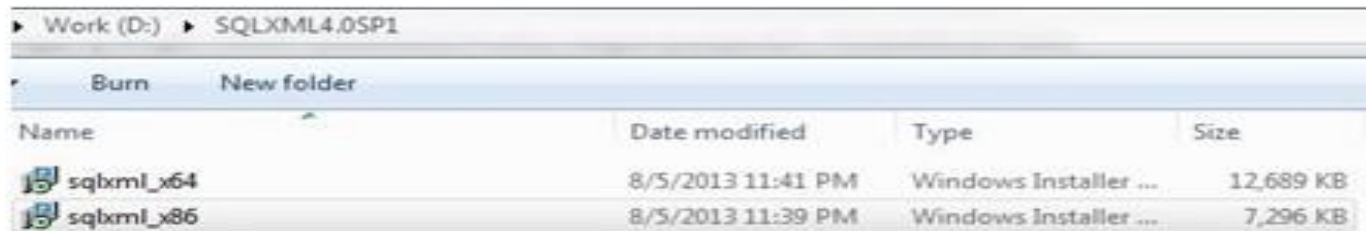


Xquery, Xpath

Cách cài đặt SQLXML 4.0 SP1

<http://www.microsoft.com/en-gb/download/details.aspx?id=30403>



Giới thiệu

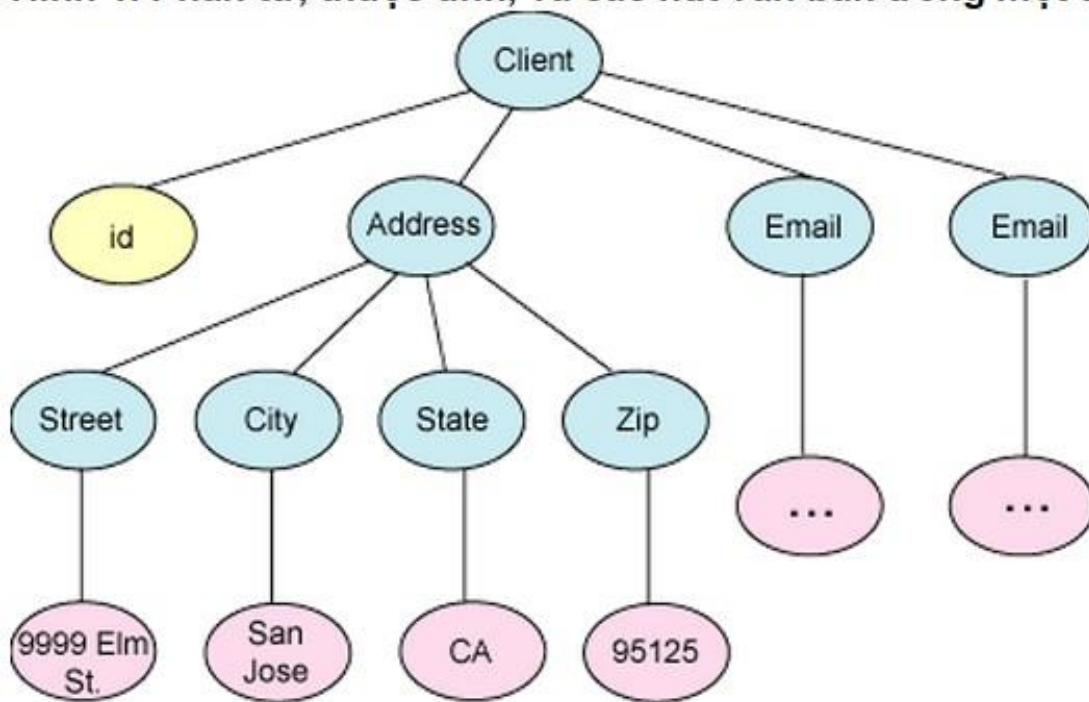
- Xpath và Xquery là hai ngôn ngữ có rất nhiều mặt giống nhau, hỗ trợ tìm kiếm thông tin trong tài liệu XML.
- Có thể xem Xpath là tập hợp con của Xquery.
- Xquery sử dụng Xpath như là một ngôn ngữ chính để định hướng tìm kiếm thay vì dùng đệ qui để duyệt cây.

