韓國 컴퓨터情報學會 論文誌
第9卷 第4號, 2004. 12.

2004-9-4-1-2

XML과 관계형 데이타베이스 매핑을 통한 자료의 변환

김 길 준*

Data Transformation through Mapping between XML and Relation Database

Gil-Choon Kim

요 약

XML과 데이터베이스간의 자료 변환의 원리는 XML과 데이터베이스 간 매칭의 원리로 이루어진다. SQL Server의 데이터에 접근하기 위한 방법은 URL에 SQL 쿼리를 지정하는 방법과 template 파일을 이용하는 방법이 있다. MS-SQL 서버는 OpenXML 기능을 이용하여 기존의 관계형 데이터베이스에 대해 SQL 쿼리를 실행한 결과를 XML 문서로 변환한다. 즉,OpenXML은 노드 트리를 생성한 후에 XML 문서의 로우셋 데이터를 반환하고, XML 데이터를 관계형 형식으로 얻게 한다. XML 데이터를 데이터베이스 데이터로 삽입하기 위해서는 sp_xml_preparedocument 프로시져를 사용하여, XML 문서를 파싱해 데이터를 추출한 후, 그 문서의 노드 구조를 메모리의 트리 구조로 매핑해서 데이터베이스 테이블에 저장하게 된다. 결국 XML과 데이터베이스간의 자료 변환의 원리는 XML과 데이터베이스간 매칭의 원리로 이루어진다. 본 논문에서는 매핑 원리를 제시한 후, SQL Server의 지원하에 두 자료간의 변환을 구현해 보임으로서 자료의 확장성과 효율성및 다양한 효과를 가져올 수 있음을 제시하고 있다.

Abstract

The data transformation between XML and Relation Database is made through the principle of mapping bewtween them. There are two ways to access SQL Server, one is to assign SQL query to URL and the other is to use template file. MS-SQL server takes advantage of OpenXML function to transform the results of executing SQL query into XML documents. That is, OpenXML first makes node tree and then transforms row set data of XML documents into XML data of relation type. In order to insert XML data into database data, data is extracted from parsing XML documents using sp_xml_preparedocument procedure, and then the document structure is mapped into tree structure and stored in a table of database. Consequently, Data transformation between XML and Relation Database is made through mapping bewtween them. This article proposes the principle of mapping between XML and Relation Database and then shows the implementation of transformation between them so that it introduces the possibilty of bringing the extension and efficiency of data and various effects.

▷ Keyword : Mapping, OpenXML, DTD

• 제1저자 : 김길준
• 접수일 : 2004.08.23, 심사완료일 : 2004.11.13
* 성결대학교 전자상거래 학부 교수

Ⅰ. 서 론

최근의 XML의 자료를 Database화로 처리하는 연구는 학문적으로 제시되고 있으나 두 자료간의 매핑 원리를 자세히 제시하고 있는 연구는 많지가 않다. 더구나 매핑의 경우의 수가 다양하여 일반적이 경우만 제시되고 있는 실정이다. 여기에서는 일반적인 매핑원리와 그에 따른 구현을 제시하고 있다. XML문서는 파일 단위로 관리하는 것보다는 데이터베이스를 사용해서 관리하는 것이 편리하다.XML문서는 데이터의 형태와 사용 목적에 따라 심문서와 문서 중심문서의 2가지 형태로 분류할 수 있다. 데이터 중심 문서는 기존의 전통적인 데이터베이스(관계형,객체지향)에 저장하는 것이 유리한 반면에 문서 중심 문서는 일반적으로 XML전용 데이터베이스(NDX:Native XML Database)에 저장하는 것이 유리하다.[9]

Ⅱ. XML과 데이터베이스

2.1 데이터 중심 문서

데이터 중심 문서는 주로 정형화된 데이터들을 포함하고 있는 문서이다. 즉 데이터 중심 문서는 조직적이고, 정규화된 구조를 가지며, 혼합된 콘텐츠가 거의 없다. 그래서 기계적인 처리가 가능하며, 미들웨이를 이용하여 자동적으로 관계형 데이터베이스에 저장할 수 있다. 특징을 알아보면

- 정형화된 데이터이고, 데이터들 간의 순서가 중요하지 않으며, 메타 데이터가 없고, 기계적으로 처리하기 쉽다.
- XML문서가 데이터베이스에 저장될 때 메타 정보들은 모두 제거되며, 문서의 DTD 정보는 저장되지 않은다. 또한 엔티티 참조와 일반 데이터는 데이터베이스에 저장될 때 구별이 없어지며, 처리 명령어는 불필요하기 때문에 데이터베이스에 저장되지 않은다. 또한 엘리먼트와 속성의 차이도 없어진다.

