관계형 데이터베이스에서 XML 뷰 기반의 질의 처리 모델+

최규원O* 정채영* 김영옥* 김영균** 강현석* 배종민*
* 경상대학교 컴퓨터과학과/컴퓨터정보통신연구소, ** 한국전자통신연구원

{jinsilkwO, wcdi}@rtp.gsnu.ac.kr, yokim@kornet.net, kimyoung@etri.re.kr, {hskang, jmbae}@nongae.gsnu.ac.kr

A Query Processing Model based on the XML View in Relational Databases

Kyu-Won ChoiO Chae-Yeong Jeong* Young-Ok Kim* Young-Kyun Kim** Hyun-Syug Kang* Jong-Min Bae*
* Department of computer science, Gyeongsang National University /
Computer Information and Telecommunications Research Institute
** Electronics and Telecommunications Research Institute

요 약

XML 기반의 데이터베이스 통합을 위한 랩퍼 시스템은 자료 저장소의 내용을 XML로 표현한다. 본 논문은 관계형 데이터베이스의 내용을 XML 스키마로 표현하는 랩퍼 시스템에서 사용자가 XML 뷰를 정의했을 때, XML 뷰에 대한 사용자 질의어 처리에 대하여 논한다. XML 뷰를 정의하는 언어와 사용자 질의어가 모두 XQuery일 때, XML 뷰와 사용자 질의어의 합성, XQuery로 표현된 사용자 질의어를 SQL로 번역, 그리고 질의 결과를 XML 문서로 변환하는 XML 질의어 처리 알고리즘을 제시한다.

1. 서 론

이질의 데이터베이스를 통합하기 위해서는 각 데이터베이스의 내용을 하나의 표준 양식으로 표현해야 한다. XML은 전자문서 교환의 표준으로 인정받고 있기 때문에 이질의 데이터베이스 내용을 XML로 표현하는 것은 자연스러운 표현 방법이 된다. 이 때 사용자는 데이터베이스 내용을 XML 스키마로 간주하기 때문에 사용자는 XML 질의어로서 질의를 하게되고, 질의 결과로 XML 문서 형태를 얻는다. 이 때 관계형 데이터베이스에서 사용자가 뷰 테이블을 정의할 수 있듯이 사용자가 XML 뷰를 정의할 수 있어야 한다. XML 뷰에는 실제 데이터와 유사한 최하위 수준으로 보는 기본 XML 뷰와 사용자가 원하는 관점으로 정의한 응용 XML 뷰가 있다. 사용자는 응용 XML 뷰를 바탕으로 질의를 하고, 제시된 질의어는 뷰와 합성되어서 목적 언어로 번역되어야 한다.

여기서는 XML 질의어로서 W3C에서 제안된 XQuery[4]를 사용하고 관계형 데이터베이스를 대상으로 XQuery를 SQL로 변환하는 알고리즘에 대해서 논한다. 그리고 변환된 SQL 문장을 실행시킨 결과 집합을 XML 문서로 생성시키는 방법에 대해서 논한다. 본 논문에서 질의 처리의 기본 방향은 첫째, 사용자 질의어와 합성되는 XML 뷰를 트리로 표현한다. 둘째, 사용자 질의어를 뷰 트리와 합성, SQL로 변환 그리고 XML 문서를 생성하기 위한 템플릿(template)을 얻기 위해 질의어를 질의 트리로 표현하여 처리한다. 셋째, 질의 트리를 순회하여 XML 문서를 생성하되, 데이터베이스 커서 개념을 도입한다.

본 논문의 구성은 2장에서 XML 뷰 기반의 질의어 처리를 위한 기존의 연구들에 대해서 알아본다. 3장에서는 XML 뷰를 처리하기 위한 뷰 트리 모델을 제시한다. 4장에서는 질의 트리 구성에 대해서 알아보고, 5장에서 합성과 SQL 변환 알고리즘과 질의 트리의 생성 예를 보인다. 6장에서 질의 트리로 XML 문서를 생성함을 보이고, 마지막으로 7장에서 결론 및 향후 연구 과제에 대해서 논한다.

