University of Jijel
Faculty of Exact Sciences and Computer Science
Department of Computer Science
L3 – Computer Systems

Semi-Structured Data
Chapter 5
XML and Databases

Tarek Boutefara
t_boutefara@univ-jijel.dz
2025/2026

Content

- Uses of XML
- Models
- XML and Databases
- Architectures
- Example: BaseX
- XQuery

Content

- **Uses of XML**
- Models
- XML and Databases
- Architectures
- Example: BaseX
- XQuery

Uses of XML

- XML is one of the most widely used languages for storing semi-structured data.
- It is used for:
  - Storing data,
  - Saving configurations (Maven, Ant, etc.),
  - Basic models for other programming languages (RDF, OWL, etc.),
  - Data modeling support (XSD, etc.), and
  - Websites.

Uses of XML

- In the context of data, XML can be used (primarily):
  - Data exchange
    - AJAX,
    - SOAP,
    - …
  - Data storage
    - XML database

Uses of XML

- Data exchange:
  - Multiple environments:
    - Client/Server exchange (front-end/back-end),
    - Exchange between Information Systems.
  - Data from both parties is not necessarily stored in XML files:
    - Schema transformations are necessary,
    - XML as an exchange medium only (no querying, indexing, or other DBMS functionalities).

Uses of XML

- Data storage:
  - Information system data is stored, at the physical level, in XML files.
  - This is referred to as an XML Database.
  - All the functionalities of a DBMS must be ensured:
    - Indexing,
    - Querying,
    - Constraints,
    - Concurrent access,
    - ...

Content

- Uses of XML
- **Models**
- XML and Databases
- Architectures
- Example: BaseX
- XQuery

Models

- XML documents can be:
  - Data-oriented,
  - Document-oriented.

Models

- Data-oriented XML document:
  - Regular structure,
  - Simple data types (fine granularity),
  - Absence of mixed elements (such as paragraphs containing text sequences and links, for example),
  - Intended for automatic processing.

Models

- Data-oriented XML documents:
  - Examples:
    - Invoice,
    - Customer data,
    - …
  - Can be easily extracted from a database (shares the same characteristics defined by the relational model and normal forms).

Models

- Document-oriented XML:
  - Less regular structure,
  - Greater granularity and presence of complex and mixed elements,
  - Intended for human use:
    - Element order is important.

Models

- XML-oriented documents:
  - Examples:
    - Course materials,
    - Blog post,
    - …
  - Generally created by a human using an editor.

Content

- Uses of XML
- Models
- **XML and Databases**
- Architectures
- Example: BaseX
- XQuery

XML and Databases

- Levels of abstraction
  - Conceptual level,
  - Logical level,
  - Physical level.

XML and Databases

- Levels of abstraction
  - Conceptual level → ERD (Entity/Relationship)
  - Logical level → (**??**),
  - Physical level → XML.

XML and Databases

- Logical Level
  - The conceptual model (CDM) must be translated into a logical model suitable for the chosen storage method.
  - The relational model is the appropriate logical model when considering the use of a relational database.
  - It is unsuitable for storage based on XML files.

XML and Databases

- Logical Level
  - Hierarchical Model
    - The most suitable logical model for an XML file,
    - also relies on a representation in the form of a tree (tree structure).

XML and Databases

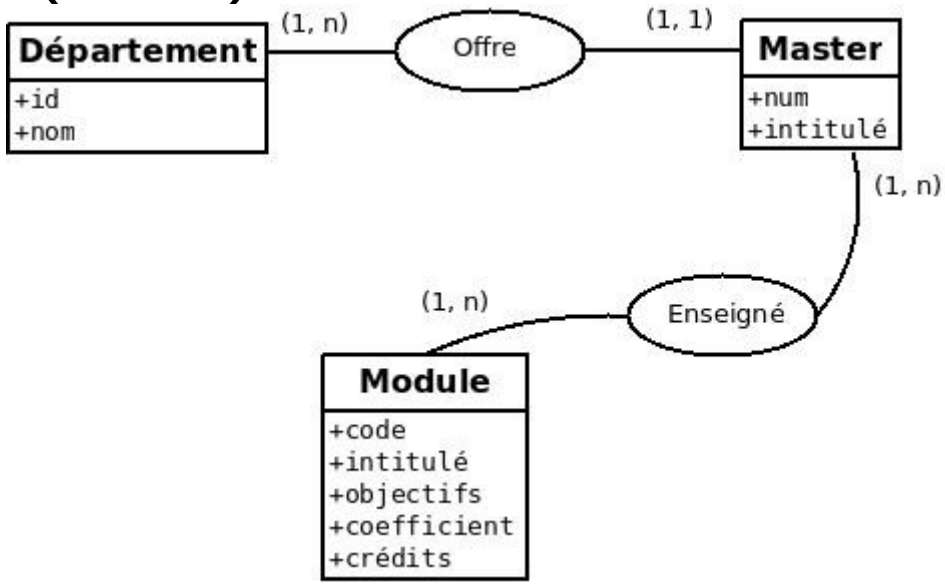
- Logical Level
  - Hierarchical Model
    - All associations are of type 1:N,
    - Associations of type N:M must be converted to a 1:N association
      - A direction of the association is chosen,
    - Associations of dimension greater than two (≥ 3) must be decomposed.

