

분산 객체 관계 데이터베이스 시스템을 이용한 분산 XQuery

질의 처리기 설계 및 구현*

이재민^o 장건업 홍의경
서울시립대학교 컴퓨터과학부
jmlee97@naver.com^o, {atlas, ekhong}@venus.uos.ac.kr

Design and Implementation of Distributed XQuery Query Processor using Distributed ORDBMSs

Jae Min Lee^o Gun Up Jang Eui Kyeong Hong
Dept. of Computer Science, University of Seoul

요 약

최근 컴퓨팅 환경은 인터넷 환경의 웹을 기반으로 한 분산 컴퓨팅 환경으로 변화하고 있다. 그에 따라 XML 문서의 사용과 XML 문서의 양이 급속하게 증가하였으며, 언제나 쉽게 필요한 XML 문서에 접근할 수 있어야 한다. 또한 다양한 형태로 분산 저장된 XML 문서에서 원하는 데이터를 추출하고 변환하며, 단편화된 XML 데이터를 통합하는 작업들이 필요하게 된다. 따라서 XML 문서를 분산 객체 관계 데이터베이스 시스템에 효율적으로 저장하는 시스템을 개발하고, 분산 저장된 XML 문서에서 사용자가 필요한 정보를 검색할 수 있도록 하기 위해 XQuery 질의어를 지원하는 연구가 필요하다.

본 논문에서는 분산 객체 관계 데이터베이스 시스템에 저장된 XML 데이터를 접근할 수 있도록 하기 위해 XPath를 분산 SQL로 변환하여 실행하는 분산 XPath 질의 처리기를 확장하여 XQuery를 분산 SQL로 변환하여 실행하는 분산 XQuery 질의 처리기를 설계 및 구현하였다.

1. 서 론

지식과 정보 교류의 기반이 웹 환경으로 바뀌면서 W3C에서는 기존의 HTML의 단점을 보완하고 SGML의 복잡성을 제거한 XML을 표준으로 제정하였다[1]. XML은 확장성과 문서 관계성 표현이 우수하여 새로운 정보 공유 환경의 표준 매체로 자리 잡아가고 있으며, 이기종 시스템 간의 정보 교환을 용이하게 하는 특성을 가지고 있어 EC/EDI, 전자 도서관, 전자 상거래 등 다양한 응용 분야에서 사용되고 있다.

XML 문서를 저장하기 위하여 관계형 데이터베이스 시스템에 XML 데이터를 통합하는 작업들이 XML 문서의 구조 정보를 저장함으로써 관계형 데이터베이스 시스템이 지니고 있는 질의 수행 능력 및 우수한 성능을 충분히 활용할 수 있고 기존의 응용 시스템의 데이터를 활용할 수 있는 장점이 있다[2].

최근 컴퓨팅 환경은 인터넷 환경의 웹을 기반으로 한 분산 컴퓨팅 환경으로 변화하고 있다. 그에 따라 XML 문서의 사용과 XML 문서의 양이 급속하게 증가하였으며, 언제나 쉽게 필요한 XML 문서에 접근할 수 있어야 한다. 또한, 다양한 형태로 분산 저장된 XML 문서에서 원하는 데이터를 추출하고 변환하며, 단편화된 XML 데이터를 통합하는 작업들이 필요하게 된다. 이를 해결하기 위해서 W3C에서는 XML 문서에 대한 표준 질의어로 구조적 질의어인 XPath를 핵심으로 하는 XQuery를 새로

운 질의어의 표준으로 제정하였다[3,4]. 따라서 XML 문서를 분산 객체 관계 데이터베이스 시스템에 효율적으로 저장하는 시스템을 개발하고, 분산 저장된 XML 문서에서 사용자가 필요한 정보를 검색할 수 있도록 하기 위해 XQuery 질의어를 지원하는 연구가 필요하다.

본 논문에서는 분산 객체 관계 데이터베이스 시스템을 이용하여 분산된 XML 데이터를 접근할 수 있도록 하기 위한 분산 XPath 질의 처리기[5]를 확장하여 XQuery를 분산 SQL로 변환하여 실행하는 분산 XQuery 질의 처리기를 설계 및 구현하였다.