2.2 문서 중심 문서

문서 중심 문서는 주로 사람이 읽을 목적으로 만들어진 문서이다. 즉 문서 중심 문서는 데이터 중심 문서 보다 좀 더 자유스러운 형식의 텍스트를 자유롭게 표현된 문서이며, 보통 사람 중심적이고 ,혼합된 컨텐츠를 충분히 사용하므로 정규화 구조가 미흡하다. 또한 데이터 중심 문서에 비해 비정형적이지만, XML 전용 데이터베이스에 저장될 수 있다.

문서 중심 문서를 XML전용 데이터베이스를 사용하는 장.단점을 알아보면

비정규화된 데이터 저장이 가능하고, 빠른 검색속도 유지하며,XML 질의어 사용이 가능한 반면 질의어의 결과가 XML문서이므로 원하는 데이터를 얻으려면 다시 SAX나 DOM을 사용해야 한다.

Ⅲ. XML과 데이터베이스와의 매핑

XML 문서와 데이터베이스 사이의 데이터 이동은 문서와 데이터베이스의 매핑을 이용하는 소프트웨어에 의해서 이루어지며, 이러한 매핑은 2가지 방법으로 분류할 수 있다.

- 템플릿 드리븐 매핑(templete-driven mapping)
- 모델 드리븐 매핑(model-driven mapping)

템플릿 드리븐 매핑은 명령어의 처리결과를 그 결과의 임의의 장소에 삽입할 수 있으며, for문과 if문과 같은 프로그램 구조를 사용할 수 있다.

모델 드리븐 매핑에서 XML 문서의 데이터는 미리 정의된 모델에 따라서 XML 문서를 하나의 테이블이나 테이블의 집합으로 매핑된다. 모델 드리븐 매핑은 다시 2가지의 매핑 기법으로 분류된다.

- 테이블 기반 매핑(table-based mapping)
- 개체-관계 매핑(entity-relational mapping)

3.1 테이블 기반 매핑

테이블 기반 매핑은 대부분의 XML을 사용할 수 있는 관계형 데이터베이스에서 사용하며, XML문서를 객체들의 트리 형태로 보고, 이들 객체들을 데이터베이스에 매핑시키는 방법을 사용한다.객채 트리는 다시 여러개의 테이블 데

이터로 변환된다.

테이블 기반 매핑은 XML문서와 관계형 데이터베이스 사이의 데이터를 이동시킬 때 많이 사용되며, XML문서를 하나의 테이블이나 테이블들의 집합으로 매핑 한다.또한 사용하는 소프트웨어에 따라 자식 엘리먼트나 속성이 테이블의 칼럼을 형성한다.

3.2 개체-관계 매핑

개체-관계 매핑은 두 단계로 이루어진다. 첫째 단계는 XML DTD가 개체 DTD로 매핑되는 단계이며, 두 번째 단계는 개체 DTD가 데이터베이스 DTD로 매핑되는 단계이다. DTD를 개체로 매핑하는 것은 원소의 타입과 데이터 타입을 인식하는 단계에서부터 시작된다. 이때 DTD내의 PCDATA를 갖는 원소와 속성은 모두 단순 타입으로 간주되고, 단순 타입은 모두 스칼라 타입(기본 타입형,string)으로 매핑된다. 자식 원소를 갖거나 속성을 갖는 원소들은 모두 복합 타입으로 간주된다. 복합 타입인 경우에는 내부에 구조적인 값을 가지고 있기 때문에 클래스로 매핑되며, 복합 타입의 내부에 있는 자식 원소들과 속성은 모두 클래스의 멤버 필드로 매핑된다. 이때 자식 원소와 속성은 모두 멤버 필드로 매핑되기 때문에 차이점이 없다. DTD를 이용해서 개체로 변환하는 경우에 이러한 데이터 타입으로 매핑되는 것은 사람의 간섭이 있어야만 가능하다. 그러나 XML DTD를 사용하는 경우에는 XML DTD의 데이터 타입이 직접 개체의 데이터 타입으로 변환되는 것이 가능하다. 개체-관계 매핑에서 두 번째 단계에서 클래스는 테이블로 매핑되고, 스칼라 타입은 칼럼으로 매핑된다. 또한 포인터와 레퍼런스 관계는 기본 키와 외래 키의 관계로 매핑한다. 이 관계에서 부모와 자식 원소는 1:1 관계인 경우에 기본 키는 어느 테이블에 있든지 관계없다. 만약 관계가 1:N 인 경우에는 원소의 부모 자식 관계에 상관없이 1 부분이 항상 기본 키를 가지고 있어야 한다. 매핑 단계에서 기본 키 칼럼이 생성된다. 만약 기본 키 칼럼이 생성되었다면, 기본 키의 값은 데이터 변환 소프트웨어에 의해서 자동적으로 생성되게 된다.