XML and Databases

- Logical Level
  - Hierarchical Model at the physical level:
    - Each entity becomes an element.
    - Each property can be translated as:
      - An element (leaf node) with content,
      - An attribute.
    - The links (which were associations at the conceptual level) can be translated as:
      - Sub-elements (most commonly used solution)
      - Xpointer links (less supported)

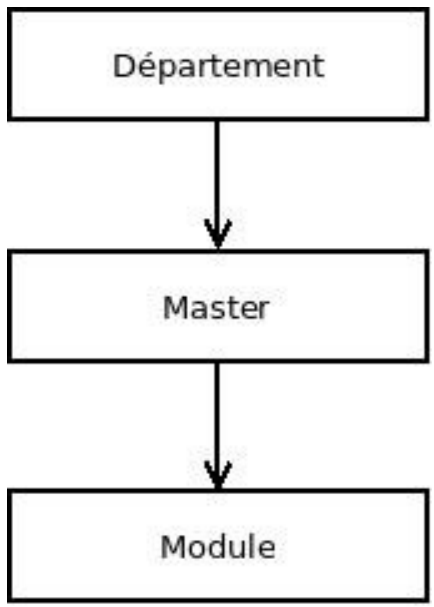
XML and Databases

- Logical Level
  - Example: Consider the conceptual data model (CDM):



XML and Databases

- Logical Level
  - Example: one possibility of the logical model (depending on the chosen direction of the association):



XML and Databases

- Physical Level
  - Example: The XML file (simplified module entity) (part 1):
    - <departements>
      - <departement id="1">
        - <nom>Informatique</nom>
        - <masters>
          - <master num="1" intitule="SIAD">
            - <modules>
              - <module>ED</module>
              - <module>BDA</module>
            - </modules>
          - </master>
          - <master num="2" intitule="RS">
            - <modules>
              - <module>Protocoles</module>
              - <module>Sécurité</module>
            - </modules>
          - </master>
        - </masters>
      - </departement>

XML and Databases

- Physical Level
  - Example: The XML file (simplified module entity) (part 2):
    - <departement id="2">
      - <nom>Math</nom>
      - <masters>
        - <master num="3" intitule="Statistiques">
          - <modules>
            - <module>Stats</module>
            - <module>Méthodes Numériques</module>
          - </modules>
        - </master>
        - <master num="4" intitule="Mathématiques Appliquées">
          - <modules>
            - <module>Analyse</module>
            - <module>Algèbre</module>
          - </modules>
        - </master>
      - </masters>
    - </departement>
  - </departements>

Content

- Uses of XML
- Models
- XML and Databases
- **Architectures**
- Example: BaseX
- XQuery

Architectures

- The integration of XML as a data storage and exchange medium can be achieved using two methods:
  - Table/XML Mapping
    - Use of middleware,
  - XML Storage
    - RDBMS with XML extension
    - XML DBMS (native XML databases)

Architectures

- Middle-ware
  - Like
    - JAXB sous J2EE,
    - Apache Axis,
    - BEA Weblogic Workshop.
  - Simpler tools allow you to retrieve data in XML format.
    - MySQL Client

Architectures

- Middle-ware : generic model
  - <database>
    - <table>
      - <row>
        - <column></column>
        - <column></column>
        - ...
      - </row>
      - ...
    - </table>
  - </database>

