

Proyecto LND: Tecnologías XML

La base de datos XML realizada para este proyecto se centra en una discográfica y los distintos álbumes que esta tiene a la venta.

Esta incluye desde los grupos que interpretan cada uno de los álbumes, el precio que alcanza en las tiendas hasta el número de ventas que estos álbumes han tenido desde su lanzamiento. También incluye el tipo de formato en el que está disponible cada álbum, para así cubrir las necesidades de los clientes que decidan comprar un álbum, las canciones que cada álbum posee, la duración total sumando todas las canciones, los integrantes que forman cada grupo, la fecha en la que el álbum fue publicado, los géneros musicales de cada uno e información del estado actual de cada grupo (activo o inactivo).

Canciones del álbum cuyo nombre es “Master of Puppets”

The screenshot shows an XML document on the left and an 'Evaluar XPath' dialog box on the right. The XML document is a list of songs under a 'canciones' root element. The dialog box has a text input field for the XPath expression.

```
1 <canciones>
2   <cancion> Battery</cancion>
3   <cancion> Master of Puppets</cancion>
4   <cancion> The Thing That Should Not Be</cancion>
5   <cancion> Welcome Home (Sanitarium)</cancion>
6   <cancion> Disposable Heroes</cancion>
7   <cancion> Leper Messiah</cancion>
8   <cancion> Orion</cancion>
9   <cancion> Damage, Inc.</cancion>
10 </canciones>
```

Evaluar XPath

XPath:

`/albumes/album[@nombre="Master of Puppets"]/canciones`

OK Cancel

Álbumes que poseen más de dos géneros musicales

The screenshot shows an XML document on the left and an 'Evaluar XPath' dialog box on the right. The XML document represents the album 'Appetite for Destruction' by Guns n' Roses, including its genres and band members. The dialog box contains an XPath expression to find albums with more than two genres.

```
1 <album nombre="Appetite for Destruction">
2   <grupo> Guns n' Roses</grupo>
3   <generos>
4     <genero> Hard Rock</genero>
5     <genero> Heavy Metal</genero>
6     <genero> Sleaze Rock</genero>
7   </generos>
8   <integrantes>
9     <integrante> Axl Rose</integrante>
10    <integrante> Slash</integrante>
11    <integrante> Duff McKagan</integrante>
12    <integrante> Izzy Stradlin</integrante>
13    <integrante> Steven Adler</integrante>
14  </integrantes>
15  <canciones>
16    <cancion> Welcome to the Jungle</cancion>
17    <cancion> It's So Easy</cancion>
18    <cancion> Nightrain</cancion>
19    <cancion> Out Ta Get Me</cancion>
20    <cancion> Mr. Brownstone</cancion>
21    <cancion> Paradise City</cancion>
22    <cancion> My Michelle</cancion>
23    <cancion> Think About You</cancion>
24    <cancion> Sweet Child o' Mine</cancion>
25    <cancion> You're Crazy</cancion>
26    <cancion> Anything Goes</cancion>
27  </canciones>
28 </album>
```

Evaluar XPath

XPath:

`/discografica/album[count(generos/genero)>2]`

OK Cancel

Álbumes a partir del tercer puesto sin incluirse él

The screenshot shows an XML document on the left and an 'Evaluar XPath' dialog box on the right. The XML document represents the album 'Master of Puppets' by Metallica, including its genres and band members. The dialog box contains an XPath expression to find albums starting from the third position.

```
1 <album nombre="Master of Puppets">
2   <grupo> Metallica</grupo>
3   <generos>
4     <genero> Heavy Metal</genero>
5     <genero> Trash Metal</genero>
6   </generos>
7   <integrantes>
8     <integrante> James Hetfield</integrante>
9     <integrante> Lars Ulrich</integrante>
10    <integrante> Kirk Hammet</integrante>
11    <integrante> Cliff Burton</integrante>
12  </integrantes>
13  <canciones>
14    <cancion> Battery</cancion>
15    <cancion> Master of Puppets</cancion>
16    <cancion> The Thing That Should Not Be</cancion>
17    <cancion> Welcome Home (Sanitarium)</cancion>
18    <cancion> Disposable Heroes</cancion>
19    <cancion> Leper Messiah</cancion>
20    <cancion> Orion</cancion>
21    <cancion> Damage, Inc.</cancion>
22  </canciones>
23 </album>
```

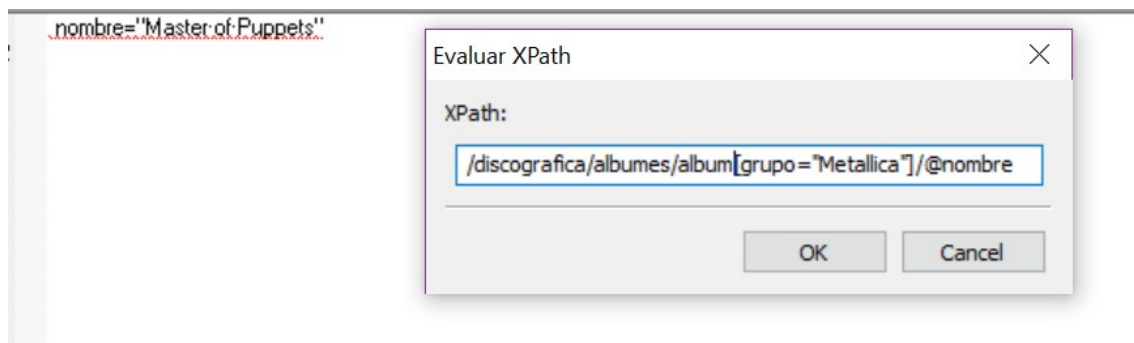
Evaluar XPath

XPath:

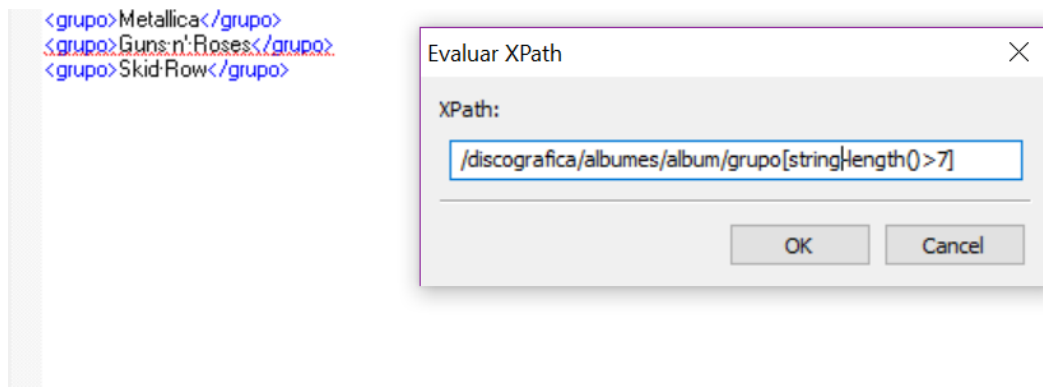
`/discografica/album[position()<3]`

OK Cancel

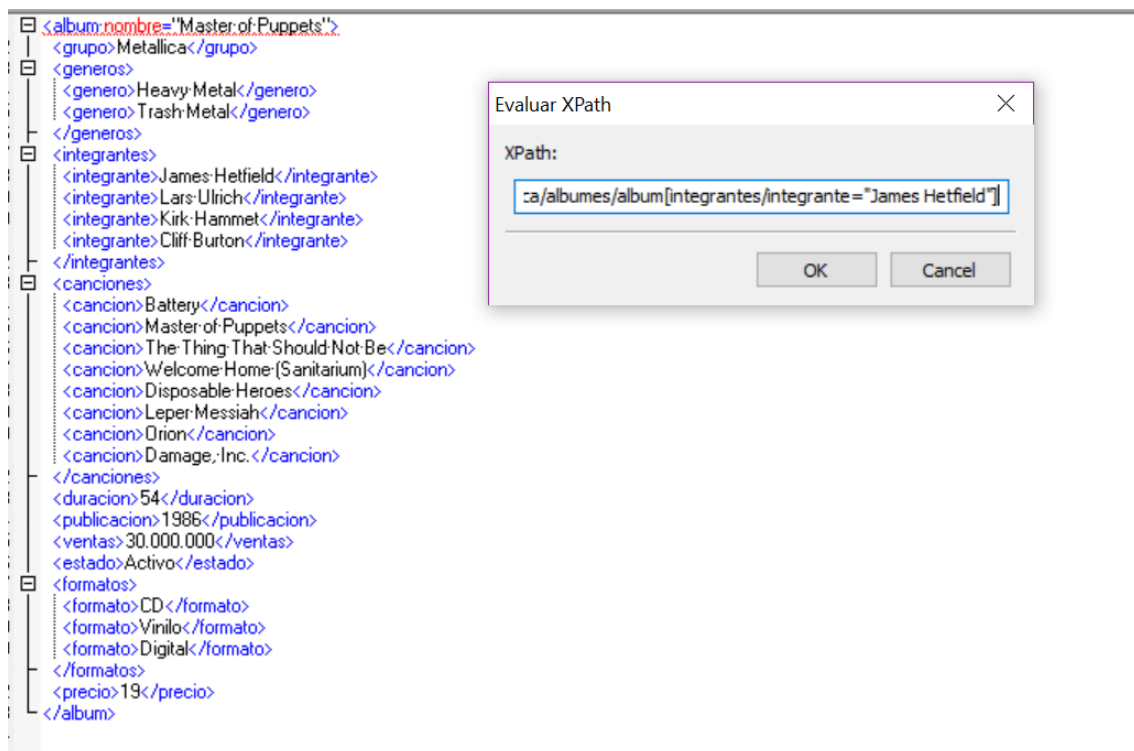
Nombre del álbum cuyo grupo es “Metallica”



Grupos cuyo nombre posea más de 7 caracteres



Álbum el cual el nombre de uno de los integrantes del grupo que lo interpreta es “James Hetfield”



Todos los datos de los álbumes cuyo intérprete esté inactivo

Editor

```

1 for $x in doc('musica.xml')/discografica/albumes/
  album
2 where $x/estado = 'Inactivo'
3 return $x

```

Resultado

```

<cancion>The Threat</cancion>
<cancion>Quicksand Jesus</cancion>
<cancion>Psycho Love</cancion>
<cancion>Get the Fuck Out</cancion>
<cancion>Livin on a Chain Gang</cancion>
<cancion>Creepshow</cancion>
<cancion>In a Darkened Room</cancion>
<cancion>Riot Act</cancion>
<cancion>Mudkicker</cancion>
<cancion>Wasted Time</cancion>
</canciones>
<duracion>50</duracion>
<publicacion>1991-08-06</publicacion>
<ventas>8000000</ventas>
<estado>Inactivo</estado>
<formatos>
  <formato>CD</formato>

```

Nombres de los álbumes ordenados alfabéticamente

The screenshot displays the BaseX GUI with three main panels:

- Editor:** Shows the XML document structure for 'musica.xml'. The content is:


```
<tr>{
  for $x in doc('musica.xml')/discografica/albumes/
    album/grupo/text()
  order by $x ascending
}
return <td>{$x}</td>
}</tr>
```
- Resultado:** Displays the query result as a table with columns: <tr>, <td>, <td>, <td>, <td>, <td>. The rows represent the groups: Guns n Roses, Metallica, Pantera, and Skid Row.
- Información de la Consulta:** Provides details about the query execution, including the total time (1.29 ms) and the compiled query plan.

