

INTERNET TABANLI BİR SORU BANKASI UYGULAMASININ ASP, ADO ve ODBC TEKNOLOJİLERİYLE GELİŞTİRİLMESİ

Mustafa Türksever *
Murat Osman ÜNALIR *

ÖZET

Günümüzde, eğitim kurumlarında, öğrencilerin ve öğretim üyelerinin kullanımına açık bir soru bankası aranan bir uygulamadır. Çoğu zaman öğrenciler derslerle ilgili soru sıkıntısı yaşamakta ya da çeşitli kaynaklardan buldukları veya kendi yarattığı soruları başkalarına ulaştırmakta güçlük çekmektedirler. Herkesin ortak kullanabildiği, öğrencilerin de öğretim üyelerinin kontrolünde katkıda bulunduğu bir soru bankası genel başarıyı arttıracak faydalı bir uygulamadır. İnternetin, bilgi çağıının en çok paylaşılan iletişim ortamı olması nedeniyle soru bankası İnternet tabanlı olacak şekilde geliştirilmiştir. Bu nedenle, İnternet tabanlı bir uygulama geliştirme amacıyla kullanılan araç takımı ASP (Active Server Pages), ADO (Active Data Objects) ve ODBC (Open Database Connectivity) teknolojilerinden meydana gelmektedir.

SUMMARY

Implementation of a Internet based question-bank application by using ASP, ADO, and ODBC technologies.

An open question-bank application is needed by both students and lecturers in todays educational foundations. Students frequently have difficulty in accessing questions that are related with their courses. An Internet based common question-bank can increase the performance of the students. In this study such a question-bank has been developed by using ASP, ADO, and ODBC technologies.

