관계 데이터베이스에서 XQuery 질의 처리*

신병주° 진 민 하경재
경남대학교 컴퓨터공학과
{challenger°, mjin, kjha}@hawk.com.kyungnam.ac.kr

Processing XQuery Queries in Relational Databases

Byungjoo Shin° Min Jin Keongjae Ha
Dept. of Computer Engineering, Kyungnam University

요 약

XML의 계층적 구조와 관계 데이터베이스의 평면적 구조가 일치하지 않기 때문에, 관계 데이터베이스에 저장된 XML 데이터에 대한 질의를 처리하여 XML 문서를 생성하기 위해서는 별도의 처리과정이 요구된다. 또한, 기존의 관계 데이터베이스가 XML 질의 언어를 지원하지 못하기 때문에 XML 질의 언어의 SQL로의 변환도 요구된다. 따라서, 본 논문은 XQuery의 경로 표현식을 패스 테이블을 이용하여 SQL로 변환하여 데이터를 추출하고 뷰를 이용하여 평면적 구조로 저장된 XML 데이터를 계층적 구조인 XML 문서로 효과적으로 변환하여 태깅하는 방법을 제안하고, 이를 이용한 XML 질의 처리 시스템을 설계한다.

1. 서 론

XML은 다양한 장점으로 인해 인터넷 기반 환경에서 데이터의 표현 및 교환을 위한 표준으로 자리잡았다. 따라서, XML 데이터를 저장하고 질의를 처리할 수 있는 XML 저장 시스템과 관련한 다양한 연구가 진행되고 있다. 관계 데이터베이스는 XML 데이터와 기존의 관계 데이터가 동시에 저장 가능하고, 질의 처리 성능면에서 타 저장 시스템에 비해 우수하다. 그러나, XML 문서는 엘리먼트들의 계층적 구조로 이루어져 있지만, 관계 데이터베이스는 테이블들의 평면적 구조로 이루어져 있기 때문에 구조적으로 일치하지 않는다. 따라서, XML 문서를 관계 데이터베이스에 저장하고 질의를 처리하기 위해서는 추가적인 처리과정이 필요하다. 또한, 관계 데이터베이스가 XML 질의 언어를 지원하지 못하기 때문에 XML 질의 언어의 SQL로의 변환이 필요하다[3][6][7][8].

따라서, 본 논문에서는 XML 구조 정보를 표현하는 DTD를 분석하여 관계 스키마를 생성하여 XML 문서를 저장하고, 이에 대해 XML 표준 질의 언어 중의 하나인 XQuery로 질의할 수 있는 방법을 제안하고자 한다. 이를 위해 XQuery를 SQL로 변환하고 평면 구조로 저장된 XML 데이터를 계층적 구조인 XML 문서로 변환하는 XML 질의 처리 시스템을 설계한다.

2. XML 문서 저장

```
<!ELEMENT Paper ( PaperTitle, Authors, Reference? )>
<!ELEMENT PaperTitle ( #PCDATA )>
<!ELEMENT Authors ( Author+ )>
<!ELEMENT Author ( Name, Country?, University? )>
<!ELEMENT Name ( #PCDATA )>
<!ELEMENT Country ( #PCDATA )>
<!ELEMENT University ( #PCDATA )>
<!ELEMENT Reference ( Book*, Paper* )>
<!ELEMENT Book ( BookTitle, Authors )>
<!ELEMENT Authors ( Author+ )>
<!ELEMENT Author ( Name, Country, University )>
<!ELEMENT Name ( #PCDATA )>
<!ELEMENT Country ( #PCDATA )>
<!ELEMENT University ( #PCDATA )>
```

