RS">
            - <module>Protocoles</module>
            - <module>Sécurité</module>
          - </master>
        - </masters>
      - </departement>

# XML et BDD

- Niveau Logique
  - Exemple : Le fichier XML (entité module simplifié) (partie 2) :
    - <departement id="2">
      - <nom>Math</nom>
      - <masters>
        - <master num="3" intitule="Statistiques">
          - <module>Stats</module>
          - <module>Méthodes Numériques</module>
        - </master>
        - <master num="4" intitule="Mathématiques Appliquées">
          - <module>Analyse</module>
          - <module>Algèbre</module>
        - </master>
      - </masters>
    - </departement>
  - </departements>

# Plan

- Utilisations du XML
- XML et BDD
- **Modèles**
- Architectures
- Exemple : BaseX
- XQuery

# Modèles

- Les documents XML peuvent être :
  - Orientés Données,
  - Orientés Document.

# Modèles

- Doc. XML orientés Données :
  - Structure régulière,
  - Données de types simples (granularité fine),
  - Absence des éléments mixtes (comme les paragraphes qui contiennent des séquences du text et des liens par exemple),
  - Destinés à un traitement automatique.

# Modèles

- Doc. XML orientés Données :
  - Exemples :
    - Facture,
    - Données d'un Client,
    - ...
  - Peuvent être facilement extraits à partir d'une base de données (partage les mêmes caractéristiques définies par le modèle relationnel et les formes normales).

# Modèles

- Doc. XML orientés Documents :
  - Structure moins régulière,
  - Granularité plus grande et présence des éléments complexes et mixtes,
  - Destiné à une utilisation humaine :
    - Ordre des éléments important.

# Modèles

- Doc. XML orientés Documents :
  - Exemples :
    - Support du cours,
    - Article de blog,
    - ...
  - Généralement créés par un humain en utilisant un éditeur.



# Plan

- Utilisations du XML
- XML et BDD
- Modèles
- **Architectures**
- Exemple : BaseX
- XQuery

# Architectures

- L'intégration du XML comme un support de stockage et d'échange des données peut être assuré en suivant deux méthodes :
  - Mapping Table/XML
    - Utilisation des middle-ware,
  - Stockage XML
    - SGBDR avec extension XML
    - SGBD XML (BDD XML natives)

# Architectures

- L'intégration du XML comme un support de stockage et d'échange des données peut être assuré en suivant deux méthodes :
  - Mapping Table/XML
    - Utilisation des middle-ware,
  - Stockage XML
    - SGBDR avec extension XML
    - SGBD XML (BDD XML natives)

# Architectures

- Middle-ware
  - Comme
    - JAXB sous J2EE,
    - Apache Axis,
    - BEA Weblogic Workshop.
  - Des outils plus simples permet de récupérer les données sous format XML
    - Client MySQL

# Architectures

- Middle-ware : modèle générique
  - <database>
    - <table>
      - <row>
        - <column></column>
        - <column></column>
        - ...
      - </row>
      - ...
    - </table>
  - </database>

# Architectures

- Client MySQL (Exemple)
  - Soit la base de données « Exemple »
  - Qui contient la table « Personne »
  - Sur un serveur avec l'utilisateur « root »

# Architectures

- Client MySQL (Exemple)
  - Pour activer le module XML, il suffit d'ajouter l'option --xml
  - `mysql -u root -p -D "Exemple" -e "Select * from Personne --xml"`

# Architectures

tarek : bash — Konsole

Fichier Édition Affichage Bookmarks Configuration Aide

```
-$ mysql -u tarek -p -D "Exemple" -e "Select * from Personne"
```

Enter password:

id	nom	prenom
1	Ahmed	Benahemd
2	Amer	Benamer

```
-$ █
```

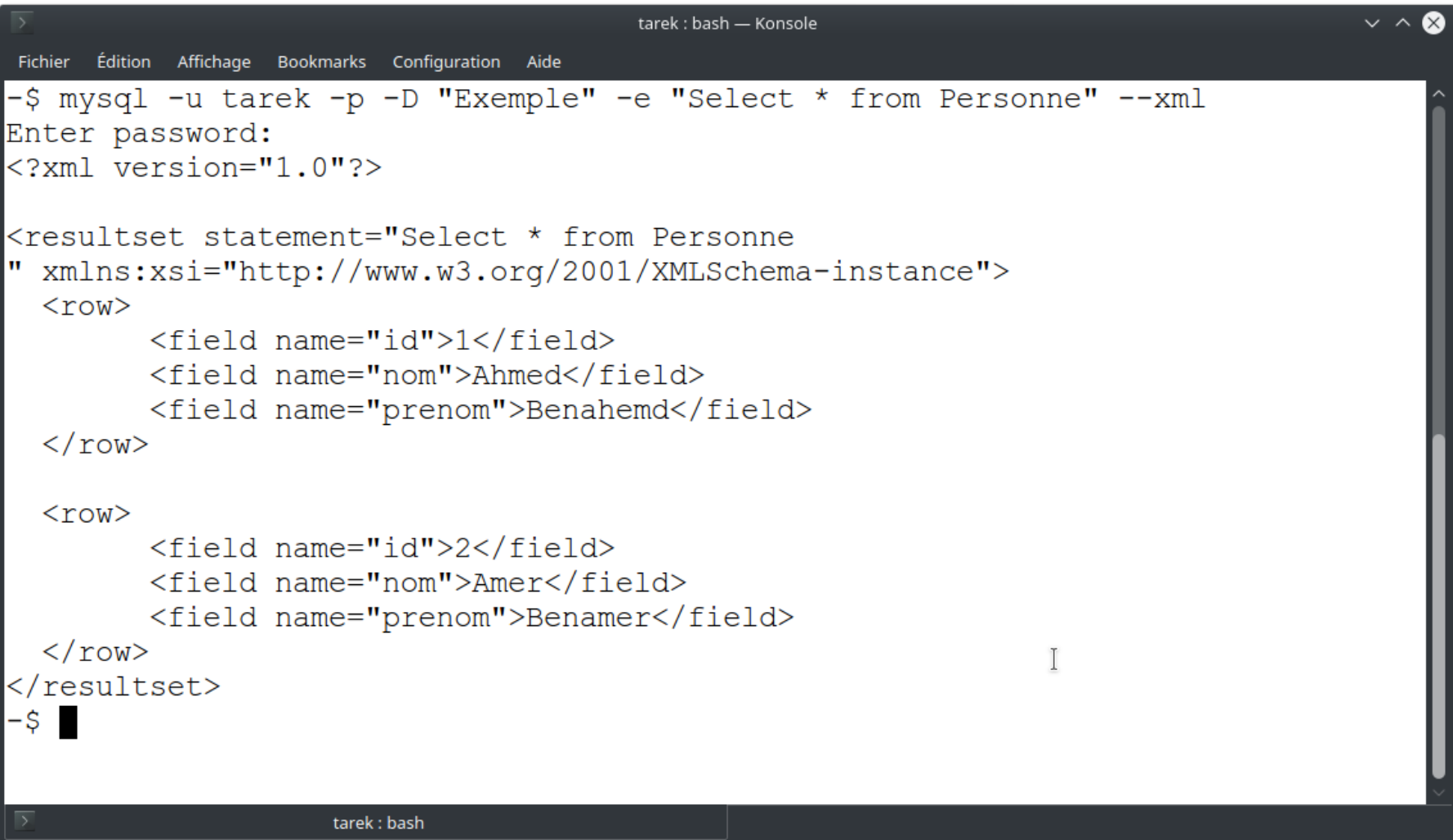
tarek : bash



# Architectures

- Client MySQL (Exemple)
  - La commande suivante permet de récupérer le contenu de la table :
  - `mysql -u root -p -D "Exemple" -e "Select * from Personne"`

# Architectures



The image shows a terminal window titled 'tarek : bash — Konsole'. The terminal displays a MySQL command and its XML output. The command is: `mysql -u tarek -p -D "Exemple" -e "Select * from Personne" --xml`. The output is an XML document with a root element `<resultset statement="Select * from Personne"` and two rows of data. The first row has fields `id` (1), `nom` (Ahmed), and `prenom` (Benahemd). The second row has fields `id` (2), `nom` (Amer), and `prenom` (Benamer). The terminal also shows the prompt `Enter password:` and the XML declaration `<?xml version="1.0"?>`.

```
tarek : bash — Konsole
Fichier  Édition  Affichage  Bookmarks  Configuration  Aide

-$ mysql -u tarek -p -D "Exemple" -e "Select * from Personne" --xml
Enter password:
<?xml version="1.0"?>

<resultset statement="Select * from Personne
" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <row>
    <field name="id">1</field>
    <field name="nom">Ahmed</field>
    <field name="prenom">Benahemd</field>
  </row>

  <row>
    <field name="id">2</field>
    <field name="nom">Amer</field>
    <field name="prenom">Benamer</field>
  </row>
</resultset>
-$
```

