바다-II에서 XML 관리자의 설계와 구현

차명훈,* 박영철**
영진전문대학,* 경북대학교**
mhcha@yeungjin.ac.kr, ycpark@knu.ac.kr

Design and Implementation of an XML Manager for BADA-II

Myung Hoon Cha,* Young Chul Park**
Yeungjin Junior College,* Kyungpook National University**

요 약

XML 문서를 관계형 데이타베이스에 저장하는 것은 관계형 데이타베이스 관리 시스템이 제공하는 안정성과 질의 능력을 활용하고자 하는 것이 주된 요인이다. 관계형 데이타베이스는 2차원의 테이블 구조를 저장하고 응용하는 것에 특화되어 있으므로 XML 문서에 표현되어 있는 계층적 구조, 중첩 구조 등 이질적인 요소들을 관계형 데이타베이스에 반영하는 모델링 방법으로 현재까지 완벽하게 실용화된 것은 없다. 본 논문은 XML 문서를 관계형 데이타베이스에 저장하기 위한 스키마 구조를 제안하고 XML 문서를 바다-II에 저장, 검색, 재구성한 구현 결과를 제시한다.

1. 서론

HTML은 인터넷의 확산을 가져오는데 큰 역할을 하였으나, 인터넷이 활성화되면서 대규모로 구축된 웹기반의 데이타를 효율적으로 검색하기에는 적합하지 않다. W3C에서는 HTML이 가지는 표현의 한계를 극복하기 위하여 SGML을 사용하기 쉽도록 단순화한 XML을 제안하였다. XML은 SGML의 부분집합으로서 SGML이 가지는 메타언어로서의 기능 등을 사용하기 용이하도록 단순화시켜 방대한 내용을 가지고 있는 문서를 효율적으로 검색할 수 있도록 표현할 수 있다[1, 2, 3, 4].

XML을 사용하여 작성된 문서를 저장하는 기법은 일반적인 화일 시스템에 저장하는 방법과 관계형 데이타베이스 또는 객체지향 데이타베이스에 저장하는 방법, 그리고 XML 문서 전용 저장 관리 시스템을 사용하는 방법의 세가지로 구분된다[4, 5, 6, 7, 9].

본 논문은 XML 문서를 관계형 데이타베이스에 저장하는데 적용할 수 있는 방법을 제시하며, 제안한 기법을 바다-II[10]에 구현하였다.

2. 관련 연구

XML 문서를 화일 시스템에 저장하는 방법은 다중 사용자 지원, 보안 문제, 데이타 비일관성, 회복 문제 등 화일 시스템에 내재한 단점때문에 사용하기에는 효율적이지 않다. 현재는 XML 문서를 주로 관계형 데이타베이스 시스템에 저장하거나 XML 문서를 전용으로 저장하기 위한 시스템을 개발하는 방향으로 관련 연구가 진행되고 있다. XML 문서를 관계형 데이타베이스에 저장하기 위한 시도로는 Oracle 9i 등을 예로 들 수 있고[6], 객체지향 데이타베이스를 활용하여 XML 문서를 저장하기 위한 시도의 결과로는 eXcelon 시스템 등이 있다[7]. XML 문서 전용 저장 관리 시스템은 저장 시스템 수준에서 XML 문서를 효율적으로 저장하기 위한 구조로 설계하는 분야로서, 스탠포드 대학의 Lore 시스템을 XML 전용 저장 관리 시스템의 예로 들 수 있다[4, 5]. 높은 시장 점유율을 보유한 기존 관계형 데이타베이스 관리 시스템을 사용하는 경우에는 XML 문서를 저장하기 위한 전용 테이블들을 미리 생성한 후 XML 문서를 파싱한 결과를 그 테이블들에 저장하거나 LONG 형 또는 CLOB(Character Large OBject)의 형태로 문서 자체를 저장한다. XML 문서를 CLOB의 형태로 저장하게 되면 정보 검색의 효율이 저하되는 것이 단점이다. 본 논문은 관계형 데이타베이스 시스템이 제공하는 질의 능력을 충분히 활용하기 위하여 XML 문서를 관계형 테이블로 매핑하는 모델링 방법을 제시하며, XML 문서의 저장, 저장된 문서에 대하여 요소(element) 및 속성(attribute)에 대한 질의, XML 문서의 재구성이 가능하도록 하였다.