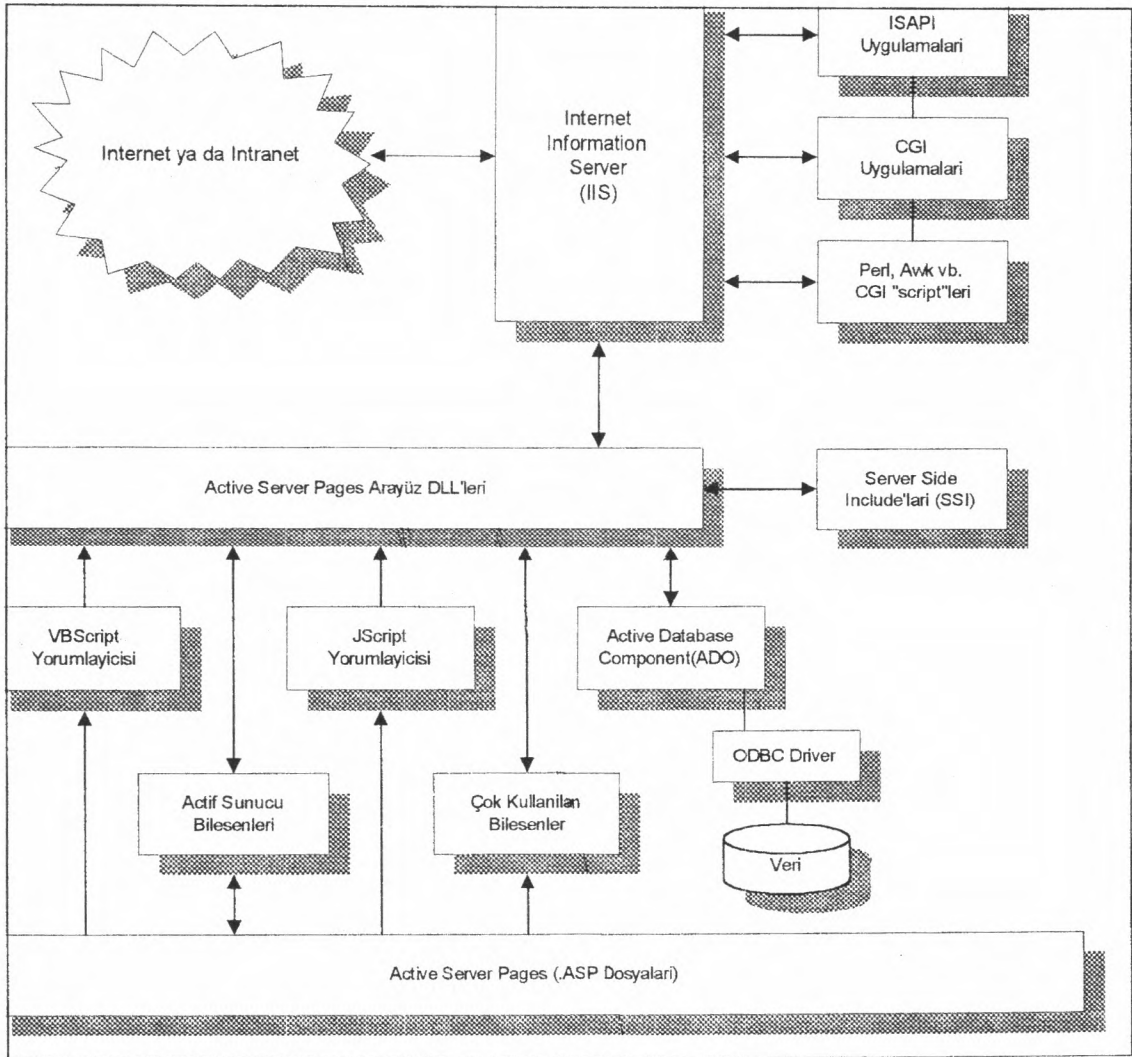
1. GİRİŞ

Günümüzde çok sayıda, değişik web siteleri ve intranet kurulumları bulunmaktadır. Bir şirketin ürünlerini tanıtan ya da servislerini pazarlayan siteler geliştirilebilir. Ya da tüm amacı, en son olaylar ve teknolojiler hakkındaki haberleri sağlamak olan bilişim tabanlı bir

*Ege Üniversitesi Bilgisayar Mühendisliği Bölümü - Bornova-İZMİR

site de olabilir. Hatta, dünyanın diğer taraflarındaki insanların, oturdukları yerden katılabildikleri konferanslar vermek için de kullanılabilir. Bir eğitim kuruluşu ya da bilimsel bir organizasyon tartışma belgelerini yayınlamak veya ticari siteler bazı araştırma ve anketler yapmak isteyebilir. ASP ile ziyaretçilerden bilgi toplamak da temel bir aktivitedir, bu yüzden değişik veritabanlarına erişebilmek gerekmektedir. Bir şirket intranetinde de bunların bazıları veya hepsi yapıyor olabilir.

Şekil-1'de, ASP etrafında geliştirilen bir sitede, değişik sistem parçalarının ve daha önceden bulunan web sunucu tabanlı uygulamaların nasıl birleştirilebileceği gösterilmektedir (Walther,1999). Bu şekil aynı zamanda, ASP'nin gerçekten de bütün bunların hepsini birarada tutan bir "yapıştırıcı" olduğu görüşünü de desteklemektedir.



Şekil-1: ASP temelli bir Internet uygulaması yapısı.

2. ASP (ACTIVE SERVER PAGES)

Active Server Pages (ASP), bir World Wide Web sitesi ya da bir şirket intraneti için dinamik HTML sayfaları yaratmak amacıyla geliştirilen Microsoft'un en son sunucu tabanlı teknolojisidir.

ASP, Microsoft'un, web sitesi tasarımcılarının ve servis sağlayıcılarının sayfalarını daha çekici, dinamik ve interaktif hale getirmelerine olanak sağlayan en son uygulamasıdır. Bu, tam olarak ziyaretçilerin ve reklam verenlerin ilgisini çekmek için ihtiyaç duyulan şeydir. Bir başka faydası ise, daha önceki CGI uygulamalarının, IDC (Internet Database Connector) "script"lerinin ve çok kullanılan bileşenlerin devre dışı bırakılmadan kullanılmaya devam edilebilmesidir.

ASP, daha önceden bir bilgisayar ağı olan ve bunun üzerinde bir intranet geliştirmek isteyen insanları hedef almıştır. Bunların, bir köşede IBM, bir diğerinde bir ICL ve birbirine bağlanacak 200 değişik tür ve markada PC'leri olabilir. Bunun anlamı, diğer sistem ve uygulamalara bağlı bulunan ayrı bir HTTP sunucusu anlamına gelir ve ASP bu konuda çok iddialıdır çünkü başka bir yolla yönetilmesi çok zor olan bütün şeyler ASP ile yönetilebilir.

ASP, insanların kolayca ve risksizce satın almasına izin veren güvenli "transaction"lar, sunucu tabanlı uygulamalar ve web siteleri yaratmak için bir yoldur. İnternet'te alışveriş merkezi kurmak isteyen iş sahiplerini hedeflemiştir. Gerçekte, İnternet üzerinde kimse o anda o kadar çok şey almaya bile, saniyede 10 milyon "transaction"a ihtiyaç duymaktadırlar.

