

Consultes de selecció avançades de eXist

Hèctor Manrique Prades

QP 2010/2011

28 de maig de 2011

Índex

1 Introducció i motiu del manual.....	2
2 D'on provenim?	2
3 Estructura de dades.....	2
4 Consultes	3
4.1 Selecció.....	4
4.2 Selecció avançada.....	5
5 Conclusions del manual i agraïments	8
6 Bibliografia.....	9

1 Introducció i motiu del manual

Aquest manual pretén ser un complement de referència per a la utilització de la base de dades eXist, complementant l'excel·lent realització del manual d'en Mutsuda, que en el seu manual explica perfectament com integrar la base de dades en una interfície Java i com comunicar la capa de domini amb la de dades de manera fàcil i ràpida.

En el seu manual no explica gaire per sobre com realitzar consultes, només les consultes bàsiques que es realitzen amb XPath i poc més, i qualsevol persona que es vulgui aventurar a fer servir eXist voldrà saber com més se li podrà treure tot el profit en aquesta base de dades que, encara que sembli que no, és molt potent i realment fàcil de fer servir.

Espero que aquest manual sigui de molt profit tant per als futurs estudiants de l'assignatura de la FIB Projecte de Xarxes de Computadors, on tenen una bona oportunitat d'aventurar-se en el món de les bases de dades en XML, com per a possibles persones que puguin trobar aquest manual buscant referències per eXist.

No s'expliquen les consultes per eliminar, actualitzar o inserir dades, doncs són consultes que ja s'expliquen amb el màxim detall possible al manual d'en Mutsuda. Només s'expliquen les consultes de selecció bàsiques per entrar en context i, tot seguit, les avançades, que tenen complicació i poden arribar a ser molt potents.

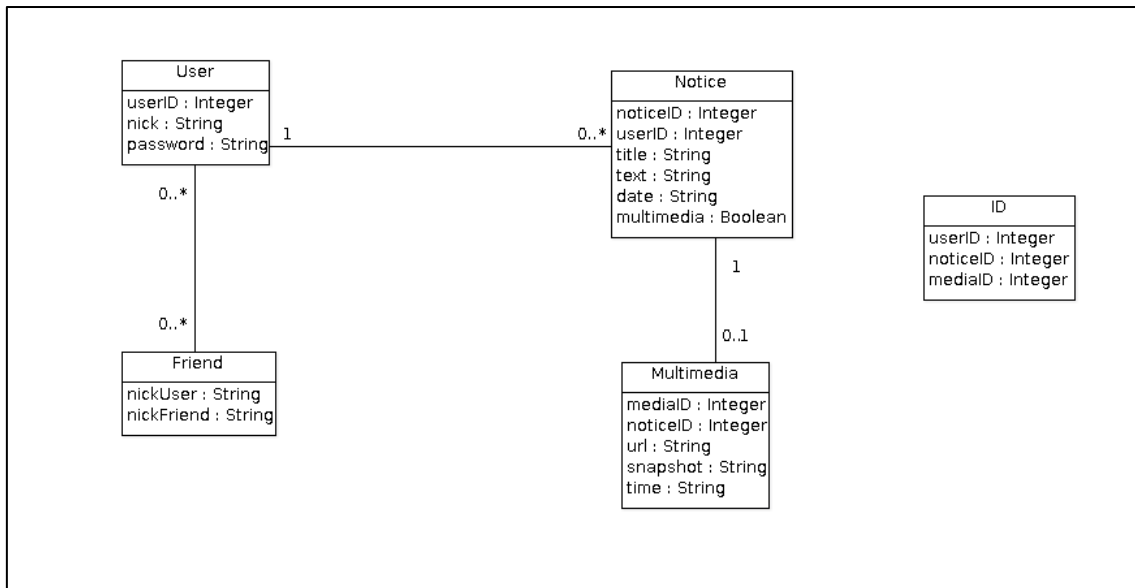
2 D'on provenim?