그림 1. DTD 예

2.1 관계 스키마 생성

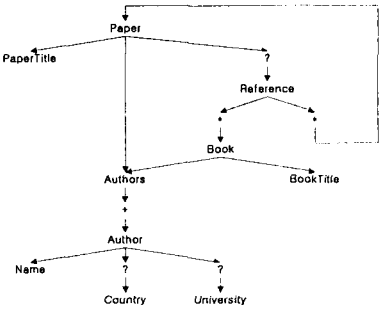

그림 2. DTD 그래프

관계 데이터베이스에 XML 문서를 저장하는 방법으로는 모델 사상 방법과 구조 사상 방법이 있다[7]. 본 논문에서는 XML 문서의 구조 정보를 나타내는 DTD를 분석하여 관계 스키마를 정의하는 구조 사상 방법에 의해 XML 문서를 저장한다. 특히, 기존의 공유 인라인 방법과 하이브리드 인라인 방법의 특성을 혼합한 연관 인라인 방법을 사용한다[3][6][9].

연관 인라인 방법에서 먼저 그림 2와 같이 DTD의 구조를 표현하는 DTD 그래프를 생성한다. DTD 그래프에서 노드는 엘리먼트, 애트리뷰트, 반복 지시자를 표현하고, 에지는 엘리먼트간의 관계를 표현한다[6]. 연관 인라인 방법은 DTD 그래프에서 노드의 in-degree의 값에 따라 엘리먼트의 관계 스키마 사상 방법을 결정한다. 즉, in-degree의 값이 0일 경우에는 릴레이션으로 사상하고, in-degree의 값이 1인 경우에는 상위 릴레이션의 속성으로 사상한다. in-degree의 값이 2인 경우에는 상위 엘리먼트와의 관계를 분석한다. 관계를 분석하는 기준은 값을 갖는 하위 엘리먼트의 개수이다. Authors 엘리먼트와 같이 값을 갖는 하위 엘리먼트가 2개 이상이면 독립된 개체로 간주하여 릴레이션으로 사상하고, 1개라면 종속적 개체로 간주하여 복수 개의 상위 릴레이션의 속성으로 분리하여 사상한다. 그러나, 다중값을 갖는 엘리먼트와 순환 구조를 갖는 엘리먼트는 in-degree의 값에 관계없이 별도의 릴레이션으로 사상한다. 그리고, 특정한 값을 가지지 않고 XML 문서의 구조만을 표현하는 엘리먼트는 관계 스키마로 사상하지 않는다. 그리고, 각 릴레이션은 기본키 역할을 하는 ID 속성을 갖고 다른 릴레이션과

* 본 연구는 정보통신부의 정보통신학술기초연구지원사업(정보통신연구진흥원)으로 연구되었음.

의 관계 표현을 위해 ParentID와 ParentCode 속성을 가지며 다중값을 갖는 경우 순서 정보의 유지를 위해 Order 속성을 추가한다. 그림 3은 그림 2의 DTD 그래프를 바탕으로 연관 인라인 방법에 의해 생성된 관계 스키마의 예이다.

```
Paper( ID, ParentID, ParentCode, Order, PaperTitle, docID )
Author( ID, ParentID, ParentCode, Order, Name, Country, University, docID )
Book(ID, Parent ID, Order, BookTitle )
```

그림 3. 관계 스키마 생성 예

2.2 패스 테이블

XML 표준 질의 언어인 XQuery는 XML 문서의 특정 정보를 가리키기 위해서 패스 표현식을 적용한다. 하지만, XQuery는 기존의 관계 데이터베이스에서 지원하지 않기 때문에 SQL로의 변환이 필요하다. 따라서, XQuery에서 사용되는 패스 표현식의 처리를 위해 발생할 수 있는 모든 경로 표현식을 그림 4와 같은 패스 테이블에 저장한다.

Path				
ID	PathExp	Table	Column	ParentCode
1	#/Paper	NULL	NULL	NULL
2	#/Paper#/PaperTitle	Paper	PaperTitle	NULL
3	#/Paper#/Authors	NULL	NULL	NULL
4	#/Paper#/Authors#/Author	NULL	NULL	NULL
5	#/Paper#/Authors#/Author#/Name	Author	Name	1
:	:	:	:	:

그림 4. 패스 테이블

3. XML 질의 처리

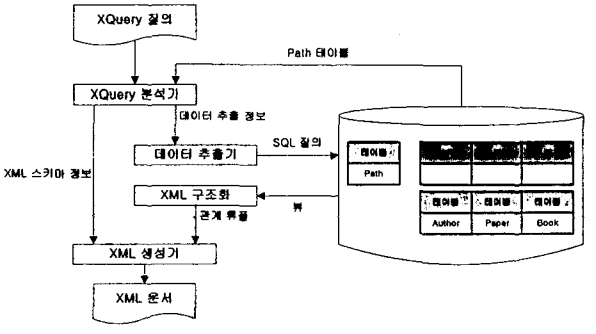

그림 5. XML 질의 처리 시스템 구성도

그림 5는 관계 데이터베이스를 이용한 XML 질의 처리 시스템의 구성도이다. 관계 데이터베이스의 여러 테이블에 분산 저장된 XML 데이터에 대한 질의를 사용자는 XQuery로 작성한다.

3.1 XQuery 분석

XQuery 분석기는 XQuery로 작성된 사용자 질의를 파싱을 통해 분석한다. 이 때, XQuery 분석기는 XML 데이터의 추출을 위한 SQL문 생성에 사용되는 정보와 구조화된 관계 튜플과 함께 XML 문서 생성에 사용되는 XML 구조 정보를 분리하여 얻는다.

3.1.1 데이터 추출을 위한 정보

XQuery로 작성된 질의는 관계 데이터베이스에서 지원되지 않기 때문에 SQL 형태로 변환해야 한다. 질의 변환시 XQuery의 경로 표현식에서 "/"는 "#/", "//"는 "#%/"로 변환하여 그림 4의 패스 테이블을 이용한다. 이 변환과정은 SQL의 LIKE문을 이용할 수 있도록 하여 XQuery 경로 표현식의 간편한 처리가 가능해진다[7].

그림 6은 XQuery로 작성된 질의이다. 본 논문에서는

```
for $p in document("paper.xml")//Paper
return
   <Paper>
      <PaperTitle>$p/PaperTitle</PaperTitle>
      <Authors>
         { for $a in $p/Authors/Author
           return
              <Author>
                 <Name>$a/Name</Name>
              </Author>}
      </Authors>
      <Reference>
         for $b in $p/Reference/Book
         return
            <Book>
               <BookTitle>$b/BookTitle</BookTitle>
               <Authors>
                  { for $ba in $b/Author
                    return
                       <Author>
                          <Name>$ba/Name</Name>
                       <Author>}
               </Authors>
            </Book>
         for $pp in $p/Reference/Paper
         return
            <Paper>
               <PaperTitle>$pp/PaperTitle</PaperTitle>
               <Authors>
                  { for $ppa in $pp/Authors/Author
                    return
                       <Author>
                          <Name>$ppa/Name</Name>
                       </Author>}
               </Authors>
            </Paper>
      </Reference>
   </Paper>
```

그림 6. XQuery로 작성된 질의

XQuery의 FLWOR 표현식에 기반한 질의에 대하여 설명한다. (1)은 그림 6의 XQuery 질의에서 사용된 경로 표현식 중 //Paper/PaperTitle과 //Paper/Authors/Author/Name 경로 표현식을 이용해 테이블과 컬럼 정보를 추출하는 SQL 문이다.

SELECT Table, Column, ParentCode (1)
FROM Path
WHERE PathExp LIKE "#%/Paper#/PaperTitle"

SELECT Table, Column, ParentCode (2)
FROM Path
WHERE PathExp LIKE "#%/Paper#/Authors/Author/Name"

(1)과 (2)와 같은 SQL 문에 의해 추출된 정보는 데이터 추출기로 전달되어 XML 데이터 추출시 이용된다.

3.1.2 XML 구조 정보

XQuery 분석기에 의해 추출되는 XML 구조 정보는 추출된 XML 데이터가 XQuery의 return 절에 의해 명시된 형태의 구조로 XML 문서를 생성하기 위해 사용된다. XML 구조 정보는 그림 7과 같은 형태의 배열로 저장되어 추출되는 관계 튜플과 함께 태깅된다.