ASP bir uygulama değildir. Microsoft'un dışında kalan bütün kullanılabilir, faydalı parçacıkları yapıştırmak için kullanılan hızlı ve etkili bir "yapıştırıcı" olarak görülebilir, ki bu da insanları Windows 2000 Server almaya ikna etmek için yeterli bir sebeptir.

Active Server Pages'in ve çoğu Microsoft teknolojisinin temeli İnternet Information Server (IIS) yazılımıdır (Mezick, 1999). Bu şimdi, tamamen Windows 2000 Server işletim sistemi içinde sunulduğundan, mümkün olan en etkin kombinasyondur. Bunun anlamı, diğer parçacıklar, ekstra özellikler sağlamak için istenildiği şekilde, etkinliği ve temiz arayüzü etkilemeden dahil edilebilir. Active Server Pages temelde, İnternet Information Server ile entegre olmuş ve diğer çok kullanılan bileşenlere bir arayüz bulunduran bir VBScript ve JScript yorumlayıcısıdır (McKinney, 1997).

3. DİNAMİK WEB SAYFALARI

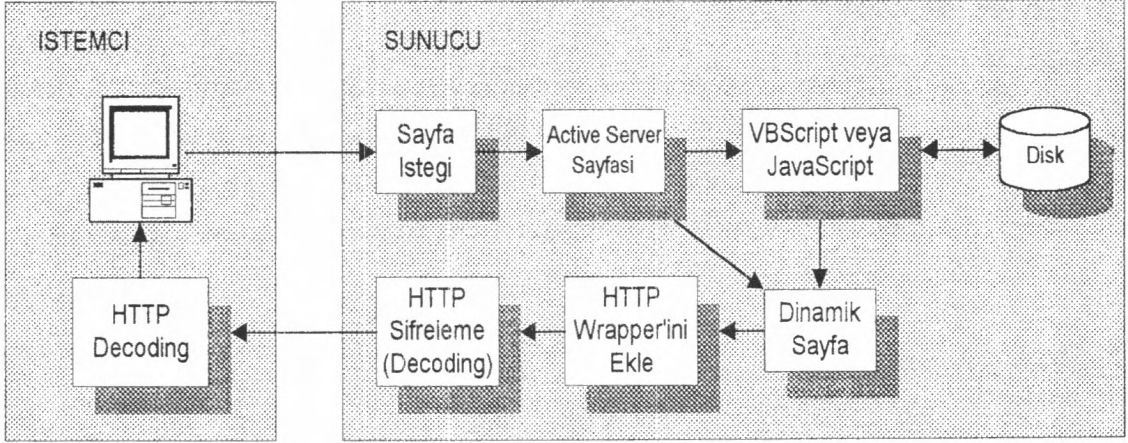
Kuşkusuz, Active Server Pages'in en çekici yönü, "browser" tarafından referans edilen dosyanın içerisine direk olarak "script"lerin eklenebilmesi ve böylece **dinamik sayfaların** yaratılabilmesidir. ASP'nin, statik sayfaların referans edilmesi, CGI ya da ISAPI uygulamalarının işletilmesi ya da geleneksel "script"lerin çalıştırılması gibi varolan metodlardan farkını anlamak önemlidir.

3.1 Active Server Pages'in Farkı

IDC ve döndürülecek sayfa için şablon kullanan diğer metodlar üretkenlikte orta dereceli bir artış sunarlar. Sayfa metninin ya da genel görünümünün değişmesi için, kodun değişmesinin ve her seferinde tekrar derlenmesinin gerektiği bir ISAPI uygulamasından farklı olarak, sadece şablon dosyasının değişmesi gerekir. Fakat, yine de iki dezavantajı bulunmaktadır: birincisi, "script" bir 'sonuç kümesi' yaratmalıdır; ancak o zaman şablon bir web sayfasına dönüşür. İkincisi, şablonun genelde sabit bir yapısı vardır ve 'sonuç kümesi'ndeki bilgilerin işleme biçimini kısıtlar. Genelde, sadece dönen değerlerin listelenmesi ve bu değerlere bağlı olarak sayfaya neyin dahil edileceği konusunda basit kararların verilmesi mümkündür.