# Architectures

- SGBDR avec extension XML
  - Plusieurs SGBDR proposent des fonctions XML et permet d'utiliser XML comme format de stockage,
    - Les différentes fonctionnalités (accès concurrent, contraintes, ...) sont assurées par les mêmes fonctions de la partie Relationnelle,
    - Des fonctions additionnelles sont offertes pour manipuler les fichiers XML (format, XPath, XQuery)

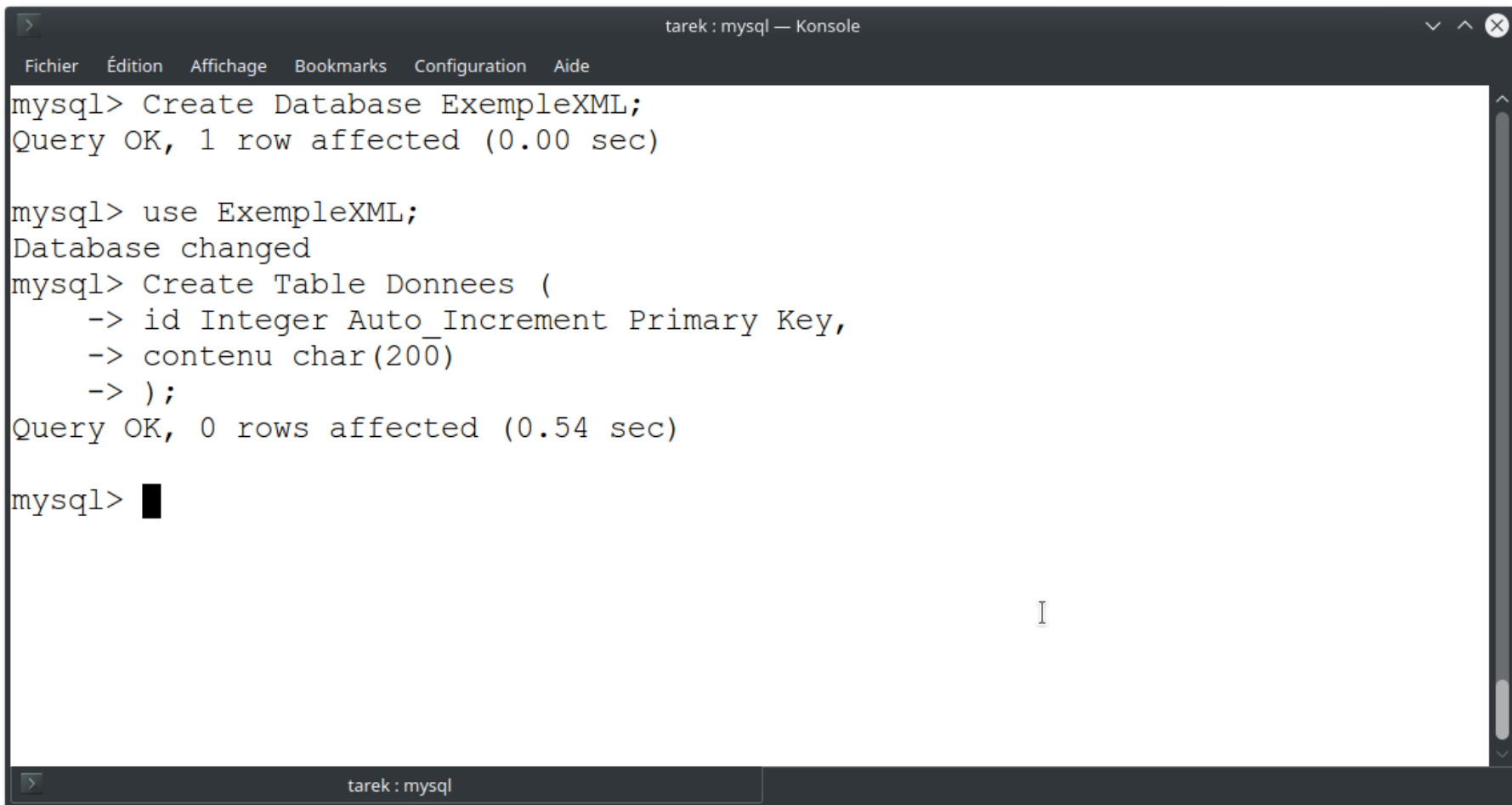
# Architectures

- SGBDR avec extension XML
  - Exemple MySQL : parmi les fonctions XML sous MySQL :
    - ExtractValue(colonne, xpath)
    - UpdateXML(colonne, xpath, nouvelle\_valeur)

# Architectures

- SGBDR avec extension XML
  - Exemple MySQL :
    - Soit la base de données « ExempleXML »
    - Qui contient une seule table « Donnees »
