

# HTTP를 이용한 XML-QL기반 데이터 접근에 관한 연구

(A Study on Accessing SQL 2000 via HTTP based on XML-QL)

박 종 략\*  
(Jong-Rak Park )

**요약** XML과 데이터베이스를 이용한 웹 애플리케이션의 개발이 증가에 따라, 데이터 관리로 많이 사용되는 Microsoft SQL 2000 서버의 이용 또한 증가하게 되었다 따라서 본 논문은 데이터베이스의 원활한 관리와 취급을 위해 웹상에서 데이터 베이스를 직접 접근하고 다루는 방법에 대하여 나타내었다. XML을 이용하여 웹 상에서 직접 데이터 베이스에 접근하는 방법은 소개함으로, 데이터베이스를 이용한 전자 상거래 구축은 물론 많은 분야에서 XML과 SQL 서버 2000의 활용하여 전자상거래에 구축에 있어서 많은 도움을 줄 것으로 기대한다.

**Abstract** As Electronic Commerce is expanding its area rapidly and various web applications are developed by using SQL Query Language. This paper described a method of data accessing and transformation via HTTP. Finally we expect that this paper method will useful in constructing EC system using XML and SQL 2000 sever in related fields.

## 1. 서론

XML(Extensible Markup Language) 스펙 1.0 발표되면서 기존 HTML로 일관하던 인터넷 문서에 큰 변화를 주게 되었다. 사용자 정의 구조와 구조적 문서의 작성은 HTML로 할 수 없었던 많은 일들을 할 수 있게 되었다. 사용자의 의도에 따라서 기존 HTML을 작성과는 달리 목적에 따라서 태그를 작성함으로 데이터 관리가 매우 유용해 졌다. 웹에서 멀티미디어 저작물을 생성하고, 인터넷을 통해서 서비스를 제공할 수 있도록 HTML을 사용하던 개발자들이 XML의 유용성과 개발에 대한 편리성으로 인하여 XML을 사용하는 경우가 늘어나고 있다. 웹의 폭발적인 보급은 이러한 편리성에서 출발한다고 볼 수 있으며, 앞으로도 각종 저작물이나 문서 등의 정보의 일방적인 제공에서 매우 중요한 수단이 될 것으로 예측된다. 웹의 성장과 함께 데이터 베이스를 사용하는 서비스가 증가하고 수신자 측에서도 데이터 베이스 가공의 필요성이 증가하게 되었다[1].

HTML의 한계점을 극복하기 위한 XML은 HTML과

같은 마크업 언어를 정의하기 위하여 특정 목적에 맞게 사용할 수 있도록 지원한다. XML의 사용에 대한 장점을 간단히 살펴보면 데이터의 의미와 구조를 표현하기 위한 태그를 정의할 수 있어서 XML 문서를 수신한 브라우저는 데이터의 의미를 이해하고 어떻게 처리해야 할지를 알 수 있다. 일례로, 상품 카탈로그를 사용자에게 전달하기 위해서는 제품명과 가격을 `<Car id="1001" Country="한국"/>`과 같은 방법을 사용할 수 있다. XML은 데이터 구조에 대한 정보 또한 DTD(Document type Definition)를 사용하여 기술할 수 있다. 따라서 이 메타 정보를 이용한 문서내 데이터의 구조적인 완성도 또한 검사 할 수 있다. 또한 XSL(XML Style Sheet)로 XML 데이터 조작이 가능하며 XML 정보를 웹 브라우저로 검색하기 위한 HTML 문서로 변환하는 것이 가능하다. 제품정보 데이터는 대부분 애플리케이션으로 처리되는데 효율적인 저장소와 데이터 처리, 데이터 통신은 전산화된 시스템으로 구현이 되며 또한, 매우 많은 비용이 요구되지만 사실상 데이터를 다루는 가장 유용한 방법이기도 하다.