Active Server Pages, ayrı bir "script" ve şablon kombinasyonuna olan ihtiyacı ortadan kaldırmaktadır (Kaufmann, 1999). "Browser" istediği sayfayı referans eder ve sayfa, normal statik bir sayfa gibi belleğe yüklenir. Fakat, "browser"a yollanmadan önce sunucu ele alınması gereken herhangi bir "script" olup olmadığını kontrol eder ve varsa işletir. Bu "script" hesaplamalar yapıp değerleri sayfaya ekleyebilir ya da gerektiği şekilde ekstra metin ve HTML kodu yaratabilir. Ve, öncelikle bir 'sonuç kümesi' yaratmak yerine, sayfanın üzerinde yüklendikten sonra işlem yaptığından, tamamlanmış sayfanın nasıl görüneceği üzerinde çok daha iyi kontrol sağlanabilir.

Sunucu yüklü sayfanın işletimini yaptıktan sonra, onu HTTP "wrapper"ının içine paketler ve "browser"a gönderir. Oraya vardığında ise, normal statik bir sayfa gibi sadece metin ve HTML kodudur. Öyleyse bu durumda, "browser" dönecek olan sayfayı referans etmekte, fakat kullanıcının gördüğü, içindeki "script"lerin işletilmesi ile oluşan sonuçları içeren bu dosyanın düzenlenmiş bir kopyası olmaktadır (Şekil-2).

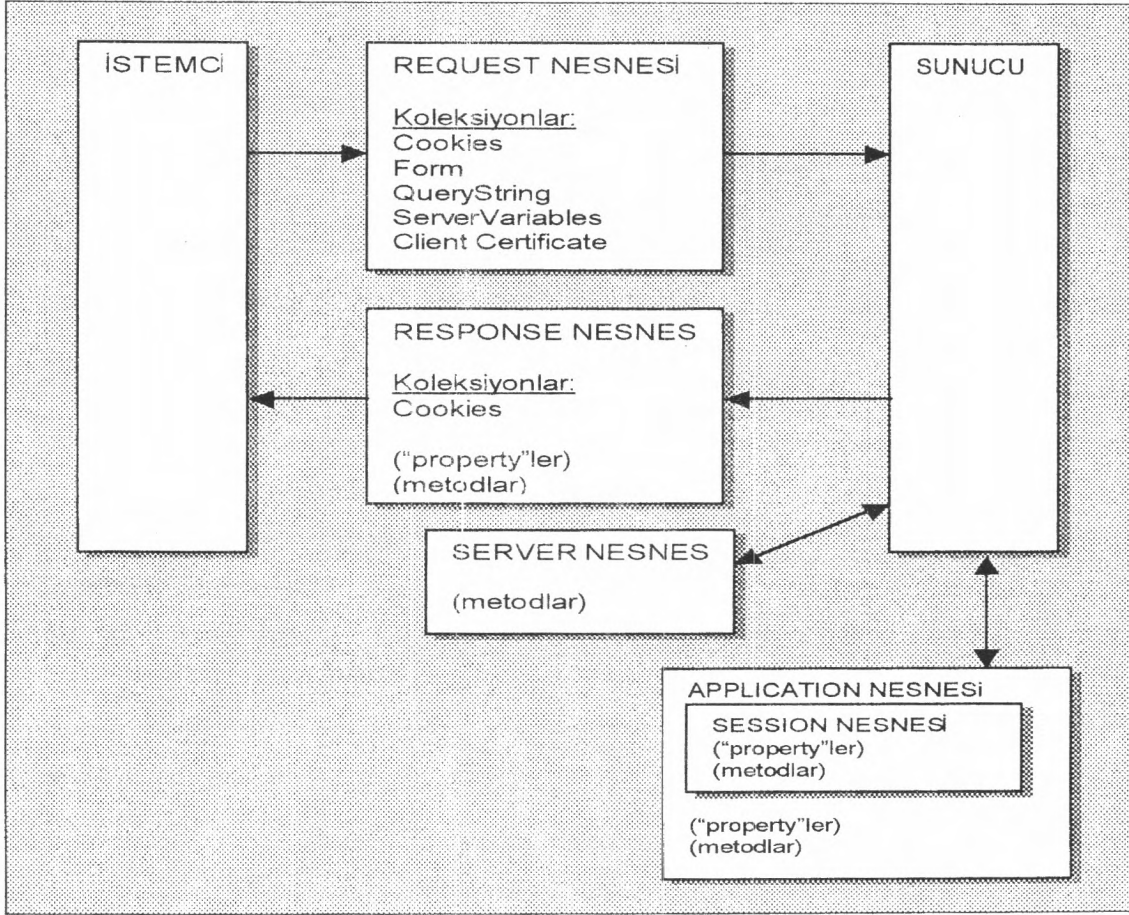


Şekil-2: ASP tabanlı dinamik bir sayfaya erişim.

