

# 관계형 데이터베이스를 이용한 XQL질의 처리시스템의 설계 및 구현

김경원<sup>0</sup> 이지훈<sup>\*</sup> 장복선<sup>\*</sup> 김천식<sup>\*\*</sup> 손기락<sup>\*</sup>  
\*한국외국어대학교 컴퓨터 및 정보통신공학과  
\*\*경동대학교 정보통신공학부

{kwstone, airer, boksun77}@orgio.net , {dol, ksohn}@san.hufs.ac.kr

## Design and Implementation of XQL Query Processing System using a Relational Database

Kyungwon Kim<sup>0</sup> , Jihun Lee, Boksun Jang, Chunsig Kim, Kirack Sohn  
\*Dept. of Computer & Information Telecommunication Engineering,  
Hankuk University of Foreign Studies

\*\*Dept. of School of Information Technology Engineering, Kyungdong University

### 요 약

XML이 웹 문서의 표준이며, 문서교환용 언어로서 사용되고 있다. 따라서, 향후 많은 XML문서의 보관 및 관리 그리고 XML문서를 위한 질의어 처리는 매우 중요할 것으로 본다. 지금까지, XML문서의 저장 및 검색과 관련한 연구 및 제품개발이 여러 업체에 의해서 있어왔고, 지금도 연구 및 개발이 진행되고 있다. 하지만, 효율적인 XML문서의 저장 및 검색을 위한 시스템은 아직까지 없었다. 따라서, 본 논문에서는 XML문서를 트리 형태로 저장하는 방식인, *DFS-numbering* 방식을 이용하여 문서를 저장하고 XQL질의를 처리할 수 있는 효율적인 시스템을 설계 및 구현하고자 한다.

### 1. 서 론

XML(eXtensible Markup Language)는 인터넷상의 표준문서 [1]로서 B2B, B2C, B2G사이의 문서 교환에 있어서 중요한 역할을 하게되었다. XML은 전자상거래 환경에서 중요한 위치를 차지하는 만큼 아직까지 XML문서 저장 및 교환을 위한 데이터베이스 시스템으로 각광을 받는 시스템은 없으며, 지금도 많은 IT기업 및 벤더들이 편리한 XML문서의 저장과 질의를 위한 시스템을 개발하기 위한 노력을 기울이고 있다.

XML이 이처럼 문서교환용으로서 각광을 받고 있는 이유는 XML은 자기서술적 능력을 갖고있기 때문이다. 이 능력을 가지고 HTML문서 태그로서는 기술할 수 없는 문서의 구조를 자유롭게 기술할 수 있을 뿐만 아니라, 문서에 의미를 포함할 수 있다는 것은 향후 효율적인 문서검색 시스템을 개발하는데도 도움일 될 것이다.

구조화되고 의미를 포함한 XML문서로부터 필요한 정보를 추출하기 위해서, XML구조에 맞는 많은 언어들 이 개발되었다. 즉, 다음과 같은 언어들 이 XML 문서에서 정보추출을 위해서 개발된 언어들 이다 : Lore[2], XML-QL[3], XML-GL[4], QUIT[5], XQL[6], XQuery[7].

XQL은 정규경로 표현에 의해서 임의의 경로로의 이동이 쉽다. 또한, 이 질의어는 질의표현이 누구나 쉽게 이해하고 습득하기에 어렵지 않다는 장점을 갖고 있다.

본 논문은 XML의 효율적인 저장을 위해서 [8]에서 제안된 *DFS-Numbering* 방법을 이용하여 관계형 데이터베이스에 XML문서를 저장하고, 질의를 위해서 XQL를 사용하며, 또한 XQL로 질의하면 본 논문에서 제안한 시스템으로부터 질의결과를 효율적으로 획득할 수 있는 시스템의 설계 및 구현에 대해서 기술한다.

### 2. 관련연구

관계형 데이터베이스에 XML 문서를 저장하고, 저장된 문서에 경로표현을 제공하는 질의언어로 질의를 하게되면 만족하는 경로를 표현하고 수행하기 위해서는 문서 트리의 길이만큼을 조인해야 하는 단점을 갖고 있다. 이 방법은 트리의 길이에 비례하여 성능이 떨어진다는 것을 나타내는 것이다.

본 논문에서와 같이 트리를 이용하는 방법으로서 과거에 제안된 [9]는 정규경로 표현을 위해서 성능이 그렇게 좋지 않았다. 왜냐하면 경로의 길이가 깊어지거나 복잡한 질의에 있어서는 트리 운행에 많은 시간이 소요된다. [8]은 *DFS-Numbering* 방식을 이용하도록 저장되고, 검색하는 방법으로서 기본적인 질의만 지원되었다. [10]은 [8, 9]의 단점인 트리 검색성능을 개선하였다.