데이터 처리 방법 중 하나로 좀더 큰 대용량 저장소와 데이터의 검색에 대한 효과적인 처리가 가능해야만

\* 대구대학교 대학원 산업공학과 박사과정

한데 지금까지 데이터 저장소에 대한 가장 보편적인 접근은 SQL 서버 2000과 같은 관계형 데이터베이스를 사용하고 데이터베이스의 내의 데이터를 조작하기 위해서 SQL을 사용할 수 있다. 데이터 관리의 여러 기술들이 새롭게 사용되면서 데이터 분석가들은 기업의 각 부분에서 사용중인 서로 다른 시스템들을, 전체적으로 효율적으로 기능하고 있다면, 통합할 필요가 없다. 그러나 서로 다른 어플리케이션을 통합하려 할 때 많은 어려움에 직면하게 되는데, 한 개의 시스템에 정의 특별한 데이터 엔티티의 정의는 다른 시스템에서 사용된 데이터 엔티티와 서로 다르게 정의 될 수가 있다.

이러한 통합의 경우 해당 어플리케이션이 다른 기업에 속해 있는 경우 더더욱 문제가 커지게 된다. 이러한 문제를 XML의 네임스페이스를 이용하여 정의할 수 있으며 많은 데이터 생성 및 관리 또한 XML과 SQL 서버와 같은 쿼리 문을 이용하여 해결할 수 있다. 따라서 본 논문은 XML과 SQL2000 서버를 이용하여 효과적인 시스템 구축에 대하여 기술하였다.

## 2. 관련 연구

XML이 단순한 콘텐츠에서 데이터베이스로까지 그 적용 분야가 확장되면서 XML로 표현된 정보들을 어떻게 효율적으로 저장하고 관리할 것인가에 대한 기술들 XML과 함께 발전되어 가고 있다. 그러나 기존의 DBMS가 XML 데이터를 효율적으로 저장·관리 할 수 있는가 하는 것이 큰 관심사이다. 많은 연구들이 복잡한 XML 데이터 관계형 데이터 베이스에 저장 가능하도록 데이터베이스와 XML 문서의 연결에 대하여 발표되었다 [3, 4].

각 논문들의 중심 내용은 XML 파일을 관계형 데이터 베이스에 어떻게 저장할 것인가 그리고 주어진 XML 데이터나 DTD로부터 정보의 의미에 대하여 데이터 베이스 제약 사항들을 다른 방법에 대하여 기술하고 있으나, 객체 지향 데이터 베이스 시스템의 경우 효과적인 방법으로 대량의 데이터를 다루기에는 적합하지 않다[5].

하지만, 인터넷의 경우, 전자상거래(EC)를 위한 플랫폼으로서 발전하고 있으며 사용자들의 다양한 요구사항을 충족시키기 위한 S/HTTP·S/MIME·XML·자바·IPv6·SQL3·SQL4 등과 같은 새로운 프로토콜이 등장하고 있다. 특히 단순한 전자문서의 틀에서 벗어난 DB와 메타데이터 구현의 필요성이 제기되고 있다. 객체지향 개념의 SQL은 기술적으로 발전하고 안정적으로 정착되는 단계에 와 있으며 SQL 표준화 활동기관은 ISO/IEC JTC1/SC32(데이터 관리 및 교환분과)에서 메타데이터·DB언어·SQL 등을 표준화하고 있으며, NIST (National Institute of Standards)에서 SQL 구현의 테스트를 담당하고 있다. 국내 기구로는 기술표준원

SC32 전문위원회와 ISO/IEC TC1/서브커미티, ISO JTC1 SC32 워킹그룹 등이 활동 중이다. W3C의 XML 표준화 활동을 살펴보면 문법적 기술이나 하이퍼링크에 관련된 XLL, 멀티미디어 문서 동기화를 위한 SMIL 등의 표준화에 집중해왔다. 앞으로 XML-SQL을 이용한 사용 가능성은 매우 밝으며, 사용 가치면에서 폭넓게 사용 될 것으로 기대 된다.

## 3. 시스템의 구성과 기능

### 3.1. 구성

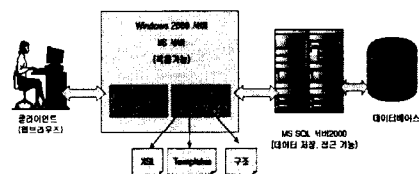
본 실험을 위해 사용된 시스템은 윈도우 2000서버와 IIS5.0 이상 그리고 SQL 서버 2000 을 사용하였다. 웹에서 SQL 서버 2000과 NT를 다양한 XML 파서 3.0을 이용하여 시스템 구성을 하였다.

