

Dữ liệu XML

- Đặt vấn đề
- Các khái niệm cơ bản trong XML
- DTD
- Truy vấn dữ liệu XML
- Truy vấn dữ liệu XML trong SQL Server
- Lược đồ XML
- Thảo luận

1

1

Một ví dụ về tài liệu HTML

```
<html>
  <head><title>Example</title></head>
  <body>
    <h1>This is an example of a page.</h1>
    <h2>Some information goes here.</h2>
  </body>
</html>
```

- HTML dùng để làm gì?
- Thẻ (tag) – đánh dấu

2

XML là gì?

XML (eXtensible Markup Language) là ngôn ngữ đánh dấu cho tài liệu dạng văn bản

XML cho phép tạo các thẻ để **mô tả (biểu diễn)** tài liệu và dữ liệu

3

3

Một ví dụ về XML

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in 10 Years</title>
</article>
```

- Các thẻ (Tags) được thêm vào tài liệu nhằm cung cấp thêm thông tin cho dữ liệu, tăng ngữ nghĩa cho dữ liệu
- Các thẻ XML cho người đọc về ngữ nghĩa của dữ liệu trong dữ liệu text - Người dùng dễ hiểu
- Có cấu trúc, các chương trình dễ đọc và viết

4

4

Một ví dụ khác về XML

... đây cũng là 1 tài liệu XML, tuy nhiên:

```
<t108>
  <x87>Gerhard Weikum</x87>
  <g10>The Web in 10 Years</g10>
</t108>
```

- Khó hiểu, tại sao?
- Ngữ nghĩa dữ liệu kém, tại sao?
- Có cấu trúc, các chương trình dễ đọc và viết

5

5

Một ví dụ tiếp về XML

... và đây lại là một ví dụ về tài liệu XML:

```
<data>
  ch37fhgks73j5mv9d63h5mgfkds8d9841gnsmcns983
</data>
```

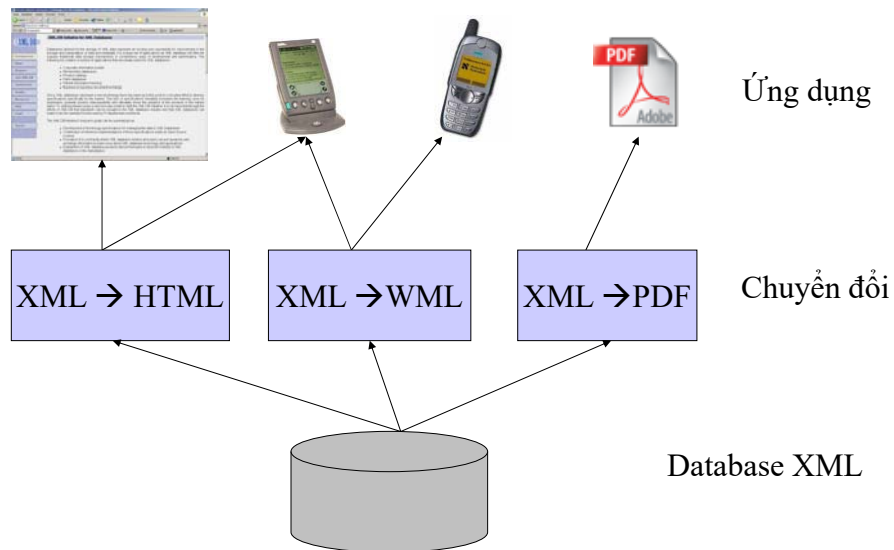
- Người dùng không thể hiểu
- Không có ngữ nghĩa dữ liệu
- Không có cấu trúc

Lợi ích thực sự mà XML mang lại phụ thuộc nhiều vào người thiết kế ứng dụng

6

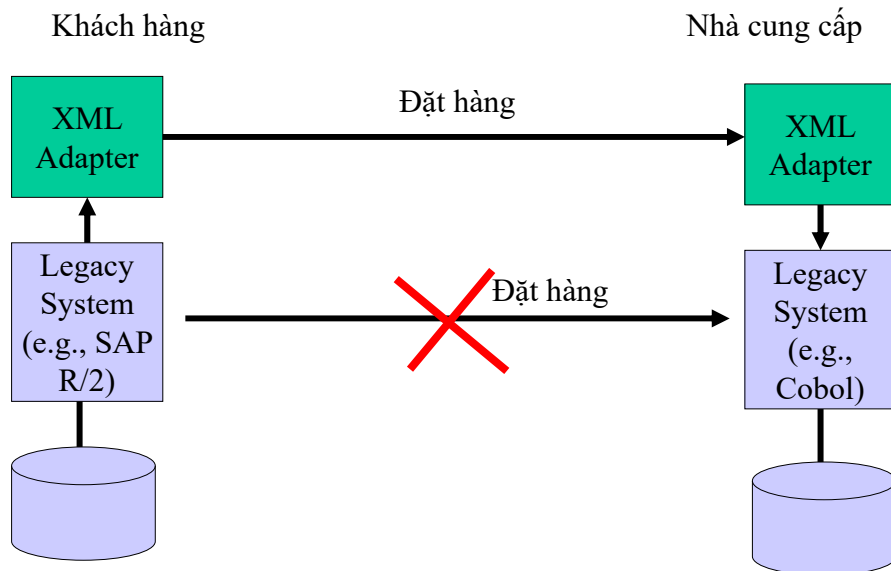
6

Ứng dụng 1: Quản trị nội dung



7

Ứng dụng 2: Trao đổi dữ liệu



8

8

Ứng dụng 3: Metadata

- Metadata/Super data: dữ liệu của dữ liệu
- XML là nền tảng cho RDF (Resource Description Framework – mô hình chuẩn để trao đổi dữ liệu trên Web)
- XML là nền tảng cho Semantic web

9

9

Ứng dụng 4: Dữ liệu bán cấu trúc

- **Dữ liệu bán cấu trúc** (semi-structured data): dữ liệu có cấu trúc một phần, khó khăn khi biểu diễn theo mô hình quan hệ cũng như mô hình hướng đối tượng
- XML là một ngôn ngữ chuẩn cho phép mô tả lược đồ và biểu diễn dữ liệu bán cấu trúc
- Dựa vào các thẻ để xác định ngữ nghĩa của các cấu trúc con

10

10

Nội dung

- Đặt vấn đề
- Các khái niệm cơ bản trong XML
- DTD
- Truy vấn dữ liệu XML
- Truy vấn dữ liệu XML trong SQL Server 2012
- Lược đồ XML
- Thảo luận

11

11

Các thành phần trong tài liệu XML

- Các thẻ
- Phân tử
- Thuộc tính
- Thực thể
- Namespace – không gian tên
- Các kiểu dữ liệu

12

12

Các thành phần trong tài liệu XML

- Ví dụ một tài liệu XML

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

13

13

Các thành phần trong tài liệu XML

- Các thẻ: **article**, **title**, **author**
 - với thẻ mở: **<article>**,...
 - và thẻ đóng: **</article>**,...
- Các phân tử: **<article> ... </article>**
 - Các phân tử có tên (article) và nội dung (...)
- Các phân tử có thể lồng nhau
- Các phân tử có thể rỗng (không có nội dung)
- Nội dung của các phân tử thường là các chuỗi ký tự và/hoặc các phân tử lồng nhau

14

14

Các quy tắc viết tài liệu XML

- Phải có một **phần tử gốc** duy nhất chứa tất cả các phần tử khác trong tài liệu
- Ví dụ:

```
<catalog>  
  <product productID="1">Chair</product>  
  <product productID="2">Desk</product>  
</catalog>
```

- Mỗi **thẻ mở** phải có một **thẻ đóng** giống hệt nhau
- Mỗi cặp thẻ mở và thẻ đóng tương ứng tạo thành 1 phần tử
- Mỗi phần tử con phải nằm trọn trong phần tử cha của nó

15

15

Chỉ thị xử lý và chú thích

- Dòng lệnh nằm ở đầu file XML
`<?xml version="1.0" encoding="utf-8" standalone="yes"?>`
- Chú thích: đặt trong cặp `<!--` và `-->`
- Một tài liệu XML có thể có hoặc không có phần chỉ thị xử lý cũng như phần chú thích
- Một số ký tự đặc biệt cần phải sử dụng theo đúng quy định:

<	→	<
&	→	&
>	→	>
"	→	"
'	→	'

16

16

Ví dụ về tài liệu XML

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Thẻ mở

Thẻ đóng

Phần tử

Nội dung của phần tử (Phần tử con hoặc/và Text)

17

17

Ví dụ về tài liệu XML