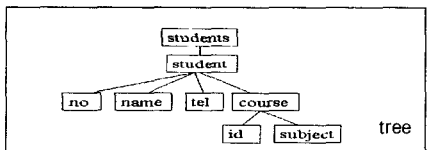
복잡한 XML문서는 여러 개의 테이블로 구성된 하나의 데이터베이스로 표현될 수 있지만 XML 문서의 물리적인 구조,문서 정보(DTD),주석,처리 명령어 등은 저장할 수 없다.테이블 기반 매핑은 미들웨어를 이용해서 XML문서와 데이터베이스 사이에 데이터를 이동시키는 경우에 많이 사용된다. 특히 웹 응용 프로그램에서 관계형 데이터베이스에 있는 정보를 XML 문서 형태로 변환할 때 많이 사용된다.[5]

Ⅳ. XML과 개체-관계 매핑

독립적으로 존재하는 개체-관계(entity-relationship)는 데이터베이스에서 개체와 개체간 또는 개체를 구성하는 속성간의 관계를 이용하여 필요한 정보를 검색하는 방법이다.

표 1. XML 문서와 테이블 간의 매핑

```
<students>
<student no="001">
 <name>hong gildong</name>
 <tel>777-1114</tel>
 <course>
 <subject id="111">OS</subject>
 <subject id="333">DS</subject>
</course>
</student>
<student no="002">
 <name>chun hyang</name>
 <tel>888-1114</tel>
 <course>
 <subject id="222">DB</subject>
</course>
</student>
</students>
```



id	sno	subject
111	001	OS
222	002	DB
333	001	DS

course

no	name	tel
001	hong gildong	777-1114
002	chun hyang	888-1114

student

개체는 독립적으로 존재하는 기본적인 대상이고, 관계는 개체들 사이에 존재하는 연관성을 말한다. 개체-관계 매핑 방법은 XML문서를 트리로 표현하고, 해당 개체를 데이터베이스로 매핑한다. 첫째 단계는 XML의 DTD를 개체 Class로 매핑하고, 두 번째 단계는 Class가 테이블로, 기본 형은 칼럼으로 매핑한다.

V. MS-SQL Server와 XML활용

SQL서버는 XML의 많은 기능중 일부를 지원하는 최초의 RDBMS으로서, SQL Server 2000에 있는OpenXML를 사용하면 XML 문서에 있는 데이터를 SQL 서버로 불러올 수도 있고, 윈도우 IIS 웹 서버를 사용하여 데이터베이스와 URL 질의어를 사용하여 불러올 수 있다.

그러면 다음과 같은 방식을 사용하여 SQL 서버의 XML 지원을 구현 해 본다.

① OpenXML을 사용한 XML 데이터 저장

② HTTP를 사용하여 SQL Server 데이터에 접근

5.1 OpenXML을 사용한 XML 데이터 저장

OpenXML()함수는 XML구조와 관계형 구조를 매칭시켜 처리하는 기능이 있어

XML 데이터를 데이터베이스에 삽입하거나 수정을 가능하게 한다. 또한 하나의 저장 프로시저를 호출하여 여러 개의 행 수정을 가능하게 한다. 또한 OpenXML을 사용하면 XML문서에 있는 데이터를 SQL서버로 불러올 수 있다. 여기에는 MS SQL서버에서 제공되는 2개의 시스템 프로시저가 사용된다.

