

T.C.
MİLLİ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

BİLİŞİM TEKNOLOJİLERİ

NESNE TABANLI PROGRAMLAMA 5

ANKARA 2007

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	ii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. ADO.NET	3
1.1. ADO.NET Veri Tabanı	3
1.2. Bağlı Olmayan Veri Tabanı Erişimi	3
1.3. Veri Sağlayıcıları (Data Providers).....	3
1.4. Connection Sınıfı	4
1.4.1. OleDbConnection Kullanarak Veri Tabanı Bağlantısı Yapmak.....	4
1.5. SQL Deyimleri.....	11
1.5.1. SQL'in Temel İfadeleri.....	11
1.6. OleDbDataAdapter Sınıfı.....	12
1.7. Dataset Nesnesi	18
1.8. Command Sınıfı	24
1.8.1. Command Nesnesi Oluşturmak	25
1.9. DataReader Sınıfları.....	29
1.9.1. OleDbDataReader Sınıfı.....	29
1.10. Veri Görüntüleme (DataView) Nesnesi	32
1.10.1. Sıralama (Sort).....	37
1.10.2. DataView'de Kayıt Aramak	39
1.10.3. DataView'e Kayıt Ekleme.....	42
1.10.4. DataView'deki Kaydı Değiştirmek	43
UYGULAMA FAALİYETİ	46
ÖLÇME VE DEĞERLENDİRME	52
ÖĞRENME FAALİYETİ-2	53
2. XML DOKÜMANLARI.....	53
2.1. XML Nedir.....	53
2.2. DataSet ve DataTable Nesneleri	54
2.3. DataTable İçeriğini XML'e Kaydetmek	55
2.4. XML Dosyayı DataSet'e Aktarmak.....	57
2.5. Projenize Xml Dosya Ekleme.....	59
2.6. XML Dosyanın Yapısını Belirlemek	59
2.7. XSLT (Extensible StyleSheet Language Transformation).....	61
2.8. XmlDataDocument Nesnesi.....	61
UYGULAMA FAALİYETİ	64
ÖLÇME VE DEĞERLENDİRME	65
MODÜL DEĞERLENDİRME	66
CEVAP ANAHTARLARI.....	67
KAYNAKÇA	68

AÇIKLAMALAR

KOD	482BK0078
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Veri Tabanı Programcılığı
MODÜLÜN ADI	Nesne Tabanlı Programlama 5
MODÜLÜN TANIMI	ADO.NET ve XML ile veri tabanı desteğiyle uygulama yapabilme ile ilgili öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	Nesne Tabanlı Programlama 4 modülünü bitirmiş olmak
YETERLİK	Veri tabanı uygulamaları yapmak.
MODÜLÜN AMACI	Genel Amaç Gerekli ortam sağlandığında, XML ve ADO.NET ile veri tabanı uygulamaları yapabileceksiniz. Amaçlar 1. ADO.NET ile veri tabanı uygulamaları yapabileceksiniz. 2. XML dokümanlarıyla veri tabanı uygulamaları yapabileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Ortam Atölye, laboratuvar, ev, bilgi teknolojileri ortamı (internet) vb. kendi kendinize veya grupta çalışabileceğiniz tüm ortamlar. Donanım Programlama dilini çalıştırabilecek yeterlikte bilgisayar, yedekleme için gerekli donanım (CD yazıcı, flash bellek), raporlama için yazıcı, sayfa için internet bağlantısı, kağıt ve kalem.
ÖLÇME VE DEĞERLENDİRME	Modülün içinde yer alan her öğrenme faaliyetinden sonra, verilen ölçme araçlarıyla kazandığınız bilgileri ölçerek kendinizi değerlendireceksiniz. Öğretmen, modül sonunda size ölçme aracı (çoktan seçmeli test, doğru yanlış testi vb.) uygulayarak modül uygulamaları ile kazandığınız bilgileri ölçerek değerlendirecektir.

GİRİŞ

Sevgili Öğrenci,

Okul yaşantınızda öğreneceğiniz her konu, yaptığınız her uygulama ve tamamladığınız her modül bilgi dağarcığınızı geliştirecek ve ileride atılacağınız iş yaşantınızda size başarı olarak geri dönecektir. Eğitim sürecinde daha özverili çalışır ve çalışma disiplini kazanırsanız, başarılı olmamanız için hiçbir neden yoktur.

Günümüzde Windows tabanlı görsel programlama dillerinin hızla gelişmekte olduğu ve kullanımının oldukça yaygınlaştığı görülmektedir. Bu programlama dilleri ile sizler programlama mantığını ve becerisini çok daha kolay kavrayacaksınız.

Bu modülle, .NET görsel programlama dillerinde kullanılan veri tabanı ve XML uygulamalarını öğreneceksiniz.

Bu modülde anlatılan konuların tümünü öğrendiğinizde, nesne tabanlı programlama dilinin veri tabanı ile ilgili konularını öğrenmiş olacak ve kendinize göre programlar yapabileceksiniz.

ÖĞRENME FAALİYETİ- 1

AMAÇ

ADO.NET ile veri tabanı nesnelerini, veri tabanına erişimi, veri sağlayıcıları ve SQL deyimlerini öğreneceksiniz.

ARAŞTIRMA

- Veri tabanıyla uygulamaları geliştirilmeden önce kayıtlar nasıl saklanırdı? Araştırınız.

1. ADO.NET

1.1. ADO.NET Veri Tabanı

ADO.NET (ActiveX Data Object), veri kaynaklarından verileri almak, işlemek, düzenlemek ve veri kaynağına yeni veriler kaydetmek için kullanılan .NET sınıflarının tümüdür. ADO.NET, ADO teknolojisinin .NET ortamındaki yeni versiyonudur.

Daha önceden veri kaynaklarına ulaşmada ortaya çıkan kütüphane problemlerini gidermek için Microsoft ve diğer firmalar ODBC (Open DataBase Connectivity) adı verilen ortak kütüphaneyi geliştirdiler. Böylece bu kütüphane, her türlü veri tabanına erişim için kullanılan bir kütüphane olmuştur.

1.2. Bağlı Olmayan Veri Tabanı Erişimi

Önceki teknoloji olan ADO'da veri tabanındaki bilgileri listelemek istediğimizde, veri tabanı bağlantısını açıp bir döngü aracılığıyla bilgileri alıp döngü sonunda veri tabanı bağlantısı kapatırdık. Bu işlem süresince veri tabanı ile olan bağlantı sürekli açık kalırdı. Bu duruma "bağlı veri tabanı erişimi" denir.

ADO.NET teknolojisi ise veri tabanından istenilen bilgileri aldıktan sonra bağlantıyı kapatmaktadır. Bu duruma da "bağlı olmayan veri tabanı erişimi" denir.

1.3. Veri Sağlayıcıları (Data Providers)

Veri sağlayıcıları kullanarak veri tabanına bağlanabilir, veri tabanından veriler alabilir veya veri tabanına veri gönderebiliriz.

.NET'te kullanılan veri sağlayıcıları şunlardır;

- **.NET Framework Data Provider for ODBC** : ODBC veri kaynaklarına ulaşmak için kullanılan bir veri sağlayıcıdır. ODBC veri sağlayıcısı ile ilgili üyeler, .Net Framework içinde, **System.Data.Odbc** isim uzayında yer almaktadır. ODBC sürücüsü yardımıyla, SQL Server'a, Access tablolarına ve ODBC'yi destekleyen veri sürücülerine erişebiliriz.
- **.NET Framework Data Provider for OLE DB** : OLE DB veri kaynaklarına ulaşmak için kullanılan veri sağlayıcıdır. Ole Db veri sağlayıcısının .NET Framework üyeleri, **System.Data.OleDb** isim uzayında yer alır. Çoğunlukla bu veri sağlayıcısını Access tablolarına erişmek için kullanmaktayız.
- **.NET Framework Data Provider for Oracle** : Oracle veri kaynaklarına ulaşmak için kullanılan veri sağlayıcıdır. Oracle servis sağlayıcısı, .NET Framework'ün **System.Data.OracleClient** isim uzayında yer alan üyelerden oluşur. Oracle veri sağlayıcısının .NET'e yerleştirilmesindeki temel amaç, Oracle veri tabanlarına OLE DB veri sağlayıcısı ile OLE DB üzerinden değil, doğrudan erişilebilmesini sağlamaktır. Bu sayede Oracle veri kaynağı ile oluşturulan etkileşimde en iyi performansın elde edilmesi sağlanmıştır.
- **.NET Framework Data Provider for SQL Server** : SQL Server veri kaynaklarına ulaşmak için kullanılan veri sağlayıcıdır. SQL veri sağlayıcısına ait tüm üyeler, **System.Data.SqlClient** isim uzayında yer almaktadır.

1.4. Connection Sınıfı

Connection sınıfı veri tabanı ile bağlantı kurmak ve bağlantıyı düzenlemek için kullanılan bir nesnedir. Bu nesnenin dört temel ögesi vardır. Bunlar;

- **OleDbConnection**
- **OdbcConnection**
- **SqlConnection**
- **OracleConnection** 'dır.

Uygulamalarımız Microsoft Access veri tabanına göre hazırlandığından OleDbConnection sınıfını kullanacağız.

1.4.1. OleDbConnection Kullanarak Veri Tabanı Bağlantısı Yapmak

Bu sınıf miras alınmaz ve sınıfın alt sınıfı olamaz. OleDbConneciton kullanarak veri tabanı ile bağlantı kurabilmek için ilk önce veritabanının oluşturulmuş olması gerekir. Bunun için Microsoft Access programını kullanmanız gerekir.

Veri tabanıyla bağlantı işlemini bir örnekle açıklayalım.

Örnek

Örneğimizde bir telefon rehberi veri tabanıyla bağlantı yapalım. Access veri tabanında telefon rehberi için gerekli olan alanları (ad, soyad, adres, telefon) oluşturarak bu alanlara kayıt girilmiş olsun. Tablodaki bir alanı birincil anahtar olarak belirlemeniz gerekir.



Alan Adı	Veri Türü	Tanım
ad	Metin	
soyad	Metin	
adres	Metin	
telefon	Metin	

Resim 1.1 : Veri tabanındaki alanlar

Access'da veri tabanı oluşturulduktan sonra bağlantının gerçekleşip gerçekleşmediğini bir konsol uygulamasıyla test edelim.

Bir konsol uygulaması başlatın ve aşağıdaki kodları Main() nesnesine yazın.

```
using System;
using System.Data.OleDb;
public class AccessOleDbOrnegi
{
    public static void Main()
    {
        string dosya="Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\\tel_rehber.mdb";
        OleDbConnection baglanti=new OleDbConnection(dosya);
        try
        {
            baglanti.Open();
            Console.WriteLine("Başarıyla OleDb kullanarak veritabanına bağlanıldı.");
            baglanti.Close();
        }
        catch(Exception hata)
        {
            Console.WriteLine(hata.Message.ToString());
        }
        Console.ReadLine();
    }
}
```

OleDbConnection sınıfını kullanabilmek için System isim uzayında bulunan Data.OleDb isim uzayını projenize eklemeniz gerekmektedir. Daha sonra bağlantıda kullanılacak veri sağlayıcısını ve veri tabanını string bir değişkene (dosya) aktarmanız gerekmektedir.

```
string dosya="Provider=Microsoft.Jet.OLEDB.4.0;Data
```

```
Source=C:\\tel_rehber.mdb";  
OleDbConnection baglanti=new OleDbConnection(dosya);
```

OleDbConnection sınıfından yeni bir nesneyi string deęiřkene gre oluřturduktan sonra artık veri tabanıyla baęlantıyı gerekleřtirmiř olursunuz. Gerekleřen baęlantıyı amak iin Open () metodu kullanılmaktadır. Open () metodunu OleDbConnection sınıfından oluřturduęunuz "baęlantı" adlı nesneyle kullanmanız gerekmektedir.

```
baglanti.Open();
```

Konsoldan bir mesaj vererek baęlantının aılıp aılmadıęını anlayabiliriz.

```
Console.WriteLine("Bařarıyla OleDb kullanarak veri tabanına  
baęlanıldı.");
```

Sonra aık olan baęlantıyı Close () metoduyla kapatmanız gerekecektir.

```
baglanti.Close();
```

Baęlantı sırasında oluřabilecek hataları engellemek iin bu kodları try bloęunda yazmanız gerekir. Hata oluřması durumunda da catch bloęuna ilgili mesajın yazdırılmasını saęlayabilirsiniz.

```
catch(Exception hata)  
{  
    Console.WriteLine(hata.Message.ToString());  
}
```

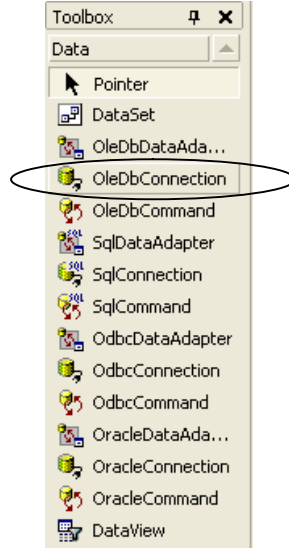
Bu iřlemlerin ardından uygulamanızı alıřtırdıęınızda ekran grnts ařaęıdaki gibi olacaktır.