본 논문의 구성은 다음과 같다. 2절에서는 XML 문서의 저장 기법 및 XPath와 XQuery에 대해서 알아보고, 3절에서는 분산 객체 관계 데이터베이스 시스템을 이용한 분산 XML 관리 시스템에 대하여 설명한다. 4절에서는 분산 XPath 질의 처리기, 5절에서는 분산 XPath 질의 처리기를 확장하여 XQuery를 분산 객체 관계 데이터베이스 시스템에서 수행 가능한 분산 SQL문으로 변환하여 처리하는 분산 XQuery 질의 처리기 설계에 대해서 설명한다. 마지막으로 6절에서는 결론을 기술한다.

2. 관련 연구

2.1 XML 문서 저장 방법

XML 문서를 저장하고 관리하기 위하여 파일 시스템을 이용한 기법, (객체)관계 데이터베이스 시스템을 이용한 기법, 네이티브 XML 데이터베이스 등이 연구되고 있다. 파일 시스템을 이용한 기법은 확장성이나 질의 처리에 제한이 많고 네이티브 XML 데이터베이스 시스템의

* 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음

경우 많은 부분에서 기술적 성숙이 이루어지지 못하였다. 그래서 기존에 많이 연구되고 기술적으로 성숙한(객체)관계 데이터베이스 시스템을 이용한 방식이 활발히 연구되었다. 기존 데이터베이스 시스템을 이용하여 XML 문서를 저장하고 관리하면 질의 처리, 트랜잭션, 권한 시스템, 인덱스 기능 등 기존 데이터베이스 시스템의 기능을 활용할 수 있다.

기존 데이터베이스 시스템을 이용한 XML 저장 기법들은 접근 방식과 데이터 모델에 따라 분류할 수 있다. 일반적으로 XML 문서는 텍스트(문서) 중심(text-centric, document-centric) XML 문서와 데이터 중심(data-centric) XML 문서로 나눌 수 있다. [8]에서는 이러한 XML 문서의 특징에 따라 가능한 저장 기법을 구분하였다. XML 문서 전체를 하나의 BLOB(Binary Large Object)이나 CLOB(Character Large Object) 형태로 저장하거나 XML 문서의 노드를 데이터베이스의 릴레이션과 애트리뷰트로 나누어 저장하는 분할 저장 기법이 주로 사용된다. 분할 저장 기법은 XML 문서가 데이터베이스에 저장되는 구조에 따라 XML 문서의 XML 스키마나 DTD를 이용하는 스키마 종속적인 기법과 스키마 독립적인 기법으로 구분할 수 있다. 스키마 종속적인 기법은 XML 스키마에 따라 데이터베이스에 생성된 데이터베이스 스키마가 비효율적으로 모델링될 수 있는 단점이 있다. 그리고 XML 스키마나 DTD가 없는 문서가 스키마가 존재하는 문서보다 많으므로 본 논문에서는 XML 문서의 스키마와 관계없이 데이터베이스의 스키마를 정의하여 XML 문서의 정보를 저장하는 스키마 독립적인 저장 기법을 사용한다. 또한 다양한 XML 문서를 처리하기 위하여 XML 문서의 노드마다 오더 레이블링(order labeling)을 사용하여 문서 중심 XML 문서와 데이터 중심 XML 문서를 동시에 처리할 수 있도록 하였다.

2.2 XML 문서 노드 오더링 기법

XML 문서의 노드 오더 인코딩 기법으로는 글로벌 오더, 로컬 오더, Dewey 오더가 있다. 글로벌 오더의 경우 XML 문서의 노드 순서대로 레이블링하는 방식으로 질의 처리 성능이 우수한 장점이 있지만 노드의 삽입, 삭제가 발생할 경우 노드의 레이블링을 다시 수행하여야 하는 단점이 있다. 로컬 오더는 노드의 갱신 성능이 우수하지만 노드들 간의 부모-자식 관계를 파악하기 어려운 단점이 있다. Dewey 오더와 같은 하이브리드 오더 기법은 갱신 연산에 따른 오버헤드를 줄이고 질의 처리 성능을 유지시키는 장점을 가진다. 본 논문에서는 XML 문서를 하이브리드 오더 인코딩을 사용하여 분산 저장하는 방법을 사용하였다[2].

2.3 XPath

XPath[3]는 1999년 W3C에서 XPath 1.0으로 권고(Recommendation)되었다. XPath는 XML 문서의 일부분을 지시하기 위해서 사용된다. XPath는 URL 경로 기법을 사용하여 XML 문서의 계층적인 구조를 논리적으로 탐색하는데 이 구조는 요소 노드(element node), 속성 노드(attribute node), 값 노드(text node)를 사용하여 트리 구조로 형성된다. 또한 XPath는 각 노드의 문자열 값

을 계산하는 방법을 정의하고 있다.