Número de integrantes que posee cada grupo

C:/Program Files (x86)/BaseX/ ...

*.xml, *.xq*

Buscar contenidos...

BaseX

bin

data

etc

ico

lib

repo

webapp

.basex (664 b)

.basexgui (2328 b)

.basexhome (0 b)

BaseX.exe (79 kB)

BaseX.jar (3780 kB)

CHANGELOG (31 kB)

helado.xml (203 b)

LICENSE (1524 b)

readme.txt (1569 b)

receta.xml (421 b)

uninst.exe (90 kB)

Editor

```
1 <tr>{
2   for $x in distinct-values(doc('musica.xml')/
3     discografica/albumes/album/grupo)
4   let $y := count(//album[grupo=$x]/integrantes/
5     integrante)
6   return <td>{$x}, {$y}</td>
7 }
8 </tr>
```

OK

6 : 18

musica.xml

discografica

albumes

album

g. gener.. canciones dura.. vent..

g. g. can. can. publ. esta..

can. can. publ. esta..

integrantes

int. int. can. can. fo. fo. fo.

int. int. can. can. CD

int. int. can. can. precio

album

grupo integrantes du. pub..

generos in. in. inte.. 53

g. g. g. inte. ven. est.

canciones

c. c. canc. canci. c. formos f.

c. c. can. canci. form.

c. c. can. ca. ca. precio

album

g. generos canciones dur.. publ..

gen.. g. c. c. canci. 57

gen.. c. c. canci. ven. est.

integrantes

integ. int. can. can. formos fo.

integ. int. ca. ca. canci. form..

integ. int. canci. canci. precio

Resultado

```
<tr>
<td>Metallica, 4</td>
<td>Guns n Roses, 5</td>
<td>Skid Row, 5</td>
<td>AC DC, 5</td>
<td>Pantera, 5</td>
</tr>
```

Tiempo Total: 2.33 ms

Información de la Consulta

Compilando:

- pre-evaluate doc("musica.xml") to document-node()
- pre-evaluate root() to document-node()
- rewrite descendant-or-self step(s)
- inline \$y_1

Consulta Optimizada:

element tr { (for \$x_0 in distinct-values(db:open-pre("musica",0)/discografica/albumes/album/grupo) return element td { (\$x_0, "", count(db:open-pre("musica",0)/descendant::album((grupo = \$x_0)/integrantes/integrante))) }) }

Número de letras que posee el nombre de cada grupo

C:/Program Files (x86)/BaseX/ ...

*.xml, *.xq*

Buscar contenidos...

BaseX

bin

data

etc

ico

lib

repo

webapp

.basex (664 b)

.basexgui (2328 b)

.basexhome (0 b)

BaseX.exe (79 kB)

BaseX.jar (3780 kB)

CHANGELOG (31 kB)

helado.xml (203 b)

LICENSE (1524 b)

readme.txt (1569 b)

receta.xml (421 b)

uninst.exe (90 kB)

Editor

```
1 <tr>{
2   for $x in doc('musica.xml')/discografica/albumes/
3     album/grupo/text()
4   let $y := string-length($x)
5   return <td>{$x} ---- {$y}</td>
6 }
7 </tr>
```

OK

6 : 21

musica.xml

discografica

albumes

album

g. gener.. canciones dura.. vent..

g. g. can. can. publ. esta..

can. can. publ. esta..

integrantes

int. int. can. can. fo. fo. fo.

int. int. can. can. CD

int. int. can. can. precio

album

grupo integrantes du. pub..

generos in. in. inte.. 53

g. g. g. inte. ven. est.

canciones

c. c. canc. canci. c. formos f.

c. c. can. canci. form.

c. c. can. ca. ca. precio

album

g. generos canciones dur.. publ..

gen.. g. c. c. canci. 57

gen.. c. c. canci. ven. est.

integrantes

integ. int. can. can. formos fo.

integ. int. ca. ca. canci. form..

integ. int. canci. canci. precio

Resultado

```
<tr>
<td>Metallica ---- 9</td>
<td>Guns n Roses ---- 12</td>
<td>Skid Row ---- 8</td>
<td>AC DC ---- 5</td>
<td>Pantera ---- 7</td>
</tr>
```

Tiempo Total: 1.57 ms

Información de la Consulta

Compilando:

- pre-evaluate doc("musica.xml") to document-node()
- inline \$y_1

Consulta Optimizada:

element tr { (for \$x_0 in db:open-pre("musica",0)/discografica/albumes/album/grupo/text() return element td { (\$x_0, " ---- ", string-length(\$x_0)) }) }

Consulta:

<tr>{ for \$x in doc('musica.xml')/discografica/albumes/album/grupo/text() let \$v := string-length(\$x)

El año de publicación de cada álbum

C:/Program Files (x86)/BaseX/ ...

*.xml, *.xq*

Buscar contenidos...

BaseX

- bin
- data
- etc
- ico
- lib
- repo
- webapp
- .baseX (664 b)
- .baseXgui (2328 b)
- .baseXhome (0 b)
- BaseX.exe (79 kB)
- BaseX.jar (3780 kB)
- CHANGELOG (31 kB)
- helado.xml (203 b)
- LICENSE (1524 b)
- readme.txt (1569 b)
- receta.xml (421 b)
- uninst.exe (90 kB)

Editor

```
1 <tr>{
2   for $x in doc('musica.xml')/discografica/albumes/
3     album
4   let $y := year-from-date($x/publicacion)
5
6   return <td>{data($x/@nombre)} ---- { $y}</td>
7 }
8 </tr>{
```