### 3.2. 시스템 기능

웹 상에서 시스템을 구축에 대해서 살펴보면 서버와 클라이언트는 하나 또는 그 이상의 로직기능을 수행하게 되는데 다음 네 가지 기능의 기본 기능들을 시스템의 적용에 있어서 수행하게 된다.

- 데이터 저장기능 : 대부분의 적용시스템은 데이터의 저장이 필요하다. 데이터가 적든지 많든지 시스템에서 저장기능이 필요하다.
- 데이터 접근 기능: 이 기능은 데이터가 저장되어 있는 속에 접근하는 기능으로 SQL 쿼리 형태를 가지고 있으며 저장 프로시저나 쿼리 분석기의 명령라인에 의해서 수행이 된다.
- 적용 기능: 적용 자체로, 적용 부분은 간단할 수도 있으며 또는 복잡할 수도 있다. 적용에 관한 법칙들은 회사의 사업 방침에 따라 적용될 수도 있다.
- 표현 기능: 이 과정은 정보를 사용자에게 나타내거나 사용자의 입력 데이터를 받아들이는 부분이다.

이와 같은 네 가지 기능을 바탕으로 다음 <그림 1>과 같이 클라이언트/서버 구조가 형성된다.



<그림 1> 클라이언트/서버 구조

## 4. HTTP를 이용한 데이터 파일 접근

### 4.1. HTTP를 통한 디렉토리 접근

URL 형태의 쿼리가 IIS에 입력이 되면, URL상의 가상 루트를 검사하게 되고, SQLISAPI.DLL 파일을 통해서 이 가상 루트가 등록되어 있는지 확인하게 된다. IIS 이를 위해서는 IIS와 가상 디렉토리를 통해서 환경 설정이 가능하다.

SQLISAPI.DLL는 다른 DLL파일들과 연계하여 가상 루트상의 SQL 서버와 연결하고, 연결이 된 후에는 URL 상의 XML명령을 SQLXMLX.DLL 전송하게 된다 [6].

가상 디렉토리의 경우 SQLVDirContorl 객체를 이용하여 제어 가능하며 이 객체는 로컬 또는 원격 웹서버 연결하기 위한 Connect 메소드를 지원한다. <그림2>는 첫 번째 웹 사이트 연결한 후 가상디렉토리 "test"에 접근해 가는 프로그램이다.

```
dim objSQLIIS
dim objVDirCol
dim objVDir
set
objSQLIIS=createobj("SQLVDir.SQLVDirContorl")
objSQLIIS.Connnet "webserv", "1"
Set objVDirCol=objSQLIIS.SQLVDir
'가상디렉토리 접근 과정
set objVDir=objVDirCol.Item("test")
.
.
.
objSQLIIS.Disconnect
```

<그림 2> 가상 디렉토리 접근 방법

가상디렉토리의 경우 객체의 이름, 형식, 경로의 속성 또한 추가 상제가 가능한데 추가하는 방법은 objVDirNW.virtualNames.AddVirtualName "Imsi\_folder", 4,"c:\폴더명\msi" 형태로 지정할 수 있다. 이 경우 "Imsi\_folder"라는 가상 디렉토리가 "c:\폴더명\msi"에 생성되어 있으며 템플릿을 위해 만들어 졌다.

또한 "Imsi\_folder" 가상 디렉토리의 경로를 변경할 경우는 다음과 같이 나타내면 된다.

```
set
objVName=objVDirNW.virtualNames.Item"Imsi_folder"
objVName.path="d:\폴더명\msi"
```

위와 같이 지정하면 경로가 "c:\폴더명\msi"가 "d:\폴더명\msi" 로 변경된다.

### 4.2. URL에 SQL쿼리문 사용

```
http://IIServer/가상폴터?sql=SELECT+**+FROM+XML+Employees+For+XML+AUTO&root=root
```

위와 같이 SQL 쿼리 문을 URL에 사용하여 "가상폴터"라는 가상디렉토리에 "?"를 사용하여 바로 SQL 쿼리 문을 입력하였다. 모든 쿼리 문의 구분은 "+"를 이용하여 표현하였다.