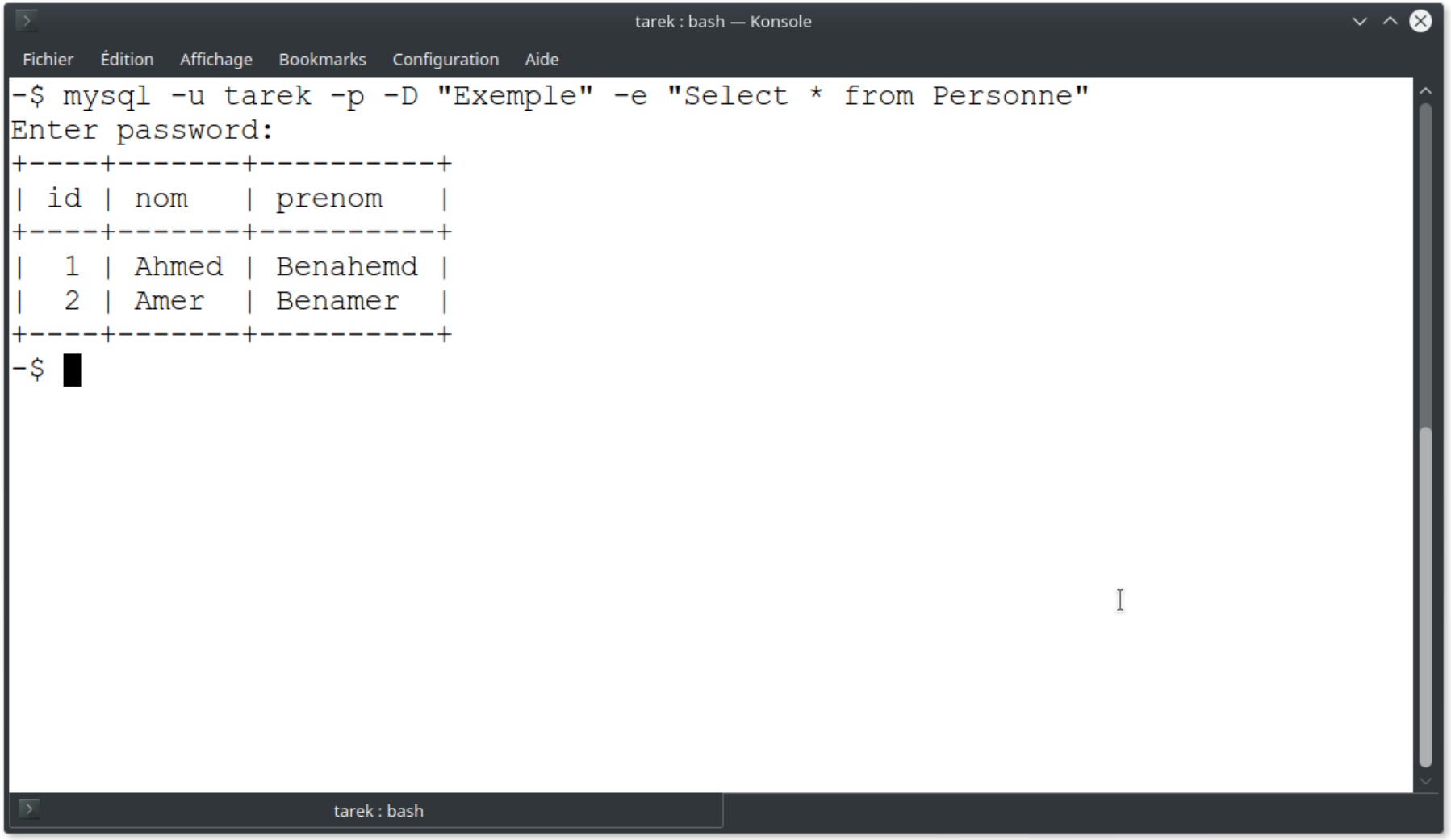
Architectures

- MySQL Client (Example)
  - Consider the database "Example"
  - Which contains the table "Person"
  - On a server with the user "root"

Architectures

- MySQL Client (Example)
  - To enable the XML module, simply add the --xml option:
  - `mysql -u root -p -D "Example" -e "Select * from Person"`