OK 8 : 6

musica.xml

albumes

album	album	album
g.. gener.. canciones dura.. vent..	gru.. integrantes	gr.. gene.. canciones durac.. ventas
g.. g.. can.. can..	in.. in.. int..	gen.. c.. c.. ca.. publi.. esta..
can.. can..	publ.. esta..	integ.. in.. canci.. ca.. formatos
integ.. int.. can.. can..	ge.. integr..	integ.. in.. canci.. ca.. formatos
int.. int.. can.. can..	can.. can..	integ.. c.. c.. ca.. fo.. fo.. for..
int.. int.. can.. can..	CD	int.. int.. canci.. ca.. formato for..
precio	ca.. ca.. c.. c..	ca.. ca.. canci.. precio
album	album	album
grupo	integrantes	g.. gene.. canciones dur.. publi..
generos	in.. in.. inte.. 53	gen.. g.. c.. c.. canci.. 57
g.. g.. g..	ven.. est..	gen.. c.. canci.. ven.. est..
integ..	integ..	gen.. c.. canci..
canciones	duraci.. formatos	c.. c.. canci.. formatos
c.. c.. canci.. canci.. c..	form.. f..	integ.. integ.. canci.. formatos
c.. c.. canci.. canci..	form..	integ.. int.. ca.. ca.. canci.. form.. fo..
c.. c.. canci.. ca.. ca..	precio	integ.. int.. canci.. form.. ..
estado	ventas	integ.. integ.. cancion precio

Resultado

```
<tr>
<td>Master of Puppets ---- 1986</td>
<td>Appetite for Destruction ---- 1987</td>
<td>Slave to the Grind ---- 1991</td>
<td>Back in Black ---- 1980</td>
<td>Cowboys from Hell ---- 1990</td>
</tr>
```

Tiempo Total: 1.48 ms

Información de la Consulta

Compilando:

- pre-evaluate doc("musica.xml") to document-node()
- inline \$y_1

Consulta Optimizada:

element tr { (for \$x_0 in db:open-pre("musica",0)/discografica/albumes/album return element td { (data(\$x_0/@nombre), " ---- ", year-from-date(\$x_0/publicacion)) }) }

Consulta:

tr { for \$x in doc('musica.xml')/discografica/albumes/album let \$y := year-from-date(\$x/publicacion)

El álbum más caro

C:/Program Files (x86)/BaseX/ ...

*.xml, *.xq*

Buscar contenidos...

BaseX

- bin
- data
- etc
- ico
- lib
- repo
- webapp
- .baseX (664 b)
- .baseXgui (2328 b)
- .baseXhome (0 b)
- BaseX.exe (79 kB)
- BaseX.jar (3780 kB)
- CHANGELOG (31 kB)
- helado.xml (203 b)
- LICENSE (1524 b)
- readme.txt (1569 b)
- receta.xml (421 b)
- uninst.exe (90 kB)

Editor

```
1 <tr>{
2   let $y := max(/discografica/albumes/album/precio)
3
4   for $x in doc('musica.xml')/discografica/albumes/
5     album
6   where $x/precio = $y
7
8   return <td>{data($x/@nombre)} ---- {data($y)}</td>
9 }
10 </tr>
```

OK 8 : 35

musica.xml

discografica

albumes

album	album	album
g.. gener.. canciones dura.. vent..	gru.. integrantes	gr.. gene.. canciones durac.. ventas
g.. g.. can.. can..	in.. in.. int..	gen.. c.. c.. ca.. publi.. esta..
can.. can..	publ.. esta..	integ.. in.. canci.. ca.. formatos
integ.. int.. can.. can..	ge.. integr..	integ.. in.. canci.. ca.. formatos
int.. int.. can.. can..	can.. can..	integ.. c.. c.. ca.. fo.. fo.. for..
int.. int.. can.. can..	CD	int.. int.. canci.. ca.. formato for..
precio	ca.. ca.. c.. c..	ca.. ca.. canci.. precio
album	album	album
grupo	integrantes	g.. gene.. canciones dur.. publi..
generos	in.. in.. inte.. 53	gen.. g.. c.. c.. canci.. 57
g.. g.. g..	ven.. est..	gen.. c.. canci.. ven.. est..
integ..	integ..	gen.. c.. canci..
canciones	duraci.. formatos	c.. c.. canci.. formatos
c.. c.. canci.. canci.. c..	form.. f..	integ.. integ.. canci.. formatos
c.. c.. canci.. canci..	form..	integ.. int.. ca.. ca.. canci.. form.. fo..
c.. c.. canci.. ca.. ca..	precio	integ.. int.. canci.. form.. ..
estado	ventas	integ.. integ.. cancion precio

Resultado

```
<tr>
<td>Slave to the Grind ---- 25</td>
</tr>
```

Tiempo Total: 1.91 ms

Información de la Consulta

Compilando:


- pre-evaluate root() to document-node()
- pre-evaluate doc("musica.xml") to document-node()
- rewrite where clause(s)

Consulta Optimizada:

Principio

Resultado

```
<album nombre="EjemploPrincipio">
  <grupo>Metallica</grupo>
  <generos>
    <genero>Heavy Metal</genero>
    <genero>Trash Metal</genero>
  </generos>
  <integrantes>
    <integrante>James Hetfield</integrante>
    <integrante>Lars Ulrich</integrante>
    <integrante>Kirk Hammet</integrante>
    <integrante>Cliff Burton</integrante>
  </integrantes>
</album>
```

 Tiempo Total: 9.78 ms

Información de la Consulta

Compilando:

- pre-evaluate doc("musica.xml") to document-node()

Consulta Optimizada:

```
insert node element album { (attribute nombre {"EjemploPrincipio"}), element grupo {"(Metallica)"}, element generos { (element genero {"(Heavy Metal)"}, element genero {"(Trash Metal)"}), element integrantes { (element integrante {"(James Hetfield)"}, element integrante {"(Lars Ulrich)"}, element integrante {"(Kirk Hammett)"}, element integrante {"(Cliff Burton)"}), element canciones { (element cancion {"(Battery)"}, element cancion {"(Master Of Puppets)"}, element cancion {"(The Thing That Should Not Be)"}, element cancion {"(Welcome Home)"}, element cancion {"(Disposable Heroes)"}, element cancion {"(Leper Messiah)"}, element cancion {"(Orion)"}, element cancion {"(Damage Inc)"} ) }, element duracion {"(54)"}, element publicacion {"(1986-02-03)"}, element ventas {"(30000000)"} }
```

Medio

C:\Program Files (x86)\BaseX/ ...

