

# XML 실체 뷰를 이용한 XQL 질의 변환\*

김수희<sup>0+</sup> 문찬호<sup>+</sup> 김영성<sup>+</sup> 강현철<sup>+</sup> 서상구<sup>++</sup>  
<sup>+</sup>중앙대학교 컴퓨터공학과, <sup>++</sup>광운대학교 경영정보학과  
<sup>\*</sup>{shkim, moonch, yskim, hckang}@rose.cse.cau.ac.kr, <sup>++</sup>skseo@daisy.gwu.ac.kr

## Transformation of XQL Queries Using XML Materialized Views

SooHee Kim<sup>0+</sup> ChanHo Moon<sup>+</sup> YoungSung Kim<sup>+</sup> Hyunchul Kang<sup>+</sup> Sang-Koo Seo<sup>++</sup>  
<sup>+</sup>Dept. of Computer Science and Engineering, Chung-Ang University  
<sup>++</sup> Dept. of MIS, Kwangwoon University

### 요 약

XML과 관련하여 XML 문서의 구조적인 특징을 고려한 XML 질의에 관한 연구가 활발히 수행되고 있다. 이들 연구들은 하나의 XML 문서 또는 XML 저장소에 저장된 여러 XML 문서들을 대상으로 효과적인 검색을 위한 XML 질의 언어의 개발이나 확장을 중심으로 연구되고 있다. 한편 XML 문서 검색의 성능 향상을 위해서 XML 저장소에는 XML 문서들 외에 그들로부터 도출된 XML 뷰를 실체 뷰로 저장해 둘 수 있다. 본 논문에서는 XQL 질의의 결과로 얻어진 XML 실체 뷰가 있다고 가정하고 이를 이용한 XQL 질의 변환에 대해 연구하였다. 이를 위해 본 논문에서는 XML 실체 뷰 영역을 포함하는 XML 저장소의 구조를 설명하고, XQL 질의의 결과와 XML 실체 뷰 간의 포함 관계를 분류하며, 그에 따른 XQL 질의 변환 알고리즘을 제안한다.

### 1. 서론

XML(eXtensible Markup Language) 이 차세대 웹문서의 표준 마크업 언어로 정해진 이후, XML과 관련하여 XML 문서로부터 사용자가 원하는 정보를 얻기 위한 XML 질의 언어들에 대한 연구가 활발히 수행되고 있다. XML 문서를 질의하기 위한 질의 언어로는 XQL, XML-QL, Lorel, UnQL 등이 있으며[1], 이 중 XQL(XML Query Language)은 1998년 W3C에 의해 제안된 범용적인 질의 언어이다[2][3]. XQL은 하나의 XML 문서 또는 XML 저장소에 저장된 여러 XML 문서를 대상으로 원하는 엘리먼트를 얻기 위해 경로 표현(path expression)과 연산자를 이용하여 질의를 작성하며, 결과로서 XML 문서를 반환한다.

본 논문에서는 XML 저장소에 저장된 여러 XML 문서를 대상으로 하는 XQL 질의 처리의 성능 향상을 위해 XQL 질의의 결과로 얻어진 XML 문서를 저장하고 있는 XML 실체 뷰(materialized view)가 있다고 가정하고, 이를 이용한 XQL 질의 변환에 대해 기술한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련 연구를 기술하며, 3절에서는 본 논문에서 제시하는 XML 저장 구조에 대해 설명한다. 4절에서는 XQL 질의 결과와 XML 실체 뷰 간의 포함 관계를 분류하고, 이에 따른 XQL 질의 변환 과정에 대해서 설명한다. 5절에서는 결론을 맺는다.

### 2. 관련 연구

질의 최적화와 물리적 데이터의 독립성 유지, 데이터 통합 등의 다양한 데이터 관리 문제와 관련하여, 주어진 질의를 처리하는 데 있어 실체 뷰를 이용하기 위한 연구들은 꾸준히 진행되어 왔다[4]. 이러한 연구들은 주로 관계형 데이터베이스를 대상으로 이루어져 왔지만, 최근에는 반구조적 데이터(semistructured data) 모델을 대상으로 많은 질의와 뷰 정의 언어들이 제안되고 있고 이에 관련된 질의 처리 방안에 관한 연구들도 진행되고 있다. [5]에서는 반구조적 데이터를 대상으로 주어진 TSL 질의를 반구조적 뷰(semistructured view)에 맞게 변환하는 TSL 질의 변환 문제를 다루고 있다.

### 3. XML 문서의 저장 구조

본 절에서는 먼저 본 논문의 기본적인 가정을 기술한 후, XML 문서의 저장 구조에 대해 설명한다.