```
<article>
  <author>Gerhard Weikum</author>
  <title>The Web in Ten Years</title>
  <text>
    <abstract>In order to evolve...</abstract>
    <section number="1" title="Introduction">
      The <index>Web</index> provides the universal...
    </section>
  </text>
</article>
```

Các thuộc tính gồm tên và giá trị

18

18

XML với cây có thứ tự

- Mỗi tài liệu XML có duy nhất 1 phần tử gốc và có thể biểu diễn như một cây có thứ tự (ordered trees)
- Cây được tạo ra từ tài liệu XML là cây có thứ tự
 - Một nút CHA có thể có nhiều nút CON.
 - Các nút CON được sắp theo thứ tự và có thể có quan hệ anh em
- Có thể sử dụng thuật toán duyệt cây theo thứ tự trước (preorder traversals) được dùng để lấy thông tin từ cây XML

19

Nội dung

- Đặt vấn đề
- Các khái niệm cơ bản trong XML
- DTD
- Truy vấn dữ liệu XML
- Truy vấn dữ liệu XML trong SQL Server 2012
- Lược đồ XML
- Thảo luận

20

20

XML chính tắc

- XML có 2 dạng: XML chính tắc (well-formed XML) và XML hợp lệ (valid XML)
- Một tài liệu XML well-formed cần thỏa các quy tắc sau:
 - Có đúng một phân tử gốc
 - Mỗi thẻ bắt đầu tương ứng với 1 thẻ kết thúc.
 - Các phân tử có thể lồng nhau nhưng không được chồng lấp nhau
 - Một phân tử không có 2 thuộc tính cùng tên
 - Các ký tự đặc biệt: <, >, &, ... không xuất hiện trong dữ liệu.
- Chỉ tài liệu XML well-formed mới được xử lý bởi trình phân tích cú pháp XML

21

21

XML hợp lệ với DTD

- Tài liệu XML quá mềm dẻo, để trao đổi dữ liệu thuận lợi thì các phân tử các thuộc tính và ngữ nghĩa của chúng cần phải thống nhất → **Định nghĩa khuôn dạng cho một tài liệu XML** (Document Type Definition – DTD)
- Một tài liệu **XML là hợp lệ** (valid XML) tương ứng với một tài liệu DTD nếu đó là một tài liệu XML well-formed và tuân thủ các quy tắc trong DTD tương ứng
- Tài liệu DTD nhằm xây dựng các từ vựng cho ứng dụng XML
- Khi đọc tài liệu XML, chỉ cần đọc phần DTD là sẽ biết được cấu trúc của tài liệu XML

22

22

Định nghĩa tài liệu DTD

- Có 2 kiểu định nghĩa tài liệu DTD: trực tiếp và gián tiếp
- **Định nghĩa DTD trực tiếp:** phần mô tả DTD nằm trong tài liệu XML và định nghĩa với cú pháp:

```
<!DOCTYPE element_root <khai_báo_các_phần_tử>
```

- **Định nghĩa DTD gián tiếp:** tài liệu DTD được mô tả chi tiết ở một file xxx.dtd nào đó (có thể trên Internet-URL), sau đó file này được khai báo ở phần đầu trong tài liệu XML với cú pháp:

```
<!DOCTYPE element_root SYSTEM "xxx.dtd">
```

23

23

Ví dụ định nghĩa DTD trực tiếp

```
<?xml version="1.0"?>
<!DOCTYPE note[<!ELEMENT note (to, from, heading, body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me</body>
</note>
```

24

24

Ví dụ định nghĩa DTD gián tiếp

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
File note.dtd
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

25

25

Định nghĩa tài liệu DTD

- Mục đích định nghĩa DTD cho 1 tài liệu XML: để tài liệu XML là hợp lệ
- Tài liệu DTD có 3 thành phần:
 - Phần tử
 - Thuộc tính
 - Thực thể
- Định nghĩa tài liệu DTD cho một tài liệu XML
 - Định nghĩa các phần tử: từ khóa <!ELEMENT>
 - Định nghĩa các thuộc tính: từ khóa <!ATTLIST>
 - Định nghĩa các thực thể: từ khóa : <!ENTITY>

26

26

Ví dụ định nghĩa DTD

Newspaper Article DTD:

```
<!DOCTYPE NEWSPAPER [  
<!ELEMENT NEWSPAPER (ARTICLE+)>  
<!ELEMENT ARTICLE (HEADLINE, BYLINE, LEAD, BODY, NOTES)>  
<!ELEMENT HEADLINE (#PCDATA)>  
<!ELEMENT BYLINE (#PCDATA)>  
<!ELEMENT LEAD (#PCDATA)>  
<!ELEMENT BODY (#PCDATA)>  
<!ELEMENT NOTES (#PCDATA)>  
<!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>  
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>  
<!ATTLIST ARTICLE DATE CDATA #IMPLIED>  
<!ATTLIST ARTICLE EDITION CDATA #IMPLIED>  
<!ENTITY NEWSPAPER "Vervet Logic Times">  
<!ENTITY PUBLISHER "Vervet Logic Press">  
<!ENTITY COPYRIGHT "Copyright 1998 Vervet Logic Press">  
>
```

27

27

Nội dung

- Đặt vấn đề
- Các khái niệm cơ bản trong XML
- DTD
- Truy vấn dữ liệu XML
- Truy vấn dữ liệu XML trong SQL Server 2012
- Lược đồ XML
- Thảo luận

28

28

Truy vấn dữ liệu XML

- Mục đích của truy vấn dữ liệu XML
 - Duyệt qua các phần tử, các thuộc tính... trong tài liệu XML để đưa ra (tìm kiếm, trích rút) thông tin theo các yêu cầu của ứng dụng
- Công cụ
 - Xpath
 - Xquery
 - SQL kết hợp với Xpath, Xquery...

29

29

Xpath

- Sử dụng **biểu thức đường dẫn** để chọn lựa các nút/tập nút trong tài liệu XML
- Gần tương tự với hệ thống file truyền thống
- Có trên 100 hàm chuẩn được tích hợp sẵn, bao gồm các hàm giá trị chuỗi, giá trị số, so sánh, ...
- Xpath coi tài liệu XML như cấu trúc cây với 7 loại nút:
 - Phần tử; Thuộc tính; Text; Namespace; Chỉ thị xử lý; Lời giải thích; Nút gốc

30

30

Cú pháp Xpath

- `<?xml version="1.0" encoding="ISO-8859-1"?>`
`<bookstore>`
`<book>`
`<title lang="en">Harry Potter</title>`
`<author>J K. Rowling</author>`
`<year>2005</year>`
`<price>29.99</price>`
`</book>`
`</bookstore>`
- Nút gốc? Nút phân tử?
- Nút thuộc tính?
- Giá trị / Text?
- ...

31

31

Cú pháp Xpath

- Đường dẫn tuyệt đối: Bắt đầu bởi dấu /
- Đường dẫn tương đối: Bắt đầu bởi dấu //
- Kí tự đại diện: Ký tự * thay cho nút bất kỳ ; Cặp ký tự @* thay cho thuộc tính bất kỳ
- Vị từ
- Một số hàm, toán tử

32

32

Xpath

Ví dụ cho tài liệu XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book>
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

/bookstore/* có nghĩa lấy tất cả nút con của bookstore

//* chọn tất cả phần tử trong tài liệu

//title[@*] chọn tất cả phần tử title với bất kỳ thuộc tính nào

33

33

Cú pháp Xpath

- **Vị từ** dùng để tìm đến nút xác định hoặc nút chứa một giá trị xác định. Vị từ được đặt trong cặp ngoặc vuông []
- Ví dụ:
 - Lấy phần tử book là con của bookstore có **price > 35**:
/bookstore/book[price>35.00]
 - Lấy tất cả phần tử title có tên thuộc tính là lang: **//title[@lang]**
 - Lấy tất cả phần tử title có **thuộc tính lang nhận giá trị 'eng'**:
//title[@lang='eng']

34

34

Cú pháp Xpath

- Một số hàm:
 - position() : vị trí của nút
 - last(): vị trí nút cuối cùng
 - ...
 - Chọn phần tử book đầu tiên là con của phần tử bookstore:
/bookstore/book[1]
 - Chọn 2 phần tử book đầu tiên là con của phần tử bookstore:
/bookstore/book[position()<3]
- Một số toán tử:
 - +, -, *, div
 - | gộp 2 tập nút
 - ...

35

35

Xpath trong SQL server

- CREATE TABLE tblBooks1(id CHAR(4), chitiet XML);
- INSERT INTO tblBooks VALUES('1',
'<book category="cooking">
 <title lang="en">Everyday Italian</title>
 <author>Giada De Laurentiis</author>
 <year>2005</year>
 <price>30.00</price> </book>
<book category="children">
 <title lang="en">Harry Potter</title>
 <author>J K. Rowling</author>
 <year>2005</year>
 <price>29.99</price> </book>
<book category="web">
 <title lang="en">Learning XML</title>
 <author>Erik T. Ray</author>
 <year>2003</year>
 <price>39.95</price> </book>')

36

36

Xpath trong SQL server

- Kết hợp SQL và biểu thức đường dẫn để truy xuất dữ liệu XML trong SQL server
 - Xét bảng tblBoooks như ở slide trước
 - Trong SQL server sử dụng một số phương thức thao tác với dữ liệu XML
 - query(): trích rút dữ liệu từ kiểu XML
 - value(): trích rút giá trị đơn từ tài liệu XML

37

37

Xpath trong SQL server

- Lấy tất cả các phần tử trong tblBooks.chitiet
`SELECT tblBooks.chitiet.query('/*') FROM tblBooks;`
- Lấy phần tử thứ nhất trong tblBooks.chitiet
`SELECT tblBooks.chitiet.query('/book[1]')`
`FROM tblBooks`
- Cho biết thông tin của 2 phần tử đầu tiên trong tblBooks.chitiet
`SELECT tblBooks.chitiet.query('/book[position()<3]')`
`FROM tblBooks`
- Cho biết thông tin phần tử cuối trong tblBooks.chitiet
`SELECT tblBooks.chitiet.query('/book[last()]')`
`FROM tblBooks`

38

38

Xpath trong SQL server

- Cho biết thông tin các cuốn sách có giá lớn hơn 30.00 trong tblBooks.chitiet