XPath는 노드의 경로를 나타내기 위해 위치 경로(Location Path)를 사용하는데, 경로 위치 표현 방식은 절대 경로와 상대 경로 두 가지 방식이 있다. 절대 경로란 / 로 시작하여 문서의 루트 노드를 시점으로 경로를 지시하는 것을 말하고, 상대 경로는 현재 노드를 기점으로 하여 요구되는 상대적인 노드의 경로를 나타낸다. 이때 경로 위치를 기초부(basis)와 술어부(predicate)로 나누어 설명할 수 있는데, 기초부는 축(axis)과 노드 검사 부분으로 구성된다. 여기서 축이란 문맥 노드(Context Node)와 경로 단계로 선택된 노드간의 관계를 명시한다. 한편, 술어부는 대괄호([,])로 묶인 표현식을 말하는데, 축에 관련된 노드집합을 새로운 노드집합으로 만들기 위한 필터링 작업을 한다. 즉, 술어부의 표현식을 이용하면 노드 집합 중에서 필요한 노드만을 선택하기 위해서 사용할 수 있으며 최종 값은 부울 값이 된다.

XPath는 XPointer, XSLT, XQuery, XSchema 등에서 문서 내 노드를 탐색하는데 주로 사용하고 있다. 기존의 XQuery에서 지원하는 생략된 표기식만을 XPath에서 사용한다. 비생략된 표기식과의 비교는 표 1을 통해 확인할 수 있다. 또한 단일 술어부를 지원하며, 술어부에 사용되는 비교 연산자는 관계 연산자를 사용하였다.

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

표 1 XPath에서 생략된 표기식

2.4 XQuery

XQuery[4]는 최초의 XML 질의 언어인 Quilt에서 유래하였으며 XPath 표현을 포함하고 있다. XQuery는 XML 문서의 구조를 이용하여 전체가 구조화되거나 부분적으로 구조화된 XML 문서에 대한 복잡한 질의를 수행할 수 있다. 그래서 관계 데이터베이스 시스템의 SQL과 같은 역할을 한다. 또한 XQuery는 W3C의 다른 표준 기술인 XML, 네임스페이스(namespaces), XSLT, XPath, XML Schema 등과 상호 호환성을 가진다.

XQuery 질의를 구성하는 기본 단위는 표현식(expression)이며, XQuery 질의 언어가 제공하는 기본 표현식으로는 경로 표현식(path expression), FLWOR 표현식(FLWOR expression) 등이 있다. 그리고 각 표현식은 다른 표현식 안에 중첩되어 나타날 수 있다. XQuery의 경로 표현식은 XPath의 경로 표현식을 따르고 있다. FLWOR 표현식은 XQuery의 핵심 표현식으로서 XQuery 질의의 기본 형식을 정의한다.

XQuery는 현재 1.0 버전이 표준으로 사용되며 2.0 버전에 대한 표준화 작업이 진행되고 있다. XQuery 2.0에서는 XML 문서의 갱신 지원과 XML 풀 텍스트 검색(full-text search) 기능에 대한 표준이 추가될 예정이다.

3. 분산 XML 관리 시스템

[5]에서는 분산 XML 데이터베이스에서 XML 문서를 저장하고 질의하기 위해 그림 1과 같은 분산 XML 문서 저장 관리 시스템을 설계하였다.