Aquest manual dona per suposat que ja es tenen nocions de com funciona la base de dades en qüestió. Aquí no vull explicar com configurar eXist ni com fer el controlador façana per comunicar-se de manera fàcil ni vull explicar les peculiaritats que comporta això. Si vols saber sobre això, recomano el manual d'en Mutsuda que, a dia d'avui, pots trobar en el següent enllaç: <http://mutsuda.llima.net/category/fib/pxc-fib/>.

En aquest manual pretenc donar una bona varietat d'exemples de consultes, tant les més fàcil com les que són una mica complexes, per tal d'introduir les XQuery una mica i que la persona que en vulgui aprendre tingui una petita referència de per on començar.

3 Estructura de dades

Per fer les consultes de manera uniforme, les faré totes sobre un exemple molt fàcil i entendre:



Bàsicament, tindrem usuaris que podran tenir tants amics com vulguin, i tantes notícies com vulguin. Aquestes notícies poden tenir contingut multimèdia si es desitja.

A més a més, hi ha un fitxer XML per controlar els ID d'usuaris, notícies i arxius multimèdia per simular el comportament de les Primary Key típiques en les bases de dades relacionals, i com que eXist això no ho suporta, estarà en els exemples, ja que s'usen en les diferents consultes, però no explicaré com actualitzar-ho quan toca, ja que són consultes molt similars a les que explicaré a continuació.

Per poder començar a treballar amb la base de dades es necessita inicialitzar els fitxers que es vulguin usar, per a que les XQuery els puguin trobar en qualsevol moment.

En aquest manual segueixo una nomenclatura bàsica: cada fitxer s'anomenarà com el que es vol guardar (User, Notices, Friends i Multimedia) i estarà inicialitzat amb, per exemple a la taula d'usuaris, <User></User>. Al manual d'en Mutsuda explica molt bé com realitzar això.

Ignorarem les dades dels ID dels elements que volem guardar, doncs la seva gestió es realitza en consultes separades i no influeix en el manual.

4 Consultes

A partir d'aquí aniré explicant les diferents consultes que es poden realitzar. Primer explicaré diferents consultes fàcils, realitzades amb XPath, però que és necessari per entendre amb profunditat les consultes de selecció avançades, fetes amb XQuery, i que es poden complicar més del que s'explicarà en el manual.

Per guardar les dades de manera entenedora es fa servir el tag `<noticesOf>`, per exemple en el cas de les notícies, per agrupar les notícies publicades per un cert usuari. Quan es veegin els exemples s'entendrà millor.

4.1 Selecció

Les consultes de selecció bàsiques es fan amb XPath, degut a que es fa una mena d'accés directe al fitxer i a la dada que es vol. Per exemple:

```
/Users/user/
```

Aquesta consulta retorna tots els usuaris del fitxer Users amb les seves característiques.

```
<user>
  <userID>1</userID>
  <nick>jordi</nick>
  <password>PXC</password>
</user>
<user>
  <userID>2</userID>
  <nick>hector</nick>
  <password>pass</password>
</user>
```

En canvi, si fem:

```
/Users/user[nick = "Jordi"]/userID/
```

Ja estem demanant exclusivament el ID d'un usuari en concret. Per fer una selecció concreta només cal indicar quin camp busques i amb quin valor. En aquest exemple, volem l'usuari que té per camp *nick* Jordi, i d'aquest usuari indiquem que només volem el camp *userID*.

A partir d'aquí es poden obtenir tot tipus de dades de manera simple a la base de dades. Però la cosa es complica quan volem que se'ns retorni conjunts de dades més restringits a un *“tot el que hi hagi sobre això”*. Aquest tipus de consultes les analitzarem en el següent apartat.