```
SELECT tblBooks.chitiet.query('/book[price>30.00]')
```

```
FROM tblBooks
```

- Cho biết tiêu đề các cuốn sách có giá lớn hơn 30.00 trong tblBooks.chitiet

```
SELECT tblBooks.chitiet.query('/book[price>30.00]/title')
```

```
FROM tblBooks
```

- Cho biết tiêu đề các cuốn sách viết bằng tiếng Anh trong tblBooks.chitiet

```
SELECT tblBooks.chitiet.query('/book/title[@lang="en"]')
```

```
FROM tblBooks
```

39

39

Xpath trong SQL server

- Cho biết thông tin các cuốn sách có giá ≤ 30.00 và > 29.99 trong tblBooks.chitiet

```
SELECT tblBooks.chitiet.query('/book[price<=30.00 and price>29.99]')
```

```
FROM tblBooks
```

- Cho biết tiêu đề các cuốn sách có giá khác 30.00

```
SELECT tblBooks.chitiet.query('/book[price!=30.00]/title')
```

```
FROM tblBooks
```

- Lưu ý: phương thức value(' ',' ') có 2 đối

- Cho biết giá của cuốn sách đầu tiên

```
SELECT tblBooks.chitiet.value('/book/price)[1]','nvarchar(100)')
```

```
FROM tblBooks
```

40

40

Xquery

- Được mở rộng từ biểu thức Xpath phiên bản 1.0
- Xquery là ngôn ngữ tìm kiếm và trích rút dữ liệu từ tài liệu XML
- Xquery có mặt trong hầu hết các các hệ quản trị CSDL thương mại (SQL server, Oracle, DB2,...)
- Xquery được thiết kế để truy vấn dữ liệu XML, tương tự SQL sinh ra để truy vấn với table trong CSDL quan hệ
- Xquery xem tài liệu XML như cấu trúc cây

41

41

Xquery

- Xquery phân biệt chữ hoa chữ thường
- Giá trị chuỗi kí tự có thể đặt trong cặp nháy đơn hoặc nháy kép
- Biến được đặt tên bắt đầu bởi kí tự \$, theo sau là tên biến.
- Lỗi giải thích được đặt trong cặp dấu hai chấm :
 - Ví dụ :lỗi giải thích:

42

42

Xquery

- XQuery là một mở rộng của Xpath bổ sung biểu thức **FLWOR**
- **Biểu thức FLWOR**
 - **for...let ... where ... order by...return...**
 - for – dùng để lặp việc gắn **mỗi biến** với mỗi phần tử trong dữ liệu XML
 - Biến được đặt tên sau dấu **\$**
 - let – gán một biến với 1 biểu thức
 - where – dùng để lọc các phần tử thỏa điều kiện
 - return – trả về thông tin cần
- Xquery sử dụng **doc("ten_file_xml.xml")** để mở tài liệu XML (file XML)

43

43

Xquery

Cho file "books.xml" như sau:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>35.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

44

44

Xquery

- Ví dụ:

Xquery với biểu thức đường dẫn:

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

Biểu thức FLWOR:

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

Ý nghĩa của 2 truy vấn trên?

45

45

Xquery

- **Mệnh đề for**

– Dùng từ khóa **to** để lặp:

```
for $x in (1 to 5)
```

```
return <test>{$x}</test>
```

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

– Dùng từ khóa **at** để đếm số vòng lặp, **data** để lấy dữ liệu

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