Architectures



Architectures

- MySQL Client (Example)
  - The following command retrieves the contents of the table:
  - 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" --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>
-$
```

tarek : bash

Architectures

- Relational Database Management Systems (RDBMS) with XML Extension
  - Several RDBMSs offer XML functions and allow the use of XML as a storage format.
  - The various functionalities (concurrent access, constraints, etc.) are provided by the same functions as the relational database.
  - Additional functions are offered for manipulating XML files (formatting, XPath, XQuery).

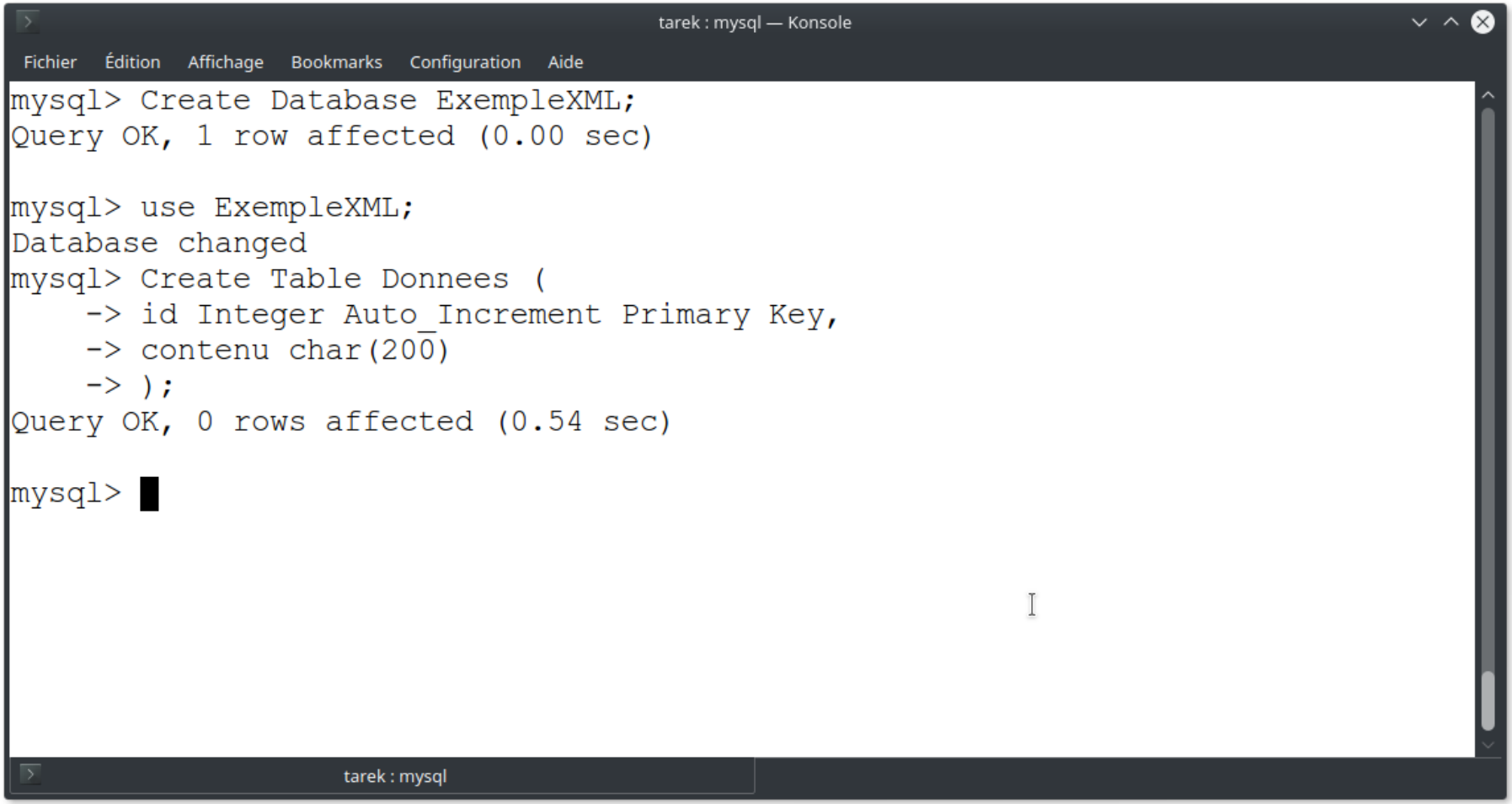
Architectures

- Relational database management system (RDBMS) with XML extension
  - Example: MySQL: among the XML functions in MySQL:
    - ExtractValue(column, xpath)
    - UpdateXML(column, xpath, new_value)

Architectures

- RDBMS with XML extension
  - Example: MySQL
    - Consider the database "ExampleXML"
    - Which contains a single table "Data"
    - Which, in turn, contains two columns:
      - Id
      - Content

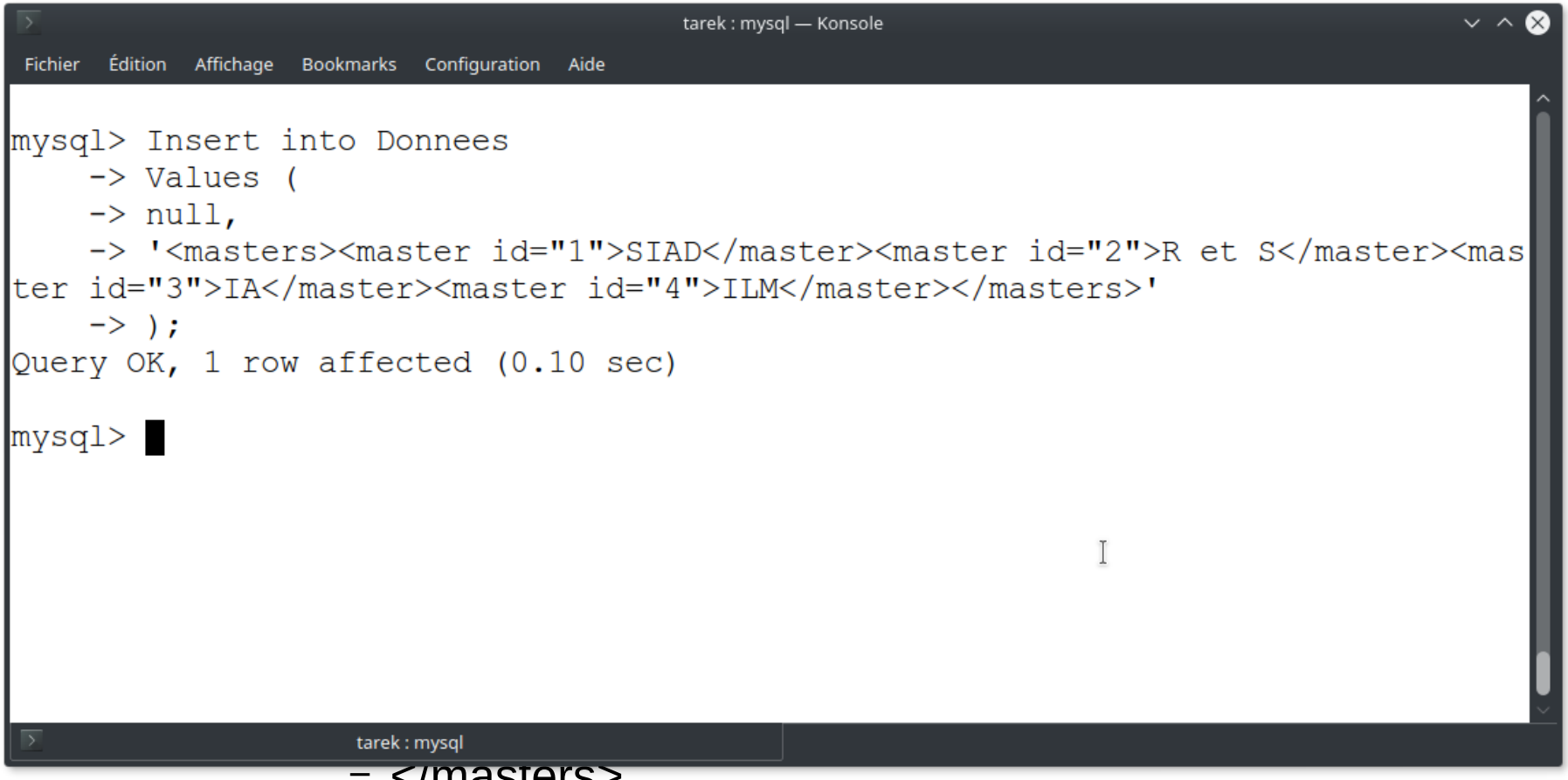
Architectures



Architectures

- RDBMS with XML extension
  - MySQL example:
    - We will insert a record:
      - The "Content" column will contain XML data (valid, single root, well-formed)
      - <masters>
        - <master id="1">SIAD</master>
        - <master id="2">R and S</master>
        - <master id="3">IA</master>
        - <master id="4">ILM</master>
      - </masters>

Architectures



- </masters>

Architectures

- RDBMS with XML extension
  - MySQL Example:
    - Demo: ExtractValue()
      - We will select the master with the "id" value equal to 2.
      - We will use the ID contained in the XML and not from the table.
      - XPath
        - //master[@id="2"]
      - MySQL Query
        - Select ExtractValue(content, '//master[@id="2"]') From Data;

Architectures

Architectures

- Native XML databases:
  - We speak of native XML databases if the data is stored, managed and queried using only XML technologies (XML is not an extension, but the core is based on XML).

Architectures

- Native XML databases:
  - Two approaches:
    - Storage using a text model
  - XML files are stored as text files.
    - Storage using a document model
      - XML files are used to construct a document object.
      - The object is stored according to an internal structure specific to the DBMS.

Architectures

- Native XML databases:
  - These DBMSs implement all the functionalities of a relational DBMS:
    - Structure: document collection (instead of tables),
    - Querying: XPath, XQL, XQuery,
    - Transaction management,
      - Locking based on an entire document.
    - Connection:
      - Dedicated API, no generic driver (like JDBC),
    - Indexes: by element and by attribute.

Architectures

- Native XML databases:
  - Several DBMSs are available:
    - BaseX
    - Exist
    - Oracle Berkeley DB XML
    - Open Text xDB
    - ...

Architectures

- Native XML databases:
  - However:
    - Native XML database management systems (DBMS) are being used less and less.
    - The most recent (NoSQL) DBMSs are based on JSON, not XML.

Architectures

- JSON
  - JavaScript Object Notation
  - A format inspired by JavaScript syntax for defining objects.

Architectures

- JSON
  - Example: Consider the following XML code:
    - <book id="1234">
      - <title>Java</title>
      - <price>1200 DZD</price>
      - <authors>
        - <author>Adam Bien</author>
        - <author>Sebastian Daschner</author>
      - </authors>
    - </book>

Architectures

- JSON
  - Its JSON equivalent will be:
    - {
      - "book": {
        - "_id": "1234",
        - "title": "Java",
        - "price": "1200 DZD",
        - "authors": ["Adam Bien", "Sebastian Daschner"]
      - }
    - }

Content

- Uses of XML
- Models
- XML and Databases
- Architectures
- **Example: BaseX**
- XQuery

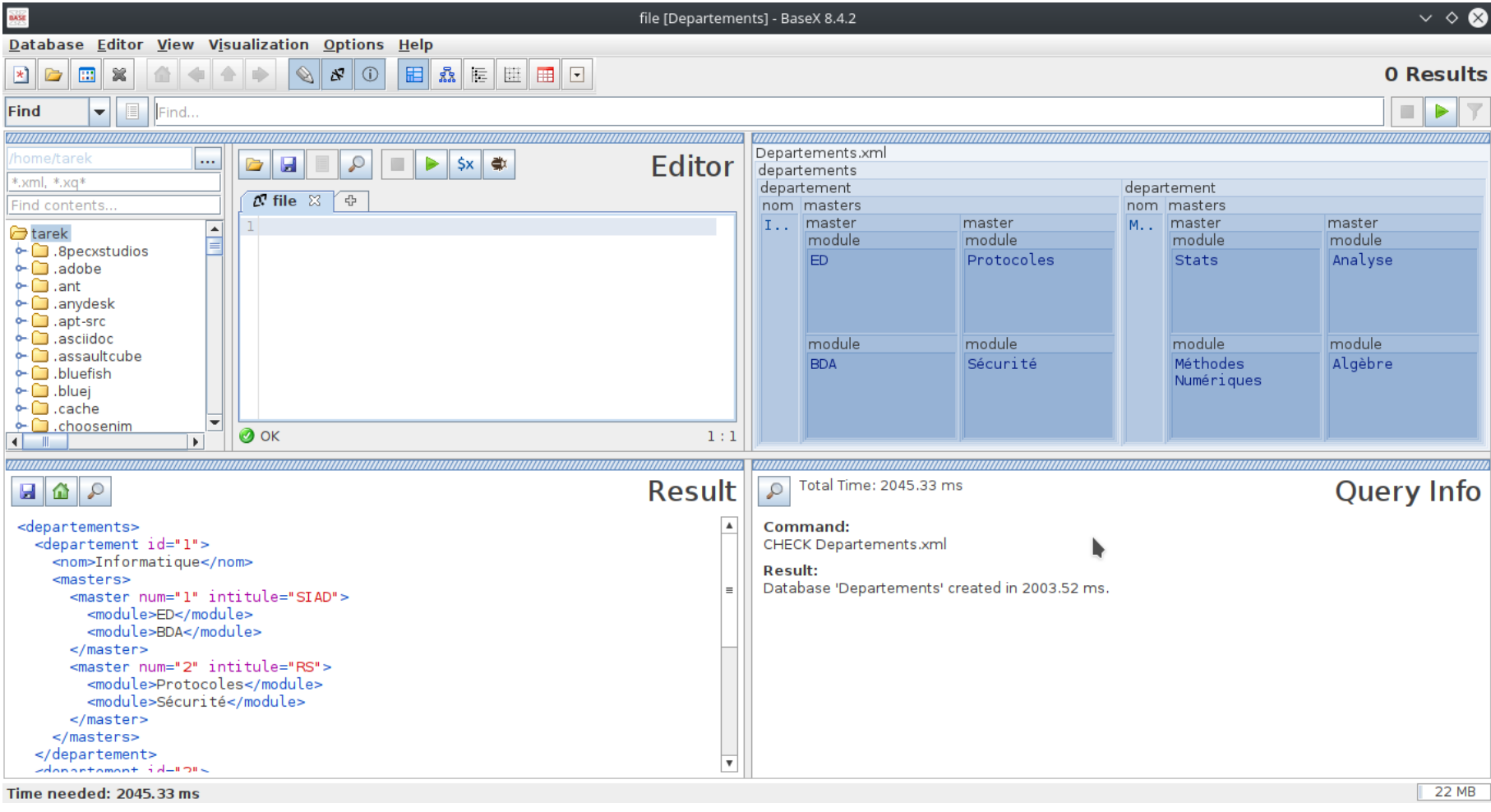
BaseX

- A native XML DBMS