3. XML 문서의 저장 및 처리

XML 문서를 저장하고 저장된 문서에 대한 효율적인 질의를 수행하기 위하여 시스템 구조 측면에서 클라이언트와 서버의 부하 분배 문제를 결정하여야 한다. XML 문서를 파싱하는 역할을 클라이언트 또는 서버 어느 한 쪽에 전담시킬 것인지 양자에게 균등 분배를 시킬 것인지를 고려하고, XML 문서에 대한 질의가 제시될 경우에는 질의의 결과를 클라이언트 또는 서버가 재구성할 것인지를 결정하여야 한다. 첫 번째 방법은 XML 문서의 파싱과 재구성 역할을 서버가 수행하도록 하는 것이다. 이 경우에는 클라이언트에서 XML 문서 자체를 CLOB의 형태로 서버에 전송하고 서버에서 문서를 파싱한 후 그 결과를 테이블들에 저장하며, 질의에 대해서는 서버가 결과를 조합하여 XML 문서를 생성한 후 그 문서를 CLOB의 형태로 클

†본 논문은 한국과학재단 목적기초연구(과제번호 : 2000-2-51200-002-3) 지원으로 수행되었음.

라이언트로 전송하게 되므로 서버의 부담이 증대되는 것이 단점이 된다. 그러나, 클라이언트의 부하가 적으며 통신 비용이 최소화되는 것이 장점이다.

두 번째 방법은 XML 문서의 파싱과 재구성 역할을 클라이언트가 수행하도록 하는 것이다. 이 경우에는 클라이언트가 문서를 파싱한 결과를 데이타베이스 관리 시스템에서 제공하는 기본 데이타형으로 변환한 후 서버에게 전송하며, 질의가 제시되면 서버가 질의를 처리한 결과를 추출한 후 데이타베이스 관리 시스템에서 제공하는 기본 데이타형으로 클라이언트에게 전송하고 클라이언트는 전송받은 데이타를 XML 문서로 재구성한다. 그 결과 서버의 부하가 최소화되지만, 통신 비용이 증가된다.

본 논문은 서버의 부하를 감소시키기 위하여 두 번째 방법의 시스템 구조를 선택하였다. 그 구조는 그림 1과 같다.

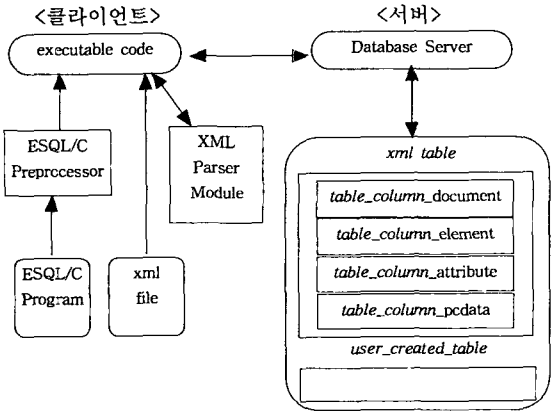

그림 1. 시스템 구조

그림 1에서 서버 측에 XML 문서 전용 테이블들이 존재하고, 서버가 질의를 처리할 때 XML 문서 전용 테이블들을 일반 테이블과 동일하게 취급한다. 클라이언트에는 파서 모듈이 있어서 XML 문서의 파싱을 수행한다. 파싱 결과 분리된 토큰들은 데이타베이스 관리 시스템이 제공하는 기본 데이타형으로서 서버에 전달된 후 XML 전용 테이블들에 저장된다.

XML 문서를 저장한 후 질의를 처리하기 위해서는 XML 문서를 구성하고 있는 선언부, 요소, 속성, PCDATA 등의 구조와 내용에 관한 정보를 모두 이용할 수 있도록 설계할 필요가 있다. XML 문서를 포함한 데이타를 처리하는 ESQL/C 코드와 API 코드의 예는 그림 2와 같다.

```
EXEC SQL BEGIN DECLARE SECTION;
    char        dept_name[20];
    int         doc_id;
EXEC SQL END DECLARE SECTION;
EXEC SQL CREATE TABLE department (
    dept_id     int,
    dept_name   varchar(20),
    employee    xml
);
doc_id = xml_new_doc_id();
EXEC SQL INSERT INTO department VALUES (
    1, "총무부", :doc_id);

xml_insert_doc("department",    "employee",    doc_id,
"chongmu_employee.xml");

EXEC SQL SELECT dept_name, employee
        INTO :dept_name, :doc_id
        FROM department
        WHERE employee.attribute_name = "hobby"
          and  employee.attribute_value = "football";

file_name = xml_reorganize_doc("department", "employee",
doc_id);
```

