

관계 데이터베이스를 이용한 XML 문서 저장 및 검색

신병주, 송 청, 진 민
경남대학교 컴퓨터공학과

Storing and Querying XML Documents Using Relatioanl Databases

Byung-Joo Shin, Chung Song, Min Jin
Dept. of Computer Engineering, Kyungnam University

요 약

본 논문은 DTD가 있는 이용한 XML 문서의 관계 데이터베이스 저장 및 검색방법을 제안한다. XML 구조와 관계 데이터베이스 스키마의 구조적 불일치로 인하여 XML 문서의 저장을 위해서는 별도의 처리과정이 필요하다. 기존의 Shared Inling 방법과 Hybrid Inling 방법의 장점을 살리고 릴레이선의 단편과 과도한 조인을 처리하기 위해 Association Inling 방법을 제안한다. 또한, XQuery를 이용한 질의를 지원하기 위해 Association Inling 방법에 의해 생성된 저장 스키마를 확장하고, XQuery로 표현된 질의를 SQL로 변환한다.

1. 서론

XML은 현재 정보통신망의 발달과 함께 인터넷 상에서 데이터 표현 및 교환의 표준으로 중요한 위치를 차지하게 되었다. 이에 따라, 전세계적으로 XML을 지원하기 위한 다양한 연구들이 진행중이고, 많은 XML 지원시스템들이 개발되었다. 그 중에서도 XML을 효율적으로 데이터베이스에 저장, 관리하기 위한 연구가 많이 진행되고 있다.

XML 저장시스템으로 다양한 시스템이 사용되어지고 있다. 특히, 그 중에서 관계형 데이터베이스 관리 시스템은 기존의 데이터 뿐만 아니라, XML 데이터와 같은 반구조적 데이터가 공존할 수 있고 대용량 데이터베이스에 대해 복잡한 질의 처리 속도가 타 저장시스템들에 비해 우수하다. 또한, 널리 사용되어지고 있다는 장점 등이 있어서 XML을 관계형 데이터베이스 관리 시스템에 저장 및 관리하기 위한 연구가 가장 활발히 진행되고 있다[1][3][7].

그러나, XML 데이터를 관계형 데이터베이스 관리 시스템에 저장 및 질의처리하기 위해서는 별도의 처리과정이 필요하다. 왜냐하면, XML의 계층적 구조와

관계 데이터베이스의 평면적인 스키마 구조가 일치하지 않기 때문이다. 따라서, XML 문서를 관계 데이터베이스에 효율적으로 저장하기 위해서는 데이터의 단편화 문제, 질의 처리의 효율성 저하 문제 등이 해결되어야 한다[3].

지금까지 XQL, XML-QL, Quilt, XQuery 등 다양한 XML 질의 언어가 개발되었으나 최근에는 XQuery로 통합되어가고 있는 추세이다. 따라서, 본 논문에서는 XML 질의 언어로 XQuery를 사용한다. 그러나, 기존의 관계 데이터베이스는 XML 표준 질의 언어인 XQuery를 지원하지 못하기 때문에 관계 데이터베이스에 저장된 XML 데이터에 대한 질의 처리를 위해서는 XQuery의 효율적인 SQL로의 변환 과정이 필요하다[4][6][7][8].

따라서, 본 논문은 XML 데이터를 관계 데이터베이스에 저장함에 있어 구조적 차이로 인해 발생하는 문제점을 해결하고, XML 표준 질의 언어인 XQuery를 지원하는 저장 스키마 생성 방법을 제안한다. 이 저장 스키마 생성 방법은 DTD가 있는 XML 문서를 대상으로 하며, DTD로부터 기본적인 관계 데이터베이스 스키마를 추출하고 XQuery 지원을 위해 이를 확장한다.

본 연구는 정보통신부의 정보통신기술기초연구지원사업(정보통신연구진흥원)으로 연구되었음.