  - Java,
  - Open Source,
  - Installer/Jar
  - GUI Mode/Server Mode
  - Management of transactions, users, access rights, etc.

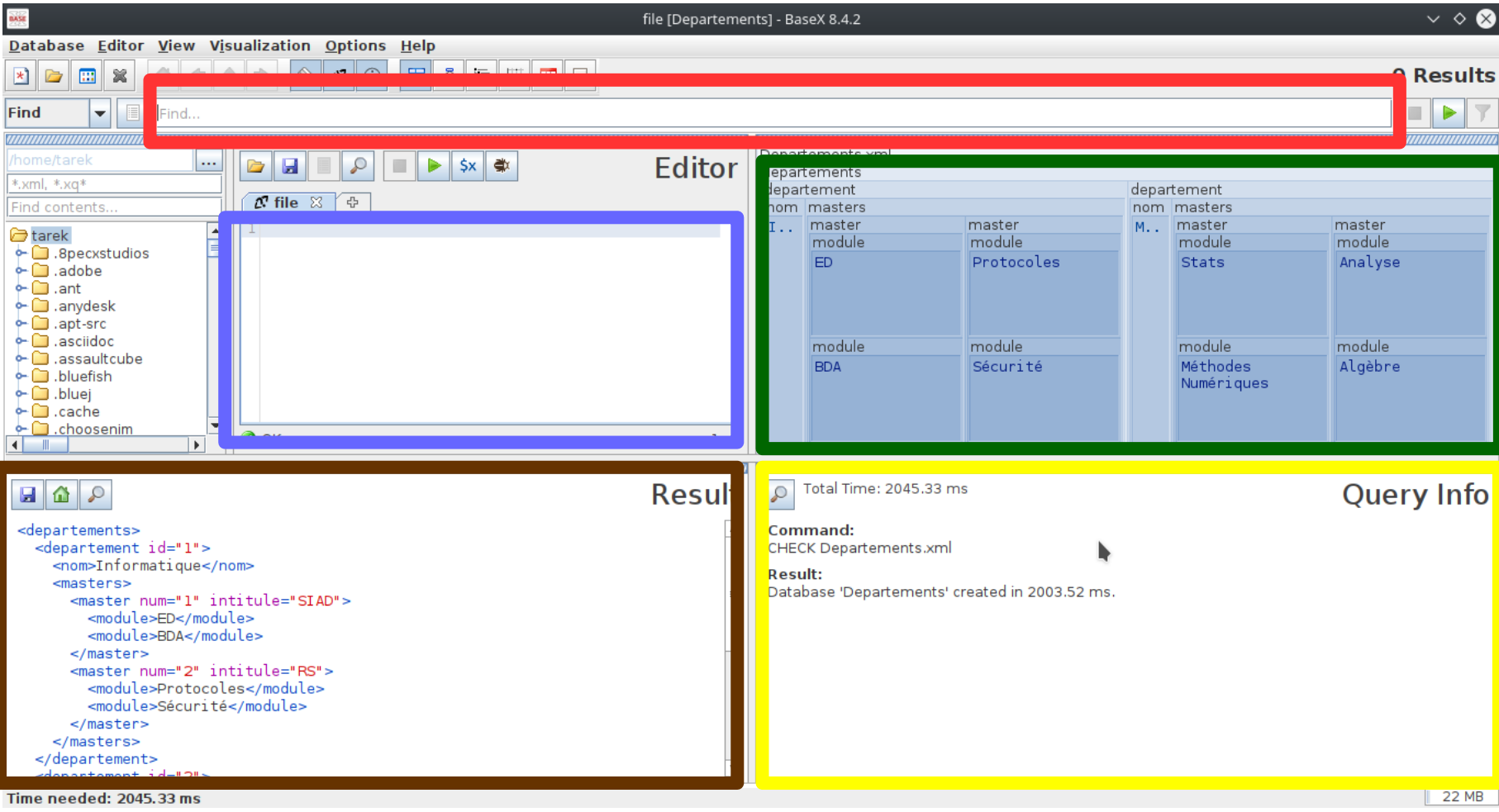
BaseX

- A Native XML DBMS
  - Launching the JAR file in GUI mode:
    - java -jar BaseX.jar filename.xml
  - Example: file Departments.xml
    - java -jar BaseX.jar Departments.xml

BaseX


file [Departements] - BaseX 8.4.2
Database Editor View Visualization Options Help
0 Results
Find
Find...
/home/tarek
*.xml, *.xq*
Find contents...
tarek
.8pecxstudios
.adobe
.ant
.anydesk
.apt-src
.asciidoc
.assaultcube
.bluefish
.bluej
.cache
.choosenim
Editor
file
OK
1 : 1
Departements.xml
departements
departement
nom
masters
I..
master
module
ED
Protocoles
BDA
Sécurité
departement
nom
masters
M..
master
module
Stats
Analyse
Méthodes Numériques
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>
Total Time: 2045.33 ms
Query Info
Command:
CHECK Departements.xml
Result:
Database 'Departements' created in 2003.52 ms.
Time needed: 2045.33 ms
22 MB

BaseX


file [Departements] - BaseX 8.4.2
Database Editor View Visualization Options Help
Find
Find...
Results
Editor
file
departements
departement
nom masters
master
module
ED
Protocoles
BDA
Sécurité
Stats
Analyse
Méthodes Numériques
Algèbre
<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>
Total Time: 2045.33 ms
Query Info
Command:
CHECK Departements.xml
Result:
Database 'Departements' created in 2003.52 ms.
Time needed: 2045.33 ms
22 MB

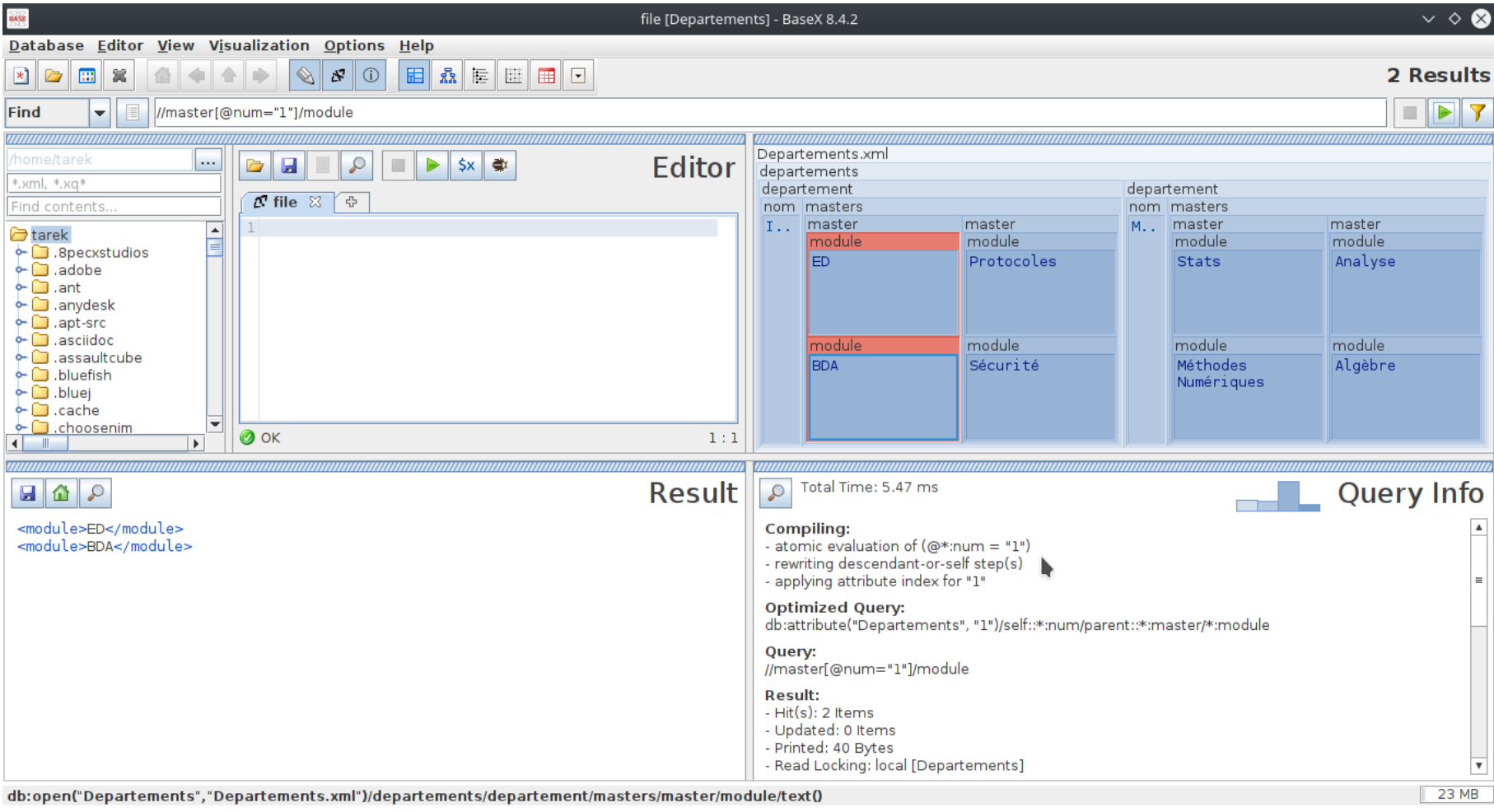
BaseX

- A Native XML DBMS
  - GUI Components:
    - **Commands and XPath,**
    - **Xquery queries,**
    - **Visualization of selected elements (query result),**
    - **Query result content,**
    - **Query stats.**

BaseX

- An XML Native DBMS
  - Example of an XPath path:
  - Display the modules taught in the master's program with the number: 1
    - //master[@num="1"]/module

BaseX


file [Departements] - BaseX 8.4.2
Database Editor View Visualization Options Help
2 Results
Find
