

# 관계 데이터베이스 시스템 기반의 XQuery 질의 처리기 설계\*

장형화<sup>0</sup> 홍의경

서울시립대학교 컴퓨터통계학과  
{hhjang97<sup>0</sup>, ekhong}@venus.uos.ac.kr

## Design of XQuery Query Processor based on Relational Database System

Hyung-Hwa Chang<sup>0</sup> Eui-Kyeong Hong

Department of Computer Science and Statistics, University of Seoul

### 요 약

XML은 고유의 확장성과 문서 관계성의 우수성을 활용하여 새로운 정보 공유 환경의 표준으로 자리잡고 있다. XML 문서의 사용이 많아지면서 대용량의 XML 문서를 효과적으로 저장, 관리 및 검색하기 위한 시스템이 필요하다.

본 연구는 관계 데이터베이스 시스템(RDBMS)을 기반으로 XML 문서를 저장하고 검색할 수 있게 하기 위해서 Numbering 기법에 의한 XML 문서 저장 기법을 사용하였다. XQuery 질의를 SQL문장으로 변환시켜서 수행함으로써 XML 데이터에 대한 검색을 가능하게 하였다.

### 1. 서 론

최근 인터넷의 발전이 진전되면서 이기종 시스템에서 작성된 문서에 대한 데이터베이스의 구축과 검색 및 상호 교환의 중요성이 높아지고 있다. 이에 따라 W3C에서는 기존의 HTML의 단점을 보완하고 SGML의 복잡성을 제거한 XML(eXtensible Markup Language)을 웹 문서의 표준으로 제정하였다. XML은 확장성과 문서 관계성이 좋기 때문에 새로운 정보 공유 환경의 표준 매체로 자리 잡아가고 있다. 이런 XML 문서를 데이터베이스 시스템에 저장하는 기법에 대한 연구가 활발히 진행 중이다. 또한 다양한 형태로 저장된 XML 문서에서 원하는 데이터를 추출하고 변환하는 작업들이 필요하게 된다. 이를 해결하기 위해서 일반적으로 질의어를 사용하게 된다. W3C에서는 XML 문서에 대한 질의어로서, 구조적 질의어인 XPath를 핵심으로 하는 XQuery를 새로운 질의어의 표준으로 발표하였다[1,2]. 따라서 XML 문서를 관계 데이터베이스 시스템(RDBMS)에 효율적으로 저장하는 시스템을 개발하고, XML 사용자가 저장된 XML 문서에서 필요한 정보를 검색할 수 있도록 하기 위해 XQuery 질의어를 지원하는 연구가 필요하다.

본 논문에서는 서울시립대학교에서 설계한 XML 문서 저장 기법[3]을 사용하였으며, XQuery 질의어를 통해 XML 데이터를 접근할 수 있도록 하기 위해서 XQuery를 SQL문으로 변환하여 실행하는 XQuery 질의 처리기를 설계하였다.

본 논문의 구성은 다음과 같다. 2절에서는 XML 문서 구조 저장 기법, XPath 및 XQuery에 대해 알아보고, 3절에서는 본 논문에서 사용하는 XML 문서 저장 시스템에 관하여 설명한다. 4절에서는 XQuery 질의 처리기의

기반이 되는 XPath 질의 처리기에 대해 살펴보고, 5절에서는 XQuery를 RDBMS에서 수행 가능한 SQL문으로 변환하여 처리하는 XQuery 질의 처리기 설계를 논의한다. 마지막으로 6절에서는 결론 및 향후 연구 방향을 기술한다.

### 2. 관련 연구

#### 2.1. XML 문서 구조 저장 기법 분류

데이터베이스 기술 분야에서는 방대한 양의 XML 데이터를 효율적으로 DBMS에 저장하고 검색하는 기술에 대한 연구가 활발히 진행되고 있다[4].