2. 관련 연구

본 논문과 유사하게 XML 뷰 기반으로 질의어 처리가 이루어지는 시스템 중에서 SilkRoute[1][3]와 XPERANTO[2]를 들 수 있다. SilkRoute는 관계형 데이터베이스 환경에서 설계된 미들웨어 시스템이다. 이는 XML 뷰를 처리하기 위해서 자체적으로 정의한 선언적인 질의어인 RXL(Relational to XML)을 사용하여 XML 뷰를 기술한다. RXL은 SQL(from과 where절) 부분과 XML-QL(construct절)의 조합으로 이루어진 질의어이다. 이는 블록 구조, 중첩된 질의 그리고 Skolem 함수를 지원하는 특징을 가지고 있다.

SilkRoute 환경에서 질의어 관리는 먼저 약속된 DTD와 관계 스키마를 참조하여 RXL로 작성된 XML 뷰가 정의된다. RXL 뷰 질의어와 사용자 질의어인 XML-QL과 합성되어 새로운 RXL이 생성된다. 그리고 합성된 RXL은 질의 변환 과정을 통해 SQL로 변환되어 관계형 데이터베이스에 접근하게 된다. 그런데, SQL로 변환하기 전에 합성된 결과를 최소화하는 단계가 필요한데, 이는 어려운 문제로 남아 있어 변환된 SQL 문장에 불필요한 내용이 존재한다.

XPERANTO 역시 관계형 데이터베이스 환경에서 설계된 미들웨어 시스템이다. 관계형 데이터를 XML 뷰로 정의하기 위해 XQuery를 사용하고 사용자 질의 또한 XQuery이다. 사용자 질의가 들어오면 먼저 파싱하여 XQGM(XML Query Graph Model)이라 불리는 중간 질의 표현으로 변환된다. XQGM은 QGM(Query Graph Model)이라 불리는 SQL 중간 질의 표현의 확장으로 볼 수 있고, XML 질의 대수의 개념이 추가되어 설계되었다. XML 뷰를 정의한 XQuery와 사용자 질의 XQeury는 각각 XQGM으로 변환되어 합성하게 된다. 합성된 결과도 중간 모델인 XQGM이고 이것은 SQL로 변환되는 부분과 XML 문서를 생성하기 위한 부분으로 분리되어 각각 실행된다.

본 논문에서는 XML 뷰와 사용자 질의어를 XQuery로 표현한다. XML 뷰를 트리로 표현한 뷰 트리를 구성하고, 또한 질의어에 대해서는 뷰 트리를 기반으로 하여 질의 트리를 구성해 나간다. 질의 트리를 구성하는 과정에서 합성과 SQL로의 번역이 이루어지며, 질의 트리는 또한 XML 문서를 생성시키는 틀 역할을 한다.

3. 뷰 트리

XQuery로 작성된 XML 뷰는 트리로 표현된다.

3.1 뷰 트리 생성

그림 1은 XQuery로 작성된 XML 뷰의 예이다. XML 뷰를 트리로 구성하기 위해서 먼저 가상 루트를 만든다. 그리고 XQuery에 나타난 모든 엘리먼트는 이 트리의 노드가 된다.

```
<Auction>
  FOR $x IN view("db")/users/tuple
  RETURN
     <Users>
        <Name ID ={$x/id/text()}> $x/name/text() </Name>
        { FOR $y IN view("db")/bids/tuple
          WHERE $y/userid = $x/id
          RETURN
        <Bids>
                <ItemNum> $y/itemno/text() </ItemNum>
                <Bid> $y/bid/text() </Bid>
        </Bids>
        }
        <Rating> $x/rating/text() </Rating>
     </Users>
</Auction>
```

그림 1 XML 뷰

그림 2는 그림 1에 대한 뷰 트리이다. Name 노드 옆의 사각형은 애

+ 본 연구는 한국전자통신연구원의 위탁연구과제인 자료 저장소 wrapper 개발의 일부로 수행된 결과임

트리뷰트를 의미하고 본 연구에서는 애트리뷰트를 엘리먼트 노드의 필드로 취급한다. Users 노드와 Bids 노드 사이의 링크에 있는 레이블은 그 하위의 노드가 리턴될 수 있는 조건을 의미한다. 하나의 조건은 그 하위에 있는 모든 노드에 적용되어야 한다.