Resim 1.2: Veri tabanı baęlantısının gerekleřtięini gsteren mesajdır

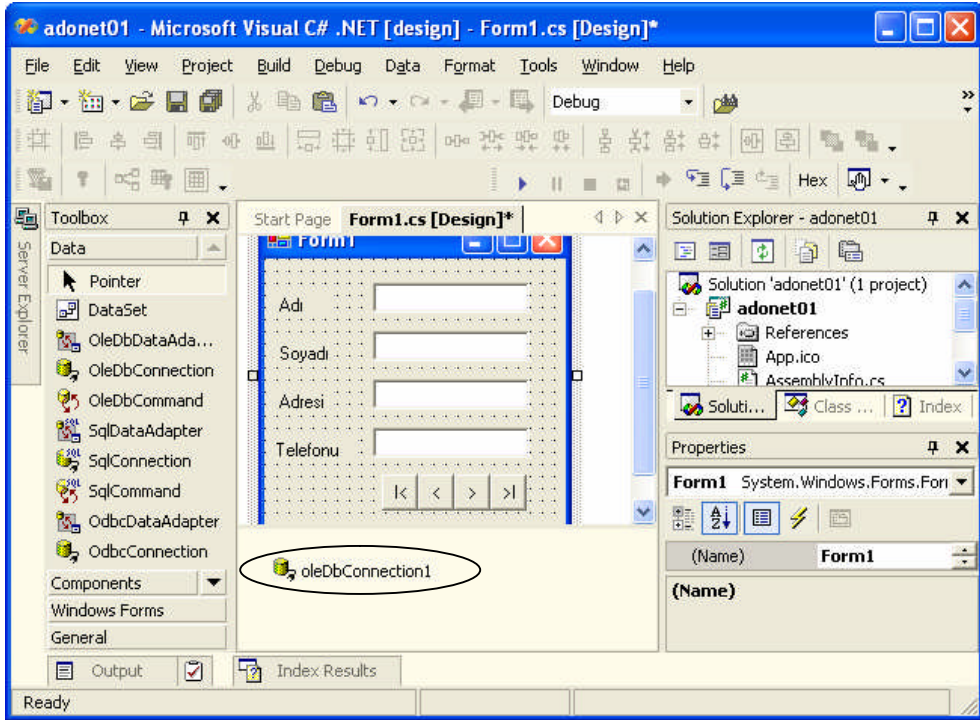
OleDbConnection sınıfını kullanarak veri tabanı baęlantısı yapmanın bir bařka yolu daha vardır. Bu da kullanacaęınız forma OleDbConnection nesnesini ekleme yntemidir.

Form'a OleDbConnection nesnesi eklemek iin Toolbox'taki Data sekmesinden OleDbConnection nesnesini tıklatarak form zerine yerleřtirmeniz gerekecektir.



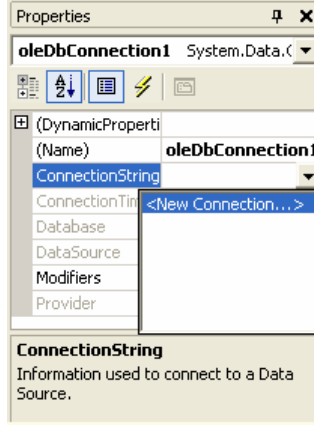
Resim 1.3: Data sekmesindeki OleDbConnection nesnesi

Form üzerine OleDbConnection nesnesini eklediğinizde formun alt tarafında size görüntülenecektir.



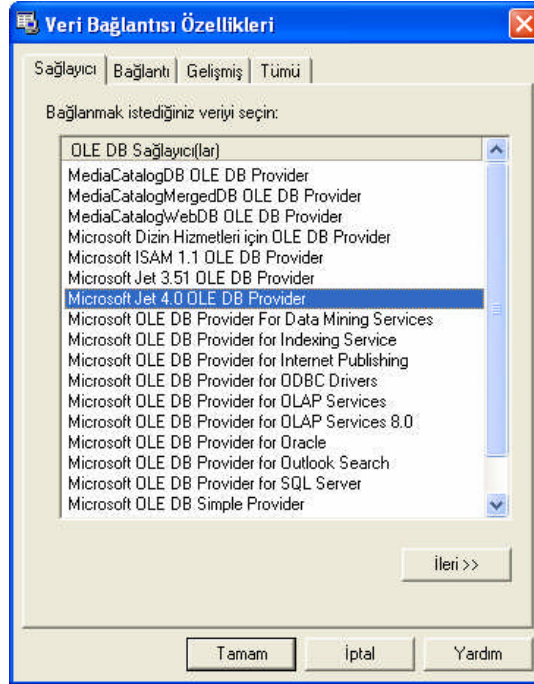
Resim 1.4: OleDbConnection nesnesinin forma eklenmesi

Eklenen Connection nesnesinin adı OleDbConnection1'dir. İsterseniz bu bağlantı adını kendinize göre değiştirebilirsiniz. Connection nesnesi projeye eklendikten sonra Properties penceresinde bağlantı özelliklerini ayarlamanız gerekecektir. ConnectionString alanındaki açılır liste kutusuna tıkladığınızda bağlantıyı gerçekleştirebileceğiniz <New Connection> seçeneği karşınıza gelecektir.



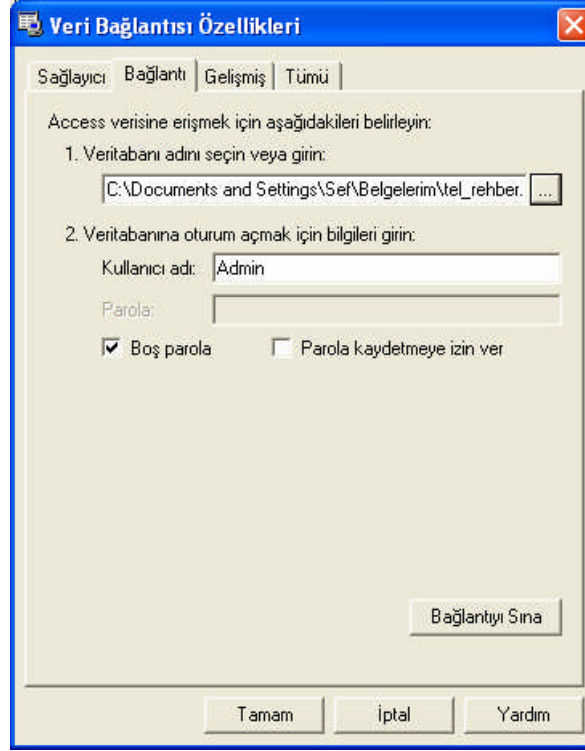
Resim 1.5: ConnectionString'in belirlenmesi

new connection seçeneğine tıkladığınızda ekrana "veri bağlantısı özellikleri" iletişim kutusu ekrana gelir.



Resim 1.6: Veri bağlantısı özellikleri

Veri tabanını Microsoft Access'ta oluşturduğunuz için Sağlayıcı sekmesinden veri sağlayıcınızı (Provider) Microsoft Jet 4.0 Ole Db Provider olarak seçmeniz gerekir. İleri düğmesine tıklayarak veri bağlantısı özelliklerinin ikinci adımına geçiniz. İkinci adım olan Bağlantı sekmesine geçtiğinizde ekran görüntüsü aşağıdaki gibi olacaktır.



Resim 1.7: Bağlantı sekmesi

Bağlantı sekmesindeki 1.bölümde veri tabanının adını seçmeniz gerekir. Veri tabanınızın bulunduğu konumu seçtikten sonra veri tabanına erişim için kullanılacak bir parola girebilir veya boş bırakabilirsiniz. Bağlantıyı Sına düğmesine tıkladığınızda veri tabanıyla bağlantının gerçekleşip gerçekleşmediğini görebilirsiniz.



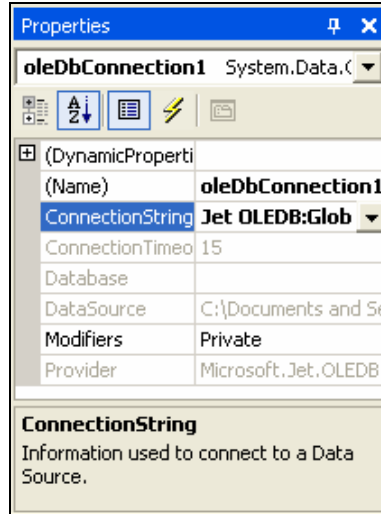
Resim1.8: Bağlantı durumu

Tamam düğmesine tıkladığınızda bağlantınızın şifre içerip içermediğini soran bir iletişim kutusu ekrana gelecektir.



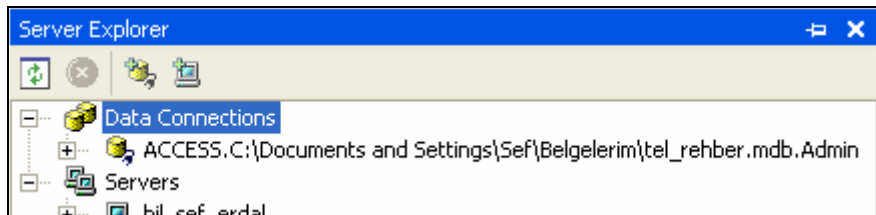
Resim 1.9: connectionString'in şifre içerip içermeyeceğinin belirlenmesi

“Don't include password” düğmesine tıklatarak bağlantının şifre içermeyeceğini belirtin. Tüm bu işlemlerden sonra Properties penceresindeki connectionString alanı otomatik olarak doldurulacaktır.



Resim 1.10: connectionString alanının son durumu

Böylece veri tabanı ile olan bağlantı sağlanmış olacaktır. Bağlantının durumu Server Explorer penceresinden de görülebilir.



Resim 1.11: Server Explorer penceresinde bağlantının görüntülenmesi

1.5. SQL Deyimleri

SQL (Structured Query Language), veritabanları için geliştirilmiş standart bir dildir. Veri tabanından verinin elde edilmesini, veri eklenmesini, veri silinmesini ve veri sorgulanmasını yöneten bir dildir. Bütün veri tabanı programlarında kullanılır.

1.5.1. SQL'in Temel İfadeleri

□ **SELECT Deyimi**

Veri tabanından veri elde etmek için kullanılır. İstenen koşullara göre uygun verilerin getirilmesi sağlanabilir.