3.2 ASP Nesne Modeli

Her ne kadar, bir ASP sayfasında “script”ler kullanılabilse de, scripting dillerinin kısıtlı özellikleri sebebiyle başarılı olacakları sınırlıdır. Fakat, web sitesinin interaktifliğinin artırılmasına yardımcı olmak gibi çok önemli bir özellikleri vardır. Active Server Pages ortamının parçası olan nesnelere üzerinde, metodlarını çağırmak ve “property”lerini kullanmak suretiyle işlem görürler. Active Server Pages, scripting dilleri kullanılarak işlenebilen bir nesne kümesi sunar.

Active Server Pages’in otomatik sağladığı beş built-in nesne, istemci “browser”ı ve sunucu arasındaki etkileşim için kullanılabilir. Bu nesnelere arasında bir hiyerarşi vardır. Kümenin en tepesinde **Server** nesnesi bulunur. **Server** nesnesi, sayfaların çalıştığı ortamı temsil eder ve diğer nesnelere biraraya gelerek Active Server uygulamasını oluşturur. Bu nesnelere; **Application**, **Session**, **Request** ve **Response**’dur. Şekil-3 bütün bu nesnelere göstermektedir.



Şekil-3: ASP nesne modeli.

Çoğu nesneyi kullanabilmek için, önce o nesnenin bir "instance"ı yaratılmalıdır. Fakat Active Server Pages'in beş nesnesi, bunu gerektirmez. Aşağıdaki tablo, bu *built-in* nesnelere ve yaptıkları işleri özetler.

Nesne	Yaptığı İş
Request nesnesi	Bir kullanıcıdan bilgi almak.
Response nesnesi	Bir kullanıcıya bilgi göndermek.
Server nesnesi	ASP işletim ortamını kontrol etmek.
Session nesnesi	Bir kullanıcı oturumu ile ilgili bilgiyi saklamak.
Application nesnesi	Bir uygulamanın kullanıcıları arasında bilgiyi paylaşmak.

4. ADO (ACTIVE DATA OBJECTS)

Bir web sitesi ve kuruluş intranetinde dinamik sayfaların kullanılmasının temel nedeni veritabanı erişim sağlamaktır. Bu, o anda mevcut ürünlerin ve en son fiyatlarının bir listesi gibi görüntülenecek değerlerin getirilmesi ya da kullanıcıdan bilgi toplanıp bunların bir veritabanına kaydedilmesi olabilir.

ASP ile sağlanan standart sunucu bileşenlerinden birisi tam anlamı ile bu işi yapmak için tasarlanmıştır. **Active Data Objects (ADO)**, ODBC (Open Database Connectivity) "driver"ı mevcut olan her veritabanına tam erişim sağlar. ODBC, bu tek bileşen ile değişik veritabanı sistemlerinin kullanılmasına izin verir ve böylece web sitesinin yaratılması ve sürdürülmesi süreçlerini kolaylaştırır.

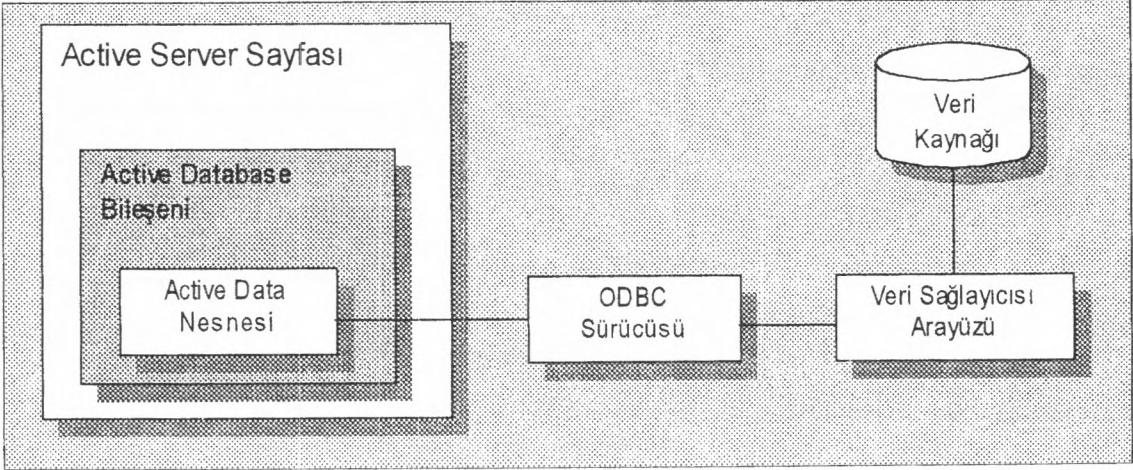
Active Data Objects (ADO), her tür veriye erişim sağlayan bir bağlantı mekanizmasıdır. En yaygın kullanımı, bir istemci uygulamanın, bir ilişkisel veritabanında tutulan veriye ulaşmasıdır. ASP'de bu, VBScript veya JScript gibi bir veritabanı ile etkileşebilen "scripting" dillerinde kod yazmaya izin verir. ASP ile daha önceden sunulmuş esneklik ile ADO, Internet üzerinden çalışan istemci-sunucu uygulamaları yaratılmasına izin verir ve geliştirilen uygulama, istemci "browser"ının türüne bağlı değildir.