그 중 관계 데이터베이스에 저장하는 방식은 비분할 저장 기법과 분할 저장 기법으로 나눌 수 있다. 비분할 저장 기법은 XML 문서 전체를 BLOB (Binary Large Object) 형태로 저장한 다음, 각각의 단말 노드는 오프셋 정보를 가지고 접근한다. 이는 문서를 한꺼번에 저장하였기 때문에 통합 과정이 필요 없이 문서 참조를 빨리 할 수 있지만, 내용의 일부만 수정되었을 때 문서 전체를 재구성해야 한다는 단점이 있다. 분할 저장 기법은 XML 문서를 엘리먼트 별로 나누어서 저장한다. 이 방법은 문서의 일부 내용이 수정되었을 때 관계되는 노드들만 수정하면 되므로 문서의 편집 및 관리가 쉽고, 동일한 내용을 갖는 노드들을 공유할 수 있는 장점을 갖는다. 그리고 사용하는 테이블의 개수에 따라 단일 테이블에 모든 정보를 저장하는 단일 간선 테이블(single edge table) 방식과 여러 개의 테이블 집합을 이용하여 저장하는 테이블 집합 방식이 있다[5].

#### 2.2. XPath

XPath는 XSLT 1.0과 함께 1999년 11월 16일 W3C에서 XPath 1.0으로 권고(Recommendation)하였다. XPath는 XML 문서의 일부분을 지시하기 위해서 사용된다. XPath는 URL 경로 표기법을 사용하여 XML 문서의 계층적인 구조를 논리적으로 탐색하는데 이 구조는 요소 노드(element node), 속성 노드(attribute node), 값 노

\* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았다

드(text node)를 사용하여 트리 구조로 형성된다.

XPath는 노드의 경로를 나타내기 위해 위치 경로(location path)를 사용하는데, 경로 위치 표현 방식에는 절대 경로와 상대 경로 두 가지 방식이 있다.

표 1. XPath의 생략된 표기식

비생략된 문법	생략된 문법
child::	(없음)
attribute::	@
/descendant-or-self::node()	//
parent::node()	..
(문서 루트 노드)	/

본 연구에서는 XPath의 모든 기능을 만족하기 보다는 XQuery에서 지원하는 생략된 표기식만을 사용한다. 비생략된 표기식과의 비교는 [표 1]을 통해 확인할 수 있다.

### 2.3. XQuery

XQuery는 XML 질의를 위한 새로운 W3C 표준으로 정의되었고, 최초로 XML 질의어인 Quilt에서 유래되었으며, 다른 언어들이 가지고 있는 몇 가지 유용한 기능들이 추가된 것이다. XQuery는 XPath에 근거한 Path 표현과 For Let When Return(FLWR)으로 표현된다. 이 논문에서는 중첩 질의를 배제하고 설계하였고, Path 표현은 2.2에서 설명한 XPath를 사용한다.

### 3. XML 저장 시스템 구조

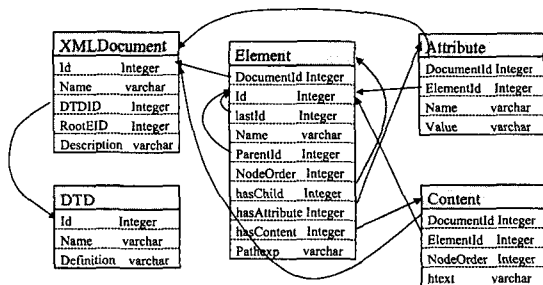


그림 1. XML 저장 스키마

본 논문에서 native-XML 문서를 DTD에 상관없이 독립적으로 RDBMS에 저장할 수 있도록 저장 스키마를 [그림 1]과 같이 설계하였다.

본 시스템에서는 XML 문서를 저장하기 위해서 다음과 같은 과정을 거친다. 우선 XML 문서를 파싱하고, 이벤트를 이용하여 엘리먼트, 애트리뷰트, 내용 등의 정보를 각각의 테이블에 사상해서 저장한다.

[그림 2]는 DFS Numbering을 이용하여 ID를 부여한 것과 경로 정보를 추가적으로 표현한다[3].

XML 문서 검색은 XQuery 질의를 통해 이루어지며, XQuery 질의는 XPath를 기반으로 XQuery 질의 처리기에 의해 적절한 SQL문으로 변환되어 수행된다. 세부적인 내용은 다음 절에서 알아본다.