Örnek

```
SELECT * FROM rehber
```

şeklindeki yazım rehber tablosundaki bütün verilerin elde edilmesini sağlar.

```
SELECT ad FROM rehber
```

ifadesiyle, rehber tablosundaki sadece "ad" alanının bilgilerinin elde edilmesini sağlar.

```
SELECT * FROM rehber WHERE ad="Ali"
```

ifadesiyle, ad alanındaki Ali ismindeki tüm kayıtların elde edilmesini sağlar.

```
SELECT * FROM rehber WHERE ad="Ali" ORDER BY ad ASC
```

Verilen koşullara göre sütundaki bilgileri artan (ASC) ya da azalan (DESC) sırada ekrana getirir. WHERE ile oluşturulan koşul ifadelerinde mantıksal operatörler de kullanılabilir (and, or,not).

```
SELECT * FROM rehber WHERE ad LIKE '*al'
```

Koşul değerlerini tam olarak değil seçimli olarak kullanabiliriz. Koşul değeri sonunda "al" olan fakat öncesinde farklı karakterleri olan bir ismi belirtmek için kullanılır.

□ **INSERT Deyimi**

Veri tabanına yeni bir kayıt eklemek için kullanılır.

```
INSERT INTO rehber (ad,soyad) VALUES ('Ceylin','Yılmaz')
```

şeklindeki ifadeyle rehber tablosunun ad ve soyad alanlarına yeni değerler ekler.

□ UPDATE Deyimi

Kayıtları güncellemek için kullanılır. Hangi kayıtların güncelleneceği bir koşul veya koşullarla belirtilebilir.

```
UPDATE rehber SET ad='Ceylin' WHERE soyad='Yılmaz'
```

biçimindeki bir bildirim soyadı Yılmaz olan kayıtların ad bilgisini Ceylin olarak değiştirir. Eğer birden fazla Yılmaz soyadı olsaydı hepsinin ad alanı Ceylin olarak değiştirilecekti.

□ DELETE Deyimi

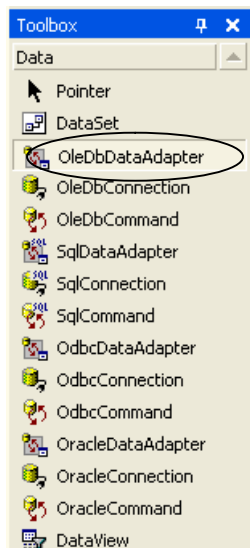
Tablodan kayıt silmek için kullanılır.

```
DELETE FROM rehber WHERE soyad='Yılmaz'
```

Soyad değeri Yılmaz olan tüm kayıtları siler.

1.6. OleDbDataAdapter Sınıfı

DataAdapter sınıfları uygulama ile veri tabanı arasında iletişim kuran sınıflardır. Kullanılmak istenen Connection nesnesinin adı ConnectionString özelliğine aktarıldığında değişik görevleri olan bağlantı metni hazırlanır. Bağlantı metnini elle hazırlamak birden fazla adım içerdiğinden bağlantı metni Server Explorer penceresinden kolayca yapılabilir. OleDbConnection nesnesi için Properties penceresinde Connection seçildikten sonra Toolbox'tan OleDbDataAdapter nesnesi seçilerek uygulamaya yerleştirilir.



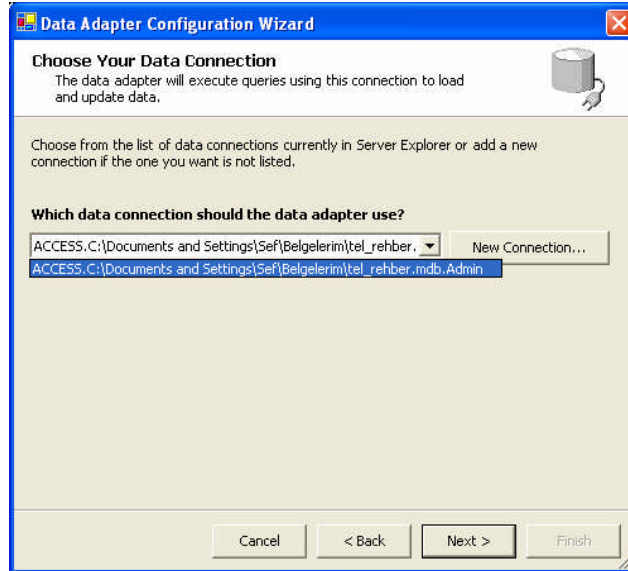
Resim 1.12: Data sekmesindeki OleDbDataAdapter nesnesi

OleDbDataAdapter uygulamaya yerleřtirildikten sonra ekrana Data Adapter Configuration Wizard iletiřim kutusu ekrana gelir. Bu sihirbaz yardımıyla veri tabanındaki kayıtlara eriřim ve deęiřikliklerin yapılması saęlanacaktır.



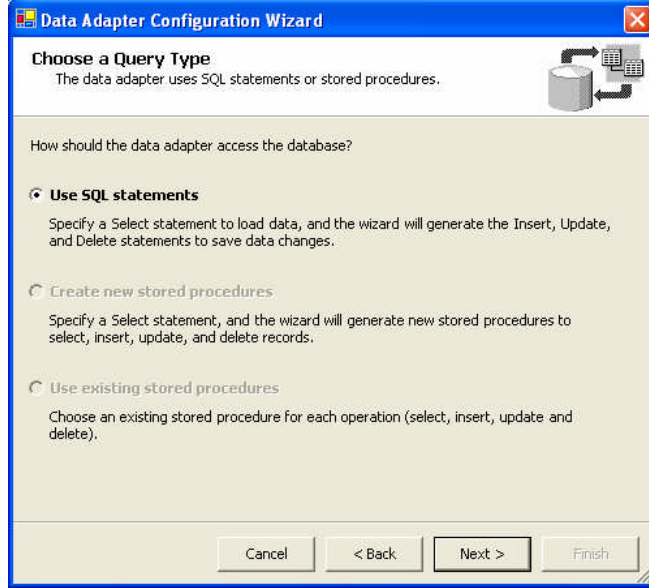
Resim 1.13: Sihirbaz ekranı

Next düęmesine tıklanarak ikinci adıma geçilir. İkinci adımda veri tabanının seçilmesi gereklidir. OleDbConnection ile baęlantı yapıldığı için baęlantı yapılan veri tabanının adı otomatik olarak görüntülenir.



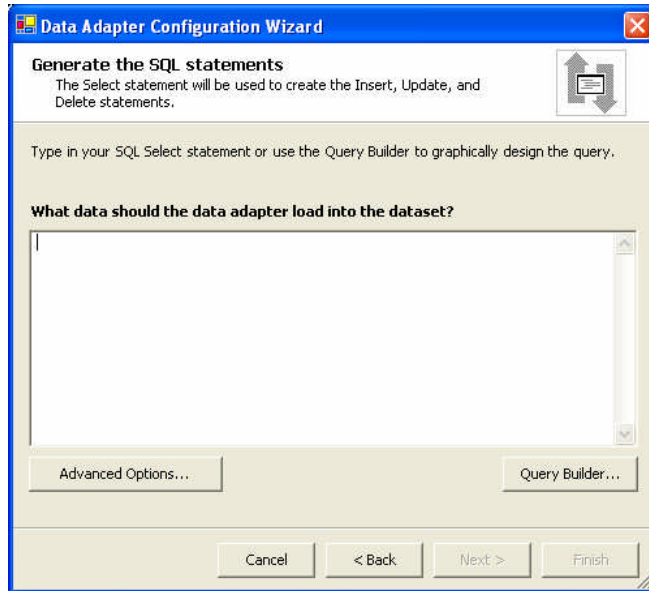
Resim 1.14: Veri tabanının belirlenmesi

Üçüncü adımda veri tabanı ile olan işlemler için sorgulama tipi seçilir. Microsoft Access ile veri tabanı oluşturulduğu için diğer iki seçenek pasif durumda olur.



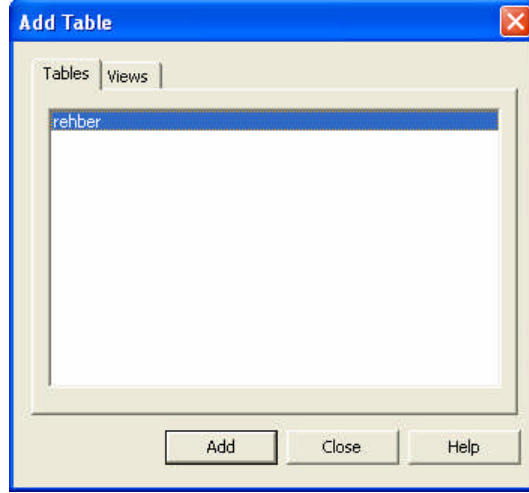
Resim 1.15: Sorgulama tipinin seçilmesi

Use SQL statements seçeneği seçili durumdayken Next düğmesi tıklanır. Dördüncü adımda SQL kodu yazılan Generate the SQL statements iletişim kutusu ekrana gelir.



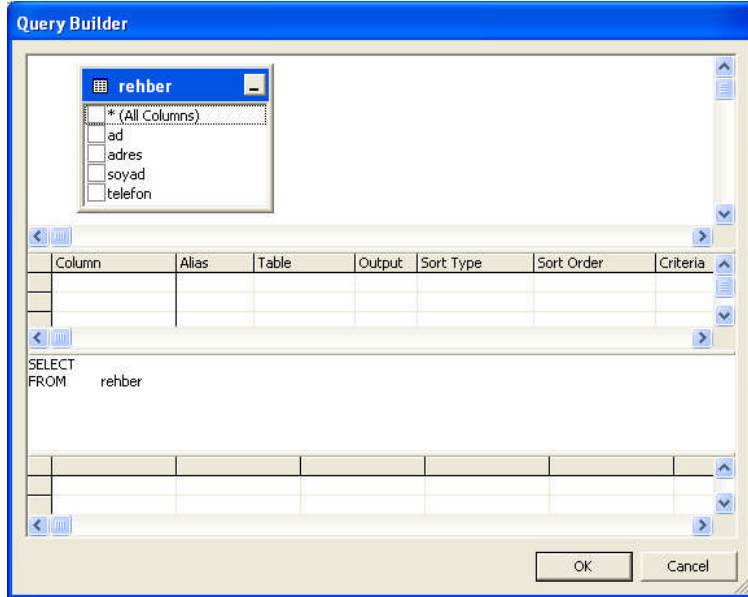
Resim 1.16: SQL kodu yazım ekranı

Yapılacak işlemlere göre Resim 1.16 ekranına SQL kodu yazılır. SQL kodunu doğrudan kendiniz yazmak yerine Query Builder düğmesine tıklatarak da yapabilirsiniz. Query Builder düğmesine tıklanınca Add Table iletişim kutusu ekrana gelir.



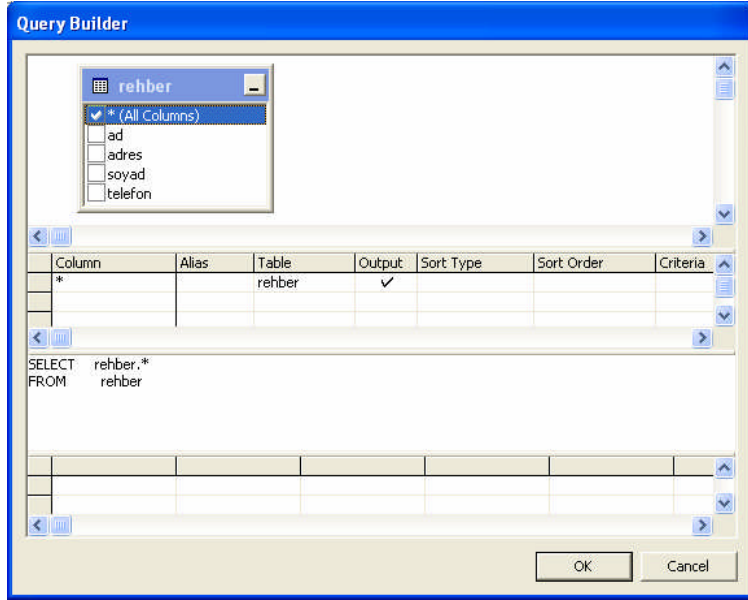
Resim 1.17: Tablo ekleme ekranı

Eklenicek tablo seçilip Add düğmesine tıklanır. Tablonun hangi alanlarıyla işlem yapılacaksa o alanlar seçilir. Eğer tüm alanlarla ilgili işlem yapılacaksa tablodan All Columns kutucuğu seçilmelidir.