0	1	2	3	4	…
For	<Paper>	<PaperTitle>	Data(1,PaperTitle)	</PaperTitle>	…

9	10	11	12	13	…
Data(2,Name)	</Name>	</Author>	EndFor(6)	</Authors>	…

:

…	45	46	47	48	49
…	</Paper>	ForEnd(31)	</Reference>	</Paper>	ForEnd(0)

그림 7. XML 스키마 정보

3.2 데이터 추출

데이터 추출기는 사용자가 원하는 XML 데이터의 추출을 위하여 XQuery 분석에서 얻은 데이터 추출 정보를 이용하여 뷰를 생성한다. 3.1.1절의 (1)과 (2)의 SQL 문에 의해 추출된 테이블과 컬럼, ParentCode 속성 정보를 이용하여 뷰를 생성하

는 SQL 문은 다음과 같다.

CREATE VIEW Paper_View(PaperId, PaperTitle) AS (3)
SELECT PaperId, PaperTitle
FROM Paper

CREATE VIEW Author_View(PaperId, AuthorId, Name) AS
SELECT P.PaperId, A.AuthorId, A.Name (4)
FROM Paper_View P, Author A
WHERE P.PaperId = A.ParentId AND A.ParentCode = 1

(3)과 (4)의 SQL 문에 의해 생성되는 뷰의 속성을 살펴보면, 각 테이블의 ID 속성, 테이블 구조상 상위 테이블들의 ID 속성들, 사용자가 원하는 데이터 속성 등으로 구성된다. 이처럼, 뷰의 생성과 생성되는 뷰의 속성은 Node Outer Union 방법이 응용에 의해 결정한다. 그림 8은 Node Outer Union 방법에 의해 XML 데이터를 추출하는 SQL 문의 실행과정을 표현하고 있다[1][5].

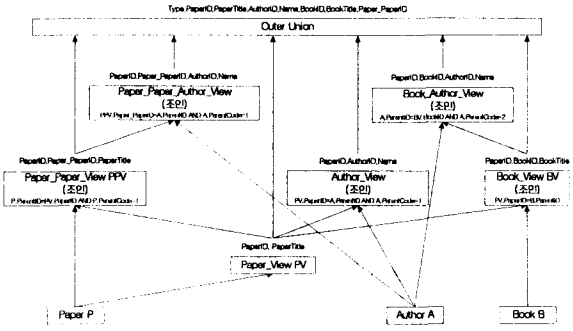

그림 8. 데이터 추출을 위한 SQL 실행 과정

3.3 XML 구조화

데이터 추출기에 의해 생성된 사용자가 원하는 데이터들로 XML 문서를 생성하기 위해서는 먼저, 사용자가 원하는 XML 문서 형태의 구조를 가져야 한다. 그러나, 데이터 추출기에 의해 생성된 결과 튜플들은 평면 구조의 관계 릴레이션의 튜플 형태이기 때문에 XML 문서 형태의 계층적 구조가 아니다. 따라서, XML 형태의 구조화를 위해 결과 튜플들에 대한 정렬 연산을 수행한다. 이 정렬 연산은 Sorted Outer Union 방법의 정렬 우선 순위 규칙을 따른다[5]. 그 정렬 우선 순위 규칙을 적용하면 PaperID, Paper_PaperID, BookID, AuthorID의 순서로 높은 정렬 우선 순위가 결정된다. 그림 9는 XML 구조화를 위한 SQL 문이다. 여기서 Type 속성은 XML 생성기에서 튜플들이 어떤 뷰의 튜플들인지를 구별할 수 있도록 하기 위한 속성이다.

```
SELECT Type,PaperID,PaperTitle,null AS AuthorID,null AS Name,
null AS BookID,null AS BookTitle,null AS Paper_PaperID
FROM Paper_View
UNION ALL
SELECT Type, PaperID, null, AuthorID, Name, null, null, null
FROM Author_View
UNION ALL
SELECT Type, PaperID, null, null, null, BookID, BookTitle, null
FROM Book_View
UNION ALL
SELECT Type, PaperID, null, AuthorID, Name, BookID, null, null
FROM Book_Author_View
UNION ALL
SELECT Type, PaperID, PaperTitle, null, null, null, null,
Paper_PaperID
FROM Paper_Paper_View
UNION ALL
SELECT Type, PaperID, null, AuthorID, Name, null, null,
Paper_PaperID
FROM Paper_Paper_Author_View
ORDER BY PaperID, Paper_PaperID, BookID, AuthorID
```