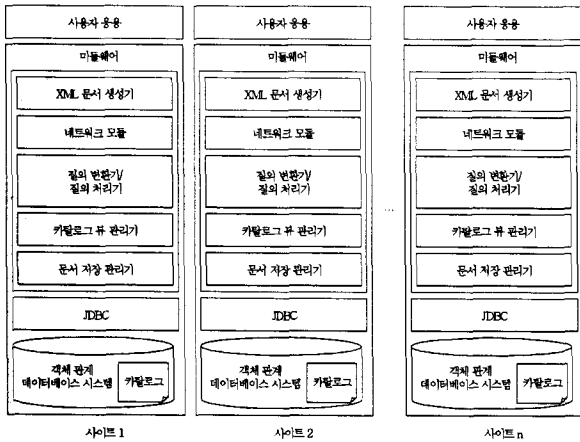


그림 1 분산 XML 관리 시스템

분산 XML 데이터베이스의 카탈로그 뷰 관리기에는 분산 XML 데이터베이스를 구성하고 있는 사이트의 카탈로그를 저장한다. 카탈로그 뷰 정보에는 각 사이트에 저장된 XML 문서와 XML 문서에 대한 XML 스키마나 DTD 정보, 각 사이트의 릴레이션에 대한 정보 등이 포함된다. 본 논문에서는 분산 XML 데이터베이스를 구성할 때 참여 사이트에 저장된 XML 문서와 관련된 카탈로그의 뷰를 다른 사이트의 카탈로그 뷰로 전송하지 않고 로컬 사이트의 카탈로그 뷰 관리기에만 저장된다. 즉 초기에는 카탈로그 뷰가 로컬 사이트에만 저장하고 있다.

본 논문의 분산 XML 데이터베이스에서 XQuery 질의는 질의가 들어온 사이트에 XML 문서가 저장되어 있는지 검사한다. 만약 XQuery 질의가 들어온 사이트에 해당 XML 문서가 있는 경우에는 해당 사이트의 질의 변환기/질의 처리기에서 질의를 수행하고 XML 문서가 없는 경우에는 XML 데이터베이스를 구성하고 있는 모든 사이트에 XQuery 질의를 전송하여 질의하고자 하는 XML 문서가 있는 사이트의 질의 변환기/질의 처리기에서 질의를 수행하여 결과를 질의가 들어온 사이트로 전송하여 준다.

네트워크 모듈은 분산 객체 관계 데이터베이스 시스템 상에서 각 사이트 간의 통신을 담당한다. 조정자와 참여자 간의 파일 이동, 질의 전달, 제어 흐름 등에 관한 정보가 네트워크 모듈을 통해 이루어진다[6].

분산 XML 문서 저장 관리기는 XML 문서의 구조 정보를 유지하기 위해 엘리먼트의 경로를 기록하는 Pathexp 정보와 부모 엘리먼트 ID를 저장하고 있으며, 문서의 구조 검색 시 검색을 용이하게 하기 위하여 하이브리드 오더를 추가적으로 사용하였다[2]. 본 시스템의 분산 XML 저장 스키마는 그림 2과 같이 구성하여 각 노드의 정보를 분산 저장할 수 있게 하였다.

XMLDocument 릴레이션의 GlobalID는 <XML 문서의 이름>.<XML 문서를 저장하는 사이트 주소> 형식으로 저장되어 XML 문서가 어느 사이트에 저장되어 있는지 검사하게 된다. 그리고 XML 문서의 이름과 DTD나 XML 스키마가 같더라도 저장되는 사이트가 다른 경우에도 XML 문서의 저장이 가능하도록 하였다.

XML 문서를 저장하는 것은 문서 저장 관리기에서 담당한다. 문서 저장 관리기는 그림 3과 같다. 문서를 저장하기 위해 먼저 XML 문서의 유효성 검증 과정을 거친다. XML 문서가 DTD나 XML 스키마에 정의된 문법과 구조화 규칙 및 그 외의 추가적인 규칙을 따르는 Well-formed XML 문서인지를 판단한다. 후에 XML 문서를 파싱해야 하는데 파싱 방법에는 노드 중심의 DOM 파서와 이벤트 중심의 SAX 파서가 있다. SAX 파서가 DOM 파서보다 문서를 파싱하는데 걸리는 시간이 더 적고 대용량의 XML 문서를 파싱하는데 유리하기 때문에 본 논문에서는 SAX2 파서를 사용하였다. 문서를 파싱해서 발생된 이벤트 순서에 따라 문서 저장 관리기를 통해 엘리먼트, 애트리뷰트, 콘텐츠 테이블에 각각의 정보를 저장한다.

