# group by를 지원하기 위한 XQuery 확장[1]

오정선[O] 조혜영 이민수
이화여자대학교 과학기술대학원 컴퓨터학과
{jsoh[O], hycho}@ewhain.net, mlee@ewha.ac.kr

## XQuery extension to support the group by clause

Jungsun Oh[O], Hyeyoung Cho, Minsoo Lee
Dept. of Computer Science & Engineering, Ewha Institute of Science and Technology, Ewha Womans University

요 약

XML 문서의 검색을 위한 질의 언어인 XQuery는 다양한 데이터 소스로부터 가져온 고유한 구조를 가진 질의 결과로 구성할 수 있도록 설계되어 XML질의 언어의 표준이 되었다. 현재 XQuery는 반복문 등을 포함하는 과다한 검색기능을 지원하도록 함으로써 질의 표현이 상대적으로 복잡한 형태를 취한다. 따라서 일반적인 SQL처럼 XQuery에 명시적인 group by절을 도입한 질의 표현기법을 모색하여 XML 데이터의 재구성과 집계함수 처리를 위한 그룹화를 보다 쉽게 구성할 수 있도록 하고자 하였다.

## 1. 서 론

XQuery(XML Query Language)는 W3C의 XML Query Working Group에서 권고하는 질의 언어로서 XPath (XML Path Language) 버전 2.0[5], XQL 및 SQL 같은 여러 가지 다른 질의 언어를 기반으로 하는 Quilt라는 질의 언어에서 발전한 것이다. XQuery는 모든 형식의 XML 데이터를 질의할 수 있게 최적화된 지능적이고 뛰어난 언어로 XML 데이터 형식과 연결된 메서드를 사용하여 XML 데이터 형식의 변수와 열에 대한 질의를 실행할 수 있다[1]. 하지만, 지금의 XQuery는 반복문 등을 포함하는 과다한 검색기능을 지원하도록 함으로써 질의 표현이 상대적으로 복잡한 형태를 취한다. 따라서 XML 데이터의 재구성과 집계함수 처리를 위한 그룹화를 효율적으로 처리해주는 질의 최적화기법에 대해 연구한다. 본 논문의 구성은 다음과 같다. 먼저 XQuery 및 그룹화 처리와 관련된 연구들에 대해 소개하고, 다음으로 XQuery 에서의 명시적인 group by 구성을 이용한 그룹화 처리가 필요한 이유와 구체적인 예제를 사용하여 group by 구성을 통한 XQuery의 확장방안에 대하여 살펴보고, 마지막으로 결론 및 향후과제를 소개하였다.

## 2. 관련연구

### 2.1 Xperanto

Xperanto 시스템은 XML Schema로 스키마 내용 표현하고, XQuery를 이용하여 관계형 데이터를 XML 뷰로 정의하고 사용자 질의를 구성한다. 관계형 데이터로부터 XML 문서를 구성하고 구현하는 방법을 명시하기 위해 SQL을 확장하였고, 대부분 경우에서 가능한 많은 계산을 관계형 엔진에서 처리하도록 하였다.

사용자 질의는 먼저 파싱 과정을 거쳐 XQGM(XML Query Graph Model)이라 불리는 중간 질의 표현으로 변환된다. XML 뷰를 정의한 Xquery와 사용자 XQuery가 각각 XQGM으로 변환되어 합성되고 합성된 결과는 SQL로 변환되는 부분과 XML문서를 생성하기 위한 부분으로 분리되어 각각 실행된다[2][3].

### 2.2 TIMBER

TIMBER 프로젝트에서는 TAX[4]라는 트리 대수(tree algebra)를 내부적으로 정의하여 잠재적으로 포함되는 그룹화 구성에 관한 내용을 지정하고, 이를 이용해 XQuery의 내포된 질의를 TAX의 그룹화 질의로 재구성하는 일련의 방식에 대해서 기술하였다.

## 3. XQuery의 그룹화

XQuery 그룹화는 데이터를 재구성하거나 집계함수를 처리하는 경우 주로 이용되며, 관계형 데이터베이스에 있어 중요한 연산 중 하나이다. 핵심 내용은 직관적으로 데이터들을 튜플들의 그룹으로 분할하고 이들에 대해 반복적인 계산을 수행하는 질의를 처리하는 것이다.

다음에서 구체적인 예제를 사용하여 group by 구성을 통한 XQuery의 확장 방안에 대해서 살펴보도록 하겠다. 그림 1은 각 도서의 도서제목, 저자, 출판사, 가격들에 대한 정보를 담고 있는 서지목록 XML데이터의 한 예이다.