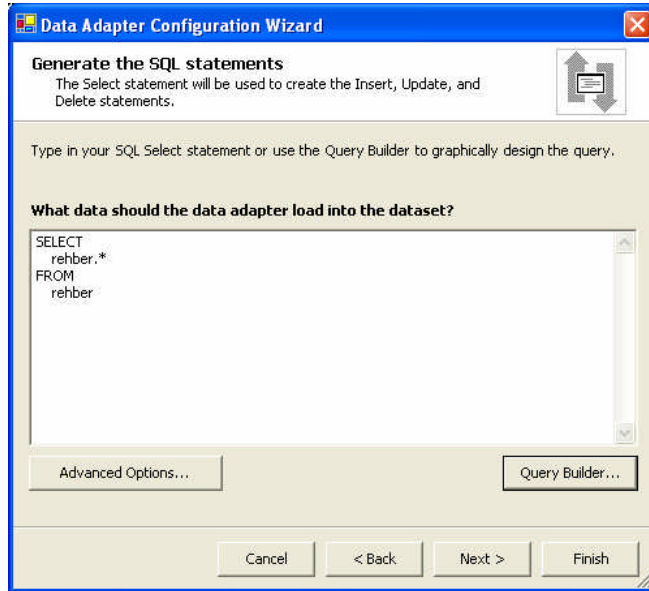
Resim 1.18: Query Builder ekranı

All Columns seçildiğinde SQL kodu sizin için oluşturulmuş olacaktır.



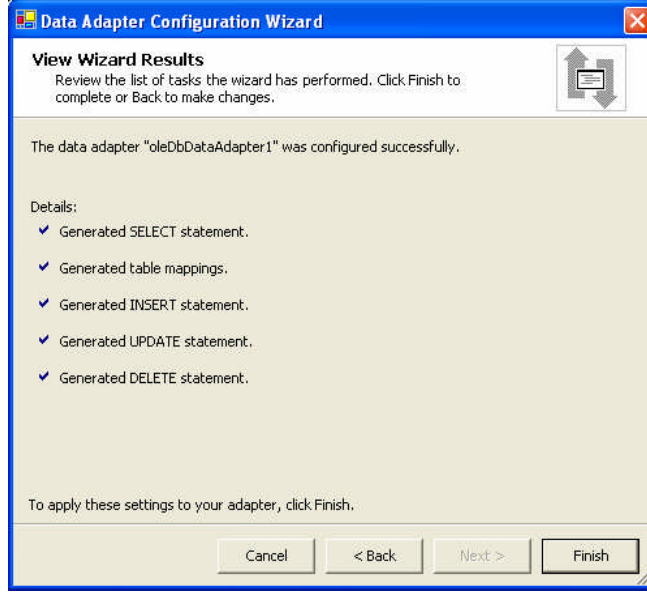
Resim 1.18: SQL kodunun oluşturulması

OK düğmesine tıklanıp Query Builder ekranı kapatılır. Böylece Generate the SQL statements penceresinde SQL kodu hazırlandıktan sonra bir sonraki adım için Next düğmesine tıklanır.



Resim 1.19: SQL kodunun oluşturulması

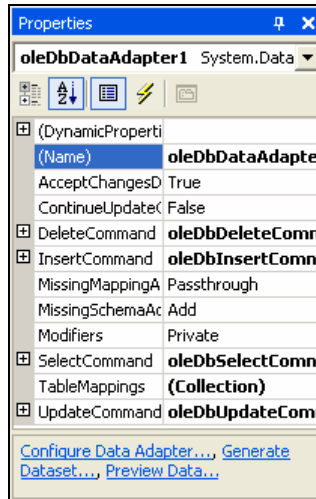
Bu kod sayesinde tablodaki bütün kayıtlara erişilip görüntülenebilir. Sihirbazın sonuçları hakkında bilgi veren View Wizard Results penceresi ekrana gelir.



Resim 1.20: Sihirbaz sonuçlarının görünümü

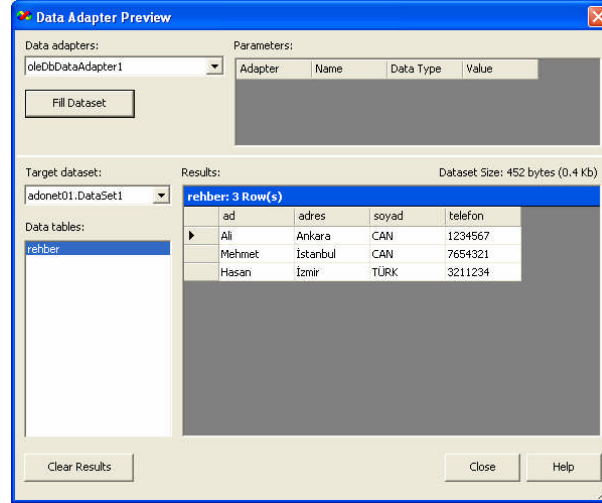
Data Adapter Wizard ile ilgili tablodaki kayıtları görüntülemek, kayıt girmek, silmek ve yapılan değişiklikleri tabloya aktarmak için gereken SQL kodlarını otomatik olarak hazırlanmış olur.

Hazırlanan SQL kodlarını OleDbDataAdapter'in Properties penceresinde görebilirsiniz.



Resim 1.21: SQL komutlarının Properties penceresinde görüntülenmesi

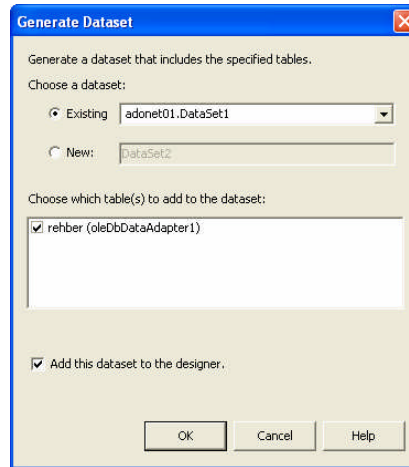
Hazırlanan OleDbDataAdapter nesnesinin özelliklerini değiştirmek için fareyle sağ tıklanınca açılan kısa yol menüsünden Configure Data Adapter komutu veya Properties penceresinin alt kısmındaki Configure Data Adapter linki tıklanır. Data Adapter nesnesiyle elde edilecek bilgiler Properties penceresindeki Preview Data linki tıklanarak görülebilir.



Resim 1.22: Verilerin görüntülenmesi

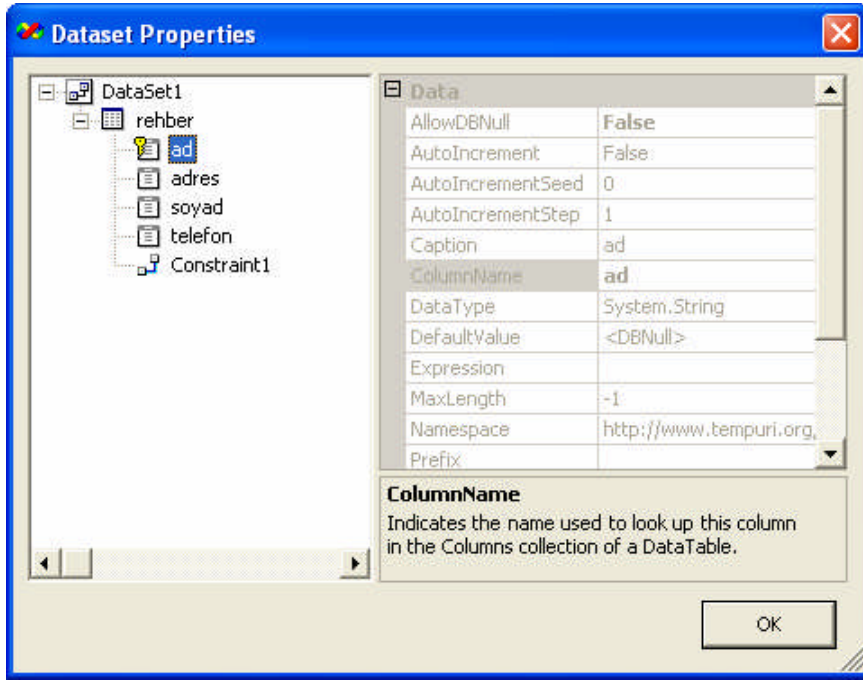
1.7. Dataset Nesnesi

Access veri tabanıyla işlem yapılırken DataSet nesnesine gerek vardır. Dataset nesnesi OleDbDataAdapter kontrolü ile hazırlanır. Properties penceresindeki Generate Dataset linkine tıkladığınızda Generate Dataset iletişim kutusu ekrana gelir.



Resim 1.23: Generate Dataset iletişim kutusu

Dataset nesnesine istenilen bir isim yazılıp ya da varolan isim kabul edilerek OK düğmesine tıklanır. Dataset nesnesi formun altına yerleşmiş olur. Dataset nesnesi seçiliyken Properties penceresinden istenilen özellikleri değiştirilebilir. Dataset nesnesine aktarılan tablonun elemanları hakkında bilgi edinmek isterseniz fare tuşuyla açılan kısayol menüsünden Dataset Properties komutu veya Properties penceresi altındaki Dataset Properties linki tıklanır.



Resim 1.24: Dataset özellikleri penceresi

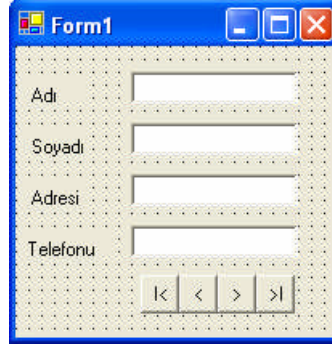
DataAdapter ile bilgiler Dataset'e aktarılırken Fill() metodundan yararlanır. Formun Load () metoduna aşağıdaki kodları yazarsanız ilgili tablonun içeriği Dataset nesnesine aktarılır.

```
private void Form1_Load(object sender, System.EventArgs e)
{
    OleDbDataAdapter1.Fill(dataSet11, "rehber");
}
```

Böylece Dataset'te rehber adında bir DataTable nesnesi hazırlanmış olur. Bundan sonra Dataset nesnesindeki kayıtları istediğiniz şekilde kullanabilirsiniz.

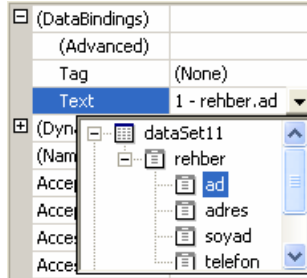
Örnek

Bu adımdan sonra forma dört adet text kutusu, dört adet label ve verilerin arasında dolaşabilmek için de 4 adet Command düğmesi ekleyin.



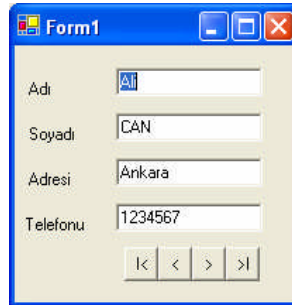
Resim 1.25: Form tasarımı

Birinci text kutusunu seçili durumdayken ToolBox'taki DataBindings alanındaki açılır liste kutusuna tıklayın. Oluşturulan Dataset içerisindeki tablo alanları size listelenecektir.