그림 9. 결과 튜플의 XML 구조화를 위한 SQL

3.4 XML 생성

XQuery 질의 분석에서 추출된 XML 구조 정보와 XML 구조화 과정에서 생성된 관계 튜플을 결합하여 사용자가 원하는 형태의 XML 문서를 생성한다. XQuery에서 표현된 XML 구조 정보를 저장한 배열을 순차적으로 순회하면서 XML 문서를 생성한다. XML 구조 정보를 저장한 배열에서 태그는 태그 생성을 위해서 사용되고 XML 데이터가 필요한 경우에는 추출된 관계 튜플의 Type 속성을 비교하여 순차적으로 데이터를 삽입한다.

4. 결론

본 논문에서는 XML 문서를 관계 데이터베이스에 저장하고 저장된 XML 데이터에 대한 XQuery 질의를 처리하여 XML 문서를 효과적으로 생성하는 시스템을 설계하였다. 평면 구조를 갖는 관계 데이터베이스와 계층 구조를 갖는 XML 문서는 구조적으로 일치하지 않고 관계 데이터베이스에서는 XML 질의 언어인 XQuery를 직접 지원하지 않기 때문에 별도의 작업이 필요하다. 따라서, 본 논문에서는 패스 테이블을 이용하여 XQuery의 경로 표현식을 SQL로 변환하고 여러 테이블에 분할 저장된 XML 데이터를 효과적으로 추출할 수 있도록 하였다. 그리고 XQuery 질의에서 추출된 XML 문서 구조 정보를 이용하여 태깅하는 방법을 제안하였다.

참고문헌

[1] M. Carey, D. Florescu, Z. Ives, Y. Lu, J. Shanmugasundaram, E. Shekita, S. SuM. Carey, D. Florescu, Z. Ives, Y. Lu, J. Shanmugasundaram, E. Shekita, S. Subramanian, "XPERANTO : Publishing Object-Relational Data as XML", *WebDB Workshop*, Dallas, 2000

[2] M. Fernandez, W. Tan, D. Suciu, "SilkRoute : Trading Between Relations and XML", *Proceedings of the International World Wide Web Conference*, 2000

[3] D. Florescu, D. Kossmann, "Storing and Querying XML Data using an RDBMS", *Data Engineering Bulletin*, Vol. 22, No. 3, 1999

[4] J. Shanmugasundaram, J. Kiernan, E. Shekita, C. Fan, J. Funderbu가, "Querying XML Views of Relational Data", *Proceedings of the 27th VLDB Conference*, Roma, 2001

[5] J. Shanmugasundaram, E. Shekita, R. Barr, M. Carey, B. Lindsay, H. Pirahesh, B. Reinwald, "Efficiently Publishing Relational Data as XML Documents", *VLDB*, 2000

[6] J. Shanmugasundaram, K. Tufte, G. He, C. Zhang, D. DeWitt, J. Naughton, "Relational Databases for Querying XML Documents : Limitations and Opportunities", *VLDB*, Edinburg, Scotland, 1999

[7] M. Yoshikawa, T. Amagasa, "XRel: A Path-Based Approach to Storage and Retrieval of XML Documents Using Relational Databases", ACM Transactions on Internet Technology, Vol. 1, No. 1, August 2001

[8] World Wide Web Consortium, *XQuery 1.0: An XML Query Language, W3C Working Draft*, November 2002. *http://www.w3c.org/TR/xquery*

[9] 신병주, 송청, 진민, "관계 데이터베이스를 이용한 XML 문서 저장 및 검색", 한국멀티미디어학회 추계학술발표논문집 제5권 제2호, 2002