

SQL SERVER 2000'den VERİLERİ XML BELGELERİNE DÖNÜŞTÜREREK E-TİCARET VE EDI (ELECTRONIC DATA INTERCHANGE) UYGULAMALARINDA KULLANMAK

M. Nusret SARISAKAL

*İstanbul Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü
34850, Avcılar, İstanbul*

e-posta: nsarisakal@istanbul.edu.tr

Anahtar Kelimeler: E-ticaret, EDI, XML, SQL Server 2000

ÖZET

Bu çalışmada Elektronik Ticaret için önemli bir yeri olan EDI uygulamaları için kullanılacak olan XML belgelerinin SQL Server 2000 veri tabanından elde edilmesi incelenmiş ve bu yöntem için örnek bir model geliştirilmiştir.

1. GİRİŞ

Günümüzde, teknolojinin etkilediği ticaretin yeni ismi olan Elektronik Ticaret, ürünlerin genelde bir ağ üzerinden elektronik olarak alım, satım, sipariş ve bazen de ulaştırılması olarak tarif edilebilir[1].

Elektronik ticaretin temel araçları olarak telefon, fax, televizyon, elektronik ödeme ve para transfer sistemleri, elektronik veri değişimi (Electronic Data Interchange - EDI) ve Internet olarak altı ana araç sayılabilir[1].

Klasik elektronik ticaret araçlarından, Telefon esnek ve interaktiftir. Faks ise interaktif olmasına rağmen gönderilen dokümanın görüntü kalitesi iyi değildir. Televizyon çok yaygın olmasına rağmen tek yönlü bir iletişim aracıdır. Ticaretin önemli destekleri olan elektronik ödeme ve fon transfer sistemleri (ATM, kredi kartları, borç kartları ve akıllı kartlar) sadece para aktarılmasında kullanıldığından ticaret sürecinde sınırlı bir bölüme hitab etmektedir[1].

Genel de elektronik ticaret, internet ve diğer ağlar üzerinden yapılan ticaret olarak anlaşılmaktadır. Elektronik ticaretin çok yeni bir kavram olmamasına karşın, ticari işlemler de bir veya daha fazla insan tarafından ses, görüntü ve yazılı metinlerin aynı anda interaktif bir biçimde iletilmesi, zaman ve mekan sınırının olmayışı ve nispeten daha düşük maliyetlerle çalışılabilmesi şeklinde internet ortamının sunduğu olanaklar, elektronik ticaret kavramını hızla gündeme getirmiştir. Bu olanaklar internetin; diğer elektronik ticaret araçlarına göre daha esnek olmasını sağlar.

Internet ortamı iletişim ve ticaretin önündeki engelleri azaltmaktadır[1].

Örneğin, internetten bir shareware tipi programın indirilip beğenildiğinde bunun ücretinin ödenmesi bir elektronik ticarettir. Bir ürünü, örneğin bir kitabı, elektronik ortam kitapçılarının raflarında bulmak, bunun hakkında bilgi almak ve siparişini vermekte bir elektronik ticarettir.

Daha detaylı olarak Elektronik Ticaret, elektronik ortamda açık ve kapalı ağlar üzerinden yapılan; mal (taşınır, taşınmaz) ve hizmet (bilgi servisleri, danışmanlık, finans, hukuk, sağlık, eğitim, ulaştırma vb.) ticareti, sayısal biçime çevrilmiş yazılı metin, ses, video görüntülerinin işlenmesi ve iletilmesi, ürün tasarımı, üretim, doğrudan tüketiciye pazarlama, üretim izleme, sevkiyat izleme, tanıtım, reklam ve bilgilendirme, sipariş verme, sözleşme yapma, banka işlemleri ve fon transferi, ortak tasarım geliştirme ve mühendislik, kamu alımları, elektronik para (sanal para) çıkarma, elektronik hisse alışverişi ve borsa, açık arttırma, sayısal imza, e-noterlik, güvenilir üçüncü taraf işlemleri, vergilendirme ve vergi toplama, fikri mülkiyet haklarının transferi, kiralanması vb. işlemler olarak belirtilebilir[1].