//master[@num="1"]/module
Editor
file
OK
1 : 1
Departements.xml
departements
departement
nom masters
I..
master
module
ED
module
BDA
master
module
Protocoles
module
Sécurité
departement
nom masters
M..
master
module
Stats
module
Méthodes Numériques
master
module
Analyse
module
Algèbre
Result
<module>ED</module>
<module>BDA</module>
Total Time: 5.47 ms
Query Info
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

Content

- Uses of XML
- Models
- XML and Databases
- Architectures
- Example: BaseX
- **XQuery**

XQuery

- XQuery is
  - A query language for XML databases,
  - It can extract elements and attributes,
  - It also allows ordering and aggregation operations,
  - It can be used to extract data for processing or to perform transformations.

XQuery

- XQuery syntax
  - Based on XPath
  - FLWOR model (read FLOWER)
    - For set_of_elements
    - Let link_variables_to_elements
    - Where conditions
    - Order by criteria_for_ordering
    - Return elements_to_return
  - XQuery is case-sensitive (x and X are considered different)

XQuery

- XQuery example (tested on: https://www.videlibri.de/cgi-bin/xidelcgi)
  - <persons>
    - <person>
      - <name>Benameur</name><firstname>Ameur</firstname><age>25</age>
    - </person>
    - <person>
      - <name>Benhmed</name><firstname>Ahmed</firstname><age>19</age>
    - </person>
    - <person>
      - <name>Benomar</name><firstname>Omar</firstname><age>31</age>
    - </person>
    - <person>
      - <name>Benreda</name><firstname>Reda</firstname><age>34</age>
    - </person>
  - </persons>

XQuery

- Query
  - for $x in /persons/person
  - return $x
- Response
  - <person>
    - <name>Benameur</name><firstname>Ameur</firstname><age>25</age>
  - </person>
  - <person>
    - <name>Benhmed</name><firstname>Ahmed</firstname><age>19</age>
  - </person>
  - <person>
    - <name>Benomar</name><firstname>Omar</firstname><age>31</age>
  - </person>
  - <person>
    - <name>Benreda</name><firstname>Reda</firstname><age>34</age>
  - </person>

XQuery

- Query