본 논문의 구성은 다음과 같다. 2장에서는 XML 데이터를 관계 데이터베이스에 효율적으로 저장하고 XQuery의 지원을 위한 관계 스키마 생성방법을 제안하고, 3장에서는 XQuery로 작성된 질의를 SQL로 변환하는 과정을 나타낸다. 마지막으로 4장에서는 결론을 맺는다.

2. 관계 스키마의 생성

2.1 Association Inling

```

<ELEMENT Book (BookTitle, Author)>
<ELEMENT BookTitle (#PCDATA)>
<ELEMENT Author (Name, Address)>
<!ATTLIST AuthorID ID #REQUIRED>
<ELEMENT Name (#PCDATA)>
<ELEMENT FirstName (#PCDATA)>
<ELEMENT LastName (#PCDATA)>
<ELEMENT Address (#PCDATA)>
<ELEMENT Article (Title, Author*, ContactAuthor)>
<ELEMENT ContactAuthor EMPTY>
<!ATTLIST ContactAuthor AuthorID IDREF IMPLIED>
<ELEMENT Author (Name, Address)>
<!ATTLIST AuthorID ID #REQUIRED>
<ELEMENT Name (#PCDATA)>
<ELEMENT FirstName (#PCDATA)>
<ELEMENT LastName (#PCDATA)>
<ELEMENT Address (#PCDATA)>
<ELEMENT Monograph (Title, Author, Editor)>
<ELEMENT Editor (Monograph*)>
<!ATTLIST Editor Name CDATA #REQUIRED>
<ELEMENT Title (#PCDATA)>
<ELEMENT Author (Name, Address)>
<!ATTLIST AuthorID ID #REQUIRED>
<ELEMENT Name (#PCDATA)>
<ELEMENT FirstName (#PCDATA)>
<ELEMENT LastName (#PCDATA)>
<ELEMENT Address (#PCDATA)>
    
```

그림 1. DTD 예

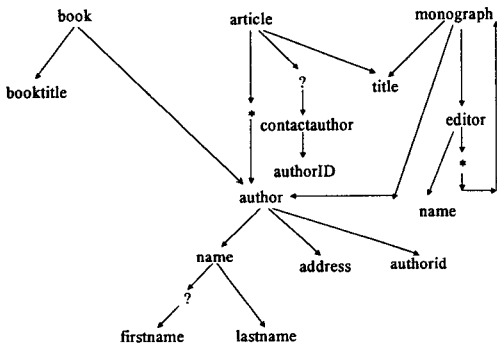


그림 2. DTD 그래프

XML 문서의 엘리먼트와 속성을 관계 스키마로 표현하기 위하여 기존의 Shared Inling 방법과 Hybrid Inling 방법[3]의 장점을 수용한 Association Inling

<pre> <Book> <BookTitle>XML</BookTitle> <Author AuthorID="1"> <Name> <FirstName>Byung-Joo</FirstName> <LastName>Shin</LastName> </Name> <Address>Kimhae</Address> </Author> </Book> <Article> <Title>Oracle</Title> <Author AuthorID="2"> <Name> <FirstName>Jong-Myung</FirstName> <LastName>Kim</LastName> </Name> <Address>Masan</Address> </Author> <Author AuthorID="3"> <Name> <FirstName>Chung</FirstName> <LastName>Song</LastName> </Name> <Address>Busan</Address> </Author> <ContactAuthor AuthorID="1" /> </Article> <Monograph> <Title>Silkroute</Title> <Author AuthorID="4"> <Name> <FirstName>Dong-Jin</FirstName> <LastName>Lee</LastName> </Name> <Address>Masan</Address> </Author> </pre>	<pre> <Editor Name="Shin"> <Monograph> <Title>XPERANTO</Title> <Author AuthorID="1"> <Name> <FirstName>Byung-Joo</FirstName> <LastName>Shin</LastName> </Name> <Address>Kimhae</Address> </Author> <Editor Name="Jin" /> <Monograph> <Title>eXcelon</Title> <Author AuthorID="2"> <Name> <FirstName>Jong-Myung</FirstName> <LastName>Kim</LastName> </Name> <Address>Masan</Address> </Author> <Editor Name="Kim"> <Monograph> <Title>Tamino</Title> <Author AuthorID="3"> <Name> <FirstName>Chung</FirstName> <LastName>Song</LastName> </Name> <Address>Busan</Address> </Author> <Editor Name="Song" /> <Monograph> <Editor> <Monograph> </Editor> </pre>
--	--

그림 3. XML 문서의 예

방법을 제안한다[9][10].