E-ticaret'te veri değişiminin yani EDI'nin öneminin ne kadar fazla olduğu artık herkes tarafından anlaşılmıştır. Çünkü firmalar ticaretini yaptıkları ürünler ile veya verdikleri hizmetler ile ilgili bilgileri, aralarında ticari bağlar olan başka firmalarla paylaşmak zorundadırlar. Böylece hem zamandan hem de paradan tasarruf edilmiş olunacaktır. Bu nedenle e-ticaret yapan ya da EDI kullanarak veri paylaşımında bulunan ticari kuruluşlar, hızlı ve kolay bir sistem üzerinden veri alışverişi yapmak istemektedirler. Kuruluşların bu ihtiyacını fazlasıyla karşılayacak

yöntemlerden birisi de XML kullanarak veri alışverişinde bulunmaktadır.

Microsoft SQL Server'ın 2000 sürümünde internet üzerinden XML tabanlı veri değişiminde çok önemli bir XML desteği sunmaktadır. SQL Server ile XML bağlantısı oluşturularak veri tabanındaki bilgiler XML belgeleri şeklinde alınıp işlenebilmektedir[2].

SQL Server'daki veritabanına XML ile ulaşmak için bir virtual web dizini tanımlanmakta ve bu web dizini içinde şemalar (Schema) ve şablonlar (template) tanımlanmaktadır. Bu şemalar ya da şablonlar sorgulanarak veritabanına ulaşabilmektedir[3]

SQL Server'ın bu imkanı sağlaması bu alanda kullanıcılara oldukça büyük bir kolaylık sağlamaktadır. Çünkü bu olanak sayesinde SQL Server'daki verilere internet üzerinden bir URL

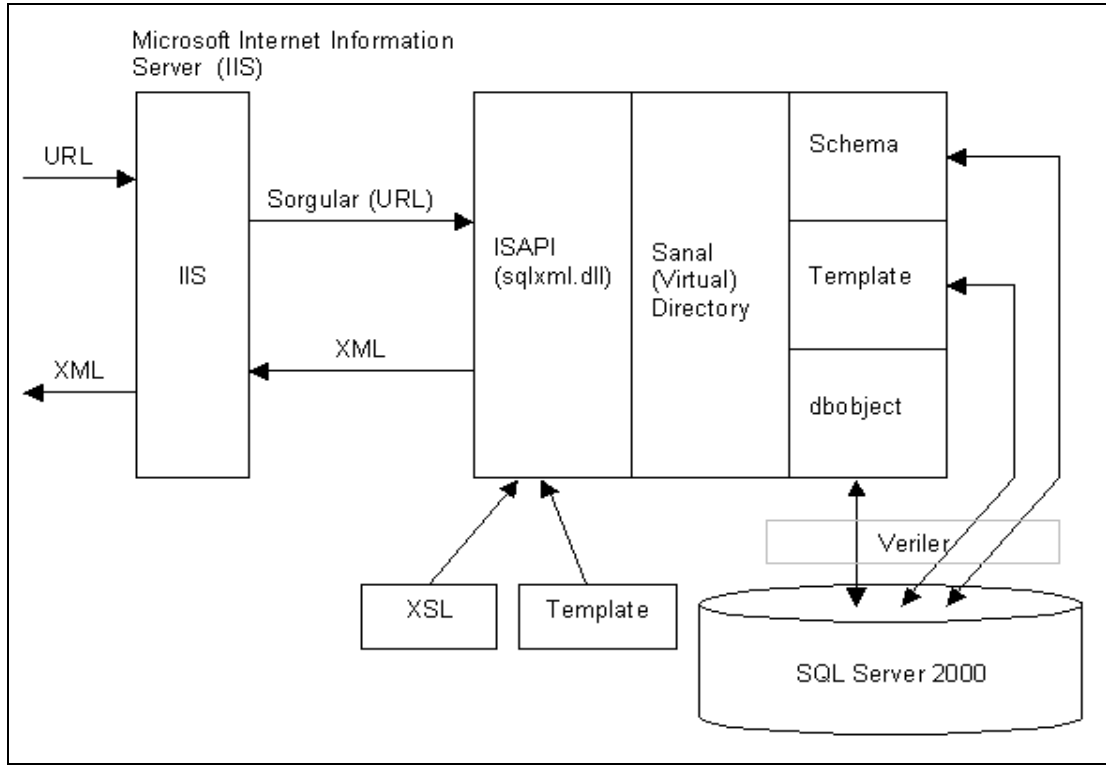
yazarak ulaşılabilir ve bu bilgiler alarak istediği şekilde kullanılabilir. Böylece diğer sistemler tarafından güncellenen SQL Server'daki bilgilere de istenildiği zaman erişilmiş olur.