XQL은 질의를 다음과 같은 범주로 분류한다.

- Collection
- Selection children and descendants expression
- Filter expression
- Boolean expression
- Equivalence and comparison expression
- Comparisons and Literals
- Union and Intersection expression

본 논문에서 제안한 시스템에서는 XQL에서 제안한 질의 범주에 해당하는 질의 연산을 제공하고 있다. 본 논문의 기여는 제안된 알고리즘에 의해서 정규경로 표현을 XML 데이터에 대한 트리 운행 없이 처리할 수 있다는 점과 XQL에서 제안한 일부 함수를 제외한 대부분의 연산을 제공한다는 점이다.

### 3. 질의 시스템 구성

본 시스템(그림 1)은 크게 문서관리부분과 XQL 처리부분으로

두 부분으로 나누어볼 수 있다. 문서관리부는 XML형태로 구성되어 있는 문서를 파싱후에 인덱싱(DFS-Numbering)[8] 하여 데이터베이스에 저장하는 부분이고, XQL 처리부분은 사용자가 입력한 XQL형태의 질의를 SQL형태로 변환하여 저장된 XML 문서를 검색하는 부분이다.

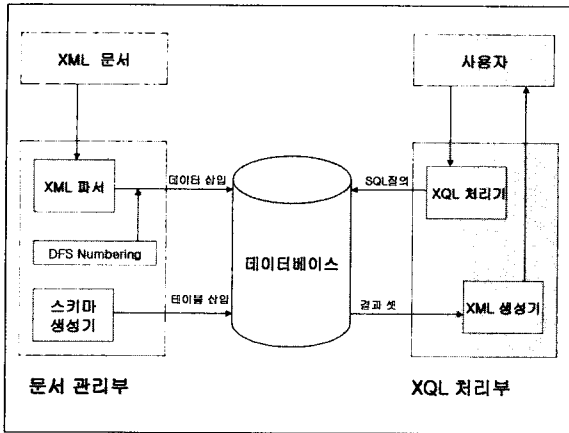


그림 1 XQL 질의 시스템 구조

### 3.1 질의를 위한 테이블 구성

XML 형태로 되어 있는 문서를 각각의 요소별로 파싱해서 미리 구성된 (표 2)와 같은 RDB 테이블에 저장한다.

테이블 이름	기능
T_PI	문서의 Process Instruction문들을 저장하기 위한 테이블
T_COMMENT	문서의 주석문을 저장하기 위한 테이블
T_DOCUMENT	문서를 처리하는데 필요한 기본정보들을 저장하기 위한 테이블
T_ELEMENT	문서의 요소(Element)들의 기본적인 정보들과 인덱스에 필요한 데이터들을 저장하기 위한 테이블
ATTRIBUTE_LIST	속성과 요소를 연결하기 위한 테이블
T_ATTRIBUTE	속성과 이름을 저장하기 위한 테이블
T_NAMESPACE	이름공간을 저장하여, T_ATTRIBUTE와 연결하기 위한 테이블

표 1 파싱한 데이터를 저장하기 위한 테이블

XML 요소들을 관계형 데이터베이스에 저장하는데 있어, 항상 문제가 되어오던 트리구조에 대한 정보는 DFS-Numbering[8] 기법을 이용하여 해결하였다.

XML문서의 요소(Element)들은 해당 문서의 트리 구조에 따라서 각각 번호를 매기(Numbering)고, 이 번호(Number)들은 관계형 데이터베이스 상에서의 XML문서 검색을 더욱 용이하게 한다. (그림 2)는 (표 1)을 스키마의 관계를 나타내는 E-R 모델링이다.