관계 데이터베이스에 XML 문서를 효율적으로 저장하기 위한 저장 스키마의 생성을 위해 먼저 DTD의 구조를 표현하는 DTD 그래프를 생성한다. <그림 2>는 <그림 1>의 DTD의 구조를 표현한 DTD 그래프이다. DTD 그래프에서는 엘리먼트, 애트리뷰트, 반복지시자 등을 노드로 표현하고 각 엘리먼트와 애트리뷰트 간의 관계는 에지로 나타낸다. 그리고, 노드 표현시 DTD 상에서 같은 이름을 가진 엘리먼트는 한 노드로 표현되고, 애트리뷰트는 이름과 관계 없이 DTD에 나타나는대로 DTD 그래프에 표현된다[3].

Association Inling 방법에서 XML 문서의 엘리먼트들을 관계 스키마의 릴레이션이나 속성으로 사상하는 기준은 DTD 그래프에서 노드의 in-degree 값이다. <그림 2>의 DTD 그래프에서 in-degree 값이 0 일 경우에는 상위 엘리먼트가 존재하지 않으므로 릴레이션으로 사상한다. 그리고 in-degree 값이 1인 경우에는 상위 엘리먼트에 포함시켜 관계 스키마의 속성으로 사상한다[3].

in-degree 값이 2 이상일 경우에는 별도의 릴레이션을 생성하는 것을 원칙으로 하지만 예외적인 경우가 있다. in-degree 값이 2 이상일지라도 단말 노드라면 상위 엘리먼트에 속성으로 표현한다. 그리고,

in-degree 값이 2 이상일 경우에는 상위 엘리먼트와 하위 엘리먼트의 상관관계의 분석을 통해 릴레이션 또는 속성으로 표현한다. 다중값을 갖는 엘리먼트와 순환 구조를 갖는 엘리먼트는 *in-degree*의 값과는 무관하게 별도의 릴레이션을 생성하여 표현한다[9][10].

복수 개의 상위 엘리먼트를 가지거나 다중값을 갖는 엘리먼트, 순환구조를 갖는 엘리먼트 등을 릴레이션으로 사상할 경우 다른 릴레이션과 관계를 갖게 된다. 다른 릴레이션과의 관계 표현을 위해 *parentID*와 *parentCode*, *Index* 속성 등을 추가한다[3][5].

<그림 4>는 <그림 1>의 DTD에 의한 *Association Inling* 방법에 의해 관계 테이블이 생성되고, <그림 3>의 XML 문서가 저장된 예이다.

Book			Article			
BookID	BookTitle	DocID	ArticleID	Title	AuthorID	DocID
1	XML and Java	1	1	Pear-to-Pear XML	1	1

Monograph						
MonographID	ParentID	ParentCode	Index	Title	Name	DocID
1	0	0	1	SilkRoute	Shin	1
2	1	1	1	XPERANTO	Jin	1
3	1	1	2	eXcelon	Kim	1
4	3	2	1	Tamino	Song	1

Author							
AuthorID	ParentID	ParentCode	Index	FirstName	LastName	Address	DocID
1	1	1	1	Byung-Joo	Shin	Kimhae	1
2	1	2	1	Jong-Myung	Kim	Masan	1
3	1	2	2	Chung	Song	Busan	1
4	1	3	1	Dong-Jin	Lee	Masan	1
1	2	3	1	Byung-Joo	Shin	Kimhae	1
2	3	3	1	Jong-Myung	Kim	Masan	1
3	4	3	1	Chung	Song	Busan	1

그림 4. 테이블 생성과 XML 문서 저장의 예

2.2 XQuery 지원을 위한 관계 스키마의 확장

본 논문에서는 다양한 XML 질의 언어로 XQuery를 사용한다. 따라서, XQuery를 지원하기 위해 *Association Inling*에 의해 생성되는 저장스키마를 확장한다.