### 4.3.URL에서 템플릿 사용

```
http://IIServer/가상폴터?template=<ROOT+xmlns:sql="urn:schemas-microsoft-com:xml-sql"><sql:query><SELECT+Employees+FOR+XML+AUTO/></sql:query></ROOT>
```

이것은 XML 문서 형태를 띄고 있으면 1개 또는 그 이상의 SQL 문을 포함하고 있다.

### 4.4.URL에서 템플릿 파일의 선언

URL에 모든 문장을 길게 쓰는 것보다 SQL 쿼리 템플릿 파일에 넣어 두고 아래 예와 같이 선언만 하여도 가능하다.

```
http://IIServer/가상폴터/템플릿가상공간/template.xml
```

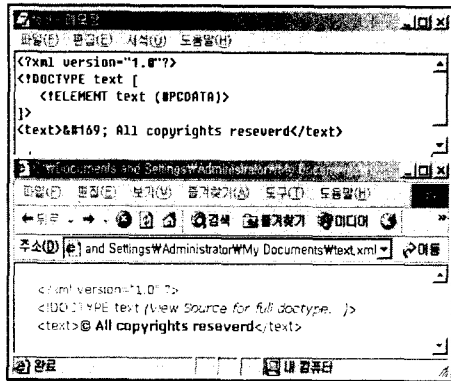
"템플릿가상공간"은 반드시 IIS의 가상디렉토리 유틸리티에 선언되어 있어야한다.

URL에서 스키마 파일에 XPath에 대한 쿼리도 나타낼 수 있으며 URL에 데이터베이스의 객체도 선언할 수 있다.

## 5.엔티티

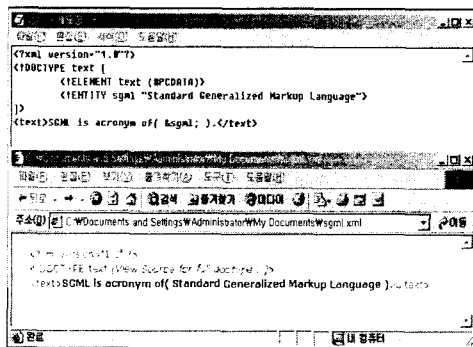
### 5.1. 엔티티 선언과 참조

XML 문서를 물리적인 구조로 구분하면 "엔티티"라고 하는 저장 단위로 구성이 되는데 이는 일종의 데이터 저장단위로서 '파싱 가능 데이터'이거나 '파싱 불가능한 데이터'로 구분이 가능하다. XML 문서는 논리적 구조에 관계없이 물리적 저장 구조가 서로 다르게 구성될 수 있다. 문자 엔티티의 경우 참조하는 형식은 '&#문자\_코드값;'의 형태로 선언할 수 있는데 "문자\_코드값"은 10진수와 16진수 모두 가능하다. 아래 <그림3> 엔티티의 선언과 실행하였을 때 화면을 나타낸다.[2]



<그림 3> 엔티티 실행 화면

다음은 엔티티의 선언과 참조에 관한 방법으로 선언하는 방법은 <ENTITY 엔티티 선언 "엔티티 값">의 형태로 나타낼 수 있으며 참조는 "&엔티티선언;"을 하게 되면 아래 그림과 같이 참조한 값이 나타나게 된다.



<그림 4> 엔티티의 선언과 참조

## 5.2. 템플릿 파일에서 엔티티

아래의 <그림5>에 나타난 데이터 구조에 대한 <그림 6>의 템플릿 쿼리문에 사용된 엔티티의 실행 결과가 <그림7>과 같이 나타난다.