2. SQL Server'dan XML Belgeleri Elde Etmek

SQL Server'a erişmek için öncelikle sanal dizin (virtual directory) oluşturmak gerekmektedir. Bu sanal dizin içinde ise template, schema ve dboject'ler içeren virtual name'ler tanımlanmalıdır.

SQL Server'dan veri elde etmenin birkaç yolu vardır.

- SQL sorgusunun URL içine yazılması ile
- SQL sorgusunun Virtual Directory'deki bir template dosyası içine yazılması ile
- Virtual Directory içindeki Schema dosyalarının kullanılması ile



Şekil.1 SQL Server 2000 ile XML ilişkisi [4]

2.1. URL'ler ile SQL Server'dan XML Verisi Elde Etme

URL'ler kullanılarak SQL Server'a erişebilmek için, veritabanı URL içine yazılan bir SQL sorgusu ile sorgulanır ve sonuçlar bir XML dokümanı olarak geri döner. Daha somut bir şekilde açıklamak için küçük bir örnek vermek gerekirse, veri tabanındaki alışveriş tablosu kullanıldı.

```
http://localhost/cpxml?sql=SELECT+*+
FROM+alisveris+WHERE+alisveris_no<4+
FOR+XML+AUTO&root=root
```

Bu sorgunun sonucu olarak aşağıdaki XML Dokümanı elde edilecektir.

```
<?xml version="1.0" encoding="utf 8" ?>
<root>
<alisveris alisveris_no="1" firma_id="1" />
<alisveris alisveris_no="2" firma_id="1" />
<alisveris alisveris_no="3" firma_id="1" />
</root>
```

Örnek1. URL ile SQL server'dan XML verisi elde etmek

Bu yöntemin kullanılması güvenlik açısından sakıncalıdır. Çünkü SQL sorgusu direk olarak URL içine yazılmıştır. Bu da veri tabanının yapısı hakkında bilgiler vermektedir. Bu nedenle veri tabanının güvenliği azalmaktadır [4].

2.2. Template'ler ile SQL Server'dan XML Verisi Elde Etme

Template'ler SQL Server ile XML bağlantısı oluşturmanın en elverişli yoludur. SQL sorgusu virtual directory içinde bulunan bir template dosyası içerisine yazılmaktadır. Bu template dosyası da XML yapısına sahiptir. Sorgu için gerekli parametreler template içinde tanımlanmaktadır. Örnekl'deki sorgu template ile aşağıdaki gibi gerçekleştirilir.

```
http://localhost/cpxml/templates/alisveris_temp.xml?alis_no=4
```

Bu URL'nin parametre yolladığı firmalartemp.xml dosyasının içeriği aşağıdaki gibidir.

```
<?xml version='1.0' ?>
<urun xmlns:sql="urn:schemas-microsoft-com:xml-sql">
<sql:header>
<sql:param name='alis_no' />
</sql:header>
<sql:query>
SELECT * FROM alisveris
WHERE alisveris_no<@alis_no
FOR XML AUTO
</sql:query>
</urun>
```

Örnek2. Template ile SQL Server'dan XML verisi elde etmek

Template'ler ile SQL server'a erişimin avantajları şöyle açıklanabilir, öncelikle birden fazla tablodan ve birden fazla sorgu yapılabilir. Bu nedenle oldukça esnek bir yapıya sahip olduğu söylenebilir. Sorgu URL içinde görülmediğinden veri tabanının güvenliği sağlanmış olur[4].