그림 2. XML 처리 ESQL/C 프로그램의 예

그림 2의 테이블 생성 구문에서 xml이라는 칼럼의 자료형을 명시적으로 제공함으로써 서버는 XML 문서가 들어온다는 것을 알 수 있고, 서버는 내부적으로 XML 데이타를 저장하기 위한 XML 전용 테이블들을 생성한다. xml 칼럼은 내부적으로 int 형으로 변경되어 XML 문서 식별자를 저장하는 공간이 된다. 그림 2에서 xml_로 시작하는 함수는 XML 문서를 처리하는 API 함수들이다. xml_new_doc_id()는 하나의 XML 문서에 대하여 시스템 전체에서 유일한 문서 식별자를 할당하는 함수로서 서버가 그 값을 반환한다. xml_insert_doc() 함수는 테이블 이름, 칼럼 이름, XML 문서 식별자와 문서 이름을 인자로 받은 후 그 문서를 파싱하고 그 결과를 XML 전용 테이블에 저장하는 역할을 수행하는 API 함수이다. 사용자가 xml이라는 칼럼형을 가지는 테이블을 생성하면 그 테이블은 그 칼럼당 하나의 XML 문서를 저장할 수 있고, 하나의 XML 문서는 시스템 전체에서 유일한 문서 번호를 가진다. WHERE절에서 사용된 employee.attribute_name과 employee.attribute_value라는 구문은 employee라는 이름을 가진 xml 칼럼을 하나의 객체로 취급하여 그 속성으로 attribute_name과 attribute_value를 지원하도록 설계한 예이다. 이러한 형식의 구문을 지원함으로써 XML 문서를 저장한 XML 전용 테이블들의 존재는 사용자에게 은닉되고, 사용자는 자신이 직접 생성한 기본 테이블 이름만 FROM 절에 명시하면 필요한 XML 문서의 문서 식별자를 구할 수 있다. xml_reorganize_doc() 함수는 테이블 이름, 칼럼 이름, XML 문서 식별자를 인자로 받은 후 XML 문서를 저장한 XML 전용 테이블에서 데이타를 인출한 후 그 결과를 재구성하여 새로운 XML 화일을 생성한다.

XML 문서를 저장하는 용도의 XML 전용 테이블들을 구성하는 방법으로 다음과 같이 세가지 방법들을 고려할 수 있다. 첫째, 시스템 전체에서 하나의 XML 전용 테이블 집합을 두어 XML 데이타를 저장한다. 둘째, 사용자가 xml 칼럼형을 가진 하나의 테이블을 생성할 때마다 하나의 XML 전용 테이블 집합을 둔다. 셋째, 하나의 xml 칼럼형 당 하나의 XML 전용 테이블 집합을 둔다. 본 논문은 세 번째 기법을 채택하여 XML 전용 테이블 집합을 구성한 것이 특징이다. 하나의 칼럼 당 XML 전용 테이블 집합을 생성하면 권한 관리의 문제점이 배제될 뿐

아니라 질의 수행 속도가 신속해지는 것이 장점이다.

그림 3은 그림 2의 department 테이블을 생성할 경우, 자료형이 xml인 employee 칼럼에 대하여 내부적으로 생성하는 XML 전용 테이블 집합의 생성 구문이다.

```
CREATE TABLE department_employee_document (
    doc_id    int  primary key,
    encoding     varchar(ENC_LEN),
    version      varchar(VER_LEN),
    xml_filename varchar(FILE_LEN),
    dtd_filename varchar(FILE_LEN)
);

CREATE TABLE department_employee_element (
    doc_id int,
    element_id  int,
    parent_id int,
    element_name varchar(ELE_LEN),
);

CREATE TABLE department_employee_attribute (
    doc_id int,
    attribute_id int,
    parent_id int,
    attribute_name varchar(NAME_LEN),
    attribute_value varchar(VAL_LEN)
);

CREATE TABLE department_employee_pcdata (
    doc_id int,
    pcdata_id int,
    parent_id int,
    pcdata varchar(VAL_LEN)
);
```