4.1 ADO Veri Arayüzü

ADO en basit şekildeyle, **Open Database Connectivity (ODBC)** metodları üzerinden veritabanları ile etkileşmek için tasarlanmıştır. Bir ODBC sürücüsü (driver) mevcut olan herhangi bir veri kaynağı ile kullanılabilir. Bu, sadece SQL Server, Oracle, Access vb. gibi 'düzgün' veritabanı uygulamaları değil, aynı zamanda Microsoft Excel gibi işlem tablosu dosyaları veya diğer yalın formatlı veri dosyaları anlamına gelir.

4.2 Veri Sağlayıcısı

ADO başka bir tabakanın üzerine kurulduğundan, bir veri kaynağına olan bağda başka bir bağlantı tabakası belirtmek gerekebilir (Şekil-4). Başka bir deyişle, artık sadece sürücü yazılımını (ODBC gibi) düşünmek yeterli değildir, verinin gerçek **sağlayıcısının** da ne olduğu gözönüne alınmak zorundadır. ODBC, OLE-DB sağlayıcılarının sadece en popüler olanıdır (Wood, 1999).



Şekil-4: ADO mimarisi.

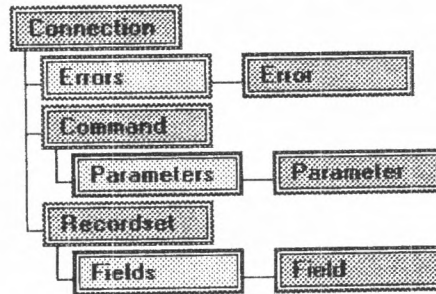
ADO'nun yapabildikleri, büyük ölçüde OLE-DB sağlayıcısının özelliklerine bağlıdır. Bazı nesnelere farklı çalışabilir, "property"ler eksik olabilir ya da belli nesnelere kullanılamaz veya hiç yoktur.

ADO'nun özellikleri, ODBC sağlayıcısını kullanırken de, ODBC sürücüsünün özelliklerine bağlı olarak çok değişiklik gösterir. Örneğin, çoğu ODBC sürücüsü her imleç türünü sağlamaz. SQL Server ve Access ODBC sağlayıcıları AbsolutePage "property"sini sağlarken, diğer ODBC veritabanı sürücülerinin çoğu sağlamaz.

4.3 ADO Nesne Modeli

ADO, Microsoft Access ve diğer nesne tabanlı uygulamalarda veritabanına erişim sağlamak için kullanılan mekanizmalardan daha basit ve anlaşılması kolaydır (Sussman, 1999). Örneğin hiyerarşide sadece üç ana nesne, **Connection**, **Recordset** ve **Command**, ve bunların altında birkaç koleksiyon, **Parameters**, **Properties** ve **Errors**, bulunur.

ADO'nun nesne modeli aşağıdaki şekilde görüldüğü gibidir.



Şekil-5: ADO nesne modeli.

Connection, **Command**, **Recordset**, ve **Field** nesnelерinin aynı zamanda bir de **Properties** koleksiyonu bulunmaktadır.

Connection nesnesi kullanılarak, herhangi bir yerde – genelde bir veritabanında – tutulan verilere erişmeye olanak tanıyan bir aktif bağlantı kurulabilir. Bu veri kaynağından kayıtları elde etmek, SQL sorguları işletmek ya da veriyi direk olarak işlemek için **Command** nesnesi kullanılır. **Recordset** nesnesi ise, bir SQL sorgusunun ya da saklı prosedürün işletimi veya bir tablonun açılması ile dönen verilere erişmeye izin verir.