```
return <book>{$i}. {data($x)}</book>
```

```
<book>1. Everyday Italian</book>
<book>2. Harry Potter</book>
```

46

46

Xquery

- **Mệnh đề let**

- Gán giá trị cho biến

let \$x := (1 to 5)

return <test>{\$x}</test>

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

- **Mệnh đề where**

- Dùng để đưa điều kiện tìm kiếm hay trích lọc. Có thể sử dụng tiêu chuẩn phức hợp với and, or,...
- Tương tự mệnh đề where trong SQL

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

where \$x/price>100 or \$x/price<30

47

47

Xquery

- **So sánh giá trị: trong Xquery có 2 cách so sánh**

- So sánh tổng quát: =, !=, <, <=, >, >=
- So sánh giá trị: eq, ne, lt, le, gt, ge

- **Sự khác nhau của 2 cách so sánh**

- Biểu thức sau trả về giá trị **đúng** nếu tồn tại thuộc tính q có giá trị lớn hơn 10

\$bookstore//book/@q > 10

- Biểu thức sau trả về giá trị **đúng** nếu tồn tại duy nhất một thuộc tính q được trả về và giá trị của nó lớn hơn 10. Nếu có nhiều thuộc tính q được trả về thì sẽ bị lỗi.

\$bookstore//book/@q gt 10

48

48

Xquery

- **Biểu thức điều kiện**

- if then else

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>

Kết quả:

<adult>Everyday Italian</adult>

<child>Harry Potter</child>

49

49

Xquery

- **Hàm có sẵn**

- Xquery có trên 100 hàm tích hợp sẵn, bao gồm các hàm cho giá trị chuỗi, số, so sánh, thời gian, thao tác với nút, thao tác tuần tự, giá trị boolean,.

- Một lời gọi hàm có thể xuất hiện trong một phần tử hoặc trong một vị tử hoặc trong mệnh đề let

- Hàm count() đếm số phần tử

- Hàm avg() tính trung bình cộng...

50

50

Xquery trong SQL server

- CREATE TABLE hosoXML(id int identity(1,1), noidung xml)
- Insert Into HosoXML Values ('<Persons>
 <Person>
 <Name>Demo1</Name> <Salary>10000</Salary>
 <Designation>Developer</Designation> <RegionId>1</RegionId>
 </Person>
 <Person>
 <Name>Demo2</Name> <Salary>12000</Salary>
 <Designation>Sr. Developer</Designation> <RegionId>2</RegionId>
 </Person>
 <Person>
 <Name>Demo3</Name> <Salary>13000</Salary>
 <Designation>Sr. Developer</Designation> <RegionId>3</RegionId>
 </Person>
 <Person>
 <Name>Demo4</Name> <Salary>14000</Salary>
 <Designation>Developer</Designation> <RegionId>5</RegionId>
 </Person> </Persons>')

51

51

Xquery trong SQL server

- Xét truy vấn SQL với Xpath sau:
Select **noidung.query('/Persons/Person[Salary>12000]')**
From hosoXML
Câu lệnh trên làm gì?
- Xét truy vấn SQL với biểu thức FLOWR sau:
Select **noidung.query('for \$A in /Persons/Person
 where \$A/Salary>=12000
 return \$A')**
From hosoXML;
Câu lệnh trên làm gì?

52

52

Xquery trong SQL server

- Xét truy vấn SQL với biểu thức FLOWR sau:
Select `noidung.query('for $A in /Persons/Person
where $A/Salary>=12000
return $A/Name')`

From `hosoXML`;

Câu lệnh trên làm gì?

- Xét truy vấn SQL với biểu thức FLOWR sau:
Select `noidung.query('for $A in /Persons/Person
where $A/Salary>=12000
order by $A/Name[1] descending
return $A')`

From `hosoXML`;

Câu lệnh trên làm gì?

53

53

Xquery trong SQL server

- Xét truy vấn SQL với biểu thức FLOWR sau:
Select
PersonCount=`noidung.value('count(/Persons/Person[Salary>12000])','int')`

From `hosoXML`

Câu lệnh trên làm gì?

- Xét truy vấn SQL với biểu thức FLOWR sau:
Select
SalaryAvg=`noidung.value('avg(/Persons/Person[Salary>12000]/Salary)','int')`

From `hosoXML`

Câu lệnh trên làm gì?

Có thể thay hàm avg trên bởi hàm `sum()`, `max()`, `min()`

54

54