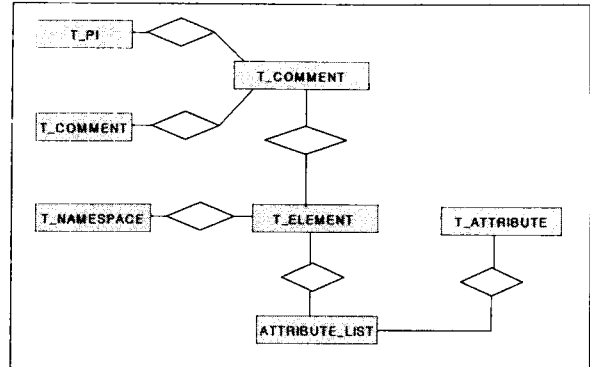


그림 2 XQL스키마를 위한 E-R모델링

### 3.2 XML 문서 삽입

XML 문서를 SAX[11] 표준 파서를 이용하여 파싱한다. 요소별로 파싱한 결과를 관계형 데이터베이스에 저장하기 전에 DFS-numbering 기법을 이용하여 요소(Element)의 태그별로 번호 매김(Numbering)한다. 이 기법은 Document의 모든 태그를 나오는 순서대로 번호 매김(Numbering)하여 요소별로 "시작태그(Start Tag)"와 "끝태그(End Tag)"의 번호와 Level을 번호로 표현하여 관계형 데이터베이스 테이블에 요소별로 삽입한다. 이 기법을 이용하면 요소들의 포함관계를 쉽게 파악할 수 있다.

### 3.3 XQL질의를 SQL질의로 변환 방법

XQL표현(Expression)을 처리하기 위해서는 먼저 XQL 표현을 파싱 해주는 파서가 필요하다. 본 논문에서는 XQL 표준 BNF을 JavaCC라는 도구를 이용하여 파서의 틀을 구성하였다.

XQL 표현이 입력되면 이미 구현된 XQL 파서를 이용하여 각각의 표현별로 파싱하여 그 표현에 해당하는 SQL 부분 문자열을 생성한다. 모든 XQL 표현의 파싱이 끝나면, 부분별로 변환된 SQL 부분 문자열들을 조합하여 하나의 SQL 문자열로 만든 후, 이 문자열을 실제로 관계형 데이터베이스에 질의하여 결과를 XML 문서 형태로 재 조합하여 돌려준다.

XQL 표현을 하나의 SQL 문자열로 변환하는 알고리즘(그림 3)은 다음과 같다.

1. XQL 표현(Expression)별로 파싱한다.
2. 파싱된 표현을 스택에 Push한다.
3. 1과 2를 XQL 표현이 끝날 때까지 반복한다.
4. 스택에서 저장된 표현들을 Pop한다.
5. 표현에 해당하는 SQL 부분 문자열로 변환·조합하여 스택에 Push한다.
6. 모든 XQL 표현이 SQL 문자열로 변환될 때까지 4, 5를 반복한다.
7. 최종적으로 변환·조합된 SQL 문자열을 실제 RDB에 질의한다.
8. 질의결과를 다시 XML 문서형태로 재 조합한다.

그림 3 XQL을 SQL로 변환하기 위한 알고리즘

예제 1 : 다음의 XQL질의가 (그림 4)와 같은 SQL로 어떻게 변환되고 변환된 SQL의 의미는 무엇인지 설명하라.

`namecard/*/phone[office]`

위의 XQL은 *namecard* 요소로부터 손자에 해당하는 경로에 존재하는 *phone* 요소를 추출하라는 것으로, 특히 *phone* 요소가 *office* 요소를 포함하는 *phone* 요소를 추출하라는 의미이다.

위 XQL질의가 SQL로 생성된 모습이 (그림 4)와 같은 모습이다.

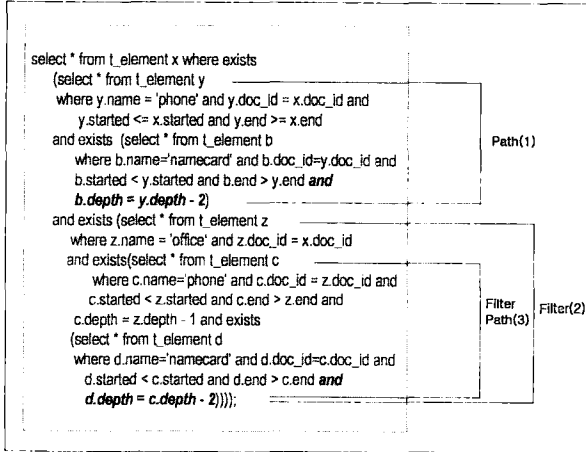


그림 4 XQL에 해당하는 생성된 SQL

(그림 4)의 Path(1)가 설명하는 것은 "name/\*/phone"에 해당하는 부분의 SQL 질의문자열이다. 이 부분은 질의가 "name/phone" 이나 "name//phone"일 경우에도 같다. 다른 부분은 이탤릭체로 되어 있는 부분으로, 이 부분은 두 요소들의 연결관계(Level 차이)를 표현하는 부분으로 위의 예제에서는 "/"\*로 연결되어 있으므로 "b.depth=y.depth-2"로 표현되어 있다.