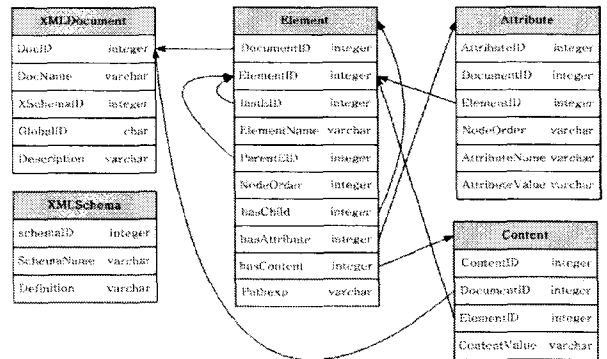


그림 2 분산 XML 저장 스키마

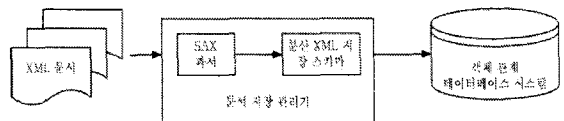


그림 3 문서 저장 관리기

SAX 파서로 XML 문서를 파싱하게 되면 문서상에서 이벤트가 발생한 순간에 미리 구현된 메소드를 수행하게 된다. 본 시스템에서는 XML 문서의 각 노드의 정보를 이벤트가 발생한 상황에 즉시 처리한다. 이벤트가 일어난 상황에 오더 레이블링이 수행되며 하이브리드 오더링 방법으로 쉽게 고유한 ElementID를 부여할 수 있다. 또한 현재 이벤트의 위치 경로를 기록하기 위해 Pathexp 정보를 저장한다. XML 문서가 분산 XML 데이터베이스 시스템에 저장된 후 XML에 관련되어 수정된 카탈로그의 변경 사항은 모든 사이트의 카탈로그 뷰 관리기에 전송되어 저장된다.

분산 저장된 XML 문서의 검색은 분산 XQuery 질의 처리기를 통해서 수행되는데, 분산 XQuery 질의 처리기

는 XQuery 질의를 분산 SQL 문으로 변환하여 처리하는 작업을 수행한다. 그리고 분산 XQuery 질의 처리기에 의해 반환된 결과 집합은 XML 문서 생성기에서 DOM 형식의 트리 구조로 재구성되어 XML 문서 조각을 형성하게 되거나 XQuery의 Return 절에서 정의한 형태의 XML 문서 형태로 반환된다. 이에 대한 세부적인 내용은 다음 절에서 알아본다.

4. 분산 XPath 질의 처리기

XQuery를 분산 SQL문으로 바꾸기 위해 XQuery에 사용되는 XPath를 뷰로 만들고 뷰를 이용하여 분산 SQL문을 생성한다. 따라서 본 절에서는 XPath 질의를 앞 장에서술한 분산 XML 저장 시스템 하에서 분산 SQL문으로 생성하는 방법에 대해서 알아본다. 본 절에서 설명하는 분산 XPath 질의 처리기는 [5]에서 개발한 질의처리기를 사용하였다.

분산 XPath 질의 처리기 구조는 그림 4와 같다. 그림 5의 dlab1 사이트에 저장되어 있는 book.xml 정보를 찾는 XPath 질의인 "/bookstore/book"을 질의 변환 단계를 거치면 그림 6과 같은 분산 SQL 문이 생성된다.

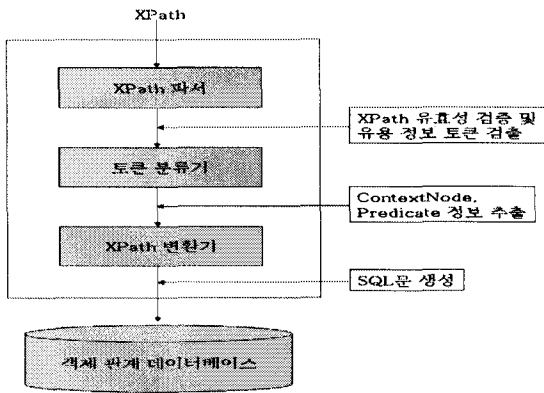


그림 4 분산 XPath 질의 처리기 구조

```
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

그림 5 book.xml

XPath 질의 처리 과정을 통해 만들어진 SQL문은 반환되어야 하는 엘리먼트의 범위 ID를 얻게 된다. 이렇게

생성된 SQL문을 XQuery에 사용한 변수에 매칭이 되도록 뷰를 생성한다. 생성된 뷰는 분산 XQuery 질의 처리기에서 이용하게 된다.