Connection, **Command** ve **Recordset** nesnelерinin bir **Properties** koleksiyonu vardır. **Connection** nesnesinin aynı zamanda, nesne kullanılırken oluşan hataların detaylarıyla ilgilenmek için bulundurduğu bir **Error** nesneleri koleksiyonu bulunmaktadır. Son olarak, **Recordset** nesnesi bir **Fields** koleksiyonu içerir. Bu koleksiyondaki her bir eleman, tahmin edilebileceği gibi, 'recordset' içindeki her bir saha ile ilgili bilgileri tutan bir **Field** nesnesidir.

5. ODBC (OPEN DATABASE CONNECTIVITY)

ODBC (Open Database Connectivity), Microsoft tarafından geliştirilmiş veritabanı uygulamaları ile uygulama programları arasında iletişimi sağlayan standart bir API'dir (Applications Programming Interfaces). ODBC, uygulama programlarının çeşitli üçüncü kuşak yada heterojen veritabanlarına erişebilmelerini sağlar (Sanders, 1999). Program geliştiren kişiler ODBC uyumlu ya da ODBC istemcisi olarak adlandırılabilen programlar geliştirebilirler. Ayrıca spesifik veritabanı uygulama sistemleri için "ODBC sürücülerini" de yazılabilir.

ODBC istemci/sunucu bilgisayarlaşmada veritabanına erişme yöntemlerini basite indirgemeyi amaç edinir. Çoklu anahtar veritabanı sunucu satıcıları ve pek çok desktop üreticileri ODBC arayüzünü kullanmaya başlamışlardır. ODBC, desktop ürünlerinin merkezi depolanmış verilere son kullanıcıların erişmesini sağlar. Microsoft'un yazılım dünyasının öncü ve itici gücü olması nedeniyle ODBC, istemci/sunucu veritabanı iletişiminde etkin bir güce sahip olmaktadır.

6. SORU BANKASI UYGULAMASI

Uygulamada, deęişik haklara sahip üç ayrı kullanıcı grubu bulunmaktadır: öğrenci, hoca ve yönetici. Öğrenciler, belli derslerin belli konuları ile ilgili soru ve cevabını hazırlayıp o dersin hocasına gönderebilmekte ve yine bir dersin herhangi bir konusu ile ilgili olarak soru bankasındaki soruları görebilmektedir. Hoca ise, öğrencilerden verdiği derslerinden birisi ile ilgili gelen soruları görüp bunları soru bankasına kaydedebilir veya beęenmezse bu soruyu deęiştirebilir ya da silebilir. Hoca kendisi de herhangi bir dersin bir konusu ile ilgili olarak soru hazırlayabilir. Yönetici, veri tabanının yönetilmesinden yani; yeni hoca ve öğrencilerin eklenmesi, yeni derslerin eklenip bunların hocalara atanması, ya da bunların hepsinin silinmesi vb. gibi işlemlerin yapılmasından sorumludur. Kullanıcıların bütün yetkilerini aşağıdaki gibi listeleyebiliriz:

YÖNETİCİ

1. Yeni ders ekleme, ders silme,
2. Bir derse konu ekleme, konu silme,
3. Yeni hoca ekleme, hoca silme,
4. Bir derse bir hocayı atama, hocayı dersten silme,
5. Yeni öğrenci ekleme, öğrenci silme,
6. Bir derse öğrenci ekleme, dersten öğrenciyi çıkarma,
7. Soru bankasındaki soruları görebilme.