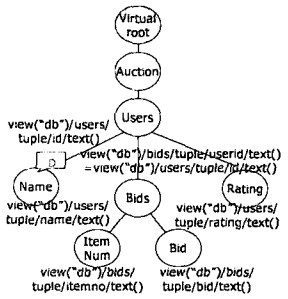

그림 2 뷰 트리

3.2 뷰 트리 노드의 구조

뷰 트리의 각 노드는 표 1과 같은 필드들로 구성된다.

필드	내용
이름	엘리먼트 이름
타입	비단말노드, 단말노드(텍스트), 단말노드(빈태그)
부모 노드	현재 노드의 부모 노드 포인터
첫 번째 자식 노드	현재 노드의 첫 번째 자식 노드 포인터
다음 형제 노드	현재 노드의 다음 형제 노드 포인터
값	경로식 스트링
조건	XQuery의 WHERE절 내용
애트리뷰트	애트리뷰트 이름, 애트리뷰트 값

표 1 뷰 트리 노드 정보

'이름' 필드는 XML 뷰에 나타나는 엘리먼트의 이름을 의미한다. '부모 노드' 필드, '첫 번째 자식 노드' 필드 그리고 '다음 형제 노드' 필드는 현재 엘리먼트를 기준으로 부모 엘리먼트, 첫 번째 자식 엘리먼트 그리고 다음 형제 엘리먼트 의미하며 각각 해당 노드의 포인터 값을 가진다. '타입' 필드는 트리를 구성하는 각 노드가 단말 노드인지 비 단말 노드인지를 구별하는데 세 가지로 분류한다. 비 단말 노드, 값을 가지는 단말 노드 그리고 값을 가지지 않는 단말 노드로 나눈다. '값' 필드는 단말 노드이면서 값을 가지는 노드인 경우에 값을 가지게 되는 필드로 경로식이 값으로 들어간다. 다음 '조건' 필드는 XML 뷰에 나타나는 WHERE절에 있는 내용이 들어가는데 해당 노드가 생성되기 위해서는 '조건' 필드에 들어있는 조건을 만족해야함을 의미한다. 마지막으로 '애트리뷰트' 필드는 엘리먼트에 애트리뷰트가 존재할 경우 값이 들어간다. 여기서 값은 애트리뷰트 이름과 애트리뷰트 값을 의미한다. 각 노드는 이러한 필드로 구성되어 정보를 유지하게 된다.

본 논문에서는 이 정보를 이용하여 사용자 질의어와 합성되고 SQL로 변환한다. 그리고 뷰 트리는 다음과 같은 역할을 한다.

- 뷰 트리에는 실제 데이터베이스의 테이블 명과 컬럼 명을 가지고 있기 때문에 SQL 문장을 생성할 수 있는 정보를 얻을 수 있다.
- 사용자 질의어에 나타나는 임의의 경로식에 적용되어야하는 조건 정보를 파악할 수 있다.
- 사용자 질의어의 경로식에 내재된 구조적인 정보를 쉽게 알 수 있다.

4. 질의 트리

사용자 질의어가 들어오면 이를 실제 데이터베이스에 실행시킬 수 있는 질의어로 변환하고 변환된 질의어를 실행시켜 그 결과를 XML 문서로 출력해야 한다. 이를 위해서는 사용자 질의어와 XML 뷰를 합성하는 문제와 SQL로 변환하는 문제 그리고 XML 문서를 생성하는 문제를 모두 고려한 질의 처리 방법을 모색해야 한다.

뷰 트리와 마찬가지로 사용자 질의어도 트리로 표현한다. 이를 질의 트리라 한다. 질의 트리를 바탕으로 XML 뷰와의 합성과 SQL로의 변환이 동시에 이루어진다. 또한 질의 트리는 XML 문서를 생성하기 위한 템플릿이 된다.

질의 트리를 구성하는 노드 형으로는 가상루트 노드, XQueryElement 노드 그리고 XQuerySQL 노드가 있다. 가상 루트 노드는 뷰 트리와 마찬가지로 모든 노드의 루트가 되는 가상적인 노드이다.

XQueryElement 노드는 사용자 질의어에 나타난 엘리먼트와 뷰 트리와의 합성으로 인해 추가되는 엘리먼트에 해당한다. 이 노드는 XML 문서를 생성하기 위한 템플릿 역할을 하여 XML 문서에 태그로 나타난다. XQueryElement 노드는 표 2와 같은 필드로 구성된다.