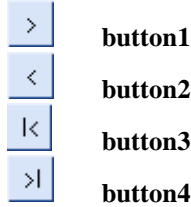
Resim 1.26: DataBindings text alanı

Birinci text kutusu için bu işlemi gerçekleştirdiğimizden tablo alanlarından ad alanını tıklatın. Artık birinci text kutusu Dataset yardımıyla ad alanıyla ilişkilendirilmiş olur. Diğer text kutuları için de aynı işlemleri yaparak ilgili alanları seçin. Uygulamayı çalıştırdığınızda (F5) tablodaki alanlara girilmiş kayıtlar form üzerinde görüntülenecektir.



Resim 1.27: Verilerin form üzerinde görüntülenmesi

Form üzerine yerleştirilen Command düğmelerinin bu aşamada bir etkisi bulunmamaktadır. Bu düğmelerin de aktif olabilmesi için veri bağlama işleminin gerçekleştirilmiş (DataBindings) olması gerekir. DataBindings, bir nesnenin bir çok özelliğine veri bağlaması yapılması işlemini yöneten nesnedir. Şimdi, formdaki Command düğmelerinin aktif olabilmesi için şu kod satırlarını yazmak gerekmektedir.



Resim 1.28 : Komut (hareket) düğmeleri

```
private void button1_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"rehber"].Position=
        this.BindingContext[dataSet11,"rehber"].Position+1;
}

private void button2_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"rehber"].Position--;
}

private void button3_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"rehber"].Position=0;
}

private void button4_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"rehber"].Position=
        this.dataSet11.Tables["rehber"].Rows.Count-1;
}
```

Bunun sonucunda tablodaki veriler arasında kolayca dolaşılır. Ancak hangi verinin kaçınıcı eleman olduğunu ve tabloda toplam kaç kaydın olduğunu görmek mümkün değildir. Aktif kaydın numarasını ve tabloda toplam kaç kaydın olduğunu görmek için komut düğmelerinin arasına bir text kutusu daha ekleyin.



Resim 1.29: Komut düğmeleri arasına text kutusunun yerleştirilmesi

Eklenecek text kutusunu adı textBox5 olacaktır. Beşinci text kutusunda aktif kaydı ve toplam kaydı görebilmek için kayıt_goster() isimli bir metod oluşturmanız gerekecektir.

```
int kayıt_no, toplam_kayıt;
void kayıt_goster()
{
    kayıt_no=this.BindingContext[dataSet11, "rehber"].Position+1;
    toplam_kayıt=this.dataSet11.Tables["rehber"].Rows.Count;
    textBox5.Text=(kayıt_no).ToString()+" / "+(toplam_kayıt).ToString();
}
```

Kayıtları görebilmek için oluşturulan kayıt_goster() metodunda iki adet değişken kullanılmıştır. Birinci değişken olan kayıt_no, o anda gösterilen aktif kaydın numarasını göstermektedir. Kayıtlar veri tabanında tutulurken kayıt numarası 0 (sıfır)'dan başlamaktadır. Kayıt numarasını 0 olarak görmek istemediğimiz için Position deyimini 1 artırmak gerekmektedir. İkinci değişken olan toplam_kayıt değişkeni de tabloda bulunan tüm kayıtların sayısını verecektir. Bu da Rows deyiminin Count özelliğiyle elde edilir. Son olarak beşinci text kutusuna bu iki değişkeni yazdırdıktan sonra aktif kaydın numarasını ve toplam kayıt sayısını text kutusunda görüntüleyebilirsiniz. Ancak bu metodun yani kayıt_goster() metodunun işletilmesi için Form1_Load() metoduna kayıt_goster() metodunu dahil etmek gerekir.

```
private void Form1_Load(object sender, System.EventArgs e)
{
    OleDbDataAdapter1.Fill(dataSet11, "rehber");
    kayıt_goster();
}
```

Bu işlem formun çalıştırıldığı anda ilk kaydın numarasını ve toplam kayıt sayısını verecektir. Yani bir başka kayıta gidildiğinde kayıt numarası değişmeyecektir. Son olarak da kayıt_goster() metodunu komut düğmelerinin Click() metoduna ekleyerek bu sorunu halletmiş olursunuz.

```

private void button1_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet1,"rehber"].Position=
        this.BindingContext[dataSet1,"rehber"].Position+1;
    kayit_goster();
}

private void button2_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet1,"rehber"].Position-=1;
    kayit_goster();
}

private void button3_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet1,"rehber"].Position=0;
    kayit_goster();
}

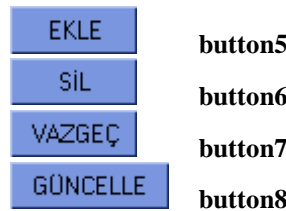
private void button4_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet1,"rehber"].Position=
        this.dataSet1.Tables["rehber"].Rows.Count-1;
    kayit_goster();
}

```

Dataset nesnesini kullanarak tabloya yeni veri girebilir, silebilir ve verileri güncelleyebilirsiniz. Bunun için form üzerine dört adet Command düğmesi daha ekleyiniz.

Resim 1.30: Ekle, sil, vazgeç ve güncelle düğmelerinin eklenmesi

Bu komut düğmeleri aracılığıyla tabloya yeni bir kayıt ekleyebilecek, bir kaydı silebilecek, yapılan işlemde vazgeçebilecek ve kayıtları güncelleyebileceksiniz.



Resim 1.31: Forma yeni eklenen komut düğmeleri

```

private void button5_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"rehber"].AddNew();
    textBox1.Focus();
}
private void button6_Click(object sender, System.EventArgs e)
{
    int kay_no=this.BindingContext[dataSet11,"rehber"].Position;
    this.BindingContext[dataSet11,"rehber"].RemoveAt(kay_no);
    this.BindingContext[dataSet11,"rehber"].Position-=1;
    this.BindingContext[dataSet11,"rehber"].EndCurrentEdit();
    oleDbDataAdapter1.Update(dataSet11);
}
private void button7_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"rehber"].CancelCurrentEdit();
}
private void button8_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"rehber"].EndCurrentEdit();
    oleDbDataAdapter1.Update(dataSet11);
}

```

Beşinci komut düğmesindeki AddNew() metoduyla tabloya yeni bir kaydın eklenmesi sağlanır. Focus() metodu ise baş tarafında belirtilen nesnenin aktif nesne olmasını sağlar.

Altıncı komut düğmesinde, aktif kaydın numarası kayno değişkenine aktarılarak RemoveAt() metoduyla tablodan silinir. Aktif kaydın silinmesinin ardından Position, bir önceki kayıta konumlandırılır. Aktif kaydı düzenleme işlemi EndCurrentEdit() metoduyla sonlandırılır ve OleDbDataAdapter'in Update() metoduyla da Dataset güncellenmiş olur.

Yedinci komut düğmesinde, mevcut düzenleme işleminden çıkılması CancelCurrentEdit() metoduyla sağlanır. Böylece kayıta yapılan değişiklik veya yeni bir kaydın girilmesi işlemi sonlandırılmış olur.

Sekizinci komut düğmesinde ise aktif kaydı düzenleme işlemi EndCurrentEdit() metoduyla sonlandırılır ve OleDbDataAdapter'in Update() metoduyla da Dataset güncellenmiş olur.

1.8. Command Sınıfı

SQL deyimlerini temsil eden bir sınıftır. SQL komutları Command nesneleri sayesinde işletilmektedir.

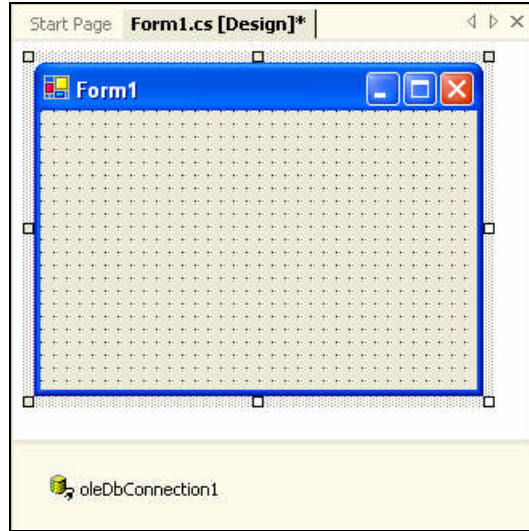
Ado.Net'teki Command sınıfları şunlardır.

- OleDbCommand**
- OdbcCommand**
- SqlCommand**

□ OracleCommand

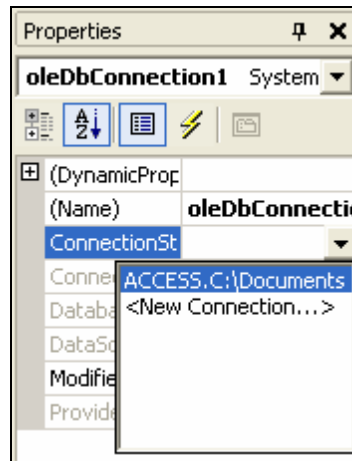
1.8.1. Command Nesnesi Oluşturmak

Command nesnesi oluşturma işlemini bir örnekle açıklayacak olursak Command nesnesi oluşturmak için forma bir OleDbConnection nesnesinin yerleştirilmiş olması gerekir.



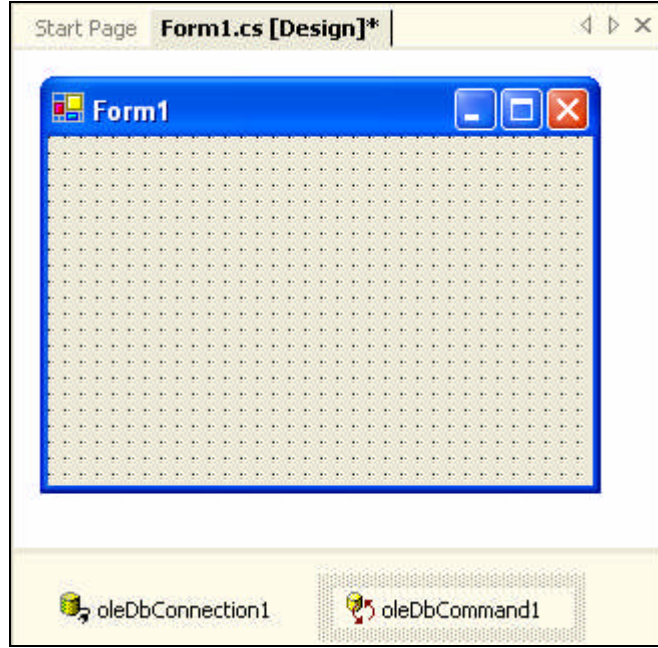
Resim 1.32: OleDbConnection nesnesinin eklenmesi

Form'a OleDbConnection nesnesini yerleştirdikten sonra Properties penceresinde Connection nesnesinin ConnectionString özelliğine daha önce Server Explorer penceresinin üst kısmında Data Connections kısmında tanımlı bağlantı bilgileri aktarılır.



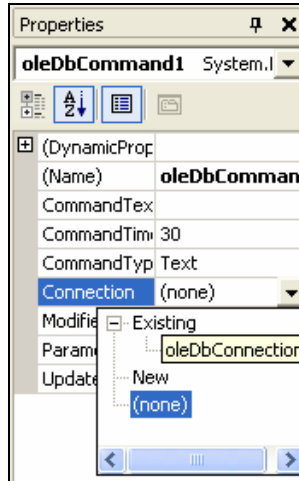
Resim 1.33: ConnectionString'in belirlenmesi

ToolBox'tan OleDbCommand nesnesi projeye eklenir.