#### 3.1 가정

본 논문에서는 XML 실체 뷰와 사용자로부터 주어지는 XQL 질의에 대해 다음과 같은 가정을 한다.

- XQL 질의는 XML 저장소에 저장된 여러 XML 문서들을 대상으로 하며, XML 저장소에 저장된 모든 XML 문서는 공통된 하나의 DTD를 만족한다.

\* 본 논문은 한국 과학 재단의 특정 기초 연구 지원 사업에 의 한 것임.

- XML 실제 뷰는 XML 저장소에 대한 XQL 질의를 통해 얻어진 것으로 고정적이다.
- 사용자로부터 주어지는 XQL 질의는 경로 표현(path expression)만을 허용하고 그 외의 연산자는 허용하지 않는다. 따라서 가능한 XQL 질의는 /book/category 또는 /book//author, /book/\*/title과 같은 형태로 제한된다.

3.2 XML 저장소의 구성

본 논문에서 가정하는 XML 문서의 저장 구조는 크게 XML 저장소의 하부 XML 문서 영역과 XML 실제 뷰 영역으로 나누어 볼 수 있다. 각 영역의 XML 문서는 엘리먼트 단위로 분할되어 엘리먼트 테이블과 뷰 엘리먼트 테이블에 각각 저장된다. (그림 2)는 (그림 1)의 XML 문서를 저장한 예를 보여준다. 이들을 각각 설명하면 다음과 같다.

```

<Catalog>
  <RecSubjCategories>
    <Category>XML</Category>
    <Category>Web Publishing</Category>
    <Category>Internet</Category>
  </RecSubjCategories>
  <Book>
    <Title>IE5 XML Programmer's Reference</Title>
    <Pages>480</Pages>
    <ISBN>1-861001-57-6</ISBN>
  </Book>
  ...
</Catalog>
    
```

(그림 1) XML 문서의 예

DTDID	Ename
01	Catalog
0101	SubjCategories
010101	Category
0102	Book
010201	Title
010202	Pages
010203	ISBN

(a) DTD 테이블

DID	DTDID	EID	Ename	Contents
1	01	01	Catalog	-
1	0101	0101	SubjCategories	-
1	010101	010101	Category	XML
1	010101	010102	Category	Web Publishing
1	010101	010103	Category	Internet
1	0102	0102	Book	-
1	010201	010201	Title	IE5 XML Programmer's Reference
1	010202	010202	Pages	480
1	010203	010203	ISBN	1-861001-57-6
...	...	...	...	...

(b) 엘리먼트 테이블

BaseDID	ViewDID	DTDID	EID	Ename	Contents
1	1	0101	01	Book	-
1	1	010101	0101	Title	IE5 XML Programmer's Reference
1	1	010102	0102	Pages	480
1	1	010203	0103	ISBN	1-861001-57-6
...	...	...	...	...	...

(c) 뷰 엘리먼트 테이블

BaseDID	ViewDID	DTDID	Path	BaseEID	ViewEID
1	1	0101	Catalog/Book	0102	01
...	...	...	...	...	...

(d) EID 매핑 테이블

(그림 2) XML 문서 저장 구조의 테이블 구성

- DTD 테이블 : (그림 2)의 (a)와 같이 XML 저장소의 모든 XML 문서가 만족하는 DTD의 구조를 트리 형태로 나타내었을 때, 각 엘리먼트에 대해 루트 엘리먼트로부터의 위치를 표시하는 DTDID(DTD ID)와 해당 엘리먼트의 이름으로 구성된다.

- 엘리먼트 테이블 : XML 저장소의 모든 XML 문서를 (그림 2)의 (b)와 같이 엘리먼트 단위로 분할하여 저장한다. 엘리먼트 각각은 XML 문서를 구분하는 DID(Document ID)와 XML 문서의 트리 구조에서의 위치를 표시하는 EID (Element ID)[6], 엘리먼트의 DTD를 나타내는 DTDID와 엘리먼트 이름, 내용으로 구성된다.

- 뷰 엘리먼트 테이블 : 미리 정의된 XQL 질의의 결과로 얻어진 XML 실제 뷰를 엘리먼트 테이블과 동일한 방식으로 저장한다. BaseDID는 해당 엘리먼트의 엘리먼트 테이블에서의 DID를 나타내며 ViewDID는 XML 실제 뷰 영역 내의 뷰를 구분한다. (그림 2)의 (c)는 XML 저장소에 'Catalog/Book'이라는 XQL 질의를 주어 그 결과로 얻어진 뷰 엘리먼트 테이블을 보여준다.

