المختصر المفيد لتعلم

XQuery

إعداد

حسام كلحسن

تشرين الأول 2011

الـ XQuery بالنسبة للـ XML هي بمثابة الـ SQL لجداول قاعدة المعطيات الـ XQuery مصممة للاستعلام من بيانات الـ XML المخزنة ليس فقط في ملفات الـ XML بل قد تكون مخزنة في قاعدة معطيات الـ XQuery 1.0 والـ XPath 2.0 تشتركان بنفس الـ data model وتستخدمان نفس التوابع والعمليات بعض الأمثلة على كيفية استخدام الـ XQuery :

بفرض لدينا مستند الـ XML التالي :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>

<book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

حتى نصل إلى محتويات المستند السابق نستخدم التابع doc() كالتالي:

```
Doc("books.xml")
```

للتنقل على العناصر نستخدم تعابير الـ XPath ,مثلا:

```
Doc("books.xml")/bookstore/book/title
```

ونتيجة الاستعلام هي :

```
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
```

كما يمكننا استخدام الـ predicates :

```
Doc("books.xml")/bookstore/book[price>40]
```

ونتيجة الاستعلام هي :

```
<book category="WEB">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
```

كيفية الاستعلام من ملف الـ XML باستخدام FLOWER :

FLOWER هي تجميع لأوائل الكلمات التالية:

For ,Let ,Where ,Order by ,Return

لنعد إلى المستند الذي استخدمناه في الأمثلة السابقة وليكن لدينا تعبير الـ Xpath التالي:

```
doc("books.xml")/bookstore/book[price>30]/title
```

تعبير الـ FLOWER المكافئ له هو:

```
For $x in doc("books.xml")/bookstore/book
Where $x/price>30
return $x/title
```

ويمكن أن نستخدم الـ FLOWER لترتيب النتائج :

```
For $x in doc("books.xml")/bookstore/book
Where $x/price>30
Order by $x/title
return $x/title
```

قواعد الـ XQuery :

- XQuery حساسة لحالة الأحرف
- العناصر والخصائص في الـ XQuery هي متحولات ويجب أن تحقق قواعد التسمية في الـ XML
- الـ string يمكن أن تكون بين إشارات تنصيص مفردة أو مزدوجة
- يتم تعريف المتحولات باستخدام الـ $ مثلا $bookstore
- التعليقات تعرف كالتالي (:XQuery Comment:)

التعابير الشرطية :

نعبر عنها باستخدام "If-Then-Else" , مثال :

```
for $x in doc("books.xml")/bookstore/book
return if ($x/@category="CHILDREN")
then <child>{data($x/title)}</child>
else <adult>{data($x/title)}</adult>
```

الأقواس في شرط الـ if مطلوبة ويجب وجود الـ else ويمكن أن تكون فقط else()

المقارنات في الـ XQuery :

يمكن استخدام : =, !=, >, =>, <, =<

أو استخدام : eq, ne, lt, le, gt, ge

الفرق بينهما :

```
$bookstore//book/@q > 10
```

هذا التعبير يكون true إذاكانت أي خاصية q تحقق الشرط

```
$bookstore//book/@q gt 10
```

التعبير السابق يعيد true فقط إذاكانت خاصية واحدة q تحقق الشرط

فيمايلي نشرح من خلال الأمثلة قواعد استخدام تعابير الـ FLOWER :

- عبارة for :

```
for $x in (1 to 5)
return <test>{$x}</test>
```

والنتيجة هي :

```
<test>1</test>
<test>2</test>
<test>3</test>
```

```
<test>4</test>
<test>5</test>
```

ويمكن استخدام الكلمة المحجوزة at لعد النتائج ,مثلا:

```
for $x at $i in doc("books.xml")/bookstore/book/title
return <book>{$i}. {data($x)}</book>
```

والنتيجة هي:

```
<book>1. Everyday Italian</book>
<book>2. Harry Potter</book>
<book>3. XQuery Kick Start</book>
<book>4. Learning XML</book>
```

ويمكن أن نستخدم تعبير in عدة مرات في عبارة الـ for ,مثلا:

```
for $x in (10,20), $y in (100,200)
return <test>x={$x} and y={$y}</test>
```

والنتيجة هي:

```
<test>x=10 and y=100</test>
<test>x=10 and y=200</test>
<test>x=20 and y=100</test>
<test>x=20 and y=200</test>
```

- عبارة let: وتستخدم لإسناد قيمة لمتحول ما

```
let $x := (1 to 5)
return <test>{$x}</test>
```

والنتيجة هي :

```
<test>1 2 3 4 5</test>
```

- عبارة where : وتستخدم لوضع شرط معين على النتيجة

```
where $x/price>30 and $x/price<100
```

- عبارة order by : وتستخدم لترتيب النتيجة

```
for $x in doc("books.xml")/bookstore/book
order by $x/@category, $x/title
return $x/title
```

والنتيجة هي:

```
<title lang="en">Harry Potter</title>
<title lang="en">Everyday Italian</title>
<title lang="en">Learning XML</title>
<title lang="en">XQuery Kick Start</title>
```

- عبارة return : تحدد النتيجة المعادة (المراد إعادتها)

```
for $x in doc("books.xml")/bookstore/book
return $x/title
```

والنتيجة هي :

```
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
```

توابع الـ XQuery :

كما قلنا سابقا الـ XQuery 1.0 والـ XPath 2.0 تشتركان بنفس التوابع

يمكن استخدام التابع في أي مكان يظهر فيه التعبير وفيمايلي بعض الأمثلة :

يمكن استخدام التابع ضمن العنصر:

```
<name>{uppercase($booktitle)}</name>
```

أو ضمن الـ predicate :

```
doc("books.xml")/bookstore/book[substring(title,1,5)='Harry']
```

أو ضمن عبارة let :

```
let $name := (substring($booktitle,1,4))
```

توابع الـ XQuery المعرفة من قبل المستخدم :

لتعريف تابع نستخدم الصيغة التالية:

```
declare function prefix:function_name($parameter AS datatype)
AS returnDatatype
{
 ...function code here...
}
```

عند تعريف التابع يجب أن نراعي ما يلي :

- استخدام الكلمة المحجوزة declare function
- يجب أن نحدد الـ prefix لاسم التابع
- جسم التابع يجب أن يوضع بين قوسين

مثال:

```
declare function local:minPrice($p as xs:decimal?,$d as xs:decimal?)
AS xs:decimal?
{
let $disc := ($p * $d) div 100
return ($p - $disc)
}
```

تم بعون الله تعالى