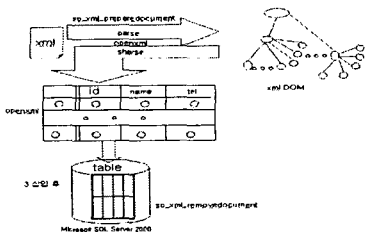


그림1. OpenXML과 2개의 시스템 프로시저

① XML를 데이터베이스의 자료로 변환

실제 테이블에 XML 형식의 데이터를 테이블에 삽입하는 경우를 알아보기 위해 먼저 XML자료를 받을 데이타베이스의 테이블을 생성한다.

```
CREATE TABLE TBLOPENXML
(title      nvarchar(100)
,author     nvarchar(10)
,price      int )
```

생성된 테이블에 XML 데이터를 입력하는 SQL문을 다음과 같이 작성한다.

▸ 프로그램(openxml2.sql)

```
DECLARE @hwd int, @openxml varchar(1000)
SET     @openxml = '<book>
<card>
<title>MSL2034</title>
<author>kim kil jun</author>
<price>25001</price>
</card>
<card>
<title>MSL2005</title>
<author>lee kil jun</author>
<price>25002</price>
</card>
<card>
<title>MSL2006</title>
<author>park kil jun</author>
<price>25003</price>
</card>
</book>'
        EXEC sp_xml_preparedocument @hwd output, @openxml
        INSERT  INTO    TBLOPENXML
                SELECT  *  FROM    OPENXML (@hwd, '/book/card', 2)
                        WITH (title nvarchar(50), author nvarchar(10), price int)
        EXEC sp_xml_removedocument @hwd
```

▸ 실행 결과

'TBLOPENXML' 테이블의 데이터('kimdb' 데...

	title	author	price
	MSL2034	kim kil ju	25001
	MSL2005	lee kil ju	25002
	MSL2006	park kil j	25003

단,select 다음 즉, with 다음의 변수가 데이터베이스의 테이블 변수와 동일해야 한다

② 데이터베이스를 XML 자료로 변환

XML형식에서 루트에 XML문서에서 사용할 namespace를 표시해야 하는데는, SQL Server의 형식을 사용하려면 다음과 같은 namespace가 사용되어야 한다.

'xmlns:sql="urn:schemas-microsoft-com:xml-sql"

그 다음에 다음과 같은 프로그램(template.xml)을 작성하여 '가상폴더/"template'폴더에 저장해야 한다.

▸ 프로그램(template.xml)