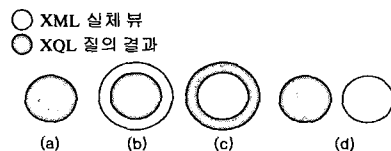
- EID 매핑 테이블 : 뷰 엘리먼트 테이블의 루트 엘리먼트에 대해 엘리먼트 테이블에서의 EID(BaseEID)와 뷰 엘리먼트 테이블에서의 EID(ViewEID) 간의 매핑 정보를 저장한다. 실제 뷰를 생성하는 XQL 질의의 결과는 트리 구조의 XML 문서에서 해당 엘리먼트와 그 자손 엘리먼트를 모두 포함하므로, 뷰 엘리먼트 테이블의 나머지 엘리먼트들의 매핑 정보도 루트 엘리먼트의 매핑 정보를 통해 알 수 있다. Path는 XML 저장소 내 XML 문서 트리 구조에서의 엘리먼트 경로를 나타낸다. 예를 들어, (그림 2)의 (c)와 같은 뷰 엘리먼트 테이블에 대해, EID 매핑 테이블은 (그림 2)의 (d)와 같이 표현된다.

4. XML 실제 뷰를 이용한 XQL 질의 변환

본 절에서는 XQL 질의를 처리하는 데 있어, 기존에 이미 존재하는 XML 실제 뷰를 이용하기 위해 XQL 질의 결과와 XML 실제 뷰 간의 포함 관계를 분류하며, 그에 따른 적절한 XQL 질의 변환 과정에 대하여 설명한다.

4.1 XQL 질의 결과와 XML 실제 뷰 간의 포함 관계

3.1절에서의 가정에 따라 XQL 질의의 결과와 XML 실제 뷰 간의 포함 관계는 (그림 3)과 같이 분류할 수 있다[7][8].



(그림 3) XQL 질의 결과와 XML 실제 뷰 간의 포함관계[7]

3.1절에서의 가정에 따라 트리 구조의 XML 문서에서 특정 엘리먼트를 찾는 XQL 질의의 결과는 해당 엘리먼트 뿐 아니라 그 자손 엘리먼트들을 모두 포함하게 된다. 따라서 (그림 3)과 같은 포함 관계에서 포함하는 쪽과 포함되는 쪽의 관계는 '부모/자식' 또는 '조상/자손'의 계층 관계를 가지게 된다. 예를 들어, (그림 3)의 (b)와 같이 XQL 질의의 결과가 XML 실제 뷰에 포함될 경우, XQL 질의가 찾는 엘리먼트는 XML 실제 뷰의 루트 엘리먼트

트의 자식 노는 자손 엘리먼트가 된다.

4.2 XQL 질의 변환

4.1절에서 서술한 바와 같이 (그림 3)의 포함 관계는 해당 엘리먼트 간의 '부모/자식' 또는 '조상/자손'의 계층 관계로 생각할 수 있다. 따라서 (그림 3)과 같은 포함 관계를 분류하기 위해서, XQL 질의가 요구하는 엘리먼트와 XML 실체 뷰의 루트 엘리먼트 간의 계층 관계를 이용하며, 이를 위해 XQL 질의와 EID 매핑 테이블의 엘리먼트 경로로부터 다음과 같은 정보를 얻는다.

- XQL 질의의 전체 경로 : 논문의 가정에 따라 XQL 질의는 경로 표현만으로 이루어진다. 경로 표현만으로 구성된 XQL 질의 중에서 '/' 또는 '\*'를 포함한 XQL 질의 경우, DTD 테이블의 정보를 이용해 루트 엘리먼트로부터 시작하여 원하는 엘리먼트까지 '/' 연산자만을 이용한 XQL 질의로 변환 가능하며, 이를 XQL 질의의 전체 경로라고 정의한다. 예를 들어, (그림 1)의 XML 문서에 대해, Catalog//Pages와 같은 XQL 질의가 주어진다면, XQL 질의의 전체 경로는 Catalog/Book/Pages와 같이 표시할 수 있다.
- 레벨 정보 : XQL 질의의 전체 경로나 EID 매핑 테이블의 엘리먼트 경로에서, 루트 엘리먼트로부터 해당 엘리먼트까지 트리의 높이를 레벨로 정의한다. 예를 들어, Catalog/Book/Pages와 같은 전체 경로나 엘리먼트 경로에 대해, 레벨은 3이 된다.
- 레벨에 따른 경로 분할 : XQL 질의의 전체 경로 또는 EID 매핑 테이블의 엘리먼트 경로를 레벨에 따라 두 개의 경로로 분할할 수 있다. 주어진 레벨을 중심으로 경로를 분할하였을 때, 분할된 경로의 앞부분을 부모 경로, 뒷부분을 자식 경로로 정의한다. 예를 들어, Catalog/Book/Pages와 같은 전체 경로나 엘리먼트 경로를 레벨 2를 중심으로 분할하였을 때, 부모 경로는 Catalog/Book 이 되며, 자식 경로는 Pages 가 된다.