필드	내용
이름	엘리먼트 이름
타입	1 (XQueryElement)
부모 노드	현재 노드의 부모 노드 포인터
첫 번째 자식 노드	현재 노드의 첫 번째 자식 노드 포인터
다음 형제 노드	현재 노드의 다음 형제 노드 포인터
값	경로식 스트링
위치	SQL의 SELECT절에서 컬럼 위치
애트리뷰트	애트리뷰트 이름, 애트리뷰트 값

표 2 질의 트리의 XQueryElement 노드 정보

XQueryElement 노드의 대부분의 필드는 뷰 트리의 노드에서 설명하였다. '위치' 필드는 SQL의 SELECT 절에 있는 컬럼 위치를 말하는데 해당 XQueryElement 노드가 태거화(tagged) 될 때 컬럼 위치에 있는 값을 가져와 삽입한다.

XQuerySQL 노드는 사용자 질의어에 나타나는 FLWR 절과 대응된다. 하나의 FLWR 절에 대해 하나의 XQuerySQL 노드를 생성시켜 질의 트리에 추가한다. FLWR 절에서 SQL 문장으로 변환된 내용은 기본적으로 해당 XQuerySQL 노드에 추가된다. 단, FLWR 절에서 사용된 변수가 다른 FLWR 절에서 선언된 변수일 경우는 변환된 SQL 내용의 일부가 다른 XQuerySQL 노드에 추가될 수도 있다. 그리고 XQuerySQL 노드는 그 하위의 모든 노드가 반복될 수 있음을 의미한다. XQuerySQL 노드는 표 3과 같은 필드로 구성된다. XQuerySQL 노드의 대부분의 필드는 뷰 트리의 노드에서 설명하였다. 표 3에서 하위 네 개의 필드에는 변환된 SQL 문장이 해당하는 필드에 추가되고 질의어를 실행시킬 때 네 개의 필드 내용을 합하여 하나의 SQL 문장으로 구성해서 실행시킨다.

필드	내용
이름	없다.
타입	2 (XQuerySQL)
부모 노드	현재 노드의 부모 노드 포인터
첫 번째 자식 노드	현재 노드의 첫 번째 자식 노드 포인터
다음 형제 노드	현재 노드의 다음 형제 노드 포인터
SELECT절	SQL의 SELECT절 내용
FROM절	SQL의 FROM절 내용
WHERE절	SQL의 WHERE절 내용
ORDER BY절	SQL의 ORDER BY절 내용

표 3 질의 트리의 XQuerySQL 노드 정보

5. 합성과 SQL 변환 알고리즘

5.1 알고리즘

합성과 변환과정의 개략적인 과정을 알고리즘으로 표현하면 다음과 같다.

```
void userXQueryProcessing(parsedUserQuery, viewTree) {
    queryTree = createQueryTreeObject();
    makeVirtualRoot(queryTree);
    while(!endOfQuery) {
        token = getNextToken(parsedUserQuery);
        if(token == startTag) {
            q = createXQueryElement();
            //질의 트리에 XQueryElement 노드 추가
            append(queryTree, q);
            while(exits attributes) process attribute;
        }
        else if(token == startOfFLWR) {
            sql = createXQuerySQLObj();
            // 질의 트리에 XQuerySQL 노드 추가
            append(queryTree, sql);
            token = getNextToken();
            do {
                if(token == "FOR" or "LET") {
                    (varName, varValue) = getVariable();
                    path = getPath(viewTree, varValue);
                    sql.setVar(varname, path);
                    process predicates;
                }
                else if(token == "WHERE") {
                    do {
                        condition = getCondition();
                        sqlCondition = translateSQL(condition);
```

```
                    sql.setWhereClause(sqlCondition);
                } until(endOfWhere)
            }
        } until(token == "RETURN")
    }
    else if(token == pathExpr) {
        targetNode = getNodePtr(viewTree, token);
        temp = makeQuerySubTree(targetNode);
        append(queryTree, temp);
        sqlcomp[] = convertSQLComponents(temp);
        sql.setAttribute(sqlcomp);
    }
    else if(token == endOfFLWR) {
        viewcond = getViewCondition();
        sqlcomp[] = convertSQLComponents(viewcond);
        sql.setWhereClause(sqlcomp);
    }
    else if(token == "SORTBY") {
        sqlsortby = translateSQL(sortbyClause);
        sql.setOrderByClause(sqlsortby);
    }
  }
}
```

질의어 처리의 첫 단계로서 질의 트리에 대한 루트 노드를 생성한다. 그리고, 합성 과정에서 생성되는 정보로써 질의 트리를 구성해 나가는 방식으로 진행된다. 질의어에서 시작 태그를 만나면, 질의 트리 노드를 생성하여 질의 트리에 추가한다.