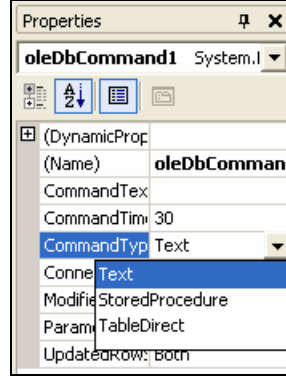
Resim 1.34: Command nesnesinin forma eklenmesi

Command nesnesinin Connection özelliğiyle bağlantı seçimi yapılır. Properties penceresinde Connection özelliğine ait liste kutusu açılarak yeni bir Connection hazırlanır veya mevcut Connection nesnelereinden biri seçilir.



Resim 1.35: Connection seçimi

Bu işlemden sonra CommandType özelliğiyle komutun tipi seçilir. Bu özellik başlangıçta Text seçeneğini içerir.

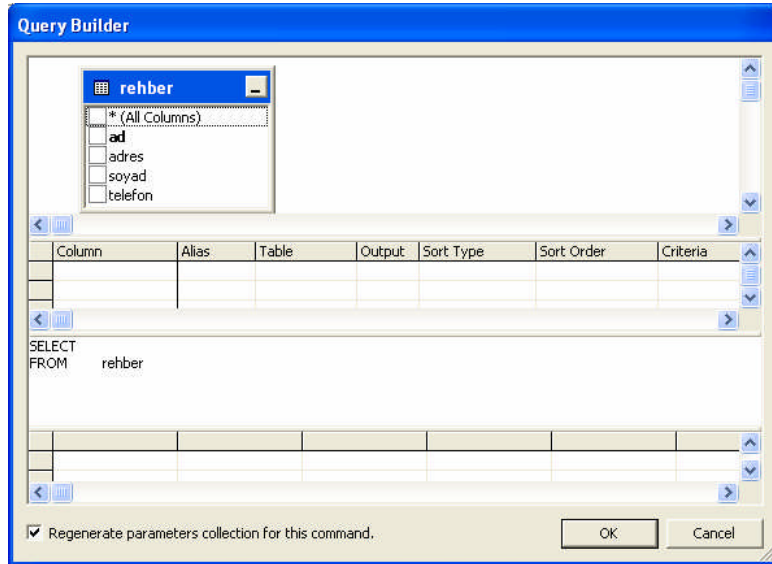


Resim 1.36: Tip seçimi

Command nesnesinin CommandType özelliğine TableDirect seçeneği aktarırsa kod yazmadan tablolar üzerinde işlem yapılabilir.

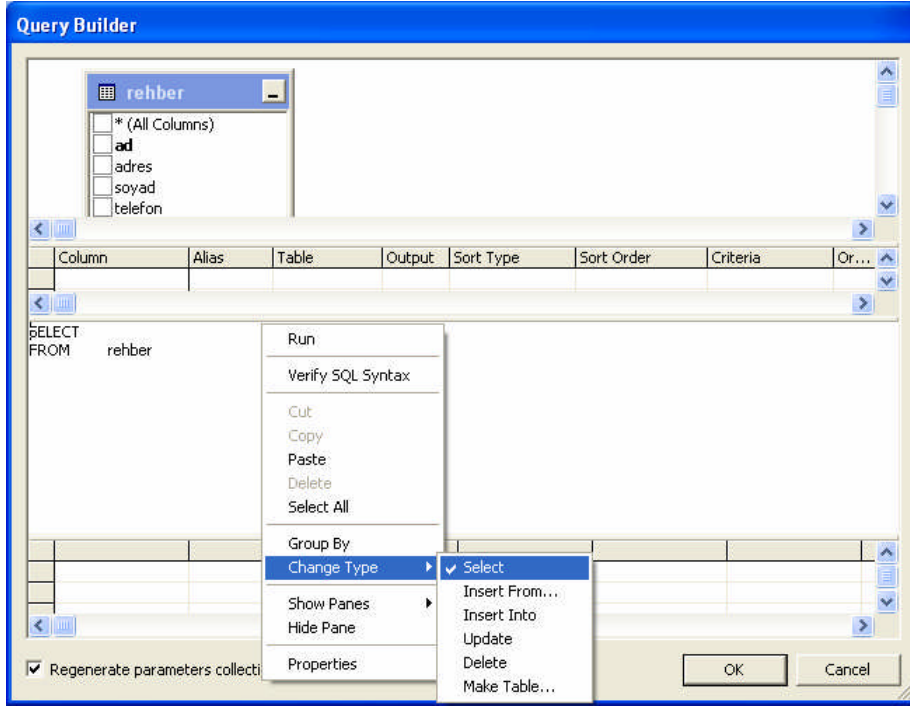
Command nesnesiyle SQL kodunu Properties penceresinde doğrudan CommandText özelliğine aktarabilir veya bu özelliğe ait düğmeyi tıklatarak Query Builder penceresini açabilirsiniz.

Query Builder çalıştırıldığında ekrana Add Table iletişim kutusu gelir ve tablo seçiminin yapılması istenir.



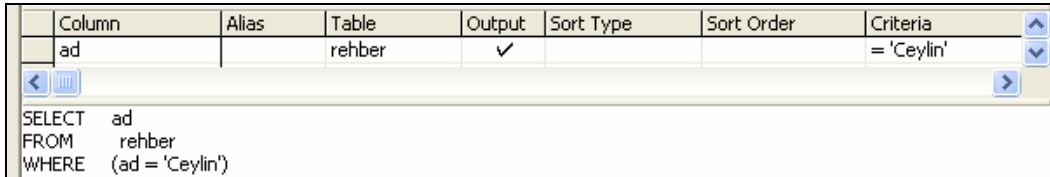
Resim 1.37: Query Builder penceresi

Query Builder başlangıçta SELECT deyimini içeren SQL kodu hazırlanmak istendiğini varsayar. İçinde INSERT INTO veya DELETE deyimleri olan SQL kodu kullanmak istenirse sağ tıkla açılan menüden sorgu tipi seçilebilir.



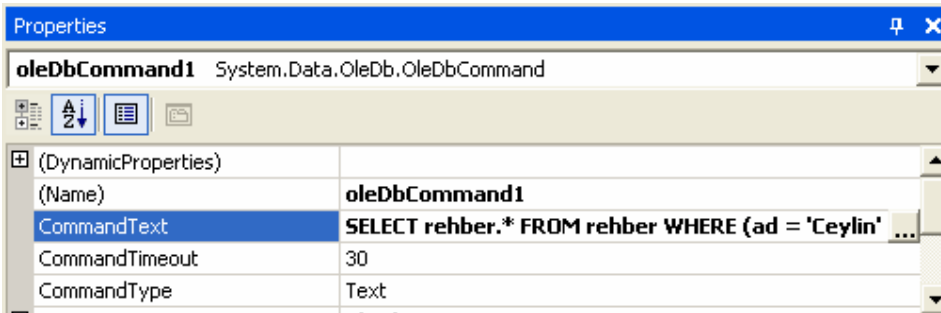
Resim 1.38: Sorgu tipinin belirlenmesi

Sadece belirli kayıtları okutmak istediğinizde WHERE deyimini SELECT deyiminin yanına yazmak yerine Criteria sütununa kriteri yazabilirsiniz. Örneğin, ad='Ceylin'



Resim 1.39: Kriter belirleme

OK düğmesine tıkladığınızda hazırlanan SQL kodu OleDbCommand nesnesinin CommandText özelliğine aktarılır.



Resim 1.40 : SQL kodunun CommandText'e aktarılması

SQL kodu yazılan Command nesnesi işletildiğinde ilgili tablonun belirtilen kritere uyan kayıtlarını geriye gönderir. Geri gönderilen satırları DataSet veya DataReader nesnesine aktarabilirsiniz.

Şu ana kadar yapılan işlemlerin sonuç ekranı DataReader konusunun örneğinde yer almaktadır.

1.9. DataReader Sınıfları

Veri okumak için kullanılırlar. Veri tabanı içerisindeki kayıtları herhangi bir tabloya aktarmadan direkt olarak okuyan sınıflardır. Bu yolla nesnelere okumak daha hızlı ve performanslıdır. Bu sınıflar verileri sadece ileriye doğru okurlar. Bu sınıflar aşağıdaki gibidir.

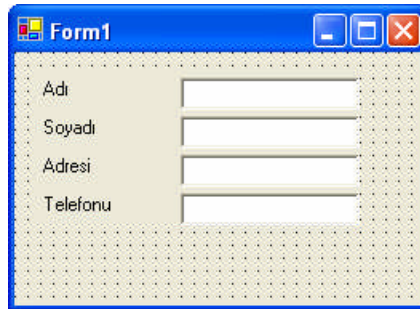
- **OleDbDataReader**
- **OdbcDataReader**
- **SqlDataReader**
- **OracleDataReader**

1.9.1. OleDbDataReader Sınıfı

OleDbConnection gibi bu sınıf da miras alınamaz ve sınıfın alt sınıfı olamaz. Mevcut bilgilerde değişiklik yapılmayacaksa OleDbDataReader nesnesi kullanılır. OleDbDataReader kullanımını, Command sınıfı örneğinin devamı olacak şekilde açıklayalım.

Örnek

Çalışmakta olduğumuz “tel_rehber.mdb” veri tabanındaki “rehber” tablosunun verilerini form üzerinde görüntüleyecek şekilde formunuza dört adet label nesnesi ve dört adet textBox nesnesi yerleştiriniz.

A screenshot of a Windows form titled "Form1". The form has a blue title bar with standard Windows window controls (minimize, maximize, close). The main area of the form is light gray with a dotted grid background. On the left side, there are four labels: "Adı", "Soyadı", "Adresi", and "Telefonu". To the right of each label is a white text box with a thin border, intended for data entry.

Resim 1.41: Formun tasarlanması

Formu tasarladıktan sonra, verilerin text kutusuna getirilmesi için formunuzun Load() metoduna aşağıdaki kodları yazınız.

```

private void Form1_Load(object sender, System.EventArgs e)
{
    OleDbDataReader okuyucu;
    OleDbCommand1.CommandText="SELECT * FROM rehber";
    OleDbConnection1.Open();
    okuyucu=OleDbCommand1.ExecuteReader();
    while(okuyucu.Read())
    {
        textBox1.Text=okuyucu.GetString(0);
        textBox2.Text=okuyucu.GetString(1);
        textBox3.Text=okuyucu.GetString(2);
        textBox4.Text=okuyucu.GetString(3);
    }
}

```

Projenizin en başına, OleDbConnection nesnesini kullanabilmek için **“using System.Data.OleDb;”** yazmayı unutmayınız.

Form1_Load() metodunda ilk önce OleDbDataReader sınıfından bir nesne oluşturmuştur.

```
OleDbDataReader okuyucu;
```

Forma eklediğimiz Command nesnesinin CommandText özelliğine de SQL kodu yazılmıştır.

```
OleDbCommand1.CommandText="SELECT * FROM rehber";
```

Eklenen OleDbConnection nesnesini Open() deyimi ile açılarak bağlantı sağlanmıştır.

```
OleDbConnection1.Open();
```

OleDbCommand1 nesnesinin ExecuteReader() metoduyla tablodaki tüm bilgilere erişilip bu bilgiler OleDbDataReader sınıfından oluşturulan okuyucu nesnesine aktarılmıştır. ExecuteReader() metodu Command nesnesiyle oluşturulan komutu çalıştırıp geriye dönen sonuçları okumak için kullanılır.

```
okuyucu=OleDbCommand1.ExecuteReader();
```

while döngüsü oluşturularak tablodaki tüm bilgilere erişim sağlanıp forma eklediğiniz text kutularına bilgilerin yazılması sağlanmıştır. Bilgilere erişimde kullanılan GetString() metodu, DataReader nesnesinden string tipteki bilgileri okumak için kullanılır. DataReader nesnesinden istenilen tipteki bilgileri okumak için GetChar(), GetBoolean() ve GetInt32() metotlarını da kullanabilirsiniz.