1 "이 논문은 2004년도 한국학술진흥재단의 지원에 의하여 연구되었음." (KRF-2004-041-D00572)

```
<bib>
   <book year="1994">
      <title>TCP/IP Illustrated</title>
      <author><last>Stevens</last><first>W.</first></author>
      <publisher>Addison-Wesley</publisher>
      <price> 65.95</price>
   </book>
   <book year="1992">
      <title>Advanced Programming in the Unix  </title>
      <author><last>Stevens</last><first>W.</first></author>
      <publisher>Addison-Wesley</publisher>
      <price>65.95</price>
   </book>
   <book year="1994">
      <title>Data Mining <title>
      <author><last>Stevens</last><first>W.</first></author>
      <publisher>Mc Graw Hill</publisher>
      <price> 70.90</price>
   </book>
   <book year="2000">
      <title>Data on the Web</title>
      <author><last>Abiteboul</last><first>Serge</first></author>
      <author><last>Buneman</last><first>Peter</first></author>
      <author><last>Suciu</last><first>Dan</first></author>
      <publisher>Morgan Kaufmann Publishers</publisher>
      <price>39.95</price>
   </book>
</bib>
```

그림 1. bib.xml의 서지목록 예제 데이터

### 3.1 유형 1 : 하나의 바인딩 변수에 의한 group by

유형 1은 하나의 바인딩 변수에 의한 group by이다. [그림 1]의 예제 데이터에 대해서 각 저자별로 출간한 도서의 제목을 검색하여 새로운 결과 XML문서를 구성하는 질의를 예제로 살펴본다.

예제1. 서지목록 데이터인 bib.xml 예제문서에서 각 저자별로 출간한 도서의 제목들을 검색하여 출력하라.

```
<results>
  <result>
    <author>Stevens W.</author>
    <titles>
      <title>TCP/IP Illustrated</title>
      <title>Data Mining</title>
      <title>Advanced Programming in the Unix Environment </title>
    </titles>
  </result>
  <result>
    <author>Abiteboul Serge</author>
    <titles>
      <title>Data Mining</title>
      <title>Advanced Programming in the Unix Environment</title>
      <title>Data on the Web</title>
    </titles>
  </result>
  <result>
    <author>Buneman Peter</author>
    <titles>
      <title>Data on the Web</title>
    </titles>
  </result>
  <result>
    <author>Suciu Dan</author>
    <titles>
      <title>Data on the Web</title>
    </titles>
  </result>
</results>
```

그림 2. 예제1의 예상 결과 문서

예상 결과 문서는 [그림 2]과 같다. 각 <result>노드 안의 <author>노드에 저자의 이름이 출력되고 <titles>노드에 해당 저자가 지은 도서의 제목들이 그룹으로 묶여 출력된다.

[그림 3]의 질의 문장은 예제 1의 결과 문서를 얻기 위해 group by절을 사용하지 않은 기존 XQuery로 작성한 것으로, W3C XML Use Cases[6]의 1.1.9.4 Q4 예제를 참고하였다.

```
<results>
{
   let $a := doc("bib.xml")//author
ⓐ for $author in distinct-values($a/text())
   return
    <result>
      <author>{ $author }</author>
      <titles>
      {
   ⓑ for $b in doc("bib.xml")/bib/book
      where some $ba in $b/author
            satisfies ($ba/text() = $author)
      return
         $b/title
      }
      </titles>
    </result>
}
</results>
```

그림 3. 예제 1을 group by절을 사용하지 않고 작성한 질의 문장

질의 수행 과정을 단계별로 살펴보면 다음과 같다. 먼저 let 구문에서 bib.xml문서내의 모든 저자노드 <author>들의 집합을 바인딩 변수 $a에 할당한다. 다음으로 ⓐ의 for 구문에서 저자노드의 <author>에 대해 distinct-values 함수를 적용하여 중복된 값을 제거하고 유일한 값만을 추출하여 저자노드 <author>안에 각각 출력한다. 마지막으로 ⓑ의 for 구문의 where절에서 some… satisfies…정량 표현식을 이용한 조건식을 구성하여 ⓐ의 return 구문에서 출력한 해당저자를 만족하는 도서노드 <book>들을 선별하여 각각의 노드가 가지는 도서 제목노드 <title>의 집합들을 도서 제목 집합노드 <titles>안에 출력한다.

```
<results>
{
 for $b in doc("bib.xml")/bib/book,
     $a in $b/author
 return group by { $a }
   <result>
     { $a }
     <titles>
        { $b/title }
     </titles>
   </result>
}
</results>
```

그림 4. 예제1을 group by절을 사용한 XQuery 질의

[그림 4]의 질의 문장은 예제 1을 group by절을 도입한 XQuery로 작성한 것이다. { } 표시 안에 그룹화 변수를 표현하며, group by절 구성에 지정된 바인딩 변수들에 해당하는 각각의 그룹들은 group by절 구성에 사용되지 않은 바인딩 변수들에 의한 일련의 노드들의 집합을 가진다.

[그림 4]에서 group by절 구성의 대상인 변수는 $a이며, group by절 구성의 대상이 아닌 변수는 $b이다. $a에 의해 유일한 저자노드 <author>의 그룹이 형성되고 각 해당 저자의 그룹에 해당하는 도서 제목노드 <title>들이 도서 제목 집합노

드 <titles>안에 반복적으로 생성되어 출력된다.