template.xml - 메모장

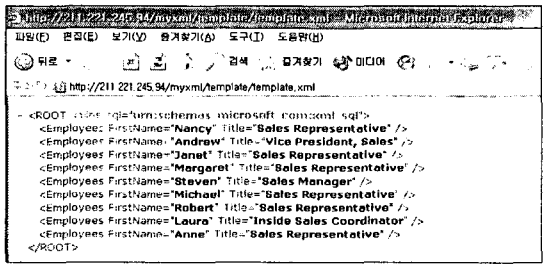
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
<ROOT xmlns:sql="urn:schemas-microsoft-com:xml-sql">
<sql:query>
SELECT FirstName,Title FROM Employees FOR XML AUTO
</sql:query>
</ROOT>
```

그리고 웹 브라우저에서 다음과 같은 형식으로 URL를 입력해서 XML자료를 확인한다.

"http://localhost(또는 작업중인 IP)/가상 디렉터리/파일명==)http://211.221.245.94/myxml/template/template.xml"

▸ 실행결과



5.2 HTP를 사용하여 SQL Server 데이터 접근

관계 데이터베이스를 XML자료로 변환하기 위해, IE를 실행시킨 다음 URL에 SQL문을 바로 입력하여 SQL Server의 데이터를 이용할 수 있다.

5.2.1 URL에 SQL 문 사용하기

① SQL 문 활용하기

SQL 서버의 HTTP 데이터 접근 형태는 IIS 서버가 URL에 지정된 가상 루트를 검사한 후,SQL Server용 OLE DB 공급자(SQLOLEDB)와 통신해 SQL서버가 연결된다.

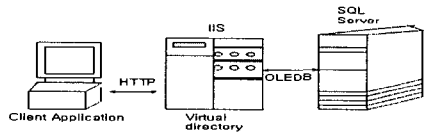
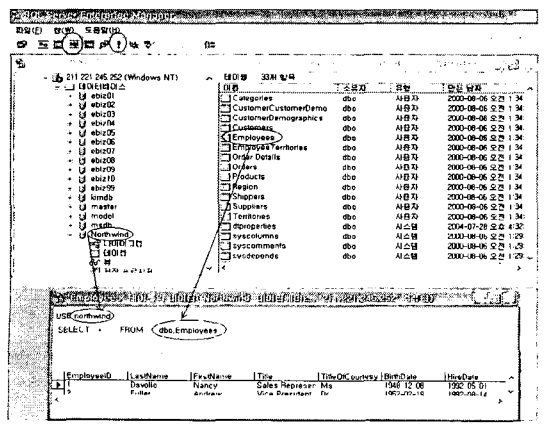


그림 2. SQL서버의 HTTP데이터 접근 구조

메뉴의 'SQL'아이콘을 클릭하여 SQL문을 입력한 후, 실행(!)아이콘을 누른다. 그러면 아래와 같이 실행 결과가 나온다.

▸ SQL 문 실행결과

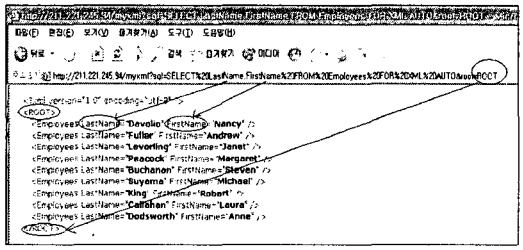


② FOR XML 사용하기

SQL 서버에서는 Select 문을 확장해서 ,데이터를 XML로 변환하는 기능을 지원하므로, 메모장에서 SQL문을 작성한 XML 프로그램을 template에 저장하여 처리하는 것과는 달리 바로 URL에 SQL문을 입력하여 처리하는 방식이다. 이러한 기능이 바로 FOR XML키워드를 확장한 T-SQL Select 문이다. 이때 사용되는 모드가 RAW,AUTO,EXPLICIT 모드이다. 그러므로 브라우저에서 쿼리를 실행하기 위해서는 FOR XML 키워드를 포함한 SQL문과 XML의 루트 요소를 생성하기 위해 '&root=루트요소 명'를 입력한 후 실행한다. 즉

"http://localhost/myxml?sql=SELECT LastName, FirstName FROM Employees FOR XML AUTO&root=ROOT"을 URL에 입력하여 실행한다. 그러면 테이블이 Element로 표현되고, 필드명은 속성으로 표현된다.

▸ XML root 생성을 위한 실행결과

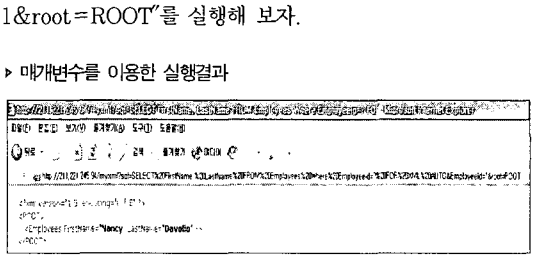


③ SQL 문에 매개 변수 전달하기

Employeeid(필드명)의 값이 SQL에 대한 입력으로 제공된다. 즉 Employeeid 필드 변수의 값이 1인 자료에 대해서 XML자료로 변환된다.

"http://localhost/myxml?sql=SELECT FirstName, LastName FROM Employees Where Employeeid=? FOR XML AUTO&Employeeid=1&root=ROOT"를 실행해 보자.

▸ 매개변수를 이용한 실행결과



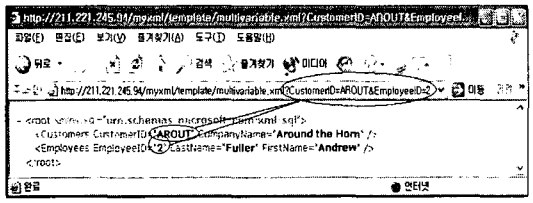
5.2.2 모든 매개변수에 인수 값 전달하기

CustomerID=ABOUT, EmployeeID=2처럼 변수에

값을 넣을 경우 URL에 입력 형태는 다음과 같다.

(http://211.221.245.94/myxml/template/multivariable.xml?CustomerID=AROUT&EmployeeID=2)

▸ 실행 결과



5.4 SQL문 Template에 저장하여 처리

URL에 SQL 쿼리를 쓰는 대신 메모장이나 저장 프로시저에 SQL 프로그램을 작성하여 template에 저장한 후, URL에서 template를 실행한 결과를 유효한 XML 문서로 만든다.

〔구문 형식〕

