

XQuery

(14주차)

학습개요

- 학습 목표
 - XQuery의 특징을 설명할 수 있다.
 - XQuery를 이용해 정보를 검색하는 방법을 설명할 수 있다.
 - XQuery의 FLWOR 표현식을 설명할 수 있다.
- 학습 내용
 - XQuery 소개
 - XQuery 문법 및 활용
 - XQuery 실습

XQuery 개요

- XML Query Language
- W3C 2007년 1월 발표
- XSLT 2.0처럼 XPath 2.0을 기반으로 XDM 데이터 모델을 사용
- XML 데이터 소스를 쿼리하거나 조작
- 사용자 정의 함수를 만들 수 있음
- XML Schema를 사용할 수 있음
- XML 구문을 사용하지 않고 SQL 구조와 유사하여 프로그래밍 경험이 없는 사람들도 쉽고 간편하게 사용할 수 있음
- 쿼리를 위해서는 XQuery를 , 문서변환을 위해서는 XPath 2.0을 사용 권장
- XQuery 프로세서가 필요

XQuery 개요

- <http://www.w3.org/TR/xquery/>



XQuery 1.0: An XML Query Language (Second Edition)

W3C Recommendation 14 December 2010 (*Link errors corrected 3 January 2011*)

This version:

<http://www.w3.org/TR/2010/REC-xquery-20101214/>

Latest version:

<http://www.w3.org/TR/xquery/>

Previous versions:

<http://www.w3.org/TR/2009/PER-xquery-20090421/>, <http://www.w3.org/TR/2007/REC-xquery-20070123/>

Editors:

Scott Boag (XSL WG), IBM Research <scott_boag@us.ibm.com>

Don Chamberlin (XML Query WG) <dchamber@us.ibm.com>

Mary F. Fernández (XML Query WG), AT&T Labs <mff@research.att.com>

Daniela Florescu (XML Query WG), Oracle <dana.florescu@oracle.com>

Jonathan Robie (XML Query WG), Red Hat, via <http://www.ibiblio.org/jwrobie/>

Jérôme Siméon (XML Query WG), IBM T.J. Watson Research Center <simeon@us.ibm.com>

Please refer to the [errata](#) for this document, which may include some normative corrections.

See also [translations](#).

This document is also available in these non-normative formats: [XML](#) and [Change markings relative to first edition](#).

[Copyright](#) © 2010 [W3C](#)® ([MIT](#), [ERCIM](#), [Keio](#)). All Rights Reserved. W3C [liability](#), [trademark](#) and [document use](#) rules apply.

XQuery 문서

- .xquery 또는 .xq 확장자를 주로 사용
- XQuery 문서 첫 라인에 xquery version "1.0"; 기술

```
xquery version "1.0";  
xquery_expression
```

- XQuery 표현식에 숫자, 문자열, 시퀀스 생성자, 엘리먼트 생성자, 빌트-인 함수, 사용자 정의 함수, 조건식, FLWOR 표현식을 사용
- XPath 2.0과 동일한 (: :) 주석을 사용

XQuery 프로세서

- <http://www.saxonica.com/download/opensource.xml>
- Saxon-HE 사용법

```
java net.sf.saxon.Query [options] -q:queryfile [ params...]
```

- java 명령어를 이용해 net.sf.saxon.Query 클래스 실행
- saxon9he.jar 자바 아카이브 파일을 클래스패스에 지정
- -q: 옵션에 xquery 파일을 기술

```
C:\Development\xml\W14>java -cp ..\SaxonHE9-6-0-5JWsaxon9he.jar net.sf.saxon.Query -  
q:books.xquery
```

[실행결과]

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<title xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">Beginning XSLT and XPath: Transforming  
XML Documents and Data</title>
```

XQuery 표현식

- 다중 표현식은 ';' 로 구분

```
xquery version "1.0";  
"Hello XQuery!"
```

```
xquery version "1.0";  
"Hello!",  
"Good luck!"
```

XQuery 표현식

- 시퀀스 생성자를 이용한 표현식

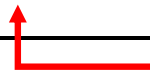
```
xquery version "1.0" encoding "utf-8";  
(1 to 5)
```

- 빌트-인 함수를 이용한 표현식

```
xquery version "1.0" encoding "utf-8";  
reverse(1 to 5)
```

- 엘리먼트 생성자를 이용한 표현식

```
xquery version "1.0" encoding "utf-8";  
<result>{(1 to 5)}</result>
```



{ } 는 텍스트와 표현식을 구분하는데 사용. { } 안의 평가식이 평가된 후 출력

XML 문서 식별

- XML 문서 식별을 위한 doc() 함수의 사용

```
xquery version "1.0" encoding "utf-8";  
doc("books.xml")
```

위치 경로를 이용한 정보 검색

- XML 데이터 소스의 콘텐츠를 선택하기 위해 XPath 구문을 사용