Veiem un altre exemple per veure més bé com funciona:

```
/Notices/noticesOf
```

Aquesta consulta retorna la següent informació:

```
<noticesOf>
  <userID>1</userID>
```

```
<notice>
  <noticeID>1</noticeID>
  <title>titol de prova</title>
  <text>text de prova</text>
  <data>22/04/2011</data>
  <multimedia>false</multimedia>
</notice>
</noticesOf>
<noticesOf>
  <userID>2</userID>
  <notice>
    <noticeID>2</noticeID>
    <title>titol diferent</title>
    <text>un altre noticia</text>
    <data>30/04/2011</data>
    <multimedia>false</multimedia>
  </notice>
</noticesOf>
```

4.2 Selecció avançada

Un cop vist com es fan les seleccions típiques en eXist, anem a veure com realitzar consultes més complexes on es retornen conjunts selectius de totes les dades que es consulten. Són les consultes pròpiament dites XQuery.

Anem a analitzar, per començar, una consulta senzilla que ens retornarà el ID de l'usuari que ha publicat una notícia:

```
for $notice in /Notices/noticesOf
where $notice/notice[noticeID = "1"]
return $notice/userID
```

En aquest exemple, demanem el ID de l'usuari que té la notícia amb identificador 1.

En la primera línia veiem com preparem totes les dades que volem analitzar. A la variable *\$notice* (la notació diu que ha d'anar precedida pel caràcter dòlar) li diem que volem totes les notícies de tots els usuaris. Aquesta part vindria a ser una primera actualització d'una consulta de selecció bàsica. Cal recordar que l'organització de la base de dades tenia el tag `<noticesOf>`, en el cas que tractem en aquesta ocasió, per

classificar i que, per tant, si fem aquesta selecció com la fem aquí retornarà totes les notícies de tots els usuaris.

En comptes de fer servir la clàusula *for*, també es pot fer servir *let* per definir un seguit de variables. La sintaxi seria idèntica.

Amb la clàusula *where* especifiquem el criteri de selecció sobre la variable que hem declarat. Podem observar que es pot substituir el *\$notice* per */Notices/noticesOf* i fer així una consulta de selecció bàsica. El format de la selecció s'explica en el punt anterior, simplement li indiquem que volem la notícia (o notícies si escollim per un altre camp) que té l'identificador 1.

Amb la clàusula *return* indiquem què és exactament el que volem retornar d'aquells camps que s'han seleccionat en la clàusula *where*. El retorn, en aquest cas, seria amb el mateix format que a les consultes de selecció bàsiques.

Una altra consulta que es pot realitzar seria la següent:

```
for $notice in /Notices/noticesOf
where $notice/userID="1"
return <notices>{$notice/notice}</notices>
```

En aquest exemple veiem dos canvis significatius respecte la consulta anterior: en la clàusula *where* i la clàusula *return*.

En el primer cas veiem com la selecció concreta es fa de manera més directa, dient directament que volem les notícies que tinguin el camp *userID* igual a 1. Això ens evita haver d'usar la sintaxi de XPath per fer la selecció concreta, agilitzant la programació i fent més fàcil la seva gestió en Java, on s'han d'escapar les cometes que pertanyen a la consulta i no és gens fàcil quan això ho sumes a la concatenació de variables que tens en el programa. Però això no ho tractaré en aquest manual.

En el segon cas veiem dues diferències: la presència de *tags* noves i la utilització dels caràcters { i }. Aquests canvis el que es fa és retornar allò que es vol retornar amb un *tag* propi, que és el que inserim. D'aquesta manera, tindrem tots els valors que volem retornar com a fill del *tag* que nosaltres hem afegit. Els caràcters { i } formen part de la notació de XQuery i són necessaris per dir-li que es vol el contingut d'aquell XPath. Així doncs, per cada node que es retorni se li afegeix el *tag* <notices>. En aquest cas, cada node que retornem són les notícies dels usuaris seleccionats. És a dir, el nostre *tag* s'afegeix al node arrel que es retorna, que en aquest cas només és 1 degut a que seleccionem segons l'identificador de l'usuari que és únic. Vegem l'exemple de retorn:

```
<notices>
  <notice>
    <noticeID>1</noticeID>
    <title>titol de prova</title>
    <text>text de prova</text>
    <data>22/04/2011</data>
    <multimedia>false</multimedia>
  </notice>
  <notice>
    <noticeID>4</noticeID>
    <title>Hola</title>
    <text>Prova</text>
    <data>223344</data>
    <multimedia>false</multimedia>
  </notice>
</notices>
```