```
〈ROOT
xmlns:sql="urn:schemas-microsoft-com:xml-sql"
sql:xml="XSL파일명"〉
〈sql:header〉 //입력 변수 부분
〈sql:param〉..〈/sql:param〉
〈sql:param〉..〈/sql:param〉
〈sql:param〉..〈/sql:param〉...n
〈/sql:header〉
〈sql:squry〉 // template 쿼리 실행 부분
 SQL문장
〈sql:query/〉
〈sql:xpath-query mapping-schema="스키마 파일명"〉
//XPATH 쿼리 실행부분
XPath 쿼리
〈/sql:xpath-query〉
〈/ROOT〉
```

① template를 이용하기

아래의 SQL(프로시져)을 작성하여 template에 저장한다.

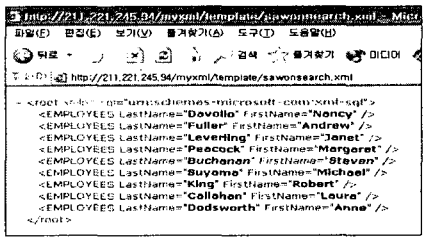
▸ 프로그램(SQL문)

```
sawonsearch.xml - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
<root xmlns:sql="urn:schemas-microsoft-com:xml-sql">
        <sql:query>
                SELECT LastName,FirstName
                FROM EMPLOYEES
                FOR XML AUTO
        </sql:query>
</root>
```

앞에서 〈root〉는 일반적인 루트와 같고, 우리가 URL뒤에 직접 길게 코딩할 때는 URL 의 SQL 문의 끝부분에 root=ROOT를 사용했는데, 그 부분이 바로 전의 프로그램(sawonsearch.xml) 부분으로 해결된다. 속성으로 사용한 xmlns:sql="urn:schemas-microsoft-com:xml-sql"은 이름 공간(namespace)을 지정하고 있어, 〈sql:query〉라는 엘리먼트를 사용할 수 있도록 했다.

그래서 〈sql:query〉 부분은 실제 SQL 구문이 들어가는 부분이다.

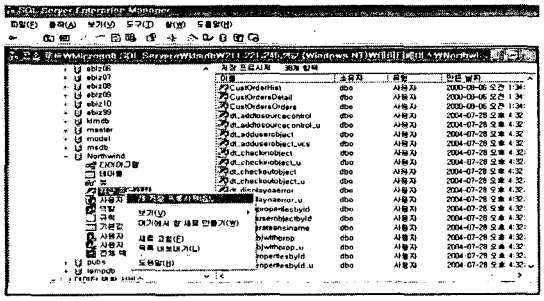
▸ 실행결과



② 저장프로시저 이용하기

저장프로시저에 SQL문을 작성하여 저장함으로서 메모장에서 SQL문을 작성하여 template에 저장하는 것을 대신한다. 데이터베이스 안에 'USING 저장프로시저'라는 프로시저를 만들어 놓았을 때, 이 프로시저도 역시 SQL쿼리 문이다. 이는 〈sql:query〉~〈sql:query〉안에 넣어두어야 한다.

㉠ 'SQL Server Enterprise Manager' 창에서 작업대상 데이터베이스의 폴더를 선택한다.



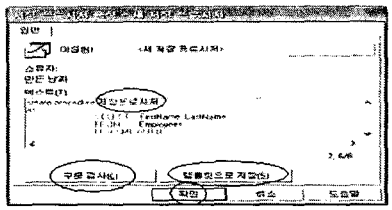
㉡ '저장프로시저'의 단축 메뉴에서 '새 저장 프로시저'를 선택한다.

㉢ '저장프로시저 속성'창에 procedure를 작성한다

㉣ 작성한 procedure의 구문을 검사하기 위해 '구문 검사'버튼을 누른다.

㉤ '템플릿으로 저장'버튼을 눌러 저장한다.

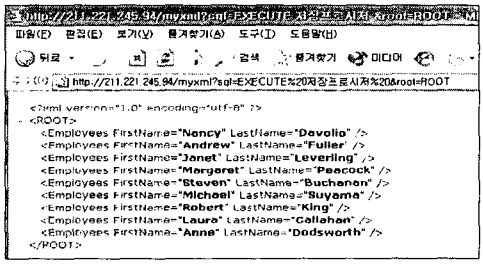
㉥ URL에서 다음과 같이 실행한다.



'http://localhost/가상폴더?sql=EXECUTE 프로시저명 &root=루트명'

저장프로시저를 실행할 때 쿼리분석기에서 실행하는 경우는 별 상관이 없지만, 외부에서 프로시저를 실행할 때 매개 변수가 있는 경는 실행되지 않을 수 있다. 이럴 때 저장프로시저를 실행하는 의미로 exec를 앞에 사용하면 이런 문제를 해결할 수 있다.

▸ 실행 결과

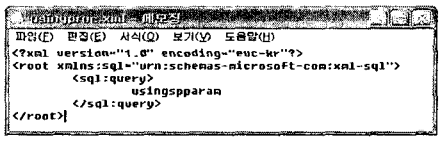


③ Call 문 이용하기

㉠ 변수를 사용하지 않은 call문 사용하기

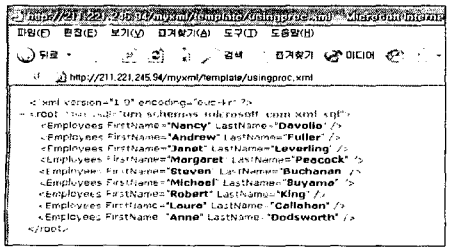
저장 프로시저에서 이미 만들어 놓은 프로시저를 호출하는 방식으로 호출하는 procedure를 '저장프로시저'의 단축메뉴 '새 저장프로시저'에서 작성한 후 URL에서 확인한다.

▸ 프로그램(usingspparam.xml)