3.2 유형 2 : 집계함수와 함께 사용된 group by

예제2. 서지목록 데이터인 bib.xml 예제문서에서 각 저자별로 출간한 도서의 제목들과 도서의 개수를 검색하여 출력하라.

```
<results>
{
 for $b in doc("bib.xml")/bib/book,
     $a in $b/author
 return group by { $a }
   <result>
     { $a }
     <title-count>
       { count($b/title) }
     </title-count>
     <titles>
       { $b/title }
     </titles>
   </result>
}
</results>
```

그림 5. 예제2를 group by절을 사용한 XQuery 질의

[그림 5] 에서 group by절 구성의 대상인 변수는 $a이며, group by절 구성의 대상이 아닌 변수는 $b이다. $a에 의한 유일한 저자 <author>의 그룹이 형성되고 각 해당 저자의 그룹이 출간한 도서의 개수가 <title-count>노드에 출력되고, 출간한 도서 제목노드 <title>들이 도서 제목 집합노드 <titles> 안에 반복적으로 생성되어 출력된다.

3.3 유형 3 : 두개 또는 그 이상의 바인딩 변수에 의한 group by

유형 3은 두개 또는 그 이상의 바인딩 변수에 의한 group by이다. 앞에서 살펴본 유형 1의 한 개의 바인딩 변수에 의한 group by 질의의 예를 확장시킨 유형이라고 할 수 있다.

예제3. 서지목록 데이터인 bib.xml 예제문서에서 각 저자별, 연도별로 출간한 도서의 제목들을 검색하여 출력하라.

```
<results>
{
for  $b in doc("bib.xml")/bib/book,
     $a in $b/author,
     $y in $b/@year
return group by { $a, {$y} }
    <result>
          $a
            <year-title>
              <year>{ $y }</year>
          $b/title
       </year-title>
    </result>
}
</results>
```

그림 6. 예제3을 group by절을 사용한 XQuery 질의

[그림 6]의 질의의 경우 group by 대상인 변수는 $a, $y이며, group by절 대상이 아닌 변수는 $b이다. 먼저 $a에 의한 유일한 <author>의 그룹이 형성되고, 다시 $y에 의해 해당저자 그룹 안에 유일한 <year> 그룹이 형성된다. 마지막으로 각 해당저자의 각 발행년도 그룹에 해당하는 도서제목 튜플들이 반복적으로 생성되어 발행년도와 함께 <year-title>노드 안에 출력된다.

## 4. 결론

XQuery의 그룹화처리 필요성에 대하여 소개하고, 실제 예제를 통한 group by를 명시적으로 도입하여, XML 데이타의 재구성과 집계함수 처리를 위한 그룹화를 보다 쉽게 구성할 수 있는 질의 표현기법을 모색하고자 하였다. 이러한 기법을 통해 다음과 같은 효과를 얻을 수 있을 것으로 기대된다. 첫째, SQL에서처럼 group by를 사용하여 그룹화를 표현할 수 있게 된다. 현재 XQuery에서는 내포된 FLWR식(nested FLWR expression)과 join을 통해 그룹화처리를 하고 있어 복잡하다. 둘째, 단일 레벨에서만 group by가 가능한 SQL과는 달리 임의의 내포 레벨에서의 그룹화가 가능해 진다. XML의 다양한 계층 구조를 고려할 때 내포 레벨에 구애받지 않는 그룹화가 반드시 필요하다. 셋째, 내포된 FLWR식 안에 다시 내포된 FLWR식 을 가지는 구조를 가진 복잡한 계층 구조의 XML 문서를 쉽게 생성할 수 있게 된다.

향후, 방대한 XQuery 문법 중에서 핵심적인 부분을 선별하여 group by 구성을 추가한 Bottom-up Query Plan으로 실제적인 질의 처리 시스템을 구현하여, 효율적으로 Query 처리해주는 질의 최적화 기법에 대한 연구를 진행할 예정이다.

### 참고문헌

[1] XML Query, http://www.w3.org/XML/Query

[2] J. Shanmugasundaram, J. Kiernan, E. Shekita, C. Fan and J. Funderburk, "Querying XML Views of Relational Data," VLDB Conference, pp.261-270, 2001.

[3] M. Carey, D. Florescu, Z. Ives, Y. Lu, "XPERANTO: Publishing Object-Relational Data as XML," WEBDB Workshop, pp.105-110, 2000.

[4] S. Paparizos, S. Al-Khalifa, H. V. Jagadish, L. Lakshmanan, A. Nierman, D.Srivastava and Y. Wu, "Grouping in XML," XMLDM, 2002.

[5] XML Path Language (XPath) 2.0, http://www.w3.org/TR/2005/WD-xpath20-20050404/

[6] Don Chamberlin, Peter Fankhauser, Daniela Florescu, Jonathan Robie, Massimo Marchiori, "XML Query Use Cases", W3C Working Draft, November 2003. http://www.w3.org/TR/xquery-use-cases/