OrderID	CustomerID	OrderDate	Freight
101	1	1996-07-04	7.45
102	1	1996-07-05	5.26
103	1	1996-07-06	5.93
104	1	1996-07-07	8.30
105	1	1996-07-08	8.49
106	1	1996-07-09	5.83
107	1	1996-07-10	8.39
108	1	1996-07-11	8.96
109	1	1996-07-12	3.83
110	1	1996-07-13	8.99
111	1	1996-07-14	3.83
112	1	1996-07-15	8.99
113	1	1996-07-16	8.99
114	1	1996-07-17	8.99
115	1	1996-07-18	8.99
116	1	1996-07-19	8.99
117	1	1996-07-20	8.99
118	1	1996-07-21	8.99
119	1	1996-07-22	8.99
120	1	1996-07-23	8.99
121	1	1996-07-24	8.99
122	1	1996-07-25	8.99
123	1	1996-07-26	8.99
124	1	1996-07-27	8.99
125	1	1996-07-28	8.99
126	1	1996-07-29	8.99
127	1	1996-07-30	8.99
128	1	1996-07-31	8.99
129	1	1996-08-01	8.99
130	1	1996-08-02	8.99
131	1	1996-08-03	8.99
132	1	1996-08-04	8.99
133	1	1996-08-05	8.99
134	1	1996-08-06	8.99
135	1	1996-08-07	8.99
136	1	1996-08-08	8.99
137	1	1996-08-09	8.99
138	1	1996-08-10	8.99
139	1	1996-08-11	8.99
140	1	1996-08-12	8.99
141	1	1996-08-13	8.99
142	1	1996-08-14	8.99
143	1	1996-08-15	8.99
144	1	1996-08-16	8.99
145	1	1996-08-17	8.99
146	1	1996-08-18	8.99
147	1	1996-08-19	8.99
148	1	1996-08-20	8.99
149	1	1996-08-21	8.99
150	1	1996-08-22	8.99
151	1	1996-08-23	8.99
152	1	1996-08-24	8.99
153	1	1996-08-25	8.99
154	1	1996-08-26	8.99
155	1	1996-08-27	8.99
156	1	1996-08-28	8.99
157	1	1996-08-29	8.99
158	1	1996-08-30	8.99
159	1	1996-08-31	8.99
160	1	1996-09-01	8.99
161	1	1996-09-02	8.99
162	1	1996-09-03	8.99
163	1	1996-09-04	8.99
164	1	1996-09-05	8.99
165	1	1996-09-06	8.99
166	1	1996-09-07	8.99
167	1	1996-09-08	8.99
168	1	1996-09-09	8.99
169	1	1996-09-10	8.99
170	1	1996-09-11	8.99
171	1	1996-09-12	8.99
172	1	1996-09-13	8.99
173	1	1996-09-14	8.99
174	1	1996-09-15	8.99
175	1	1996-09-16	8.99
176	1	1996-09-17	8.99
177	1	1996-09-18	8.99
178	1	1996-09-19	8.99
179	1	1996-09-20	8.99
180	1	1996-09-21	8.99
181	1	1996-09-22	8.99
182	1	1996-09-23	8.99
183	1	1996-09-24	8.99
184	1	1996-09-25	8.99
185	1	1996-09-26	8.99
186	1	1996-09-27	8.99
187	1	1996-09-28	8.99
188	1	1996-09-29	8.99
189	1	1996-09-30	8.99
190	1	1996-10-01	8.99
191	1	1996-10-02	8.99
192	1	1996-10-03	8.99
193	1	1996-10-04	8.99
194	1	1996-10-05	8.99
195	1	1996-10-06	8.99
196	1	1996-10-07	8.99
197	1	1996-10-08	8.99
198	1	1996-10-09	8.99
199	1	1996-10-10	8.99
200	1	1996-10-11	8.99

<그림 5>MS-SQL 데이터 구조

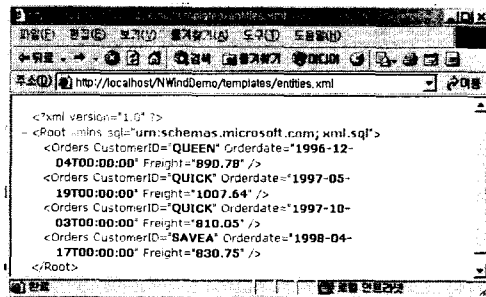
<그림6>은 <그림5> 와 같은 "orders"라는 데이터 파일에서 'CustomerID', 'OrderDate', 'Freight'에 대한 자료를 추출하여 표현을 하는데 'Freight >800'라는 조건을 만족해야만 한다.

```

<Root xmlns:sql="urn:schemas.microsoft.com:xml:sql">
  <sql:query>
    SELECT CustomerID, OrderDate, Freight
    FROM Orders
    WHERE Freight >800
  For XML Auto
  </sql:query>
</Root>

```

<그림 6> 주문에 대한 쿼리문



<그림 7> 쿼리문의 실행 결과

HTTP를 통한 SQL문의 실행도 가능하데 http://iisserver/가상루트/가상이름[/경로정보]/[XPath][?param=value[&param=value]...]과 같이 표현할 수있다.