```
<?xml version="1.0" encoding="euc-kr"?>
<root xmlns:sql="urn:schemas-microsoft-com:xml-sql">
        <sql:query>
                usingsppparan
        </sql:query>
</root>
```

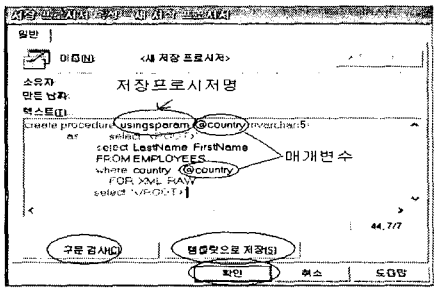
▸ 실행결과



㉡ template에 매개 변수 지정

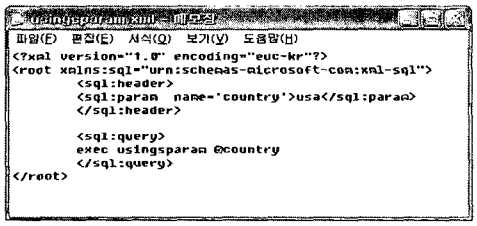
저장프로시저(usingspparam)에 매개변수를 사용한다.

▸ 매개변수를 사용한 저장프로시저(usingsparam)



▸ 프로그램(Usingsparam.xml)

저장프로시저 매개 변수를 사용한 프로그램이다.



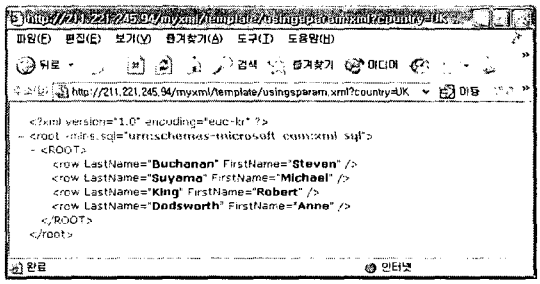
```
<?xml version="1.0" encoding="euc-kr"?>
<root xmlns:sql="urn:schemas-microsoft-com:xml-sql">
        <sql:header>
        <sql:param  name='country'>usa</sql:param>
        </sql:header>