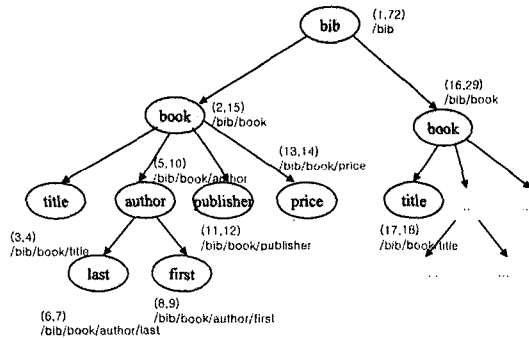


그림 2. DFS Numbering과 Pathexp 정보 표현 트리

### 4. XPath 질의 처리기 구조

XQuery를 SQL문으로 바꾸기 위해 XQuery에 사용되는 XPath를 뷰로 만들고 뷰를 이용해서 SQL문을 생성한다. 따라서 본 절에서는 XPath 질의를 앞 절에 서술한 저장 시스템 하에서 SQL문으로 생성하는 방법에 대해서 알아본다.

질의 처리기 구조는 [그림 3]과 같다. XPath 질의의 예제를 통해서 간단히 알아보자.

각 단계별 처리 과정에 관한 설명을 돕기 [그림 2]에서 book 정보를 찾는 XPath 질의인 "/bib/book"을 사용한다.

XPath 파싱 단계에서 질의의 적합성을 검증하고 { /, bib, /, book }과 같은 토큰을 발생시킨다.

토큰 분류 단계에서는 파싱 단계에서 발생된 토큰이 해당 문맥 노드를 구하는데 필요한지, 서술부를 나타내기 위해서 사용되는지 정보를 확인하여 ContextNode와 Predicate 정보로 분류하는데 ContextNode에는 질의에 의해 반환될 엘리먼트 집합의 최상위 엘리먼트의 위치 정보를 저장하고, Predicate에는 질의의 요구를 만족하는 정보만을 검색할 수 있도록 Predicate 정보를 저장한다. 위 예제의 경우 ContextNode에는 "/bib/book"이라는 정보가 저장된다.

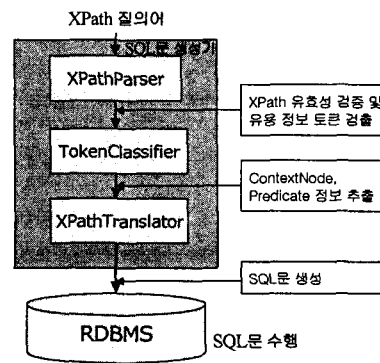


그림 3. XPath 질의 처리기 구조

XPath 질의 변환 단계에서는 토큰 분류 단계에서 구성한 ContextNode 정보와 Predicate 정보를 활용하여 RDBMS에서 수행될 실질적인 SQL문으로 변환하는데,

Element 테이블과 Content 테이블을 사용한다. 위 세 단계를 모두 거치면 최종적인 SQL 질의가 다음과 같이 생성된다[3].

```
SELECT e1.id startId, e1.lastId lastId, e1.parentId parentId,
       e1.xmlDocumentId xmlDocumentId
FROM   helement e1
WHERE  e1.xmlDocumentId = 5 and e1.name='book'
       and e1.pathexp = '/bib/book'
```

본 과정을 통해 만들어진 SQL문은 반환되어야 하는 엘리먼트의 범위 ID를 얻게 된다. 이 SQL문을 뷰로 만들어서 XQuery 질의 처리기에서 이용한다.

### 5. XQuery 질의 처리기 설계

관계 데이터베이스에 저장된 XML 문서 정보에 대한 질의 검색을 위해 사용자는 W3C 표준인 XQuery질의어를 통해서 XML 문서 검색을 할 수 있다. 따라서 본 절에서는 XQuery 질의를 SQL문으로 변환하는 XQuery 질의 처리기를 설계하였다. 변환된 SQL문은 적은 조인으로 결과를 검색할 수 있는 장점이 있다.