*.xml, *.xq*

Buscar contenidos...

BaseX

bin

data

etc

ico

lib

repo

webapp

baseX (664 b)

baseXgui (2328 b)

baseXhome (0 b)

baseX.exe (79 kB)

BaseX.jar (3780 kB)

CHANGELOG (31 kB)

helado.xml (203 b)

LICENSE (1524 b)

readme.txt (1569 b)

receta.xml (421 b)

uninst.exe (90 kB)

Editor

```
17 <cancion>The thing that should not be</cancion>
18 <cancion>Welcome Home</cancion>
19 <cancion>Disposable Heroes</cancion>
20 <cancion>Leper Messiah</cancion>
21 <cancion>Orion</cancion>
22 <cancion>Damage Inc</cancion>
23 </canciones>
24 <duracion>56</duracion>
25 <publicacion>1977-02-03</publicacion>
26 <ventas>30000000</ventas>
27 <estado>Activo</estado>
28 <formatos>
29 <formato>CD</formato>
30 <formato>Vinilo</formato>
31 <formato>Digital</formato>
32 </formatos>
33 <precio>20</precio>
34 </album>
35 before doc('musica.xml')//album[3]
```

OK 35 : 26

Resultado

```
<estado>Activo</estado>
<formatos>
  <formato>CD</formato>
  <formato>Vinilo</formato>
  <formato>Digital</formato>
</formatos>
<precio>15</precio>
</album>
<album nombre="EjemploMedio">
  <grupo>Metallica</grupo>
  <generos>
    <genero>Heavy Metal</genero>
    <genero>Trash Metal</genero>
  </generos>
  <integrantes>
    <integrante>James Hetfield</integrante>
    <integrante>Lars Ulrich</integrante>
    <integrante>Kirk Hammet</integrante>
    <integrante>Cliff Burton</integrante>
  </integrantes>
```

musica.xml

discoqrafica

album

grupo	integrantes	dur.	ven.
generos	int.. int..	pub..	est..
g.. g..	int.. int..	pub..	est..
canciones	ca.. ca.. ca.. ca..	f.. f.. fo..	
ca.. ca.. ca.. ca..			
		precio	

album

grupo	integrantes	dur.	pub.
generos	int.. i.. inte..	ven..	es..
g.. g..	int.. int..	ven..	es..
canciones	can.. can.. can.. c..	fo.. fo..	
can.. can.. can..			
		precio	

album

grupo	integrantes	dur.	pu.
generos	in.. in.. int..	ve..	es..
g.. g.. g..	int.. int..	ve..	es..
canciones	c.. c.. can.. c..	for.. f..	
c.. c.. can..			
		precio	

Información de la Consulta

Tiempo Total: 13.12 ms

Compilando:

- pre-evaluate doc("musica.xml") to document-node()

Consulta Optimizada:

insert node element album { (attribute nombre { ("EjemploMedio") }, element grupo { ("Metallica") }, element generos { (element genero { ("Heavy Metal") }, element genero { ("Trash Metal") }) }, element integrantes { (element integrante { ("James Hetfield") }, element integrante { ("Lars Ulrich") }, element integrante { ("Kirk Hammet") }, element integrante { ("Cliff Burton") }) }, element canciones { (element cancion { ("Battery") }, element cancion { ("Master Of Puppets") }, element cancion { ("The Thing That Should Not Be") }, element cancion { ("Welcome Home") }, element cancion { ("Disposable Heroes") }, element cancion { ("Leper Messiah") }, element cancion { ("Orion") }, element cancion { ("Damage Inc") }) }, element duracion { ("56") }, element publicacion { ("1977-02-03") }, element ventas { ("30000000") }, element estado { ("Activo") }, element formatos { (element formato { ("CD") }, element formato { ("Vinilo") }, element formato { ("Digital") }) }, element precio { ("20") }) } into db:open-pre("musica".0)/descendant-or-self::node()/album[position() = 3]

Consulta:

insert node <album nombre="EjemploMedio"> <grupo>Metallica</grupo> <generos> <genero>Heavy Metal</genero> <genero>Trash Metal</genero> </generos> <integrantes> <integrante>James Hetfield</integrante> <integrante>Lars Ulrich</integrante> <integrante>Kirk Hammet</integrante> <integrante>

Final

C:\Program Files (x86)\BaseX/ ...

*.xml, *.xq*

Buscar contenidos...