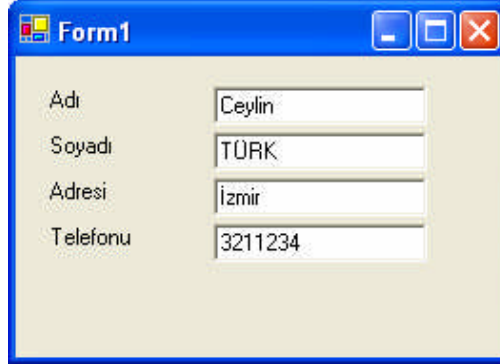
```

while(okuyucu.Read())
{
    textBox1.Text=okuyucu.GetString(0);
    textBox2.Text=okuyucu.GetString(1);
}

```

```
textBox3.Text=okuyucu.GetString(2);  
textBox4.Text=okuyucu.GetString(3);  
}
```

Tüm bu işlemlerin ardından uygulamanızı çalıştırdığınızda tablodaki kayıt size gösterilecektir.

A screenshot of a Windows application window titled "Form1". The window has a standard Windows title bar with minimize, maximize, and close buttons. The main area of the form is light beige and contains four text boxes arranged vertically. The first text box is labeled "Adı" and contains the text "Ceylin". The second text box is labeled "Soyadı" and contains the text "TÜRK". The third text box is labeled "Adresi" and contains the text "İzmir". The fourth text box is labeled "Telefonu" and contains the text "3211234".

Resim 1.42: Kaydın görüntülenmesi

Formda son kayıt gösterilmiştir. Tablodaki diğer kayıtları da görebilmek için daha önce anlatılmış olan veriler arasında hareketi sağlayan `DataBindings ()` metodunu kullanabilirsiniz. Bunun için formunuza komut düğmeleri ekleyip, `Click()` metodlarını uygun şekilde yazmanız gerekir (Resim 1.28).

Ayrıca, `CommandText` özelliğine yazılan SQL koduna `WHERE` komutunu ekleyerek şartınıza bağlı olan veriyi görüntüleyebilirsiniz.

```
oleDbCommand1.CommandText="SELECT * FROM rehber WHERE  
ad='Ceylin'";
```

Bu şekildeki yazımla adı Ceylin olan kayıt text kutularında size gösterilecektir.

Örnek

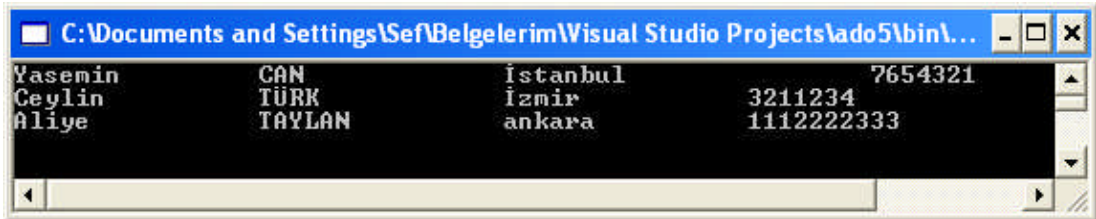
`oleDbCommand` ve `OleDbDataReader` nesnesinin kullanımını bir konsol uygulaması ile göstereyim. Yine aynı veri tabanındaki bilgileri kullanarak konsola yazdıralım.

```

using System;
using System.Data;
using System.Data.OleDb;
namespace ado5
{
    class Program
    {
        static void Main(string[] args)
        {
            OleDbConnection baglanti = new OleDbConnection(@"Provider=
                Microsoft.Jet.OLEDB.4.0;Data Source=C:\tel_rehber.mdb");
            OleDbCommand komut=new OleDbCommand();
            komut.Connection=baglanti;
            komut.CommandText = "SELECT * FROM rehber";
            baglanti.Open();
            OleDbDataReader oku;
            oku=komut.ExecuteReader();
            while (oku.Read())
            {
                Console.Write(oku["ad"]+"\t\t");
                Console.Write(oku["soyad"]+"\t\t");
                Console.Write(oku["adres"]+"\t\t");
                Console.Write(oku["telefon"]+"\t\t");
                Console.WriteLine();
            }
            baglanti.Close();
            Console.Read();
        }
    }
}

```

Uygulamayı çalıştırdığınızda aşağıdaki ekran görüntüsünü elde edeceksiniz.



Resim 1.43: Konsol uygulamasının ekran görüntüsü

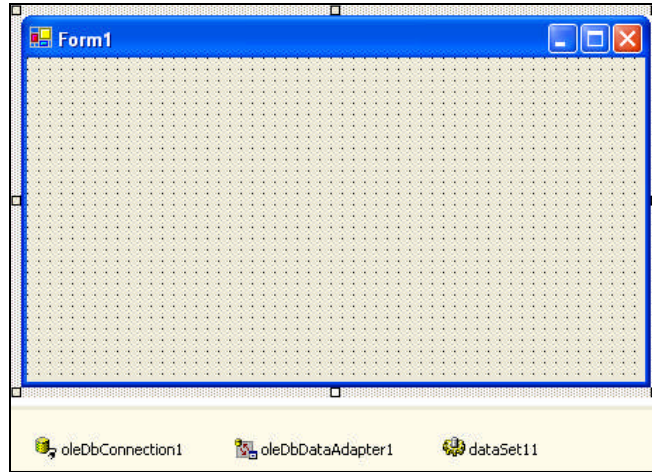
1.10. Veri Görüntüleme (DataView) Nesnesi

Veri tabanındaki bilgileri görüntülemek için kullanılırlar. Microsoft Access programındaki sorgular ile benzer işleyişe sahip olan DataView sayesinde belirtilen tablo ya da tablolardan belli bilgilere kolaylıkla ulaşılabilir. Bu sayede DataSet'te bulunan tabloları (DataTable) sıralayabilir, sorgulayabilir ve kayıt arayabilirsiniz.

DataView kullanımını bir örnekle açıklayalım.

Örnek

- DataView kullanımı için yeni bir WindowsApplication oluşturunuz.
- Oluşturduğunuz WindowsApplication'a ToolBox'taki Data sekmesinden bir OleDbConneciton nesnesi oluşturunuz ve ConnectionString'ini hazırlayınız.
- Daha sonra bir OleDbDataAdapter nesnesi ekleyiniz.
- Veri tabanı olarak yine en başta oluşturduğunuz tel_rehber.mdb veri tabanını kullanınız.
- OleDbDataAdapter nesnesini de ekledikten sonra Properties penceresindeki Generate DataSet linkini tıklayarak DataSet'i hazırlayınız.

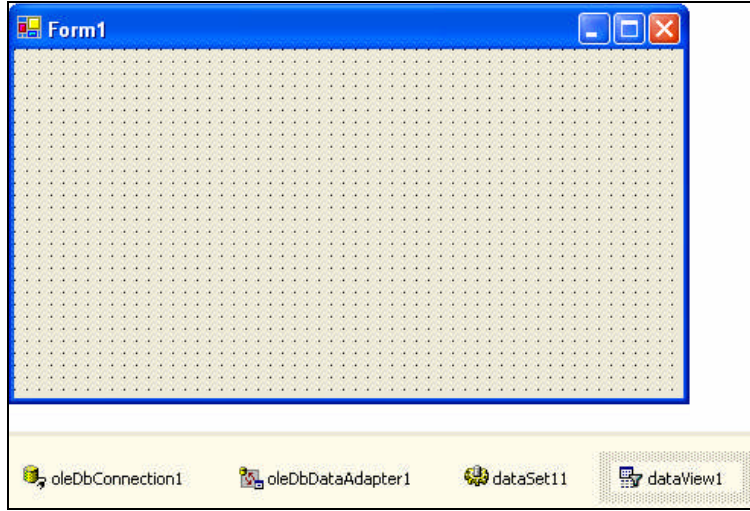


Resim 1.44: Formun hazırlanması

- Proje çalıştırıldığında OleDbDataAdapter ile erişilen tablonun içeriğinin DataSet'e aktarılması için formun Load() metodunu aşağıdaki gibi düzenleyiniz.

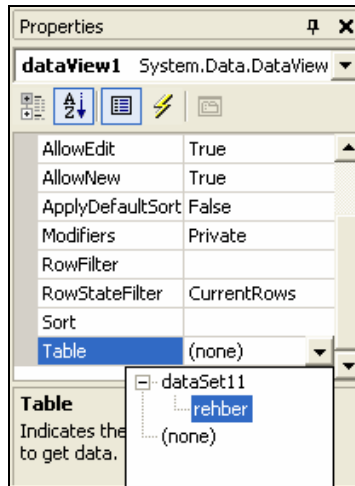
```
private void Form1_Load(object sender, System.EventArgs e)
{
    OleDbConnection1.Open();
    OleDbDataAdapter1.Fill(dataSet11, "rehber");
}
```

- ToolBox'taki Data sekmesinden DataView nesnesini forma yerleştiriniz.



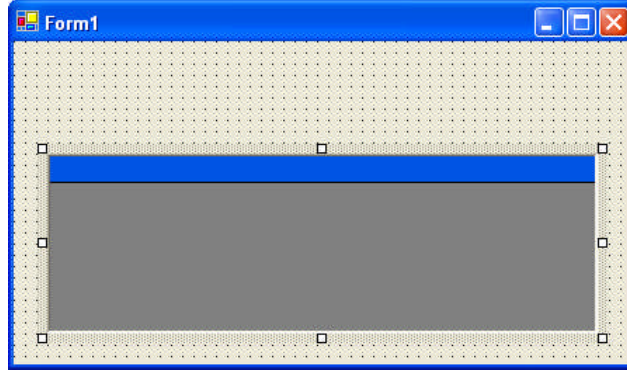
Resim 1.45: DataView nesnesinin forma yerleştirilmesi

- DataView nesnesinin Properties penceresindeki Table özelliğiyle DataTable nesnesinin seçimini yapınız.
-



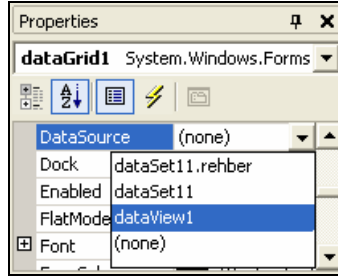
Resim 1.46: DataTable nesnesinin seçilmesi

- Table özelliğinden DataTable nesnesini ekleyerek DataView ile ilgili tablodaki bilgileri elde etmiş olursunuz.
- DataView nesnesinin tablodan elde ettiği bilgileri görmek için formunuza Toolbox'taki Windows Forms sekmesinden bir DataGridView ekleyiniz.



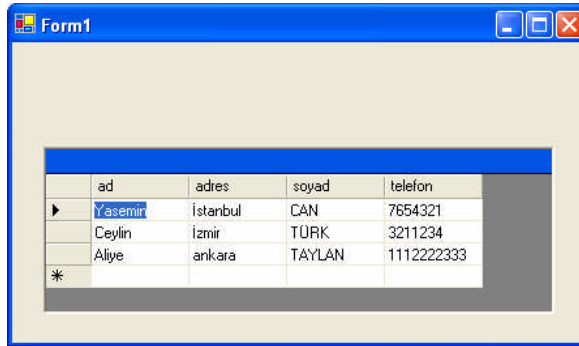
Resim 1.47: Forma DataGrid nesnesinin eklenmesi

- DataGrid nesnesinin DataSource özelliğine DataView nesnesinin adını aktarınız.