    - Qui contient, à son tour, deux colonnes :
      - Id
      - Contenu

# Architectures



The image shows a screenshot of a MySQL console window. The window has a dark title bar with the text "tarek : mysql — Konsole" and standard window controls. Below the title bar is a menu bar with options: "Fichier", "Édition", "Affichage", "Bookmarks", "Configuration", and "Aide". The main area of the window contains the following text:

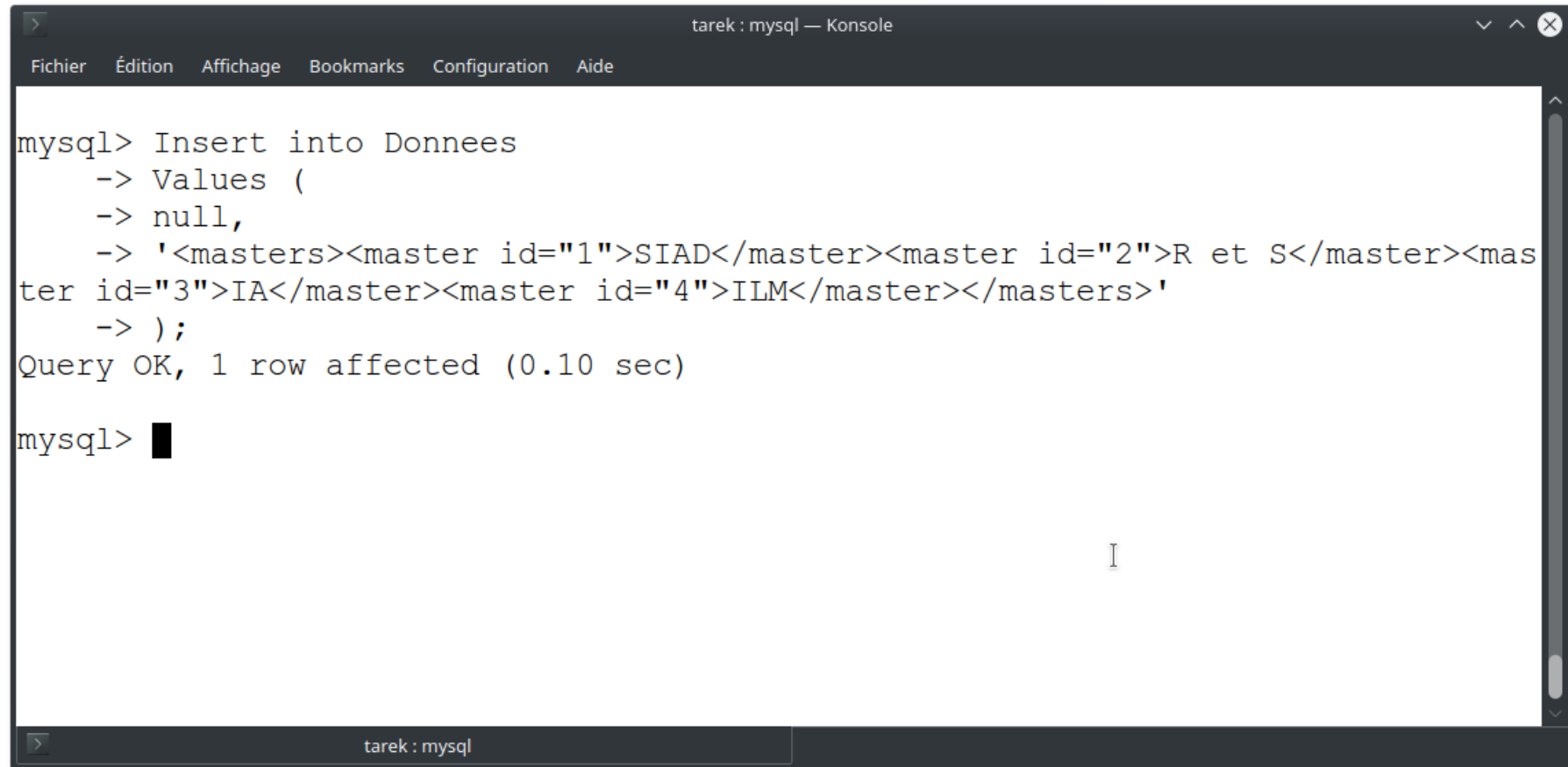
```
mysql> Create Database ExempleXML;  
Query OK, 1 row affected (0.00 sec)  
  
mysql> use ExempleXML;  
Database changed  
mysql> Create Table Donnees (  
    -> id Integer Auto_Increment Primary Key,  
    -> contenu char(200)  
    -> );  
Query OK, 0 rows affected (0.54 sec)  
  
mysql> █
```