2.3. Schema'lar ile SQL Server'dan XML Verisi Elde Etme

SQL Server'dan XML verisi elde etmenin diğer bir yoluda XML Schemalar kullanılarak SQL Server'a

ulaşmaktır. Virtual Directory içindeki Schema dosyası SQL Server'da bir tablo ile ilişkilendirilir. Bu XML Schema URL içine yazılan bir XQL sorgusu ile sorgulanabilir.

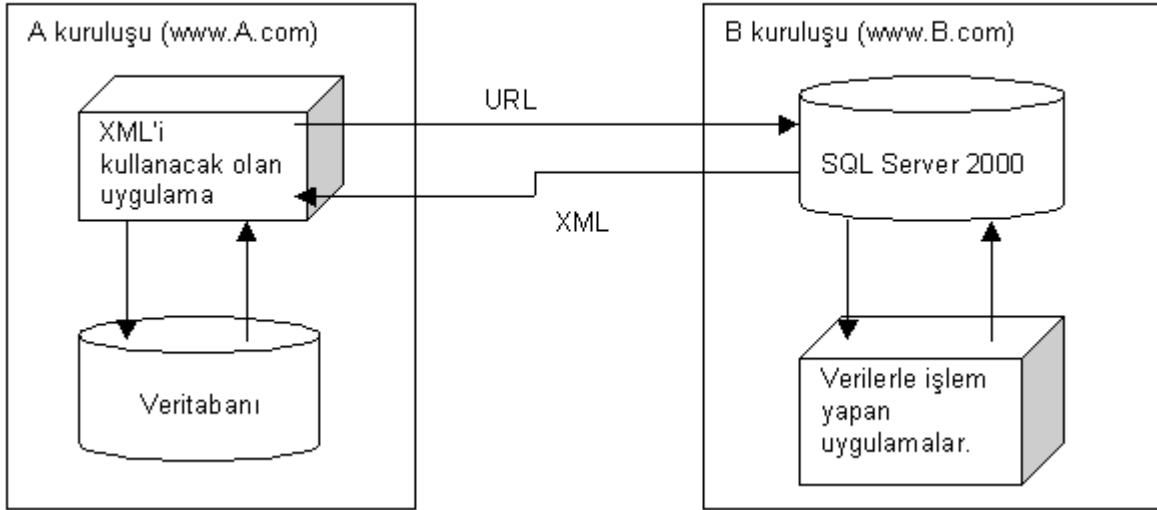
```
URL:http://localhost/cpxml/schemas/alisveris_schema.xml/alisveris[alisveris_no<4]?root=root
```

```
<?xml version="1.0" ?>
<Schema name="alisveris_schema.xml"
xmlns="urn:schemas-microsoft-com:xml-data" xmlns:dt="urn:schemas-microsoft-com:datatypes"
xmlns:sql="urn:schemas-microsoft-com:xml-sql">
<ElementType name="alisveris"
sql:relation="alisveris"
content="mixed" order="many">
<element type="alisveris_no" />
<element type="firma_id" />
</ElementType>
<ElementType name="alisveris_no"
content="textOnly" />
<ElementType name="firma_id"
content="textOnly" />
</Schema>
```

SQL Server'dan veri elde etmenin yanında, bilgileri güncelleme işlemi de yapılabilir. Bu işlem SQL Server'a gönderilecek olan bilgileri, bir template içindeki SQL cümlesine parametre olacak şekilde vererek yapılabilir.