먼저, XQuery에서는 구조화된 XML 문서의 특정 정보를 가리키기 위해서 패스 표현식을 사용한다. 따라서, XQuery로 작성된 질의문에서 사용되는 패스 표현식을 *Path* 테이블에 나타낸다. 이 *Path* 테이블에는 XML 문서에서 나타날 수 있는 모든 패스 정보를 저장하고, 각 패스가 가리키는 정보가 저장된 곳에 대한 정보를 표현한다. <그림 5>는 <그림 1>의 DTD에서 나타날 수 있는 모든 패스 정보를 저장한 예이다. 단, 순환 구조를 갖는 엘리먼트에 대한 패스 정보는 순환 구조의 발생시에 추가적으로 *Path* 테이블에 그 패스

정보를 입력한다[2][7].

Path Table

PathID	PathExp	Table	Column	ParentCode
1	#/Book#/BookTitle	Book	BookTitle	0
2	#/Book#/Author#/@AuthorID	Author	AuthorID	1
3	#/Book#/Author#/Name#/FirstName	Author	FirstName	1
4	#/Book#/Author#/Name#/LastName	Author	LastName	1
5	#/Book#/Author#/Address	Author	Address	1
6	#/Article#/Title	Article	Title	0
7	#/Article#/Author#/@AuthorID	Author	AuthorID	2
8	#/Article#/Author#/Name#/FirstName	Author	FirstName	2
9	#/Article#/Author#/Name#/LastName	Author	LastName	2
10	#/Article#/Author#/Address	Author	Address	2
11	#/Article#/ContactAuthor#/@AuthorID	Article	AuthorID	2
12	#/Monograph#/Title	Monograph	Title	0
13	#/Monograph#/Author#/@AuthorID	Author	AuthorID	3
14	#/Monograph#/Author#/Name#/FirstName	Author	FirstName	3
15	#/Monograph#/Author#/Name#/LastName	Author	LastName	3
16	#/Monograph#/Author#/Address	Author	Address	3
17	#/Monograph#/Editor#/@Name	Monograph	Name	0

그림 5. 패스 표현식의 저장을 위한 *Path* 테이블

3. XQuery의 SQL로의 변환

XML 표준 질의 언어인 XQuery는 기존의 관계 데이터베이스에서 지원되지 못하고 있다. 따라서, XQuery로 작성된 질의는 관계 데이터베이스에 저장된 XML 데이터에 대해 질의 처리하기 위해서는 SQL 질의의 형태로 변환되어야 한다. 질의 변환시에 필요한 XQuery의 경로 표현식은 약간의 변환 과정이 필요하다. <그림 5>의 *Path* 테이블에서 보는 것처럼, 경로 표현식은 “#”로 각 노드들을 구분하고 있다. 이는 경로 표현식에서 나타날 수 있는 “//”과 같은 경로 표현식에 대한 처리를 위해 사용하고 있다. 즉, XQuery로 작성된 질의에서 “/”는 “#”로, “//”는 “#%/”로 변환한다. 이 변환과정과 SQL 문의 *LIKE* 문을 이용하면 간편한 질의 변환이 가능하게 된다[7].