다음으로 질의어에서 FLWR 절을 만나면 SQL 문장을 생성하기 위한 준비를 하고, FOR나 LET 절에서 제시된 변수나 조건을 처리한다. 이를 위하여 먼저 SQL로 변환하기 위한 정보를 모아둔 노드 구조인 XQuerySQL 형의 객체를 생성하여, 이를 질의 트리에 추가한다.

FOR나 LET절에서 변수를 만나면 변수에 대한 값(경로식을 나타내는 스트링)을 얻어서, 변수의 값 즉, 경로식에서 나타난 목적 노드(target node)를 뷰 트리에서 찾아서 그 노드의 위치를 XQuerySQL 형의 객체에 기억시킨다. 이는 나중에 질의어에서 변수에 대한 참조가 발생할 때 그 변수를 재작성(rewriting)하는데 사용된다. 그리고 WHERE 절을 만나면 조건에 대응되는 SQL 문의 일부를 생성시켜서, XQuerySQL 형의 객체에 기억시킨다.

질의어에서 경로식에 대한 참조가 발생하면, 뷰 트리에서 경로식에 해당하는 노드를 찾아서, 이 노드에 대응하는 질의 트리를 구성한 다음, 질의 트리에 추가한다. 그리고 이 경로식에 대응되는 SQL 문장 요소를 생성시켜서 XQuerySQL 객체에 저장한다.

5.2 질의 트리 생성 예

여기서 사용자 질의어에 대한 XML 뷰는 3.1절의 예를 사용한다. 그림 3은 경매에 참가한 사람들 중에 등급이 A보다 큰 사람의 이름과 경매 내역을 검색하는 사용자 질의어이다.

```
<Result>
  FOR $u IN view("auction")//Users
  WHERE $u/Rating > "A"
  RETURN
    <User>
      <Name> $u/Name/text() </Name>
      { $u/Bids }
    </User>
  SORTBY(Name)
</Result>
```

그림 3 사용자 질의어

질의 트리를 구성하기 위해 먼저 가상 루트를 만든다. 사용자 질의어에서 'Result' 엘리먼트로 시작하므로 'Result'라는 이름을 가지는 XQueryElement 노드를 만들어 가상 루트 아래에 추가한다. 다음 FLWR 절이 나타나므로 XQuerySQL 노드를 만들어 추가한다. FOR 절에서 변수 '$u'의 값은 'view("auction")//Users'이므로 XQuerySQL 노드의 해시 테이블에 변수 이름과 뷰 트리에서 이 경로식에 해당하는 노드의 포인터를 기억시킨다.

XQuery의 WHERE 절의 내용은 SQL의 WHERE 절로 변환되어 XQuerySQL 노드에 SQL 문장으로 추가한다. 뷰 트리에서 그 경로식에 해당하는 노드를 찾아서 노드의 정보를 유지하는 필드 중에 '값' 필드에 있는 내용에서 'tuple'을 기준으로 tuple 바로 앞이 테이블 명이 되고 tuple 바로 뒤의 내용이 컬럼 명이 된다. 따라서 경로식 '$u/Rating'에서 변수 '$u'에 대한 포인터 값을 가져와 뷰 트리의 해당 포인터 위치에서 'Rating'이름을 가지는 자식 노드를 찾는다. 찾은 노드의 '값' 필드 내용'view("db")/users/tuple/rating/text()'에서 tuple 앞의 users 테이블 명과 tuple 뒤의 rating 컬럼 명을 추출한다. 따라서, XQuery의 WHERE 절은 SQL의 WHERE절(users.rating > 'A')로 변환되어 XQuerySQL 노드에 기억된다.