3. YÖNTEM İÇİN ÖRNEK MODEL

Örneğin aralarında ticari bir ilişki olan iki kuruluş bulunsun A Firmasının, B Firmasına ait ürün bilgilerine ihtiyacı olduğu düşünölsün. Bu durumda A firmasının SQL server veri tabanında tutulan bilgilere ulaşmak için bir URL'ye ihtiyacı vardır. B firması tarafından, A firmasına hangi URL'yi kullanacağı bildirilir. A firmasında bu URL'yi kullanarak bir web tarayıcısı (browser) yardımıyla veya XML'i parse ederek işleyen bir uygulama tarafından, SQL server veri tabanındaki bilgileri bir XML dokümanı olarak alır ve bu bilgileri kendi sisteminde kullanabilir. Bu yöntemi anlatan yapı Şekil 2'de görölmektedir.



Şekil 2. SQL Server'dan XML verisinin elde edilmesine örnek model

Aslında bu yöntem Microsoft'un SQL Server 2000 için geliştirmiş olduğu bir ISAPI uygulamasıdır[2].

SQL server'ın XML bağlantısı için yukarıda ayrıntılı olarak açıkladığımız gibi bir **virtual web directory** oluşturulması gerekmektedir. Bu sanal web dizini **sqlsapi.dll** isimli activex'i kullanmaktadır. Yani web dizininin davranışlarını bu activex belirler.

Verilere erişim için sanal nesne isimleri (virtual names) tanımlanmalıdır. Bunlar, Schema, Template ve dbject'tir. Biz eğer SQL Server'a XML Schema'ları kullanarak erişmek istiyorsak, o zaman sanal web dizininin içinde bir sanal Schema ismi tanımlamamız gerekmektedir. Eğer template'ler kullanarak verilere ulaşmak istiyorsak sanal template isimleri tanımlamamız gerekmektedir. Sanal schema isimleri SQL Server'a ulaşırken XML Schema'ları kullanır. Schema'lar sorgu yapacağı SQL Server tablosunun ilişkili ve bu tablonun yapısını temsil eden bir schema yapısını içermektedir. Bu durumda Şekil.2'deki URL aşağıdaki şekilde olmalıdır.

```
http://www.B.com/virtualDirectory/
virtualschema/Schema1.xml/
ana_element?root=root
```

ana_element kök elemetten sonra gelen XML elementidir. Bu elementi schemayı yazan kişi belirleyebilir. Fakat şema içindeki diğer elementler sorgulanacak tablonun alan isimlerinden oluşmalıdır.

Schema'lar ile SQL Server'a erişmenin dezavantajı, aynı anda birden fazla SQL tablosundan veri çekilmesinin zorluğudur. Bunun yapılması için SQL'server da daha önceden birden fazla tabloyu birleştiren bir VIEW oluşturulmalıdır.

Template'ler ise içinde kullanılacak parametrelerin ve SQL cümlesinin bulunduğu bir XML dosyasıdır. Bu sorgu sonucunda tablo ismi element ve alan isimleri de elementin attribute'ları olacak şekilde bir XML dokümanı olarak oluşur. Template'ler kullanıldığında Şekil.2'deki URL aşağıdaki şekilde olmalıdır.

```
http://www.B.com/virtualDirectory/
virtualtemplate/
template1.xml?param1="..."&...
```

Template'lerle SQL server'a ulaşmak Schema'lara göre çok daha elverişlidir. Çünkü template'lerde SQL cümleleri ile sorgulama yapıldığı için, tabloları ilişkilendirerek birden çok tablodan veri elde edilebilir. Hatta aynı anda birden fazla sorgu yapılarak XML dokümanının içeriği zenginleştirilebilir. Aynı anda çok fazla sorgu kullanıldığı zaman da doğal olarak veri akışı yavaşlayacaktır. Daha doğru bir deyişle SQL Server'm bu sorguları çalıştırması uzun süreceği için verilerin karşı tarafa ulaşması biraz gecikecektir[3].