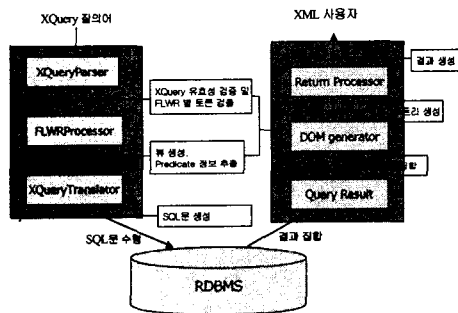


그림 4. XQuery 질의 처리기 구조

질의 처리기의 구조는 [그림 4]와 같고, 질의를 처리하기 위해 크게 XQueryParse, FLWRProcessor, XQueryTranslator로 나눈다. 각 단계별 처리 내용은 book 정보에서 publisher가 "Addison-Wesley"인 book 정보를 찾는 XQuery 예제를 통해 설명한다.

```
FOR $b IN document("bib.xml")/bib/book
WHERE $b/publisher = "Addison-Wesley"
RETURN
  <book year="{ $b/@year }">
    { $b/title }
  </book>
```

XPathParser 단계에서는 XQuery에 대한 유효성 검증과 FLWR 절 별로 토큰을 생성한다. FOR 절에서 { \$b, IN, document("bib.xml"), ./, bib, /, book } 토큰이 생성되고 WHERE 절에서 { \$b, /, publisher, =, "Addison-Wesley" } 토큰이 생성된다. RETURN 절에서 { <book year=", \$b/@year, ">, \$b/title, </book>

}과 같이 문자열과 변수를 구분해서 토큰이 생성된다.

FLWRProcessor 단계에서는 FOR 절의 변수 b에 대한 XPath 표현식 document("bib.xml")/bib/book을 XPath 질의 처리기를 이용해서 대응하는 뷰 b\_view를 생성한다. 그리고 WHERE 조건절에 대한 Predicate 정보인 \$b/publisher = "Addison-Wesley"를 추출해서 저장한다.

XQueryProcessor 단계에서는 뷰와 Predicate 정보를 이용해서 찾고자 하는 엘리먼트의 범위를 구하는 질의를 생성한다. 위의 세 단계를 모두 거쳐서 다음과 같은 SQL문이 생성된다.

```
SELECT b_view.*
FROM   b_view, helement e1, content con1
WHERE  b_view.xmlDocumentId = e1.xmlDocumentId
       AND b_view.startId <= e1.id
       AND b_view.lastId >= e1.lastId
       AND e1.xmlDocumentId = con1.xmlDocumentId
       AND e1.id = con1.elementId
       AND e1.name = 'publisher'
       AND con1.htext = 'Addison-Wesley';
```

이 SQL문을 이용해서 원하는 엘리먼트를 검색하고, 검색된 결과는 DOM 생성기를 통해서 DOM 트리를 생성한다. DOM 트리는 RETURN 절의 XPath 표현식에 대한 결과를 얻기 위해 사용된다. 마지막으로, 생성된 DOM 트리과 RETURN 절의 토큰을 이용해서 결과를 생성한다.

### 6. 결론 및 향후 연구 방향

본 논문에서는 DTD 독립적으로 저장하는 분할 방식 저장 시스템에서 XQuery 질의를 RDBMS의 SQL문으로 변환시키는 XQuery 질의 처리기를 설계하였다. 그리고 XPath를 기반으로 XQuery를 처리하기 때문에 XPath 질의에 대해서도 검색이 가능하다.

XQuery 질의 처리기의 성능은 XPath의 영향을 많이 받기 때문에 향후에는 통계 정보를 이용해서 데이터베이스 접근을 최대한 줄이는 방법과 Path 표현식의 인코딩을 통한 성능을 향상시키는 방법을 연구할 예정이다.

### 7. 참고 문헌

- [1] J.Clark and S. DeRose, "XPath 1.0: XML Path Language," W3C Recommendation, <http://www.w3.org/tr/XPath>, 1999.
- [2] D.Chamberlin, et al., "XQuery 1.0: An XML Query Language," W3C Working Draft, <http://www.w3.org/TR/2001/WD-xquery-20010607>, 2001.
- [3] 고영기, 홍의경, "분할 저장 시스템에 적합한 XPath 질의 처리기 설계," 한국정보과학회 가을학술대회 논문집 29(2), pp.52-54, 2002.
- [4] J. Shanmugasundaram, et al., "Relational Databases for Querying XML Documents: Limitations and Opportunities," Proc. of the 25th VLDB Conf., 1999.
- [5] D.Florescu and D.Kossman, "Storing and Querying XML Data Using an RDBMS," IEEE Data Engineering Bulletin 22(3), pp.27-24, 1999.