

Le langage XQuery Update Facility (XQUF)

Pr. Sidi Mohammed Benslimane
École Supérieure en Informatique
08 Mai 1945 – Sidi Bel Abbès –

s.benslimane@esi-sba.dz

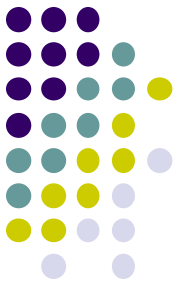


XQuery Update Facility (XQUF)



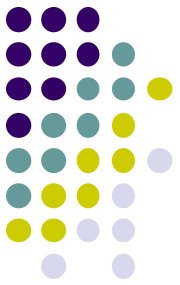
- Le langage XQuery Update Facility (XQUF) est une recommandation du W3C (Juin 2009)
- XQuery Update Facility 3.0 (January 2017)
- Il propose de nouvelles requêtes qui permettent de modifier les données de la base XML :
 - Insertion de nouveaux *nœuds dans la base XML*.
 - Suppression de *nœuds de la base XML*.
 - Modification de *nœuds dans la base XML*.
- Ces requêtes modifient toutes sortes de nœuds dans l'arbre XML concerné : éléments, attributs, textes, commentaries, etc.

Utilisation de BaseX



- En TP, nous utiliserons BaseX. Il y a également eXist. Tous deux sont gratuits et open source.
- Le logiciel BaseX propose :
 - Interrogation à l'aide de XQuery (y compris version 3),
 - Modification à l'aide de XQuery Update Facility (XQUF),
 - Une interface utilisateur complète.
- La documentation complète est disponible au format pdf sur la page: ***<http://docs.basex.org/wiki/Documentation>***

Interface de BaseX



C:/Program Files (x86)/BaseX/es16.xq - BaseX 8.5.3

Database Editor View Visualization Options Help

Command Command...

2 Results

C:/Program Files (x86)/BaseX/

*.xml, *.xq*

Find contents...

BaseX

- bin
- data
- etc
- ico
- lib
- repo
- webapp
- .basex (670 Bytes)
- .basexgui (4 KB)
- .basexhome (0 Bytes)
- adr.xml (604 Bytes)
- BaseX.exe (79 KB)

bib2.xml albums.xml ventes.xml es19.xq es18.xq es17.xq es16.xq

```
1 for $b in doc("bib.xml")//book
2 return element livre {
3   attribute titre {$b/@title},
4   for $a in $b/author return
5   for $p in doc("adr.xml")//person
6   where $a/la = $p/name
7   return <author>
8   {$a/la, $p/institution}
9   </author>}
```

OK 9 : 20

Result

```
<livre titre="Comprendre XSLT">
  <author>
    <la>Amann</la>
    <institution>CNAM</institution>
  </author>
  <author>
    <la>Rigaux</la>
    <institution>CNRS</institution>
  </author>
</livre>
<livre titre="Spatial Databases">
  <author>
    <la>Rigaux</la>
    <institution>CNRS</institution>
  </author>
```

Time needed: 47.48 ms

Query Info

Total Time: 45.48 ms

Compiling:

- pre-evaluating doc("bib.xml")
- rewriting descendant-or-self step(s)
- converting descendant::*:book to child steps
- pre-evaluating doc("adr.xml")
- rewriting descendant-or-self step(s)
- converting descendant::*:person to child steps
- swapping operands: (*:name = \$a_1/*:la)
- applying text index for \$a_1/*:la
- rewriting where clause(s)
- flattening nested GFLWOR expression: \$p_2

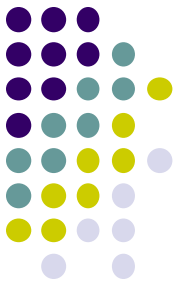
Optimized Query:

```
for $b_0 in document-node ("bib.xml")/*:bib/*:book return element livre { ((attribute titre { ($b_0/@*:title) }, for $a_1 in $b_0/*:author for $p_2 in db:text("adr", $a_1/*:la)/parent::*:name/parent::*:person return element author { (($a_1/*:la, $p_2/*:institution)) } ) }
```

26 MB

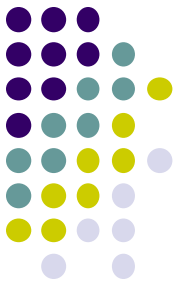
FR 23:11 30/03/2017

Interface graphique (suite)



- L'interface est composée de plusieurs panneaux :
 - Une barre pour saisir directement des recherches texte ou des requêtes XQuery ou des commandes sur la base,
 - Un éditeur de requête XQuery,
 - Une vue tabulaire de la base XML,
 - Les résultats de la requête, qu'on peut aussi afficher graphiquement selon le type des données,
 - Des statistiques sur l'exécution de la requête.
 - Elle est documentée sur la page:
http://docs.basex.org/wiki/Graphical_User_Interface
- En fait, c'est tellement complet qu'on ne pourra faire que quelques manipulations de découverte en TP.