XPath

- XML Path language (gọi tắt là Xpath) là một chuẩn để xử lý tài liệu XML (cũng như SQL là một chuẩn để làm việc với csdl)
- Dùng để xử lý nhiều kiểu truy vấn trong tài liệu XML
- Là ngôn ngữ rất phổ biến
- Tiết kiệm thời gian trích xuất dữ liệu

```
<Client id="123">  
  <Address>  
    <street>9999 Elm St.</street>  
    <city>San Jose</city>  
    <state>CA</state>  
    <zip>95141</zip>  
  </Address>  
  <email>anyemail@yahoo.com</email>  
  <email>anotheremail@yahoo.com</email>  
</Client>
```

Hình 1. Phần tử, thuộc tính, và các nút văn bản trong một tài liệu XML



Blue = element nodes; Yellow = attribute node; Pink = text nodes

Ví dụ

Thực thi Query
dưới đây:

Kết quả truy vấn trả về:

```
SELECT * FROM  
SINHVIEN
```

```
FOR XML RAW
```

```
<row MSSV="13520001" TENS="Nguyen Van An" SODT="0906762255" LOP="SE103.U32 " DIACHI="Thu Duc " />  
<row MSSV="13520002" TENS="Phan Tan Dat" SODT="0975672350" LOP="IE204.T21 " DIACHI="Quan 1 " />  
<row MSSV="13520003" TENS="Nguyen Anh Hai" SODT="0947578688" LOP="IE205.R12 " DIACHI="Quan 9 " />  
<row MSSV="13520004" TENS="Pham Tai" SODT="0956757869" LOP="IE202.A22 " DIACHI="Quan 1 " />  
<row MSSV="13520005" TENS="Le Thuy Hang" SODT="0976668688" LOP="SE304.E22 " DIACHI="Thu Duc " />  
<row MSSV="13520006" TENS="Ung Hong An" SODT="0957475898" LOP="IE208.F33 " DIACHI="Quan 2 " />
```

Ví dụ

Thực thi Query
dưới đây:

**SELECT * FROM
SINHVIEN**

**FOR XML RAW,
ELEMENTS**

Kết quả truy vấn trả về:

```
<row>
  <MSSV>13520003</MSSV>
  <TENSX>Nguyen Anh Hai</TENSX>
  <SODT>0947578688</SODT>
  <LOP>IE205.R12 </LOP>
  <DIACHI>Quan 9 </DIACHI>
</row>
<row>
  <MSSV>13520004</MSSV>
  <TENSX>Pham Tai</TENSX>
  <SODT>0956757869</SODT>
  <LOP>IE202.A22 </LOP>
  <DIACHI>Quan 1 </DIACHI>
</row>
```

Chú ý rằng XPath là *cấu trúc động*.

```
<Client id="123">
  <Address>
    <street>9999 Elm St.</street>
    <city>San Jose</city>
    <state>CA</state>
    <zip>95141</zip>
  </Address>
  <email>anyemail@yahoo.com</email>
  <email>anotheremail@yahoo.com</email>
</Client>
```

- **Duyệt các phần tử email:**
/Client/email
- **Duyệt phần tử email đầu tiên:** /Client/email[1]
- **Xác định một nút thuộc tính và giá trị:**
/Client[@id='123']/email[1]
- **Sử dụng dấu sao (*):**
/Client/*/city

- **//*** _Trả về tất cả các node trong dữ liệu.
- **//email** _Tìm phần tử email.
- **/Client/email[1]/text()** _Trả về dữ liệu kiểu text của email đầu tiên.
- **/Client/Address/*** _Chọn tất cả các con của Address.
- **/Client/data(@id)** _Trả về giá trị của thuộc tính id.