Per últim, veurem com realitzar les consultes de selecció més complicades, que usen funcions dins de la pròpia consulta per seleccionar elements determinats. L'exemple és:

```
<Notices> {
for $notices in /Notices/noticesOf/notice
where $notices[contains(lower-case(text), "hola")]
or $notices[contains(lower-case(title), "titol")]
return $notices }
</Notices>
```

Primer de tot, es pot observar que els caràcters { i } ara engloben tot el contingut de la consulta. Això és perquè ara el *tag* propi el volem com a pare de tot el retorn de la consulta, ja que en aquest cas no tenim un sol bloc de dades retornat com en el cas anterior (baixem al nivell inferior de *noticesOf*, *notice*).

Un altre canvi significatiu és la utilització de la clàusula *or*. Remarcar que també es poden fer servir les clàusules *and* i *not*. Amb aquesta clàusula podem enllaçar les condicions del *where* com es vulgui per aconseguir una selecció més acurada d'entre totes les dades.

Per últim i el canvi més important, és la utilització de funcions. En aquest exemple s'usen dos, *contains* i *lower-case*. En ambdós casos funcionen com en Java, miren que el segon *String* estigui contingut en el primer i converteix totes les lletres majúscules a

minúscules del *String* que es demana. L'única diferència està en que pots passar com a paràmetres un *tag* concret que tinguis en el XML (ho passes sense les cometes, que indiquen que és un *String* i no un *tag*) i ell mateix ja accedeix al contingut d'aquell *tag* en concret que li has indicat, facilitant l'accés al contingut propi de la base de dades.

Per realitzar-ho, ja no es pot fer la selecció com es feia en la consulta anterior, ara és necessari fer-ho com en el primer exemple, és a dir, s'han de cridar les funcions amb els caràcters [i], que s'executaran amb el XPath que tinguin al darrere, en aquest exemple el que conté la variable *\$notices*. D'aquesta manera es poden realitzar, per exemple, cerques de contingut en la base de dades per trobar nodes concrets que es vulguin aconseguir. Vegem un exemple de com seria aquesta cerca:

```
<Notices>
  <notice>
    <noticeID>4</noticeID>
    <title>Hola</title>
    <text>Prova</text>
    <data>223344</data>
    <multimedia>false</multimedia>
  </notice>
</Notices>
```

5 Conclusions del manual i agraïments

Amb aquest petit manual es vol intentar explicar el funcionament de les consultes de selecció més complexes de Xquery, i facilitar així l'inici a les XQuery a les persones que així ho desitgin.

Gràcies a aquestes consultes podem obtenir resultats molt similars als que obtindríem amb consultes SQL, explotant tota la potència que té eXist i facilitant així la programació posterior que s'hagi de fer pel tractament d'aquestes dades, sigui amb la navegació per l'arbre XML que es retorni com si es vol fer una transformació amb XSLT.

He de donar les gràcies a en Mutsuda per fer un manual d'introducció tant detallat i precís per implementar aplicacions Java amb comunicació a una base de dades eXist, que em va ajudar molt per poder embarcar-me de manera fàcil en aquesta aventura d'aprenentatge i aconseguir uns molt bons resultats finalment.

6 Bibliografia

Manual d'en Mutsuda: <http://mutsuda.llima.net/category/fib/pxc-fib/>

Guies bàsiques d'eXist:

- http://exist.sourceforge.net/update_ext.html
- <http://www.w3schools.com/xquery/>

Web d'eXist: <http://exist.sourceforge.net/>