```
SELECT e1.id startID, e1.lastid lastID, e1.parentID
       parentID, e1.documentID xmldocumentID
FROM   helement@dblab1.book e1
WHERE  e1.xmldocumentID = 5 and e1.name='book'
       and e1.patexp='/bookstore/book'
```

그림 6 XPath 질의를 변환된 분산 SQL문

5. 분산 XQuery 질의 처리기

분산 XML 데이터베이스 시스템에 저장된 XML 문서 정보에 대한 질의 검색을 위해 사용자는 W3C 표준인 XQuery 질의어를 통해서 XML 문서 검색을 할 수 있다. 따라서 본 절에서는 XQuery 질의를 분산 SQL문으로 변환하는 분산 XQuery 질의 처리기를 설계하였다. 질의 처리기에 의해 변환된 SQL문은 적은 조인으로 결과를 검색할 수 있는 장점을 가지고 있다.

분산 SQL문은 그림 7의 분산 XQuery 질의 처리기에서 파서 단계, 토큰 분류 단계, FLWOR절 처리 단계, XQuery 질의 변환 단계의 4 단계를 거쳐서 생성된다. 각 단계별 처리 방법은 그림 8의 예를 통해서 설명한다.

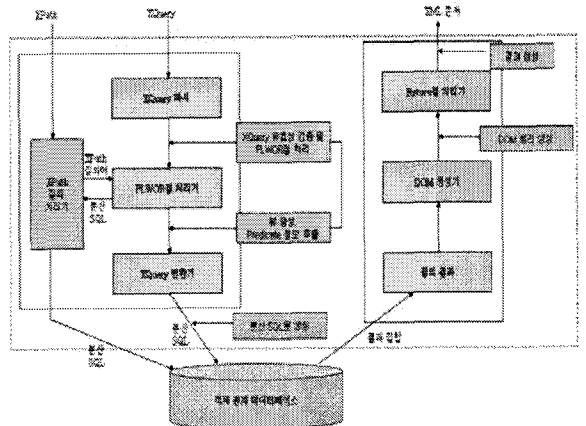


그림 7 분산 XQuery 질의 처리기

```
FOR $b in doc("books.xml")/bookstore/book
WHERE $b/price>30
ORDER BY $b/title
RETURN $b/title
```

그림 8 XQuery 질의 예

XQuery 파서 단계는 사용자의 XQuery 질의의 유효성을 검증하고 파스 트리를 생성한다.

XQuery 토큰 분류 단계에서는 XQuery 파서 단계에서 발생된 토큰을 FLWOR 절 별로 노드에 저장한다. For 절은 ForNode, Let 절은 LetNode, Where 절은 WhereNode에 저장하고, Order by 절은 OrderByNode에 저장하고, 마지막으로 Return 절은 ReturnNode에 저장

을 한다. 각 노드에 저장 되는 값은 파스 트리에 의해 생성된 이름과 값이 저장된다. Where 절에 대한 파스 트리에서 (Where, where), (ComparisonExpr, =), (VarName, b), (Slash, /), (Qname, title)이 저장된다.

FLWOR절 처리 단계에서는 먼저 FOR절에 있는 변수 b에 대한 XPath 경로 표현식 document("book.xml")/bookstore/book을 만들고 분산 XPath 질의 처리기에서 XPath 표현식에 대한 그림 6과 같은 분산 SQL 질의를 얻는다. 이 과정에서 질의를 하고자 하는 XML 문서가 저장된 사이트를 찾기 위해서 카탈로그 뷰 관리기의 정보를 조회한다. 이렇게 얻어진 질의를 사용해 변수 b에 대응하는 b_view 뷰를 생성한다. 그리고 변수 이름은 (변수,경로 표현식)을 쌍으로 하는 var_map에 저장한다. FLWOR 절에서 사용하는 변수에 대해서 var_map 을 찾으면 변수와 관련된 뷰를 가져올 수 있다. 또한 For 절의 토큰 정보를 SqlNode에 추가한다. SqlNode는 XQuery에 대한 SQL을 생성하기 위해 사용한다. 그리고 Let 절은For 절에서 변수를 뷰로 만드는 작업을 하고 Where 절은 조건에 대한 predicate 정보인 \$b/title과 var_map 에서 \$b 에 대한 경로 표현식 /bookstore/book을 추출하여 SqlNode의 Where 절에 토큰 정보를 추가한다. 마지막으로 Return 절은 { <book lang=", \$b/@lang, ">, \$b/title, </book> }과 같이 문자열과 변수를 구분해서 토큰을 생성하고 ReturnNode에 저장한다.

XQuery 질의 변환 단계는 반환해야 하는 요소 노드 정보를 추출하기 위해 SqlNode의 select 객체, from 객체, where 객체를 이용하여 분산 XML 데이터베이스에서 실행될 수 있는 분산 SQL문으로 변환한다. 각 객체는 FLWOR 처리 단계에서 생성한 토큰 정보가 저장되어 있다. Where 객체는 비교 연산자를 기준으로 경로 표현식과 검색 값으로 나누어진다. 경로 표현식의 유형에 따라 달리 분류가 되는데 만약 요소 노드의 내용을 검색하는 경우에는 엘리먼트 테이블과 콘텐츠 테이블을 참조하고, 두 테이블간의 조인을 위해 조인 조건이 필요하다. 만약 속성 노드 정보에 대한 검색일 경우에는 엘리먼트 테이블과 애트리뷰트 테이블이 사용되고, 두 테이블간의 조인을 위해 조인 조건이 필요하다. 만약 경로 표현식에 "/" 토큰이 존재하면 경로 표현식을 나타내는 필드인 PathExp 필드에 대해 단일한 경로 표현식으로 사상할 수 없으며 여러 개의 PathExp 필드에 대해 연산을 처리해야 한다. 예를 들면 경로 표현식 "/bookstore//book" 인 경우 "/bookstore" 와 "/book" 인 두 개의 부 요소 노드 (Sub-ContextNode) 정보로 나눠서 각각 문자열 검색을 통해 검색해야 한다.

앞의 단계를 거치면 예제 XQuery 질의는 뷰와 조건 정보를 이용해서 찾고자하는 엘리먼트의 범위를 구하는 질의를 생성한다. 그림 8과 같은 분산 SQL문이 생성된다. 이 분산 SQL문을 사용하여 원하는 엘리먼트를 검색하고, 검색된 결과는 DOM 생성기를 통해서 DOM 트리를 생성한다. DOM 트리는 RETURN 절의 표현식에 대한 결과를 얻기 위해 사용된다. 마지막으로 생성된 DOM 트리와 RETURN 절의 토큰을 이용해서 결과를 생성한

다.