Insertion de nœuds



Syntaxe

`insert ("node" | "nodes") SourceExpr TargetChoice TargetExpr`

SourceExpr décrit un nouveau *nœud* (élément ou attribut) qui doit être inséré dans

TargetExpr désigne les emplacements cibles pour l'insertion.

TargetChoice ::= `((as ("first" | "last")?) into) | after | before`

Important : l'expression après le **into** doit désigner un élément unique. Pour systématiser l'insertion sur plusieurs éléments, il faut faire une boucle.

Insertion d'éléments: as first

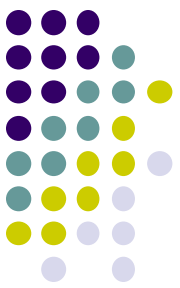


```
insert node <movie title="Before Night Falls"/>  
as first into doc("movies.xml")/movies
```

```
<movies>  
  <movie title="The Big Lebowski">  
    <director>E. Cohen</director>  
    <director>J. Cohen</director>  
  </movie>  
  <movie title="All President's Men">  
    <director>A. J. Pakula</director>  
  </movie>  
</movies>
```

```
<movies>  
  <movie title="Before Night Falls"/>  
  <movie title="The Big Lebowski">  
    <director>E. Cohen</director>  
    <director>J. Cohen</director>  
  </movie>  
  <movie title="All President's Men">  
    <director>A. J. Pakula</director>  
  </movie>  
</movies>
```

Insertion d'éléments: before



```
insert node <movie title="Before Night Falls"/>  
before doc("movies.xml")//movie[director="A. J. Pakula"]
```

```
<movies>  
  <movie title="The Big Lebowski">  
    <director>E. Cohen</director>  
    <director>J. Cohen</director>  
  </movie>  
  <movie title="All President's Men">  
    <director>A. J. Pakula</director>  
  </movie>  
</movies>
```

```
<movies>  
  <movie title="The Big Lebowski">  
    <director>E. Cohen</director>  
    <director>J. Cohen</director>  
  </movie>  
  <movie title="Before Night Falls"/>  
  <movie title="All President's Men">  
    <director>A. J. Pakula</director>  
  </movie>  
</movies>
```


Insertion sur plusieurs éléments



➤ Il faut simplement utiliser une expression FLWOR.

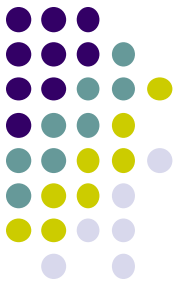
➤ Exemple

Ajouter l'élément `<achat date="2015-12-24"/>` dans tous les albums à partir du n°10

```
for $album in /albums/album
where $album/@numero >= 10
return insert node <achat date="2015-12-24"/> into $album
```

L'astuce est de mettre le insert en tant que return.

Insertion d'attributs



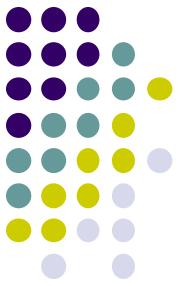
Exemple

➤ attribut *editeur* pour l'album n°1 dans albums.xml

```
insert node attribute editeur {'Casterman'}  
into /albums/album[@numero="1" and @serie="Tintin"]
```

Important : l'expression **into** doit désigner un emplacement unique.

Suppression d'éléments: delete



Syntaxe

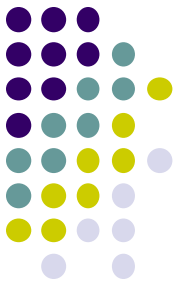
`delete ("node" | "nodes") TargetExpr`

```
delete node doc("movies.xml")//movie[director="E. Cohen"]
```

```
<movies>
  <movie title="The Big Lebowski">
    <director>E. Cohen</director>
    <director>J. Cohen</director>
  </movie>
  <movie title="All President's Men">
    <director>A. J. Pakula</director>
  </movie>
</movies>
```

```
<movies>
  <movie title="All President's Men">
    <director>A. J. Pakula</director>
  </movie>
</movies>
```

Suppression d'éléments ou d'attributs



```
delete node /albums/album[@numero="4"]
```

- Supprimer l'album n°4,

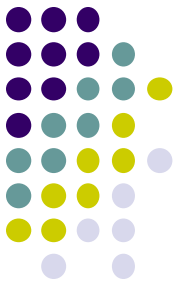
```
delete node /albums/album[date/mois="janvier"]
```

- Supprimer tous les albums parus en janvier,

```
delete node /albums/album[date/annee>1950]/@numero
```

- Supprimer l'attribut numero des albums parus après 1950.