  - for $x in /persons/person
  - where $x/age = 25
  - return $x/name
- Response
  - <name>Benameur</name>

XQuery

- Query
  - for $x in /persons/person
  - where $x/age = 25
  - return data($x/name)
- Response
  - Benameur

XQuery

- Query
  - for $x in /persons/person
  - order by $x/age
  - return $x/name
- Response
  - <name>Benhmed</name>
  - <name>Benameur</name>
  - <name>Benomar</name>
  - <name>Benreda</name>

XQuery

- Query
  - for $x in /persons/person
  - where $x/age > 25
  - order by $x/first name
  - return $x
- Response
  - <person>
    - <name>Benomar</name><firstname>Omar</firstname><age>31</age>
  - </person>
  - <person>
    - <name>Benreda</name><firstname>Reda</firstname><age>34</age>
  - </person>

XQuery

- Query
  - for $x in /persons/person
  - where $x/age > 25
  - order by $x/first name
  - return $x
- Response
  - <person>
    - <name>Benomar</name><firstname>Omar</firstname><age>31</age>
  - </person>
  - <person>
    - <name>Benreda</name><firstname>Reda</firstname><age>34</age>
  - </person>

XQuery

- Query
  - for $x in /persons/person
  - where $x/age != 25
  - return {data($x/name)}
- Response
  - Benhmed
  - Benomar
  - Benreda

XQuery

- Query
  - for $x at $i in /personnes/personne
  - return <li id="{$i}">{data($x/nom)} {data($x/prenom)}
- Response
  - <li id="1">Benameur Ameur
  - <li id="2">Benhmed Ahmed
  - <li id="3">Benomar Omar
  - <li id="4">Benreda Reda

XQuery

- Query
  - for $x in /persons/person
  - return
  - if ($x/age < 30)
    - then <young>{data($x/name)}</young>
    - else <old>{data($x/name)}</old>
- Response
  - <young>Benameur</young>
  - <young>Benhmed</young>
  - <old>Benomar</old>
  - <old>Benreda</old>

XQuery

- Request
  - for $x in (1 to 5)
  - return {$x}
- Answer
  - 1
  - 2
  - 3
  - 4
  - 5

University of Jijel
Faculty of Exact Sciences and Computer Science
Department of Computer Science
L3 – Computer Systems

Semi-Structured Data
Chapter 5
XML and Databases

Tarek Boutefara
t_boutefara@univ-jijel.dz
2025/2026