BaseX

- bin
- data
- etc
- ico
- lib
- repo
- webapp
 - .baseX (664 b)
 - .baseXgui (2328 b)
 - .baseXhome (0 b)
 - BaseX.exe (79 kB)
 - BaseX.jar (3780 kB)
 - CHANGELOG (31 kB)
 - helado.xml (203 b)
 - LICENSE (1524 b)
 - readme.txt (1569 b)
 - receta.xml (421 b)
 - uninst.exe (90 kB)

Editor

fichero

```

17 < Cancion>the thing that should not be</
    Cancion>
18 < Cancion>Welcome Home</Cancion>
19 < Cancion>Disposable Heroes</Cancion>
20 < Cancion>Leper Messiah</Cancion>
21 < Cancion>Orion</Cancion>
22 < Cancion>Damage Inc</Cancion>
23 </Canciones>
24 < duracion>54</duracion>
25 < publicacion>1986-02-03</publicacion>
26 < ventas>3000000</ventas>
27 < estado>Activo</estado>
28 < formatos>
29 < formato>CD</formato>
30 < formato>Vinilo</formato>
31 < formato>Digital</formato>
32 </formatos>
33 < precio>19</precio>
34 </album>
35 as last into doc('musica.xml')
```


OK

2: 32

Resultado

```
<formato>CD</formato>
<formato>Vinilo</formato>
<formato>Digital</formato>
</formatos>
<precio>17</precio>
</album>
</albumes>
</discografias>
<album nombre="EjemploFinal">
  <grupo>Metallica</grupo>
  <generos>
    <genero>Heavy Metal</genero>
```

musica.xml										album									
discografica																			
albumes																			
album					album					album					album				
grupo	integrantes				g.. generos	canciones				grupo	integrantes				dur..	vent..			
generos	inte..	inte..			gen.. g..	c.. c.. canc..				gene..	int.. i.. inte..				publ..	est..			
g.. g..	inte..	inte..			gen..	c.. c..	canc..			gen..	int..		inte..						
canciones					integrantes		canc..			canciones					formatos				
ca.. ca.. ca.. ca..					integr.. in..					c.. c.. c.. c.. c.. c..					fo.. fo.. fo..				
ca.. ca.. ca.. ca..					integ..	c.. c.. can..				CD					fo..				
du.. pub.. format.. p..					integ.. in..	cancion	can..			ca.. ca.. ca.. ca..					forma..				
54	fo.. fo..				dur.. vent..	formatos	p..								precio				
ven.. es..	CD				publ.. est..	fo.. fo.. fo..				album									
	formato					CD				grupo	integrantes				du.. pu..				
										generos	in.. in.. int..			57					
										g.. g.. g..				ve.. es..					
											integr..		int..						
album										canciones					formatos				
g.. gener..	canciones				durac..	format..				c.. c.. can.. can.. c..				for.. f..					
gene..	can.. can.. c.. c..				publi..	fo.. fo..													
integrantes	can.. can..	canci..			CD						can.. can..			for..					
in.. in.. int..					ventas	fo.. fo..				c.. c..	can.. can..								
	c.. c.. c.. c.. c..				esta..					c.. c..	can.. ca..								
integ..														precio					



Tiempo Total: 39.74 ms

Información de la Consulta

Compilando:

- pre-evaluate doc("musica.xml") to document-node()

Consulta Optimizada:

```
insert node element album { (attribute nombre { ("EjemploFinal") }, element grupo { ("Metallica") },
element generos { (element genero { ("Heavy Metal") }, element genero { ("Trash Metal") } ) }, element
integrantes { (element integrante { ("James Hetfield") }, element integrante { ("Lars Ulrich") }, element
integrante { ("Kirk Hammett") }, element integrante { ("Cliff Burton") } ) }, element canciones { (element
cancion { ("Battery") }, element cancion { ("Master Of Puppets") }, element cancion { ("The Thing That
Should Not Be") }, element cancion { ("Welcome Home") }, element cancion { ("Disposable Heroes") },
element cancion { ("Lepor Messiah") }, element cancion { ("Orion") }, element cancion { ("Damage Inc") }
) }, element duracion { ("54") }, element publicacion { ("1986-02-03") }, element ventas { ("300000000") },
element estado { ("Activo") }, element formatos { (element formato { ("CD") }, element formato { ("Vinilo") }
```

MODIFICAR

Valor

C:/Program Files (x86)/BaseX/...
*.xml, *.xq*
Buscar contenidos...

BaseX

bin
data
etc
ico
lib
repo
webapp
basex (664 b)
basexgui (2328 b)
basexhome (0 b)
BaseX.exe (79 kB)
BaseX.jar (3780 kB)
CHANGELOG (31 kB)
helado.xml (203 b)
LICENSE (1524 b)
readme.txt (1569 b)
receta.xml (421 b)
uninst.exe (90 kB)

Editor

1 replace value of node doc("musica.xml")//album[@nombre='EjemploPrincipio']/precio with 50

OK 1 : 90

musica.xml

albumes
album
grupo integran... grupo integran... grupo integra... album
gener... int... int... gener... int... int... gene... in... in... du... p...
g... g... int... int... g... g... int... int... g... g... in... in... v... es...
canciones canciones canciones format...
c... c... c... c... c... c... c... c... c... c... fo... f...
c... c... c... c... c... c... c... c... c... c... fo...
d... p... forma... d... p... forma... album
56 f... f... 54 f... f... g... ge... canciones
v... es... v... es... g... ge... canciones
form... form... integran... g... ge... canciones
album g... generos canciones du... ven...
ge... g... c... c... can... pu... es... integran... g... ge... canciones
ge... c... c... can... pu... es... integran... g... ge... canciones
integran... c... c... can... pu... es... integran... g... ge... canciones
i... i... in... ca... c... ca... f... f... f... du... ven... formatos p...
integ... in... ca... ca... precio pu... es... for... for... i... i... i... c... c... c... v... e... for...

Resultado

<cancion>urion</cancion>
<cancion>Damage Inc</cancion>
</canciones>
<duracion>56</duracion>
<publicacion>1977-02-03</publicacion>
<ventas>30000000</ventas>
<estado>Activo</estado>
<formatos>
<formato>CD</formato>
<formato>Vinilo</formato>
<formato>Digital</formato>
</formatos>
<precio>50</precio>
</album>
<album nombre="Master of Puppets">
<grupo>Metallica</grupo>

Compilando:
- pre-evaluate doc("musica.xml") to document-node()
- atomic evaluation of (@nombre = "EjemploPrincipio")
- rewrite descendant-or-self step(s)
Consulta Optimizada:
replace value of node db:open-pre("musica",0)/descendant::album[(@nombre = "EjemploPrincipio")]/precio with 50
Consulta:
replace value of node doc("musica.xml")//album[@nombre="EjemploPrincipio"]/precio with 50
Resultado:
- Impacto(s): 0 Items
- Actualizado: 3 Items
- Impreso: 0 b
- Bloqueo de lectura: (none)

Información de la Consulta

Tiempo Total: 42.48 ms

Completo

C:\Program Files (x86)\BaseX\

*.xml, *.xq*

Buscar contenidos...