```
SELECT b_view.*
FROM b_view, helement@dblab1.order e1,
      content@dblab1.order con1
WHERE b_view.documentID = e1.documentID
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 = 'title'
AND con1.ContentValue>30
```

그림 9 XQuery 질의의 변환된 분산 SQL문

6. 결론

본 논문에서는 분산 XML 데이터베이스를 기반으로 한 기존의 시스템에서 XML 문서를 분할하고 분산 저장하는 분산 XML 관리 시스템에서 XML 데이터를 접근할 수 있도록 하기 위해 XQuery를 분산 SQL문으로 변환하여 실행하는 분산 XQuery 질의 처리기를 설계 및 구현하였다. 분산 XQuery 질의 처리기는 XQuery 파서와 토큰 분류기, FLWOR절 처리기, XQuery 질의 변환기로 이루어져 있으며, XQuery를 분산 SQL로 변환할 때 분산 저장된 XML 문서의 정보를 확인하기 위해 각 사이트의 카탈로그 뷰 관리기를 사용하도록 하여 효율적이고 정확하게 분산 SQL로 변환하도록 하였다. 그리고 분산 SQL문을 수행하여 얻은 결과를 XQuery의 Return절에 정의한대로 XML 문서를 반환하도록 구현하였다.

7. 참고 문헌

- [1] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0," W3C Recommendation 4, Feb. 2004.
- [2] 김영우, 홍의경, "객체 관계 데이터베이스 시스템과 하이브리드 오더 인코딩을 이용한 XML 저장 시스템 설계 및 구현," 한국정보과학회 추계학술발표회 논문집 32(2), pp. 154-156, 2005.
- [3] World Wide Web Consortium, XML Path Language (XPath) 1.0, W3C Recommendation 16, Nov. 2004.
- [4] World Wide Web Consortium, XQuery 1.0: An XML Query Language W3C Candidate Recommendation 8, June 2006.
- [5] 이창주, 홍의경, "분산 객체 관계 데이터베이스 시스템을 이용한 분산 XML 문서 저장 시스템과 분산 XPath 질의 처리기 설계 및 구현," 한국정보과학회 가을학술발표대회 논문집 33(2), pp. 235-239, 2006.
- [6] 정성룡, 홍의경, "분산 환경에서 객체 관계 데이터베이스 시스템을 이용한 XML 문서 저장 시스템 구현 및 성능 비교," 한국정보과학회 추계학술발표회 논문집 32(2), pp. 151-153, 2005.