        <sql:query>
        exec usingsparam @country
        </sql:query>
</root>
```

▸ URL로 본 결과

저장 프로시저에 'usingsparm'프로시저를 작성 및 저장하고, usingsparam.xml에서는 이 저장 프로시저를 실행하는 내용의 SQL문을 작성한 후, usingsparam.xml에서 사용된 변수(country)에 값을 할당하여 usinsparam.xml을 url에서 실행하면 된다.

▸ 실행결과



Ⅵ. 결 론

OpenXML은 노드 트리를 생성한 후에 XML문서의 로우셋 데이터를 반환하고, XML 데이터를 관계형 형식으로 얻게 한다. SQL 서버에서 데이터를 삽입하기 위해서는 sp_xml_preparedocument 프로시져를 이용하여, XML 문서를 파싱해 데이터를 추출한 후, 그 문서의 노드 구조를 메모리의 트리 구조에 매핑해서 데이터베이스 테이블에 저장하게 된다. 결국 XML과 데이터베이스간의 자료 변환의 원리는 XML과 데이터베이스간 매칭의 원리로 이루어진다. 두 자료간의 변환에 따라, 상용 데이터베이스에 XML을 저장하는 이유는 XML 자료는 데이터베이스로 자료를 보관/관리 함으로서 자료의 완전성을 보장하도록 하고, 자료 전송이나 교환시는 데이터베이스 자료를 XML로 생성하여 활용도록 하기 위해서다. XML 문서를 관계형 데이터베이스로 저장하기에는 테이블 수가 너무 많거나 null 값을 갖는 칼럼이 너무 많기 때문에 비효율적이므로, XML전용 데이터베이스에 저장함으로서 공간의 효율성을 높일 수 있다. XML전용 데이터베이스에는 XML문서 전체가 한 곳에 저장되기 때문에 여러 테이블에 저장되는 관계형 데이터베이스에 비해서 검색 속도가 훨씬 빨라진다. 또한 관계형 데이터베이스도 XML 질의어도 지원하지만 질의어를 사용하여 만들어진 XML자료는 원하는 데이터를 다시 얻기 위해선 DOM이나 SAX를 이용해야 된다. 전자상거래상의 필수 불가결한 XML 자료를 안전하게 관리하기 위해서는 이를 데이터베이스로 저장 및 관리해야 되며, 자료의 통신을 위해서는 데이터베이스의 자료를 XML로 변환하여 활용하는 경우가 많다 즉, 이기종에 데이터를 전송할 때 보안 벽에 의해 전송이 불가능하기 때문에 유일하게 시스템마다 열어 놓은 포트가 80번의 포트이다. 단지80포트를 통하여 전송할 수 있는 자료는 XML 자료 뿐이다.. 본 논문은 두 자료관의 변환을 위한 매핑 관계를 설명하였으며,SQL Server라는 미들웨어를 이용하여 XML과 데이터베이스 간 자료를 변환하는 구현의 결과를 보임으로서 자료의 확장성과 효율성및 다양한 효과를 가져올 수 있음을 제시하고 있다 또한 Office자료나 Project 자료등도 XML로 표현이 가능하므로, 도형등의 자료도 Database 화하여 자료를 관리 할 수 있음을 보여 주었다. 앞으로 XSL 스타일시트를 XML 데이터에 적용해서 HTML 문서로 변형한 후, 브라우저 기반의 클라이언트로 전송하거나 WML(Wireless Markup Language)과 같은 다른 문서 형태로 변환해서 WAP(Wireless Application Protocal) 형식으로 휴대폰, PDA 등에 사용할 수 있는 솔루션 개발도 가능하다

참고문헌

[1] XML,SGML모임, "표준 XML",대청미디어,2001
[2] 김영철외2, "XML DTD 기반의 구문지향 문서 작성기", 컴퓨터 교육학회 논문지,vol.7,NO.4, JULY 2004, PP67
[3] 구덕회외1, "멀티미디어 자료의 교육적 활용을 위한 메타데이터 요소 정의 및 XML DTD 설계", 컴퓨터교육학회논문지,vol.7,NO.4,JULY 2004,pp131
[4] 신민철, "XML 웹 서비스",FREELEC,2003
[5] 홍성용외1, "XML원리와 응용",한빛미디어,2003
[6] 이하영, "XML",가메출판사,2003
[7] 송정길, "XML 프로그래밍",생능출판사,2003
[8] 박재성, "XML실전 프로그래밍",가메출판사,2003
[9] 최종명외2, "XML + Java",홍릉과학 출판사,2003
[10] 신철민외2," XML,FREELEC",2003
[11] 서보원외1, "XML입문",도서출판 대림,2002
[12] 김길준,"Visio를 이용한 기자재 및 쇼핑몰에 대한 정보 전달",한국 컴퓨터 정보 학회,2004.9
[13] 김길준,"ICT를 통한 교수-학습 효율화 방안",한국 컴퓨터 정보 학회,2003.12

저 자 소 개

김 길 준
숭실대학교 전자계산학과 박사
현재 성결대학교 전자상거래 학부 교수