Nhận xét Xpath

- Biểu thức Xpath ngắn gọn, rõ ràng.
- Hiểu được các node phức tạp trong tài liệu XML và biết các mối quan hệ giữa chúng.
- Một hạn chế của Xpath là không cung cấp cách chuyển đổi tập kết quả trả về.
- Không thể sắp kết quả hiện thị tăng dần theo tên.

XQuery

- Làm được những gì Xpath đã làm.
- Phức tạp hơn so với Xpath.
- Sử dụng cú pháp pha trộn XML và Xpath.
- Làm được những điều Xpath không làm:
 - Sắp xếp kết quả của câu truy vấn hoặc chuyển chúng thành HTML, CSV, SQL, XML ...
 - Cung cấp tính năng biểu thức FLWOR.
 - Sử dụng hàm và đệ quy.
 - Diễn tả các phép nối.

Biểu thức FLWOR

- Biểu thức FLWOR dùng để liên kết các tiêu chí rút trích dữ liệu và chuyển đổi tập kết quả trả về của câu truy vấn.
- FLWOR là viết tắt của các từ for, let, where, order by và return.
- Bắt đầu bằng một biểu thức for hoặc let và kết thúc bằng một biểu thức return .

Biểu thức FLWOR

- **For**: Lặp trên một chuỗi đầu vào, liên kết lần lượt một biến với mỗi danh mục (item) đầu vào.
- **Let**: Khai báo một biến và gán cho nó một giá trị, có thể là một danh sách các mục.
- **Where**: Xác định điều kiện để lọc các kết quả truy vấn.
- **Order by**: Xác định trật tự sắp xếp kết quả.
- **Return**: Xác định kết quả được trả về.

```

<users>
  <user>
    <name>
      <first>Lola</first>
      <last>Solis</last>
    </name>
    <age>2</age>
  </user>
  <user>
    <name>
      <first>Nina</first>
      <last>Serafina</last>
    </name>
    <age>4</age>
    <visits>
      <first>2008-01-15</first>
      <last>2008-02-15</last>
    </visits>
  </user>
  <user>
    <name>
      <first>Tracy</first>
      <last>Keller</last>
    </name>
    <age>35</age>
  </user>
</users>

```

Ví dụ

Kết quả truy vấn trả về:

```

let $xml:= _XML from Listing 1_
for $user in $xml//user[age lt 18]
order by $user/name/last
return $user/name/last/text()

```

```

(: Result
   Serafina
   Solis
:.)

```

```
let $xml:= _XML from Listing 1_
return
  <ol>{
    for $user in $xml//user[age < 18]
    order by $user/name/last
    return <li>{$user/name/last/text()}</li>
  }</ol>
```

```
(: Result
  <ol><li>Serafina</li><li>Solis</li></ol>
:)
```

```
let $xml:= _XML from Listing 1_
return
  <ol>{
    for $user in $xml//user
    where $user/age < 18
    order by $user/name/last
    return <li>{$user/name/last/text()}</li>
  }</ol>
```

Sử dụng các hàm và đệ quy

- Xquery:
 - Cung cấp các hàm, các phép toán được xây dựng sẵn và cho phép định nghĩa các hàm riêng.
 - Hỗ trợ đệ quy: tiện lợi khi làm việc với XML (có thể chứa các node lồng nhau tùy ý).
- Định nghĩa hàm transform-names dùng để thay đổi tên các node trong bất kỳ tài liệu XML nào.

Ví dụ

(: Part 1 :)

```
define function transform-names($node as node()) as node() {  
  element{replace(name($node), "_", "-")} {  
    $node/text(), for $subnode in $node/* return transform-names($subnode)  
  }  
}
```

(: Part 2 :)

```
let $xml:=  
<item>  
  <item_type>book</item_type>  
  <contributors>  
    <author>  
      <first_name>Charles</first_name>  
      <last_name>Edward</last_name>  
      <home_address>  
        <home_street>206 S. Solomon St.</home_street>  
        <home_city>New Orleans</home_city>  
        <home_state>LA</home_state>  
        <home_zip>70119</home_zip>  
      </home_address>  
    </author>  
    <artist>  
      <last_name>Salinas</last_name>  
    </artist>  
  </contributors>  
</item>  
return transform-names($xml)
```