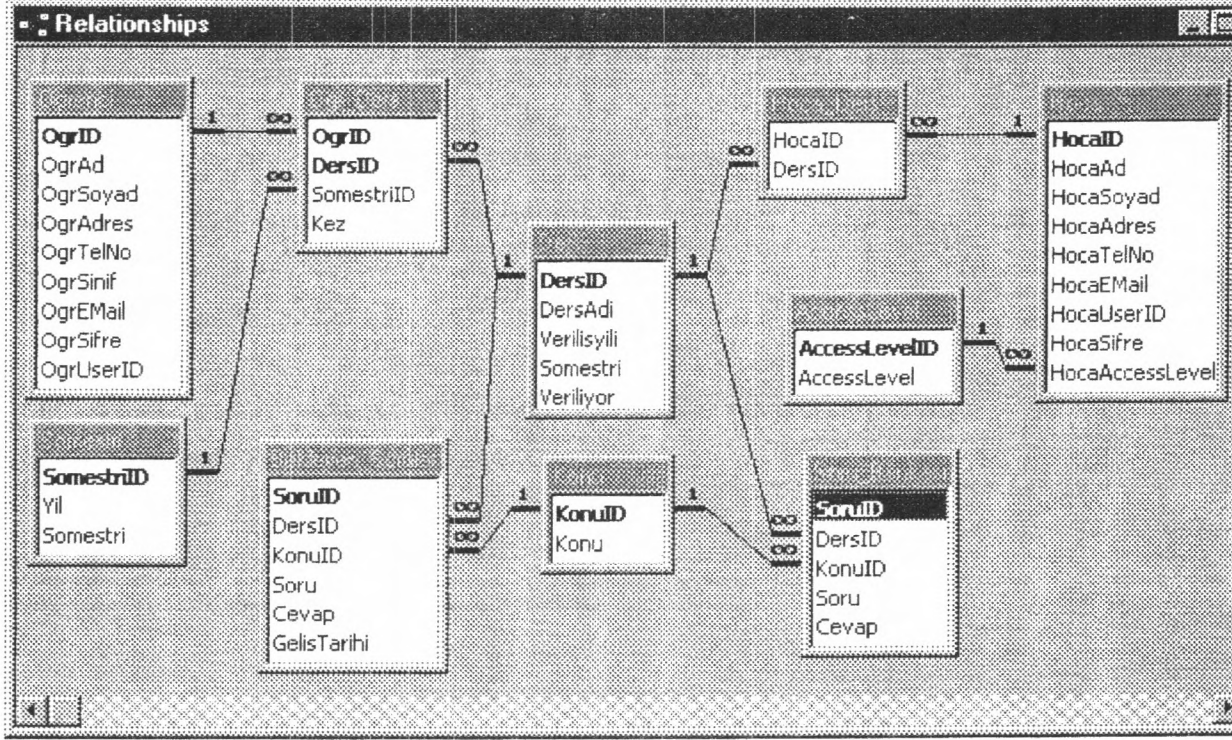
HOCA

1. Kendi verdiği bir dersle ilgili gelen soruları görebilme,
2. Bu sorulardan istediğini soru bankasına kaydedebilme, istediğini silebilme,
3. Kendisi de soru hazırlayıp soru bankasına kaydedebilme,
4. Soru bankasındaki soruları görebilme,
5. Soru bankasından kendi dersleri ile ilgili soruları silebilme,

ÖĞRENCİ

1. Belli bir dersin bir konusu ile ilgili soru yazıp ilgili hocaya gönderebilme,
2. Soru bankasındaki soruları görebilme.

Uygulamada bu dosyalardan başka, dosyaların ADO'yu kullanarak sık sık eriştikleri bir de veri tabanı mevcuttur. Şekil-6'da, kullanılan veritabanındaki tablolar ve bu tablolar arasındaki ilişkiler belirtilmektedir.



Şekil-6: Veritabanındaki tablolar ve ilişkiler.

7. SONUÇ

İnternet tabanlı bilgi sistemleri, bilginin paylaşılması ve bilginin çoğalması açısından bilgi çağına geçişte oldukça önemli rol oynamaktadır. Bu tür bilgi sistemlerinin geliştirilmesi o sistemi geliştiren ve o sistemi kullananlar arasında doğal bir bilgi akışı sürecini başlatmaktadır. Bu çalışmada soru bankası uygulamasının yapılmasının en önemli nedeni de bu tür bir süreci desteklemesidir. Uygulamanın geliştirilmesi sırasında teknolojinin sunduğu imkanlardan faydalanılmıştır. Uygulamada ASP, ADO ve ODBC'nin İnternet kapsamında entegre edilmesi oldukça önemlidir. Uygulama açık bir mimariyi desteklediğinden farklı veritabanlarıyla da çalışabilir hale getirilmesi de kolaydır. ODBC, teknolojik olarak bu tür bir imkanı sunmaktadır.

8. KAYNAKLAR

Mezick, D., (1998), "Visual Basic 5 Certification Exam Guide", McGraw-Hill.

Kaufmann, J., (1999), "Beginning ASP Databases", Wrox Press.

Walther, S., (1999), "Active Server Pages 2.0", Sams Publishing.

McKinney, B., (1997), "Hardcore Visual Basic", Microsoft Press.

Wood, C., (1999), "OLE DB and ODBC Developers Guide", M&T Books.

Sanders, R., (1999), "ODBC 3.5 Developer's Guide", McGraw Hill.

Sussman, D., (1999), "Professional ADO 2.5", Wrox Press.