그리고 계속해서 User 엘리먼트 노드와 Name 엘리먼트 노드가 추가되고 Name 노드에 대한 SQL 문장이 생성되어 추가된다. 그리고, 경로식 '$u/Bids'는 뷰 트리에 조건 정보가 있으므로 SQL로 변환되어 추가된다. 뷰 트리에서 Bids 노드 아래에 ItemNum 노드와 Bid 노드가 있으므로 질의 트리에 추가되고 변환된 SQL 문장은 XQuerySQL 노드에 추가된다. XQuery의 SORTBY는 SQL의 ORDER BY로 변환된다. 이러한 과정을 거치면 XML 문서를 생성하기 위한 템플릿으로 사용될 질의 트리가 그림 4와 같이 구성되고 합성과 변환 작업이 끝난다. 그림 5는 질의 트리에서 XQuerySQL 노드에 기억되어 있는 SQL로 변환된 질의어이다.

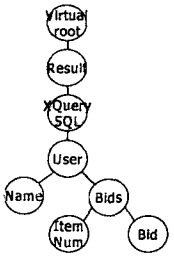

그림 4 질의 트리

```
SELECT users.name, bids.itemno, bids.bid
FROM users, bids
WHERE users.rating > 'A'
        AND bids.userid = users.id
ORDER BY users.name
```

그림 5 SQL 변환 결과

6 XML 문서 생성

질의 트리는 사용자가 제시한 질의어에 대한 구조를 나타내는 것이기 때문에 그 자체가 XML 문서 생성을 위한 틀이 된다. 따라서 질의 트리를 순회함으로써 쉽게 XML 문서를 생성시킬 수 있다. 질의 트리를 구성하는 노드 중에 XQuerySQL 노드가 SQL 문장을 가지고 있기 때문에 질의어를 실행시켜 결과 셋(Result Set)을 얻을 수 있다. XML 문서를 생성할 때 하나의 문서로 만들 수도 있지만 검색 결과의 크기가 큰 경우를 대비하여 데이터베이스 커서의 개념을 이용한다. 본 논문에서는 XQuery에서 최상위 수준의 FLWR 절을 커서 단위로 정의한다. 질의 트리를 깊이 우선 탐색하여 XML 문서를 생성하는데, XQuerySQL 노드를 만나면 SQL 문장을 실행시켜서 반환된 결과 셋의 크기만큼 트리의 하위 노드를 반복 순회하여 XML 문서를 구성한다.

7 결론 및 향후 과제

본 논문에서는 XML 뷰 기반의 질의어 처리 모델을 설계하였다. XQuery로 작성된 XML 뷰를 트리로 표현하여 구조적인 정보와 각 엘리먼트에 대한 조건 정보 등을 유지하여 사용자 질의어와 합성 시 뷰 트리에서 정보를 얻을 수 있게 하였다. XQuery로 작성된 사용자 질의어 또한 트리로 표현하여 뷰 트리와의 합성, SQL로의 변환 그리고 템플릿 구성이 동시에 이루어지는 질의어 처리 알고리즘을 개발하였다. 그리고, 질의 트리를 순회하여 XML 문서를 생성하고 커서를 정의하여 사용하였다. 설계된 모델은 Microsoft SQL Server 2000 데이터베이스를 사용하고 자바로 구현되었다.

향후 과제로는 변환된 SQL 질의어의 최적화 문제에 대해서 연구하고 XQuery로 작성된 사용자 질의어를 관계형 데이터베이스 외에 다른 이질적인 데이터베이스에 실행시킬 수 있는 질의어로 변환하기 위한 질의어 처리 알고리즘을 연구해야 한다.

참고 문헌

[1] M. Fernandez, W. Tan, and D. Suciu, "SilkRoute : Trading between Relations and XML", WWW9, pp. 723-745, 2000.
[2] J. Shanmugasundaram, J. Kiernan, E. Shekita, C. Fan, and J. Funderburk, "Querying XML Views of Relational Data", VLDB Conference, pp. 261-270, 2001.
[3] M. Fernandez, A. Morishima, and D. Suciu "Efficient Evaluation of XML Middle-ware Queries", SIGMOD, pp. 103-114, 2001.
[4] World-Wide Web Consortium, "XQuery 1.0: An XML Query Language", http://www.w3.org/TR/xquery/.
[5] World-Wide Web Consortium, "XML Query Use Cases", http://www.w3.org/TR/xmlquery-use-cases.