BaseX

bin

data

etc

ico

lib

repo

webapp

.baseX (664 b)

.baseXgui (2328 b)

.baseXhome (0 b)

BaseX.exe (79 kB)

BaseX.jar (3780 kB)

CHANGELOG (31 kB)

helado.xml (203 b)

LICENSE (1524 b)

readme.txt (1569 b)

receta.xml (421 b)

uninst.exe (90 kB)

Editor

fichero*

```
1 replace node doc("musica.xml")//album[@nombre='
EjemploPrincipio'] with
2 <album nombre='EjemploPrincipio2'>
3 </album>
```

OK 3 : 5

musica.xml

discografica

albumes

album	g.. generos	canciones	grupo	integrantes	du.. p..	album	gru.. integrantes	grupo	integrantes
g.. g..	can.. canc..	gen.. g..	i.. i.. int..	53	gen..	int.. i.. int..	gen.. g..	i.. i.. int..	
int.. int..	can.. canc..	gen..	int..	v.. es..	gen..	int.. int..	gen.. g..	int..	
int.. int..	can.. canc..	canciones	can.. can..	format..	canc..	c.. can.. c..	canciones	can.. can..	
dur.. ven..	formatos p..	ca.. can..	fo.. fo..	forma..	ca.. ca..	ca.. ca..	can.. can..	can..	
publ.. est..	f.. f.. f..	c.. c..	ca.. ca..	precio	d.. p..	formatos	can.. can..	c.. c..	
album	g.. generos	canciones	grupo	integrantes	du.. pu..	41	f.. f.. f..	du.. pub..	format.. p..
g.. g..	can.. canc..	gene..	int.. i.. int..	ve.. es..	v.. e..	fo.. fo..	ven.. es..	for.. f..	
int.. int..	can.. canc..	gen..	int.. int..	album	g.. generos	canciones	dur.. vent..		
int.. int..	can.. canc..	can.. can..	ca.. c..	formatos	g.. ge..	c.. c.. c.. c..	publ.. esta..		
dur.. ven..	formatos p..	can.. can..	ca..	fo.. fo..	int.. int..	c.. c.. c.. c..	formatos		
publ.. est..	f.. f.. f..	c.. c.. c.. c..	c..	precio	i.. i.. i.. i..	c.. c.. c.. c..	fo.. fo.. fo..		

Resultado

Tiempo Total: 10.24 ms

Información de la Consulta

Compilando:

- pre-evaluate doc("musica.xml") to document-node()
- atomic evaluation of (@nombre = "EjemploPrincipio")
- rewrite descendant-or-self step(s)

Consulta Optimizada:

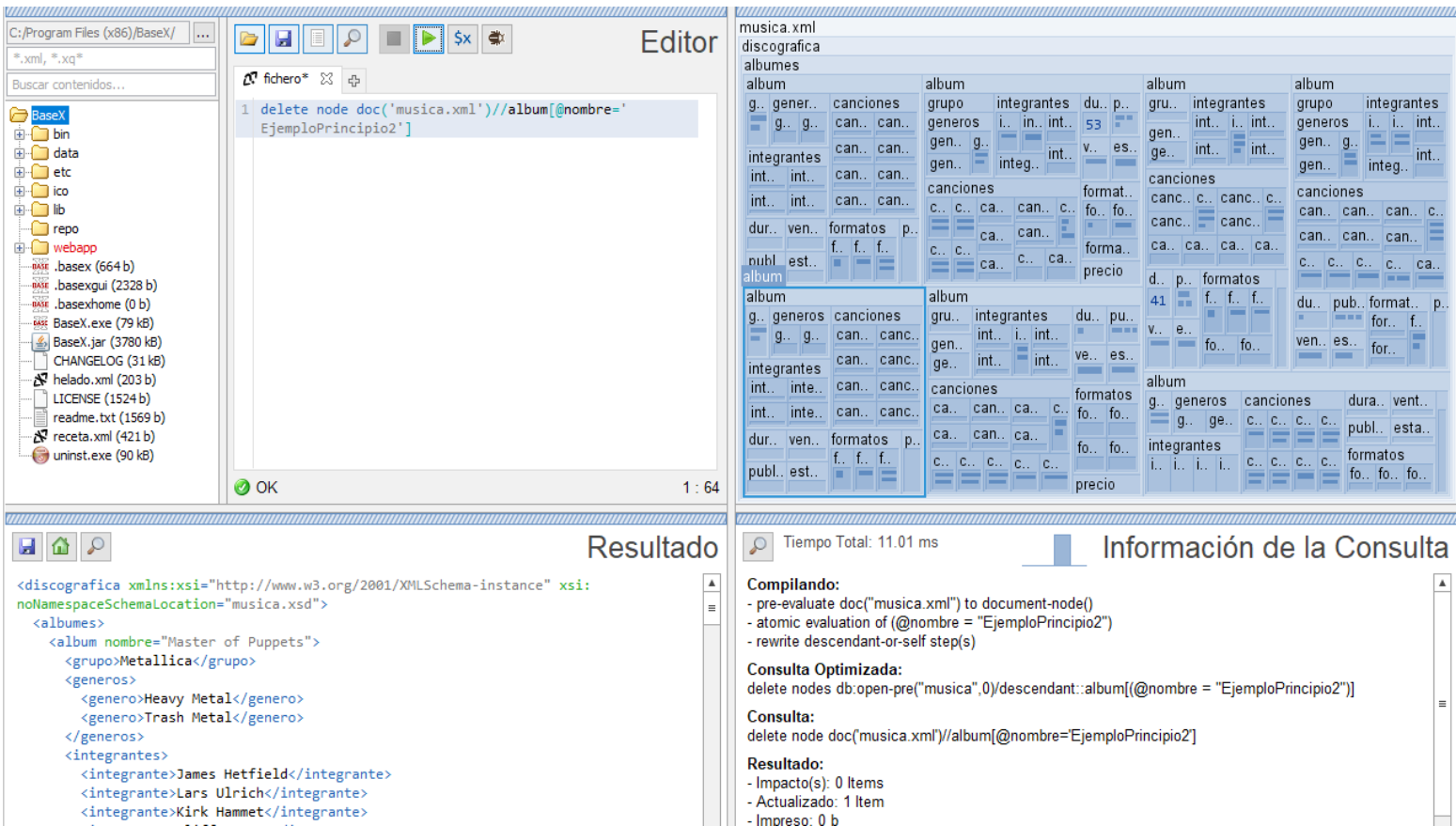
replace node db.open-pre("musica",0)/descendant::album[(@nombre = "EjemploPrincipio")] with element album { (attribute nombre { ("EjemploPrincipio2")) }