(그림 4)의 Filter(2)가 설명하는 것은 "[office]"에 해당하는 부분의 SQL 질의문자열이다. Filter부에 해당하는 요소는 필터 (/) 표현 앞부분에 나와 있는 경로 정보를 모두 포함하고 있어야만 한다. 따라서 필터부는 앞부분에서 표현된 경로정보를 모두 가지고 있어야만 한다.

(그림 4)의 Filter Path(3)가 설명하는 것은 Path(1)을 기반으로 생성되기 때문에 Path(1)이 바뀌면 이 부분도 바뀌어야 한다. 따라서, 이탤릭체로 되어 있는 부분이 위와 같이 변경되어야 한다.

(그림 3)을 이용하여 위와 같은 SQL 문자열(A)가 생성되면, 결과를 XML 문서로 재구성하기 위하여 다음과 같은 형태로 변환한다.

```

SELECT a.doc_id, a.start AS num, a.content, a.name, 's'
AS tag FROM t_element a
where exists ( A Query )
UNION
SELECT a.doc_id, a.end AS num, a.content, a.name, 'e'
AS tag FROM t_element a
where exists ( A Query )
ORDER BY 1, 2
    
```

위와 같은 형태의 질의의 수행결과는 위의 질의에서 이탤릭체로 표현된 "a.start AS num"과 "a.end AS num"에 의해서

XML 문서의 Tag 순서와 같은 순서로 정렬되게 된다. 이 정렬된 결과를 하나씩 가져와서 위의 질의에서 이탤릭체로 표현된 "s' AS tag"와 "e' AS tag"를 이용하여 시작태그(Start Tag)와 끝태그(End Tag)를 구분하여 붙여주고 태그 사이에 내용(Content)을 넣어주면 XML 문서형태로 결과를 재조합할 수 있게 된다.

#### 4. 결론 및 향후 연구

본 논문에서는 XML문서의 요소(Element), 속성(Attribute)의 특성 및 부모와 자식 요소사이의 관계를 유지하고, 관계형 데이터베이스에 XML문서를 저장하기 위한 트리 형태의 저장 방식인 "DFS-Numbering" 방식을 사용하여 XML문서를 저장하였다. 또한, 관계형 데이터베이스에 저장된 XML문서에 XQL질의어로 질의하여 질의 결과로 XML문서를 획득하기 위한 시스템을 설계 및 구현하였다.

향후 계획으로는 본 논문에서 XQL[6]에서 제안한 질의 연산의 완벽한 지원과, 질의 처리성능을 개선하여 상업적으로 이용 가능한 시스템을 구현하는 것이다.

#### 5. 참고문헌

- [1] Tim Bray, Jean Paoli, C.M.Sperberg-McQueen, and EveMaler. Extensible Markup Language (XML) 1.0 second edition W3C recommendation. Technical Report REC-xml-20001006, World Wide Web Consortium, October 2000.
- [2] S. Abiteboul, D. Quas, J. McHugh, J.Widom, J.Widom, and J.L. Wiener. The Lore query language for semistructured data. International Journal on Digital Libraries, 1(1):68-88, April 1997.
- [3] Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levy, and Dan Suciu, A query language for XML. In Proceedings of the 8th International World Wide Web Conference, pages 77-91, Toronto, Canada, May 1999.
- [4] Stefano Ceri, Sara Comai, Ernesto D'Amiani, Piero Fraternali, Stefano Paraboschi, and Letizia Tanca. XML-GL: A graphical language for querying and restructuring XML documents. In Proceedings of the 8th International World Wide Web Conference, pages 93-109, Toronto, Canada, May 1999.
- [5] Don Chamberlin, Jonathan Robie, and Daniela Florescu. Quilt: An XML query language for heterogeneous data sources. In International Workshop on the Web and Databases(WebDB'2000), Dala, TX, May 2000.
- [6] Jonathan Robie, Joe Lapp, David Schach, XML Query Language (XQL), <http://www.w3.org/TandS/QL/QL98/pp/xql.html#>
- [7] Don Chamberlin, Daniela Florescu, Jonathan Robie, Jrme Simon, and Mugur Stefanescu. XQuery : A Query Language for XML W3C working draft. Technical Report WD-xquery-20010215, World Wide Web Consortium, February 2001.
- [8] 이용석, "XML문서 저장시스템의 설계 및 구현", 한국외국어대학교 석사학위 논문,1999, 2
- [9] Jason McHugh and Jennifer Widom. Query optimization for XML. In Proceedings of the 25th VLDB Conference, pages 315-326, Edinburgh, Scotland, September 1999.
- [10] Quanzhong Li, Bongki Mon : Indexing and querying XML data for regular path expressions. VLDB 2001.
- [11] David Megginson, <http://www.megginson.com/SAX/>