Ví dụ

```
(: Result
  <item>
    <item-type>book</item-type>
    <contributors>
      <author>
        <first-name>Charles</first-name>
        <last-name>Edward</last-name>
        <home-address>
          <home-street>206 S. Solomon St.</home-street>
          <home-city>New Orleans</home-city>
          <home-state>LA</home-state>
          <home-zip>70119</home-zip>
        </home-address>
      </author>
      <artist>
        <last-name>Salinas</last-name>
      </artist>
    </contributors>
  </item>
:)
```

Diễn tả các phép nối

- Xquery còn cung cấp phương tiện để nối các node XML trong một câu truy vấn (giống như diễn tả các phép nối bảng trong câu truy vấn của SQL).

Ví dụ

```
(: Part 1 :)
let $authors:=
  <authors>
    <author>
      <name>Harold Abelson</name>
      <books>
        <isbn>978-0-07-000422-1</isbn>
        <isbn>978-0-262-01063-4</isbn>
      </books>
    </author>
    <author>
      <name>Paul Graham</name>
      <books>
        <isbn>978-0-13-370875-2</isbn>
        <isbn>978-0-13-030552-7</isbn>
        <isbn>978-0-596-00662-4</isbn>
      </books>
    </author>
    <author>
      <name>Apostolos-Paul Refenes</name>
      <books>
        <isbn>978-0-471-94364-8</isbn>
        <isbn>978-981-02-2819-4</isbn>
      </books>
    </author>
  </authors>
```

(: Part 2 :)

let \$books:=

<books>

<book>

<title>Structure and Interpretation of Computer Programs</title>

<isbn>978-0-07-000422-1</isbn>

</book>

<book>

<title>Turtle Geometry</title>

<isbn>978-0-262-01063-4</isbn>

</book>

<book>

<title>ANSI Common LISP</title>

<isbn>978-0-13-370875-2</isbn>

</book>

<book>

<title>On LISP</title>

<isbn>978-0-13-030552-7</isbn>

</book>

<book>

<title>Hackers and Painters</title>

<isbn>978-0-596-00662-4</isbn>

</book>

<book>

<title>Neural Networks in the Capital Markets</title>

<isbn>978-0-471-94364-8</isbn>

</book>

</books>

Ví dụ

(: Part 3 :)

return

```
<books-complete-info>{  
  for $book in $books/*  
    for $author in $authors/*  
      where $book/isbn = $author//isbn  
        and (  
          contains($book/title, "LISP")  
          or contains($book/title, "Neural"))  
    order by $book/title  
  return <book>{$book/*, $author/name}</book>  
</books-complete-info>
```


Kết quả

```
<books-complete-info>
  <book>
    <title>ANSI Common LISP</title>
    <isbn>978-0-13-370875-2</isbn>
    <name>Paul Graham</name>
  </book>
  <book>
    <title>On LISP</title>
    <isbn>978-0-13-030552-7</isbn>
    <name>Paul Graham</name>
  </book>
  <book>
    <title>Neural Networks in the Capital Markets</title>
    <isbn>978-0-471-94364-8</isbn>
    <name>Apostolos-Paul Refenes</name>
  </book>
</books-complete-info>
```

Một số lệnh (phương thức) khác

- query() - (phương thức cho phép bạn thực hiện một Xquery)
_Cú pháp: DbObject.query('XQuery')
 - value() _Cú pháp: DbObject.value('XQuery', 'SqlType')
 - modify() _Cú pháp: DbObject.modify('XQuery')
- Trong đó: replace value of, Delete, Insert,...
- exist() _Cú pháp: DbObject.exist('XQuery')

Nhận xét

XQuery mở rộng XPath rất nhiều, nổi lên như là công cụ ưa thích khi bạn có các yêu cầu lựa chọn dữ liệu phức tạp hoặc bạn cần phải trả về các kết quả đã được sắp xếp, được định dạng đặc biệt, hoặc đã được chuyển đổi.