```
xquery version "1.0" encoding "utf-8";  
doc("books.xml")/books/book
```

```
xquery version "1.0" encoding "utf-8";  
doc("books.xml")/books/book/title  
[contains(., "Beginning")]
```

FLWOR 표현식을 이용한 정보 검색

- SQL Select문을 기반으로 for, let, where, order, return 문을 구성

```
xquery version "1.0" encoding "utf-8";  
for $x in doc("books.xml")/books/book  
where $x/translator/name="이순신"  
return $x/title
```

- \$x : 범위변수로 소스문서 books.xml의 각 book 엘리먼트로 설정
- doc("books.xml")/books/book : 바인딩 시퀀스
- \$x/translator/name="이순신" : 바인딩 시퀀스가 설정한 범위변수를 필터링할 표현식
- \$x/title : 필터링된 결과에서 \$x/title을 출력

FLWOR 표현식을 이용한 정보 검색

- for 절은 반복 수행하면서 표현식에 의해 반환된 각 항목을 변수와 바인드함
- for 절은 중첩 사용이 가능

```
xquery version "1.0" encoding "utf-8";  
for $x in (1 to 5)  
return <test>{$x}</test>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<test>1</test>  
<test>2</test>  
<test>3</test>  
<test>4</test>  
<test>5</test>
```

FLWOR 표현식을 이용한 정보 검색

- for절에서 at 키워드를 이용해 반복 횟수를 출력할 수 있음

```
xquery version "1.0" encoding "utf-8";  
for $x at $i in doc("books.xml")/books/book/title  
return <book>{$i}. {data($x)}</book>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<book>1. Beginning XML, 5th Edition</book>  
<book>2. Beginning XSLT and XPath: Transforming XML Documents and Data</book>  
<book>3. MongoDB: The Definitive Guide, 2nd Edition</book>
```

FLWOR 표현식을 이용한 정보 검색

- for 절에서 ','를 사용하면 하나 이상의 범위변수와 바인딩 시퀀스를 사용할 수 있음

```
xquery version "1.0" encoding "utf-8";  
for $x in (10,20), $y in (100,200)  
return <test>x={$x} and y={$y}</test>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<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>
```

FLWOR 표현식을 이용한 정보 검색

- let 절은 for 절과 다르게 동일한 표현식을 반복하지 않으면서 범위변수 할당 결과를 만듦

```
xquery version "1.0" encoding "utf-8";  
let $x := (1 to 5)  
return <test>{$x}</test>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<test>1 2 3 4 5</test>
```

FLWOR 표현식을 이용한 정보 검색

- where 절은 결과를 위한 조건을 기술함

```
xquery version "1.0" encoding "utf-8";  
for $x in doc("books.xml")/books/book  
where $x/price>30000 and $x/price<40000  
return ($x/title, $x/price)
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<title>Beginning XML, 5th Edition</title>  
<price currency="원">35000</price>
```


FLWOR 표현식을 이용한 정보 검색

- order by 절로 결과의 정렬 순서를 기술할 수 있음

```
xquery version "1.0" encoding "utf-8";  
for $x in doc("books.xml")/books/book  
order by $x/@category, $x/title  
return $x/title
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<title>Beginning XML, 5th Edition</title>  
<title>Beginning XSLT and XPath: Transforming XML Documents and Data</title>  
<title>MongoDB: The Definitive Guide, 2nd Edition</title>
```

FLWOR 표현식을 이용한 정보 검색

- return 절로 반환될 결과를 기술함

```
xquery version "1.0" encoding "utf-8";  
for $x in doc("books.xml")/books/book  
return $x/title
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<title>Beginning XML, 5th Edition</title>  
<title>Beginning XSLT and XPath: Transforming XML Documents and Data</title>  
<title>MongoDB: The Definitive Guide, 2nd Edition</title>
```

사용자 정의 함수

- 사용자 정의 함수 문법

```
declare function prefix:function_name($parameter as datatype)
as returnDatatype
{
    ...
};
```

사용자 정의 함수

- 사용자 정의 함수 문법

```
xquery version "1.0" encoding "utf-8";
declare function local:salesPrice($p as xs:decimal?, $d as xs:decimal?)
as xs:decimal?
{
    let $disc := $d
    return ($p - $disc)
};

for $book in doc('books.xml')/books/book
return <salesPrice>{local:salesPrice($book/price, 5000)}</salesPrice>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<salesPrice>30000</salesPrice>
<salesPrice>22000</salesPrice>
<salesPrice>40000</salesPrice>
```

학습정리

- XQuery는 XLST 2.0처럼 XPath 2.0을 기반으로 XDM 데이터 모델을 사용합니다.
- XQuery는 XML 데이터 소스를 쿼리하거나 조작할 수 있습니다.
- XQuery는 사용자 정의 함수를 만들 수 있습니다.
- XQuery를 사용하기 위해서는 XQuery 프로세서가 필요합니다.
- XQuery는 XML 데이터 소스의 콘텐츠를 선택하기 위해 XPath 구문을 사용합니다.
- XQuery는 XML 데이터 소스의 콘텐츠를 선택하기 위해 FLWOR 표현식을 사용합니다.