XQL 질의와 EID 매핑 테이블의 엘리먼트로부터 얻어진 각각의 레벨 정보와 경로 정보를 비교하여 (그림 3)과 같은 포함 관계를 분류할 수 있다. 또한 이러한 포함 관계에 따라 주어진 XQL 질의를 하부 XML 문서에 대한 질의나 XML 실체 뷰에 대한 질의로 변환할 수 있다.

XQL 질의 결과와 XML 실체 뷰 간의 포함 관계의 분류에 따른 질의 변환 알고리즘은 (그림 4)와 같으며, 사용된 표기 (notation)는 (표 1)에 정리하였다.

(표 1) 표기 (notation)

표기	설명
q	XQL 질의의 전체 경로
q.level	XQL 질의의 레벨 정보
v	EID 매핑 테이블의 엘리먼트 경로
v.ename	EID 매핑 테이블의 엘리먼트 이름
v.level	EID 매핑 테이블 엘리먼트의 레벨
p_path(path.level)	주어진 path를 level로 분할 시 부모 경로
c_path(path.level)	주어진 path를 level로 분할 시 자식 경로
target	변환된 질의의 대상 데이터 (0 : XML 실체뷰, 1 : 하부 XML 문서, 2 : 둘 다)

```

/* XQL 질의 결과가 XML 실체 뷰와 일치 */
if (q.level == v.level && q==v) {
    query = v.ename
    target = 0
}
/* XQL 질의 결과가 XML 실체 뷰에 포함 */
else if (q.level > v.level && p_path(q, v.level)==v) {
    query = v.ename + "/" + c_path(q, v.level)
    target = 0
}
/* XQL 질의 결과가 XML 실체 뷰를 포함 */
else if (q.level < v.level && q==p_path(v, q.level)) {
    query1 = v.ename
    query2 = q+"{ $not$"+v+" }"
    target = 2
}
/* XQL 질의 결과가 XML 실체 뷰와 일치하지 않음 */
else {
    query = q
    target = 1
}
    
```

(그림 4) XQL 질의 변환 알고리즘

5. 결론

본 논문에서는 XML 저장소 외에 XML 실체 뷰가 있다고 가정하고 이를 이용한 XQL 질의 변환에 대해 연구하였다. XQL 질의가 주어졌을 때 XML 실체 뷰를 이용할 수 있다면, XQL 질의의 결과를 빠르게 반환함으로써, 성능을 향상시킬 수 있다. 이를 위해 본 논문에서는 XML 문서 저장 구조로 엘리먼트 테이블과 뷰 엘리먼트 테이블, 이들 사이의 매핑 관계를 나타내는 EID 매핑 테이블을 구성하였고, 이를 이용하여 XML 질의 결과와 XML 실체 뷰 간의 포함 관계를 분류하였으며, 그에 따른 XQL 질의 변환 알고리즘을 제시하였다.

6. 참고문헌

- [1] L. Altinel and M. Franklin, "Efficient Filtering of XML Documents for Selective Dissemination of Information," Proc. of the 26th Int'l Conf. on VLDB, 2000.
- [2] J. Robie et al., "XML Query Language (XQL)," <http://www.w3.org/TandS/QL/QL98/pp/xql.html>, 1998.
- [3] J. Robie et al., "The Design of XQL," <http://www.texcel.no/whitepapers/xql-design.html>, 1998.
- [4] R. Pottinger and A. Levy, "A Scalable Algorithm for Answering Queries Using Views," Proc. of the 26th Int'l Conf. on VLDB, 2000.
- [5] Y. Papakonstantinou and V. Vassalos, "Query Rewriting for Semistructured Data," SIGMOD Record, Vol. 28, No. 2, 1999, pp. 455-466.
- [6] 연제원 외3명, "XML 문서 구조검색을 위한 저장 시스템 설계," 한국정보과학회, '99년 봄 학술발표논문집, 26권, 1호, 1999, pp. 3-5.
- [7] D. Lee and W. Chu, "Conjunctive Point Predicate-based Semantic Caching for Web Databases," Extended Abstract, Tech. Rep. TR-980030, UCLA, Sep. 1998.
- [8] D. Lee and W. Chu, "A Semantic Caching Scheme for Wrappers in Web Databases," Tech. Rep. TR-990004, UCLA, Feb. 1999.