<표 1> HTTP 구문에 대한 설명

키워드	설명
iisserver	IIS 서버에 대한 접근
가상루트	가상 디렉토리 관리자에 의해 설정된 가상 디렉토리로 문법적으로 혹은 GUI 환경에서 설정이 가능
가상이름	가상이름은 가상루트가 정의될 때 발생한다. 이것은 템플릿, 스키마, DB객체 중의 하나가 된다.
[/경로정보]	경로정보는 템플릿 파일이나 스키마 파일이 위치할 곳으로 경로에 대한 정보는 가상이름이 설정될 때 정해진다.
[/XPath]	스키마 파일이나 DB객체 형태의 파일로 기술된다.
?sql	SQL 쿼리문의 범위를 나타낸다.
sqlString	SQL 쿼리나 저장된 프로시저 이름을 나타낸다.
?template	XML 문서로 SQL 쿼리 문자의 한계를 나타낸다.
param	파라미터 이름이거나 Content type 또는 output encoding을 나타낸다.
root	잘 형성된 XML 문서의
xsl	반환되는 데이터를 표현하기 위한 XSLT 스타일시트 파일이 있는 위치를 나타낸다.

## 6. 결론

XML은 웹에서 구조화된 정보나 반 구조화된 정보를 교환하기 위한 표준마크업 언어로 자리 잡아가고 있다 [7]. 또한 웹에서 정보교환과 저장을 위해 수 많은 데이터 베이스에 대한 제어, 삭제, 추가, 갱신 등과 같은 기능들은 없어서는 안될 요소이며 XML과 데이터 베이스의 원활한 연계도한 필수적이라고 할 수 있다. 따라서 XML과 데이터 베이스관리 많이 사용되는 SQL서버를 이용하여 본 논문에서는 웹을 통한 접근 방법들을 소개하였다. 실험의 결과 HTML과 다른 DB 못지 않게 XML과 MS-SQL의 연동이 매우 다양함을 알 수 있었으며 특히 웹을 통해 XML-QL문을 이용하여 직접 데이터 베이스를 접근 제어 할 수 있음을 실험을 통하여 보였다.

## 7.추후 연구 과제

본 실험은 프로그램의 수행의 관점에서 중점적으로 다루었는데 앞으로 실효성과 기능의 효율성을 분석을 통해 연구해 문제점들에 대한 새로운 대안의 제시가 필요할 것 같다. 또한 앞으로 추후 수행 과제로는 XML - 쿼리 문을 이용한 데이터의 삭제, 갱신, 추가 등에 관한

효율적인 방안에 대한 분석도 더 필요할 것 같다.

## 참고 문헌

- [1] 김형도, B2B 전자상거래 @XML, 배움터, 2000년, P40
- [2] 조은영, 초보자를 위한 XML, 가남사, 2001.
- [3] James Rumbaugh, Objected-oriented modeling and design, Englewood Cliffs, N.J, Prentice Hall, 1991.
- [4] Michel Goosens, Janne Saarela, A Practical Introduction to SGML, volume 16(3), in TUGboat, 1995.
- [5] Joo Kyung-SOO, "A design of Middleware Components for the connection between XML and RDB", 2001 IEEE International symposium on Industrial Electronics Processings, Pusan, Korea, June 2001.
- [6] John, Griffin, XML and SQL Sever 2000, 2002.
- [7] <http://www.dpc.or.kr/sqlxml/>



박 종 략(Park Jong-Rak)  
 1996.2 대구대학교 산업공학과  
 1998.2 대구대학교 대학원 공학석사  
 2000. 8 대구대학교 박사과정 수료  
 주요 관심분야 : 전자상거래, XML,  
 Machine vision  
 pjr0127@chol.com