Consulta:

replace node doc("musica.xml")//album[@nombre='EjemploPrincipio'] with <album nombre='EjemploPrincipio2'>

```
<discografica xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
noNamespaceSchemaLocation="musica.xsd">
  <albumes>
    <album nombre="EjemploPrincipio2"/>
    <album nombre="Master of Puppets">
      <grupo>Metallica</grupo>
      <generos>
        <genero>Heavy Metal</genero>
        <genero>Trash Metal</genero>
      </generos>
    </album>
  </albumes>
</discografica>
```

Principio



Medio

[illegible]

Final

C:/Program Files (x86)/BaseX/...
*.xml, *.xq*
Buscar contenidos...

BaseX

bin
data
etc
ico
lib
repo
webapp
basex (664 b)
basexgui (2328 b)
basexhome (0 b)
BaseX.exe (79 kB)
BaseX.jar (3780 kB)
CHANGELOG (31 kB)
helado.xml (203 b)
LICENSE (1524 b)
readme.txt (1569 b)
receta.xml (421 b)
uninst.exe (90 kB)

Editor

1 delete node doc('musica.xml')//album[last()]

OK 1 : 45

musica.xml

albumes

album	album	album
g.. gener.. canciones dura.. vent..	gru.. integrantes	gr.. gene.. canciones durac.. ventas
g.. g.. can.. can..	in.. in.. int..	gen.. c.. c.. ca..
can.. can..	gen.. ge..	publi.. esta..
integrantes	integr..	integrant..
int.. int.. can.. can..	fo.. fo.. fo..	integ.. in.. canci..
int.. int.. can.. can..	CD	fo.. fo.. for..
precio	canciones	ca.. ca.. c.. c..
	ca.. ca..	int.. int.. canci..
album	grupo	integrantes
generos	in.. in.. inte..	53
g.. g.. g..	ven.. est..	canci.. c.. canci..
integr..		canci.. canci..
canciones	duraci.. formatos	g.. generos
c.. c.. canci.. canci.. c..	form.. f..	gen.. g.. c.. c.. canci..
canc.. canci..	public..	can.. ven.. est..
c.. c.. canci..	CD	gen.. c.. c.. canci..
ca.. ca..	ventas	integ.. int.. canci..
precio	estado	integ.. int.. canci..

Resultado

<cancion>Heresy</cancion>
<cancion>Cemetery Gates</cancion>
<cancion>Domination</cancion>
<cancion>Shattered</cancion>
<cancion>Clash with Reality</cancion>
<cancion>Medicine Man</cancion>

Tiempo Total: 7.8 ms

Información de la Consulta

Compilando:
- pre-evaluate docj("musica.xml") to document-node()

Consulta Optimizada:
delete nodes db:open-pre("musica",0)/descendant-or-self::node()/album[last()]

Consulta:

XSLT

Discografica

Album	Grupo	Generos	Integrantes	Precio
Master of Puppets	Metallica	Heavy Metal Trash Metal	James Hetfield Lars Ulrich Kirk Hammet Cliff Burton	19
Appetite for Destruction	Guns n Roses	Hard Rock Heavy Metal Sleaze Rock	Axl Rose Slash Duff McKagan Izzy Stradlin Steven Adler	15
Slave to the Grind	Skid Row	Heavy Metal	Sebastian Bach Rachel Bolan Dave Sabo Scotti Hill Rob Affuso	25
Back in Black	AC DC	Hard Rock	Brian Johnson Angus Young Malcolm Young Cliff Williams Phil Rudd	20
Cowboys from Hell	Pantera	Groove Metal Trash Metal Heavy Metal	Phill Anselmo Dimebag Darrell Rex Brown Vinnie Paul Terry Date	17

Grupos ordenados alfabeticamente

AC DC

Guns n Roses

Metallica

Pantera

Skid Row

Albumes ordenados por precio

Appetite for Destruction = 15

Cowboys from Hell = 17

Master of Puppets = 19

Back in Black = 20

Slave to the Grind = 25

Albumes con duracion mayor a 50 minutos en verde y menores en rojo

Nombre	Duracion
Master of Puppets	54
Appetite for Destruction	53
Slave to the Grind	50
Back in Black	41
Cowboys from Hell	57

Estado actual de los grupos

- Metallica = Activo
- Guns n Roses = Activo
- AC DC = Activo
- Pantera = Activo
- Skid Row = Inactivo