그림 3. 하나의 xml 칼럼형마다 생성하는 XML 전용 테이블 집합의 생성 구문

그림 3의 department_employee_document 테이블은 시스템 전체에서 하나의 XML 문서를 유일하게 식별하는 doc_id, 문자집합 값을 설정하는 encoding, 문서의 버전을 나타내는 version, 저장될 XML 문서 화일의 이름을 나타내는 xml_filename, DTD 화일의 이름을 저장하는 dtd_filename으로 구성된다. department_employee_element 테이블은 department_employee_document 테이블의 doc_id 필드를 참조하는 doc_id, XML 문서를 파싱하면서 하나의 요소마다 부여받는 그 문서 안에서 유일한 값인 element_id, 부모 요소의 element_id를 가리키는 parent_id, 요소의 이름을 저장하는 element_name 필드를 가진다. department_employee_attribute 테이블과 department_employee_pcdata 테이블의 경우에도 department_employee_document 테이블의 doc_id 필드를 참조하는 doc_id, XML 문서를 파싱하면서 할당받는 그 문서 안에서 유일한 값을 가지는 attribute_id(pcdata_id) 필드 등으로 구성된다.

XML문서를 파싱한 후 그 결과를 DOM[2] 형식의 트리로 구성하고 XML 전용 테이블 집합에 그 트리의 구성 항목들을 저장한다. XML 전용 테이블 집합에 저장된 자료 값들을 이용하여 문서를 재구성하기 위해서는 클라이언트가 질의를 통하여 데이타베이스 시스템의 기본 데이타형으로 전달된 결과값들을 조합하여 DOM 형식의 트리를 구성한 후 그 트리의 노드들을 조회하여 XML 문서를 생성한다.

본 논문에서 수행된 작업은 Solaris 2.7 운영체제를 가지는 Sun Ultra Sparc 기종에서 파서는 lex, yacc을 이용하여 구현하였고, 파서 이외의 모듈은 C 언어를 사용하여 바다-II에 구현하였다.

4. 결론

XML로 작성된 문서의 수요가 증가함에 따라 대량으로 생성된 XML 문서들을 저장하고 관리하는 방법이 필요하며, 현재 XML 문서를 화일 시스템, 관계형 데이타베이스, 객체지향 데이타베이스, XML 전용 저장 관리 시스템에 저장하는 방법들이 주로 연구되고 있다. XML 문서를 관계형 데이타베이스에 저장하는 기존 방법들은 문서 전체를 LONG 또는 CLOB 형태로 저장하거나, 시스템 전체에 하나의 테이블 집합을 두어 XML 문서에서 튜플들로 모델링하여 저장하였다. 본 논문은 질의의 속도와 권한 관리의 문제점을 배제하기 위하여 하나의 xml 칼럼마다 하나의 XML 전용 테이블 집합을 생성하고 XML 문서 조회를 위한 질의문을 설계함으로써 관계형 데이타베이스 관리 시스템이 제공하는 질의의 다양성과 시스템의 안정성을 활용할 수 있도록 하였다.

참고문헌

[1] W3C, Extensible Markup Language(XML) 1.0, http://www.w3.org/TR/2000/REC-xml-20001006.html, Oct, 2000

[2] Jonathan Robie, Texcel Research, "What is the Document Object Model?", http://www.w3.org/TR/WD-DOM/introduction.html

[3] Frank Boumphrey, et al., XML Application, Wrox Press Inc, 1998

[4] Serge Abiteboyul, et al., Data on the Web, Morgan Kaufmann Publishers, 2000

[5] Jennifer Widom, et al., Lore, http://www-db.stanford.edu/lore/, 2000

[6] Oracle, XML in the Oracle9i Database, http://www.oracle.com/ip/deploy/database/9i/xmljava/index.html?xml.html, 2000

[7] eXcelon, http://www.exceloncorp.com, 2002

[8] R. Elmasri and S. Navathe, Fundamentals of Database Systems, Addison-Wesley, 2000

[9] 류진영, 김찬홍, 유영호, 김경석, "관계형 데이타베이스에서 XML 문서 저장 방안 연구", 한국정보과학회 '98 가을 학술발표 논문집(I) 제 25권 2호, 1998

[10] 서경식, "바다-II에서 SQL 숫자형의 설계 및 구현", 경북대학교 석사학위 논문, 1998