The cursor is positioned at the end of the last line, indicated by a small vertical bar. The bottom of the window shows a status bar with the text "tarek : mysql".

# Architectures

- SGBDR avec extension XML
  - Exemple MySQL :
    - Nous allons insérer un enregistrement :
      - La colonne « Contenu » va contenir des données XML (valide, une seule racine, bien formé)
      - <masters>
        - <master id="1">SIAD</master>
        - <master id="2">R et S</master>
        - <master id="3">IA</master>
        - <master id="4">ILM</master>
      - </masters>

# Architectures



The screenshot shows a MySQL console window titled 'tarek : mysql — Konsole'. The window has a menu bar with 'Fichier', 'Édition', 'Affichage', 'Bookmarks', 'Configuration', and 'Aide'. The main area contains the following text:

```
mysql> Insert into Donnees
-> Values (
-> null,
-> '<masters><master id="1">SIAD</master><master id="2">R et S</master><mas
ter id="3">IA</master><master id="4">ILM</master></masters>'
-> );
Query OK, 1 row affected (0.10 sec)

mysql> █
```

The cursor is positioned at the end of the second line of the SQL statement. The status bar at the bottom of the window shows 'tarek : mysql'.



# Architectures

- SGBDR avec extension XML
  - Exemple MySQL :
    - Démo : ExtractValue()
      - Nous allons sélectionner le master avec la valeur du « id » qui égale à 2
      - Nous allons utiliser le id contenu en XML et non pas de la table,
      - XPath
        - //master[@id= « 2 »]
      - Requête sous MySQL
        - Select ExtractValue(contenu, '//master[@id="2"]')  
From Donnees;

# Architectures



The image shows a terminal window titled "tarek : mysql — Konsole". The terminal has a menu bar with "Fichier", "Édition", "Affichage", "Bookmarks", "Configuration", and "Aide". The command prompt "mysql>" is followed by the SQL query: "Select ExtractValue(contenu, '//master[@id="2"]') From Donnees;". The output is a table with one row and one column. The column header is "ExtractValue(contenu, '//master[@id="2"]')". The row contains the value "R et S". Below the table, it says "1 row in set (0.00 sec)". The prompt "mysql>" is followed by a black square cursor.

```
tarek : mysql — Konsole
Fichier  Édition  Affichage  Bookmarks  Configuration  Aide

mysql> Select ExtractValue(contenu, '//master[@id="2"]') From Donnees;
+-----+
| ExtractValue(contenu, '//master[@id="2"]') |
+-----+
| R et S                                     |
+-----+
1 row in set (0.00 sec)

mysql> ■
```

# Architectures

- BDD XML natives :
  - On parle des bases de données XML natives si les données sont stockées, gérées et interrogées en utilisant des technologies XML seulement (XML n'est pas une extension, mais le noyau est basé sur XML).