Remplacement d'éléments ou d'attributs



Syntaxe

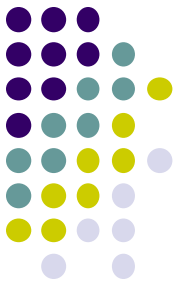
replace "value of node" TargetExpr "with" ExprSingle

Exemple

➤ remplacer le contenu de l'élément <titre> de l'album n°1

```
replace value of node /albums/album[@numero=1 and @serie="Tintin"]/titre  
with "nouveau titre"
```

Remplacement d'éléments ou d'attributs



- Il est aussi possible de remplacer un élément par autre chose.

Exemple

```
for $v in doc("catalogue.xml")//product
return
  replace value of node $v/price with $v/price * 1.1
```

- Augmenter les prix des produits d'un catalogue de 10%

Remplacement d'éléments ou d'attributs



```
replace node doc("movies.xml")//movie[1]
with doc("documentaries.xml")//documentary[2]
```

```
<movies>
  <movie title="The Big Lebowski">
    <director>E. Cohen</director>
    <director>J. Cohen</director>
  </movie>
  <movie title="All President's Men">
    <director>A. J. Pakula</director>
  </movie>
</movies>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<documentaries>
  <documentary title="The Corporation"/>
  <documentary title="Enron, the smartest guys in the room"/>
</documentaries>
```

```
<movies>
  <documentary title="Enron, the smartest guys in the room"/>
  <movie title="All President's Men">
    <director>A. J. Pakula</director>
  </movie>
</movies>
```

Renommer des éléments



Syntaxe

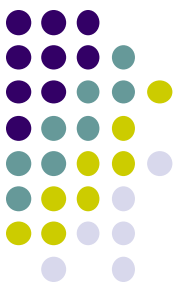
`rename "node" TargetExpr "as" NewNameExpr`

```
rename node doc("bib.xml")/bib/book[1]/author[1]  
as "principal-author"
```

- Renommer le premier élément *author* du premier élément *book* en *principal-author*

Contrairement aux insert, replace et rename, un delete peut concerner plusieurs nodes.

Renommer des éléments

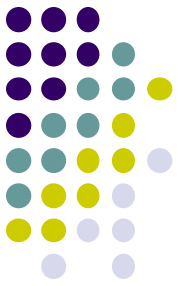


```
rename node doc("movies.xml")//movie[2]/director[1]  
as "main-director"
```

```
<movies>  
  <movie title="The Big Lebowski">  
    <director>E. Cohen</director>  
    <director>J. Cohen</director>  
  </movie>  
  <movie title="All President's Men">  
    <director>A. J. Pakula</director>  
  </movie>  
</movies>
```

```
<movies>  
  <movie title="The Big Lebowski">  
    <director>E. Cohen</director>  
    <director>J. Cohen</director>  
  </movie>  
  <movie title="All President's Men">  
    <main-director>A. J. Pakula</main-director>  
  </movie>  
</movies>
```

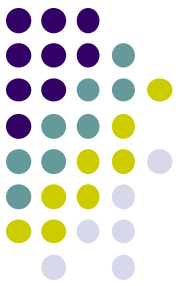
Implémentation & liens



❑ Implémentation

- **BaseX**, (Open Source Database for XML Documents) <http://basex.org/>
- **eXist**, (Open Source Database for XML Documents) <http://exist.sourceforge.net/>
- **Galax**, implémentation en Ocaml : <http://www.galaxquery.org/>
- **Qizx/open**, interpréteur open source écrit en Java : <http://www.qizx.com/>
- **Saxon**, processeur open source XQuery et XSLT <http://www.saxonica.com/>
- **MonetDB?** (Open Source Database System)
<http://monetdb.cwi.nl/projects/monetdb/XQuery/index.html>

Références bibliographiques



Titre :	BASES DE DONNEES XQuery pour interroger des donnees XML
Titre original :	ELelements du langage et mise en oeuvre cours et exercices corriges
Type de document :	texte imprimé
Auteurs :	<u>JACQUES LE MAITRE</u> , Auteur ; <u>EMMANAUL BRUNO</u> , Auteur
Editeur :	<u>ellipses</u>
Année de publication :	2013
ISBN/ISSN/EAN :	978-2-7298-8348-5
Langues :	Français (<i>fre</i>)

Titre :	XML cours et exercices : modélisation.schéma et DTD.design patterns.XSLT.DOM .relaxNG.XPath.SOAP.XQuery.XSL-FO.SVG.exist/2e EDITION
Type de document :	texte imprimé
Auteurs :	<u>ELEXANDRE BRILLANT</u> , Auteur
Editeur :	<u>EYROLLES</u>
Année de publication :	2010
ISBN/ISSN/EAN :	978-2-212-12691-4
Langues :	Français (<i>fre</i>)

Titre :	MODELISATION XML
Type de document :	texte imprimé
Auteurs :	<u>ANTOINE LONJON</u> , Auteur ; <u>JEAN-JACQUES THOMASSON</u> , Auteur
Editeur :	<u>EYROLLES</u>
Année de publication :	2006
ISBN/ISSN/EAN :	978-2-212-11521-5
Langues :	Français (<i>fre</i>)