Bu yapı iki ticari kuruluşun veri alışverişinde kullanılacağı gibi, bir şirketin kendi içindeki birimleri arasındaki Elektronik Veri Alışverişinde (EDI) de kullanılabilir. Örneğin; bir firmanın depo birimi ile pazarlama birimleri birbirine uzak yerlerde olabilir. Dolayısıyla bu iki birimin aynı bilgisayar ağına dahil olmaları zor olacaktır. Bu da birimlerin veri alışverişini zorlaştıracaktır. Eğer burada yukarıdaki yöntemde olduğu gibi bir yapı kullanılırsa veri transferindeki zorlukların üstesinden gelinebilir.

Depo birimi kendi verilerini kendi bilgisayar sistemindeki SQL Server 2000'de ki bir veri tabanına kaydedecek, güncellemelerini burada yapacaktır. Bu sistemde yukarıdaki yapı kurularak, pazarlama birimine verilerin transferi sağlanabilecektir.

Pazarlama birimi tarafında ise depo biriminden gelen XML dokümanlarını parse ederek onları kendi veri tabanına kaydeden ya da yöneticilerin anlayabileceği bir şekle çevirecek veya raporlar hazırlayabilen bir uygulama kullanılabilir. Bu şekilde depo biriminin SQL Server 2000 sistemi sanki pazarlama birimine doğrudan bağlıymış gibi işlemler kolaylıkla yapılacaktır.

Buna benzer pekçok kullanım modeli oluşturulabilir.

4. SONUÇ

SQL Server'dan XML dokümanları elde ederek ile Elektronik Veri Değişimine (EDI –Electronic Data Interchange) yeni bir bakış açısı getirileceği açıktır. Böylece daha hızlı ve kolay bir veri alışverişi sağlanarak E-ticaretin gelişimine katkıda bulunulacaktır. Bunların yanında sistemin güvenliğinin sağlanması da önemli bir konuyu teşkil etmektedir.

EDI'nin E-ticaret'teki yerinin ne kadar önemli olduğu aşikardır. Bu anlamda veri alışverişinde güvenliğin çok önemli bir yeri bulunmaktadır. Çünkü veri değişimi yapan kuruluşlar, verilerin çalınmasını yada değiştirilmesini veya sistemin güvenliğinin tehlikeye

girmesini istemezler. Bahsedilen bu yöntemle etkin bir EDI sistemi gerçekleştirilebilir.

Örneğin böyle bir EDI sistemi oluşturup kullanılmaya başlansın, burada önemli bir soru ortaya çıkacaktır: “Bu sistemin güvenliğini nasıl sağlayacağız?”. Burada kullanılan virtual directory aynı zamanda bir web directory olduğu için öncelikle onun güvenliğinin sağlanması gerekmektedir. İçerdiği template ve Schema dosyalarının başkaları tarafından ele geçirilmesi sistem yapısının öğrenilmesi demektir. Bu virtual directory'lerin güvenliği SSL gibi güvenlik protokolleri yardımıyla sağlanmalıdır. Bundan sonra ise SQL sorgularının mümkün olduğunca template'ler içinde yapılması gerekmektedir.

KAYNAKLAR

1. SARISAKAL M. Nusret (2001) : “Veri Tabanlarının E-Ticaret Uygulamalarında Kullanılması – MNS Sigorta Şirketi Otomasyonu”, I.U. Journal of Electrical & Electronics, Vol. 1, No. 1, pp 41-48, 2001, ISSN 1303 – 0914.
2. Jake Sturm, Developing XML Solutions, Microsoft Press, ISBN:0-7356-0796-6.
3. Rob Vieira, SQL Server 2000 XML Integration, <http://www.wrox.com>
4. <http://msdn.microsoft.com/xml/>