# Architectures

- BDD XML natives :
  - Deux approches :
    - Stockage en utilisant le modèle textuel
      - Les fichiers XML sont stockés sous forme de fichiers textes.
    - Stockage en utilisant un modèle de document
      - Les fichiers XML sont utilisés pour construire un objet document,
      - L'objet est stocké suivant une structure interne propre au SGBD.

# Architectures

- BDD XML natives :
  - Ces SGBDs implémentent toutes les fonctionnalités d'un SGBDR :
    - Structure : collection de documents (au lieu de tables),
    - Interrogation : Xpath, XQL, Xquery,
    - Gestion des transactions
      - Verouillage basé sur un document entier.
    - Connexion :
      - API dédié, pas de pilote générique (comme JDBC)
    - Indexes : par éléments et par attributs.

# Architectures

- BDD XML natives :
  - Plusieurs SGBD sont disponibles
    - BaseX
    - Exist
    - Oracle Berkeley DB XML
    - Open Text xDB
    - ...

# Architectures

- BDD XML natives :
  - Néanmoins :
    - Les SGBD XML natives sont de moins en moins utilisés,
    - Les SGBD (NoSQL) les plus récents reposent sur JSON et non pas XML.

# Architectures

- JSON
  - JavaScript Object Notation
  - Format inspiré du syntaxe JavaScript pour la définition d'objets.



# Architectures

- JSON
  - Exemple : soit le code XML suivant
    - `<livre id="1234">`
      - `<titre>Java</titre>`
      - `<prix>1200 DZD</prix>`
      - `<auteurs>`
        - `<auteur>Adam Bien</auteur>`
        - `<auteur>Sebastian Daschner</auteur>`
      - `</auteurs>`
    - `</livre>`

# Architectures

- JSON
  - Son équivalent en JSON sera
    - {
      - "livre": {
        - "\_id": "1234",
        - "titre": "Java",
        - "prix": "1200 DZD",
        - "auteurs": ["Adam Bien", "Sebastian Daschner"]

# Plan

- Utilisations du XML
- XML et BDD
- Modèles
- Architectures
- **Exemple : BaseX**
- XQuery

# BaseX

- Un SGBD XML Native
  - Java,
  - Open Source,
  - Installeur/Jar
  - Mode GUI/Mode Serveur
  - Gestion des transactions, des utilisateurs, des droits d'accès, etc.

# BaseX

- Un SGBD XML Native
  - Lancement du fichier Jar en mode GUI :
    - `java -jar BaseX.jar nom_fichier.xml`
  - Exemple : fichier Departements.xml
    - `java -jar BaseX.jar Departements.xml`

# BaseX

file [Departements] - BaseX 8.4.2

Database Editor View Visualization Options Help

Find Find... 0 Results

Editor

Departements.xml

departements			departement		
departement			departement		
nom	masters		nom	masters	
I..	master module ED	master module Protocoles	M..	master module Stats	master module Analyse
	module BDA	module Sécurité		module Méthodes Numériques	module Algèbre

Result

```
<departements>
  <departement id="1">
    <nom>Informatique</nom>
    <masters>
      <master num="1" intitule="SIAD">
        <module>ED</module>
        <module>BDA</module>
      </master>
      <master num="2" intitule="RS">
        <module>Protocoles</module>
        <module>Sécurité</module>
      </master>
    </masters>
  </departement>
  <departement id="2">
```

Query Info

Total Time: 2045.33 ms

Command:  
CHECK Departements.xml

Result:  
Database 'Departements' created in 2003.52 ms.

Time needed: 2045.33 ms 22 MB

# BaseX

The screenshot displays the BaseX 8.4.2 application window. The title bar indicates the current file is 'Departements.xml'. The interface is divided into several panels:

- Database**: A sidebar on the left showing a file tree for the user 'tarek'. The tree includes folders like '.8pecxstudios', '.adobe', '.ant', '.anydesk', '.apt-src', '.asciidoc', '.assaultcube', '.bluefish', '.bluej', '.cache', and '.chosenim'.
- Find**: A search bar at the top with a red rectangular highlight around it.
- Editor**: The central workspace, outlined with a blue rectangle, which is currently empty.
- Results**: A panel on the right showing two XML tables. The first table, titled 'Departements.xml', has columns 'departement', 'nom', and 'masters'. It contains two rows: one for 'I...' with modules 'ED' and 'BDA', and another for 'M..' with modules 'Protocoles' and 'Sécurité'. The second table, titled 'departement', has columns 'nom', 'masters', and 'module', with rows for 'Stats' (module 'Méthodes Numériques') and 'Analyse' (module 'Algèbre'). This panel is outlined with a green rectangle.
- Query Info**: A panel at the bottom right, outlined with a yellow rectangle, showing the command 'CHECK Departements.xml' and the result: 'Database 'Departements' created in 2003.52 ms.'.
- Result**: A panel at the bottom left, outlined with a brown rectangle, displaying the XML content of the 'Departements.xml' file. The XML structure is as follows:

```
<departements>
  <departement id="1">
    <nom>Informatique</nom>
    <masters>
      <master num="1" intitule="SIAD">
        <module>ED</module>
        <module>BDA</module>
      </master>
      <master num="2" intitule="RS">
        <module>Protocoles</module>
        <module>Sécurité</module>
      </master>
    </masters>
  </departement>
  <departement id="2">
```

The status bar at the bottom indicates 'Time needed: 2045.33 ms' and '22 MB'.

# BaseX

- Un SGBD XML Native
  - Parties du GUI :
    - **Commandes et XPath,**
    - **Requêtes XQuery,**
    - **Visualisation de la sélection (résultat de la requête),**
    - **Contenu du résultat de la requête,**
    - **Statistiques sur l'exécution de la requête.**



# BaseX

- Un SGBD XML Native
  - Exemple d'un chemin XPath :
  - Afficher les modules enseignés dans le master qui a comme numéro : 1
    - `//master[@num= « 1 »]/module`

# BaseX

file [Departements] - BaseX 8.4.2

Database Editor View Visualization Options Help

Find //master[@num="1"]/module 2 Results

/home/tarek

\*.xml, \*.xq\*

Find contents...

tarek

.8pecxstudios

.adobe

.ant

.anydesk

.apt-src

.asciidoc

.assaultcube

.bluefish

.bluej

.cache

.choosenim

Editor

file

1

Departements.xml

departements

departement

nom masters

I.. master module

ED

Protocoles

module

BDA

Sécurité

departement

M.. master module

Stats

Analyse

module

Méthodes Numériques

Algèbre

Result

<module>ED</module>

<module>BDA</module>

Query Info

Total Time: 5.47 ms

Compiling:

- atomic evaluation of (@\*:num = "1")
- rewriting descendant-or-self step(s)
- applying attribute index for "1"

Optimized Query:

db:attribute("Departements", "1")/self::\*:num/parent::\*:master/\*:module

Query:

//master[@num="1"]/module

Result:

- Hit(s): 2 Items
- Updated: 0 Items
- Printed: 40 Bytes
- Read Locking: local [Departements]

db:open("Departements", "Departements.xml")/departements/departement/masters/master/module/text() 23 MB

# Plan

- Utilisations du XML
- XML et BDD
- Modèles
- Architectures
- Exemple : BaseX
- **XQuery**

# XQuery

- XQuery est
  - Un langage d'interrogation des BDD XML,
  - Il peut extraire les éléments et les attributs,
  - Il permet aussi d'ordonner et d'effectuer des opérations d'agrégation,
  - Il peut être utilisé pour extraire des données pour des traitements ou bien pour effectuer des transformations.

# XQuery

- XQuery syntaxe
  - Repose sur XPath
  - Modèle FLWOR (lu FLOWER)
    - For ensemble\_d\_elements
    - Let lier\_variables\_a\_elements
    - Where conditions
    - Order by critere\_pour\_ordonner
    - Return elements\_a\_retourner
  - XQuery est sensible à la case (x et X sont considérés comme différents)

# XQuery

- XQuery exemple
  - Les module du master qui a comme numero la valeur 1 :
    - for \$m in doc("dep.xml")//master
    - where \$m/@num="1"
    - return \$m/module

**Université de Jijel**  
**Faculté des Sciences Exactes et d'Informatique**  
**Département d'Informatique**  
**L3 – Systèmes Informatiques**

# **Données Semi-Structurées**

## **Chapitre 05**

### **XML et les BDD**

**Tarek Boutefara**  
[t\\_boutefara@univ-jijel.dz](mailto:t_boutefara@univ-jijel.dz)  
**2020/2021**