다음과 같은 XQuery 예제를 보자.

```
FOR $b IN document("A.xml")/Book
WHERE $b//FirstName/text()="Byung-Joo"
RETURN
  <Book>
    <BookTitle>$p/BookTitle</BookTitle>
  </Book>
```

위의 예제에서는 두 개의 경로 표현식이 나타난다. 즉, */Book//FirstName(text)="Byung-Joo"*와 */Book/BookTitle*의 두 개의 경로 표현식이 나타나고 있다. 이 경로 표현식은 위에서 언급한 것처럼 변환되어 다음과 같은 SQL 질의가 생성된다.

```
SELECT Table, Column, ParentCode
```

```

FROM Path
WHERE PathExp LIKE "#/Book#/FirstName"

SELECT Table, Column, ParentCode
FROM Path
WHERE PathExp LIKE "#/Book#/BookTitle"
    
```

(1)과 (2)의 질의문에 의해 WHERE 조건절에 해당하는 경로 표현식은 Author 테이블의 FirstName 컬럼을 검색하고, 데이터 추출 컬럼은 Book 테이블의 BookTitle 컬럼이라는 정보를 획득하게 된다. 이 추출된 정보를 바탕으로 최종적으로 (3)과 같은 SQL 질의문이 생성된다.

```

SELECT B.BookTitle
FROM BookTitle B, Author A
WHERE A.FirstName = "Byung-Joo"
    
```

그리고, XQuery의 AFTER, BEFORE 연산과 RANGE 연산 등은 Path 테이블의 Start와 End 컬럼의 값과 각 테이블의 Index 컬럼의 값의 비교 연산에 의해 처리한다.

4. 결론

XML 문서와 관계 데이터베이스의 구조적 불일치로 인해 XML 문서를 관계 데이터베이스에 저장할 경우 별도의 처리과정이 필요하다. 또한, 표준 질의 언어인 XQuery는 기존의 관계 데이터베이스에서 지원해 주지 못하고 있다. 본 논문은 DTD 기반의 XML 문서를 관계 데이터베이스에 저장하고 질의 처리하는 방법을 제안하였다.

XML 문서와 관계 데이터베이스의 구조적 불일치 문제를 해결하기 위해 Association Inling 방법에 의해 관계 스키마를 생성하였고, 기존의 관계 데이터베이스가 XQuery를 지원하지 못하기 때문에 관계 스키마의 확장과 SQL로의 효율적 변환을 나타내었다.

향후 XML 데이터의 구조정보에 대한 질의의 효율적인 처리를 위해 구조정보를 저장하는 인덱스 스킵키의 결합 방안에 대한 연구가 필요하다. 또한, DTD 그래프에서 in-degree 값이 2 이상인 엘리먼트의 관계 스키마로의 사상 방법을 결정하는 체계적인 연구가 필요하다.

[참고문헌]

[1] D. Florescu, D. Kossmann, "Storing and

Querying XML Data using an RDBMS", Data Engineering Bulletin, Vol. 22, No. 3, 1999

[2] I. Tatarinov, S. D. Viglas, K. Beyer, J. Shanmugasundaram, E. Shekita, C. Zhang, "Storing and Querying Ordered XML Using a Relational Database System, ACM SIGMOD, Madison, Wisconsin, June 2002

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

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

[5] K. Williams, M. Brundage, P. Dengler, J. Gabriel, A. Hoskinson, M. Kay, T. Maxwell, M. Ochoa, J. Papa, M. Vanmane, Professional XML Databases, Wrox Press, 2000

[6] M. F. Fernandez, A. Morishima, D. Suciu, W. Tan, "Publishing Relational Data as XML : The SilkRoute Approach", IEEE Data Engineering Bulletin 24(2), 2001

[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] S. Boag, D. Chamberlin, M. F. Fernandez, D. Florescu, J. Robie, J. Simeon, "XQuery 1.0 : An XML Query Language", www.w3.org/TR/xquery

[9] 신병주, 진민, 정민수, "XML 문서의 관계 데이터베이스 저장", 한국정보처리학회 추계학술발표논문집 제8권 제2호, pp.55-58, 2001

[10] 신병주, 진민, "XML 구성요소의 릴레이션으로의 변환", 한국정보처리학회 추계학술발표논문집 제8권 제2호, pp.35-38, 2001