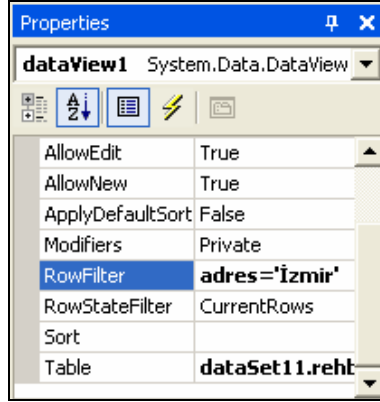
Resim 1.48: DataGrid'e DataView'in aktarılması

- Projenizi bu haliyle çalıştırdığımızda ekran görüntüsü Resim 1.49'daki gibi olacaktır.



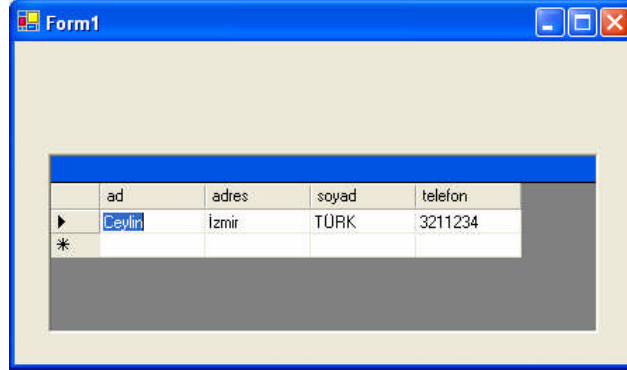
Resim 1.49: Formun çalıştırılmış hali

- Ancak DataGridView'de sadece belirttiğiniz bir alana göre listeleme yapmak için DataView nesnesinin RowFilter özelliğine bir sorgu yazmalısınız. Örneğin, sadece adresi İzmir olanları DataGridView'de görüntülemek için RowFilter alanına adres='İzmir' yazınız.



Resim 1.50: RowFilter alanına sorgu yazılması

- Projenizi bu haliyle çalıştırdığımızda adres alanı sadece İzmir olan kayıtlar DataGridView'de listelenecektir.



Resim 1.51: RowFilter özelliğinin kullanılması

Örneği biraz geliştirelim. Form üzerinde bir açılır liste kutusu (ComboBox) olsun ve bu listede bulunan adreslere göre bir filtreleme yapılsın.

- RowFilter alanına yazdığımız sorguyu siliniz ve formunuza bir ComboBox yerleştiriniz.



Resim 1.52: Forma ComboBox'ın yerleştirilmesi

- ComboBox'ın Items özelliğine birkaç tane adres yazınız ve OK düğmesine tıklayınız.

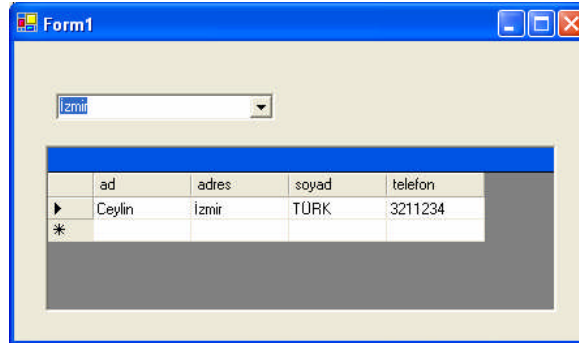


Resim 1.53: ComboBox'ın Items özelliğine elemanların girilmesi

- ComboBox'ın Text özelliğini de “Adres seçiniz...” yazın.
- Proje çalıştırılıp ComboBox'tan bir adres seçimi yapıldığında seçilen adresin bulunduğu kayıttın DataGridView'de gösterilmesi için ComboBox'ın SelectedIndexChanged() metodunun düzenlenmesi gereklidir. Bunun için ComboBox üzerinde fareyle çift tıklayınız. Açılan kod sayfasına aşağıdaki kodları yazınız.

```
private void comboBox1_SelectedIndexChanged(object sender, System.EventArgs e)
{
    string adr;
    adr=comboBox1.Text;
    adr="adres =" + "'" + adr + "'";
    dataGridView1.RowFilter=adr;
}
```

- Projenizi çalıştırıp ComboBox'tan bir seçim yaptığınızda DataGridView nesnesinde yaptığınız seçim size gösterilecektir. Eğer yaptığınız seçimle ilgili bir bilgi tabloda yoksa DataGridView'in içeriği boş olarak geriye döndürülecektir.



Resim 1.54: Seçime göre DataGridView'de listeleme

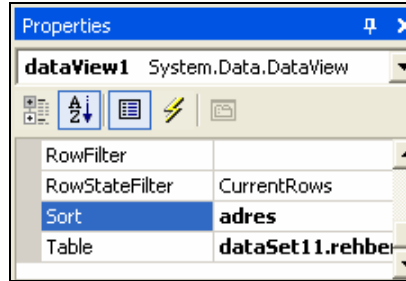
1.10.1. Sıralama (Sort)

Veri tabanındaki bilgileri DataView ile görüntülerken sıralanmış olarak da listeleyebilirsiniz. Bunun için DataTable'daki alan adlarını kullanmanız gerekecektir.

Bilgileri sıralamanın iki yöntemi vardır. Birinci yöntem; DataView1'in Sort özelliğini kullanmak, ikinci yöntem ise kod satırları yazarak DataView'in sıralama yapmasını sağlamaktır.

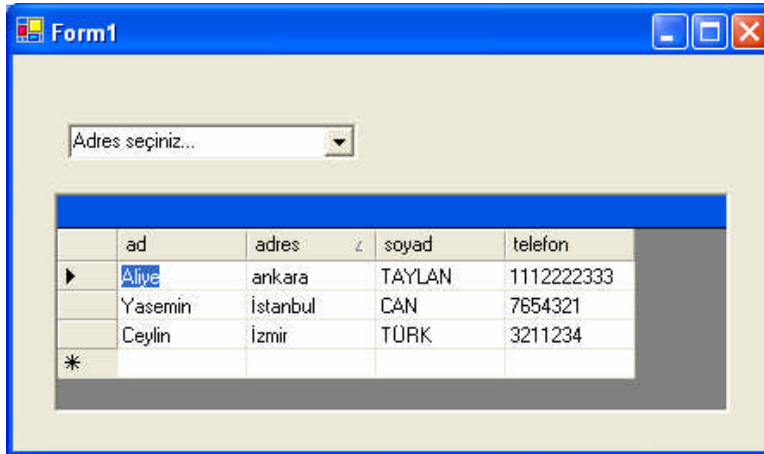
□ Sort Özelliğini Kullanmak

DataView'in Sort özelliğine tabloda bulunan ve sıralama yapmak istediğiniz alanın adını yazmak suretiyle sıralama yapabilirsiniz. Sıralama A'dan Z'ye gerçekleşecektir.



Resim 1.55: Sort özelliğine tablodaki alan adının girilmesi

Uygulama çalıştırıldığında bilgilerinizin “adres” alanına göre sıralandığını göreceksiniz.



Resim 1.56: Bilgilerin sıralanmış hali

□ Kod Yazarak

DataView nesnesinin Sort özelliği kullanılarak bilgilerin sıralanmasını sağlayabilirsiniz. Sıralama işlemi hangi metotta yapacaksanız kod satırlarını o metodun içerisine yazmanız gereklidir. Örnek uygulamamızın sıralamasını yapacak olursak sıralamanın projenin başlatıldığı anda yapılmasını varsayalım. Bunun için, kod satırlarını formun Load() metoduna yazmak gerekir.

```
private void Form1_Load(object sender, System.EventArgs e)
{
    OleDbConnection1.Open();
    OleDbDataAdapter1.Fill(dataSet11, "rehber");
    dataGridView1.Sort="adres";
}
```

Projenizi çalıştırdığınız anda bilgilerinizin sıralanmış olduğunu göreceksiniz.

1.10.2. DataView'de Kayıt Aramak

Veri tabanındaki bilgilerden istediğiniz bir kayda ulaşabilirsiniz. Bunun için kaydı aratmanız gerekecektir. DataView ile arama yapmak istiyorsanız bu nesnenin Find() metodunu kullanmanız gerekecektir. Find() metodu, aranan kaydı DataView nesnesinde bulunamazsa geriye -1 değerini döndürür.

Find() metoduyla hangi alana göre arama yapılacaksa o alanı Sort() metoduyla önceden sıralamanız gerekir.

Bu işlem için formunuzu aşağıdaki gibi düzenleyiniz.

ad	adres	soyad	telefon
*			

Resim 1.57: Formun düzenlenmiş hali

Form'a, arama yapmak için bir Ara düğmesi, aranacak bilgiyi girebilmeniz için de bir Text kutusu ekleyiniz. Arama işlemi sonucunda bulunan bilgilerin gösterilmesi için yine dört adet Text kutusunu formunuza ekleyiniz.

Arama işlemini gerçekleştirecek olan “Ara” düğmesinin kod satırlarını aşağıdaki gibi düzenleyiniz.

```
private void button1_Click(object sender, System.EventArgs e)
{
    string ara;
    ara=textBox5.Text;
    dataView1.Sort="ad";
    int kayit_numara;
    kayit_numara=dataView1.Find(ara);
    DataRowView kayit;
    if (kayit_numara!=-1)
    {
        kayit=dataView1[kayit_numara];
        textBox1.Text=kayit["ad"].ToString();
        textBox2.Text=kayit["soyad"].ToString();
        textBox3.Text=kayit["adres"].ToString();
        textBox4.Text=kayit["telefon"].ToString();
    }
    else
    {
        MessageBox.Show("Aradığınız kayıt bulunamadı..");
    }
}
```

Aranacak bilgi formdaki beşinci text kutusuna girilmiştir ve girilen bilgi string tipteki “ara” değişkenine aktarılmıştır.

```
string ara;
ara=textBox5.Text;
```

Arama yapılacak alan Sort() metoduyla sıralanmalıdır. Aramayı “ad” alanına göre yapacağımız için DataView’in Sort özelliğine “ad” alanı yazılmıştır.

```
dataView1.Sort="ad";
```

Find() metodu aranacak bilgi olan “ara” değişkenine göre yapılacağı için “ara” değişkeni Find() metoduna parametre olarak verilmiştir. Geriye dönen değer yani kaydın sıra numarası int tipindeki “kayit_numara” değişkenine aktarılmıştır.

```
int kayit_numara;
kayit_numara=dataView1.Find(ara);
```

Kaydın içeriğini text kutularında göstermek için DataRowView tipinde “kayit” değişkeni tanımlanmıştır.

```
DataRowView kayıt;
```

Kayıt bulunamadığı zaman DataView'in geriye -1 değerini döndürdüğünü söylemiştik. DataView'in -1'den farklı olduğu yani kaydın bulunduğu anda kayıtların gösterilebilmesi için if deyimiyle bir koşul oluşturulmuştur. Kayıt bulunamadığı zaman ise "Aradığınız kayıt bulunamadı" mesajı verdirilmiştir. DataView nesnesinin kayıt numarası, kayıtların gösterilebilmesi için DataRowView tipindeki kayıt değişkenine aktarılmıştır.

```
if (kayıt_numara!=-1)
{
    kayıt=dataView1[kayıt_numara];
    textBox1.Text=kayıt["ad"].ToString();
    textBox2.Text=kayıt["soyad"].ToString();
    textBox3.Text=kayıt["adres"].ToString();
    textBox4.Text=kayıt["telefon"].ToString();
}
else
{
    MessageBox.Show("Aradığınız kayıt bulunamadı..");
}
```

Aranacak bilgiyi girip Ara düğmesine tıkladığınızda bilgiler text kutularına yerleşecektir.

ad	adres	soyad	telefon
Aliye	ankara	TAYLAN	1112222333
Ceylin	izmir	TÜRK	3211234
Yasemin	İstanbul	CAN	7654321

Resim 1.58: Kaydın bulunması

Eğer bulunamazsa mesaj kutusuyla ilgili mesaj ekrana gelecektir.

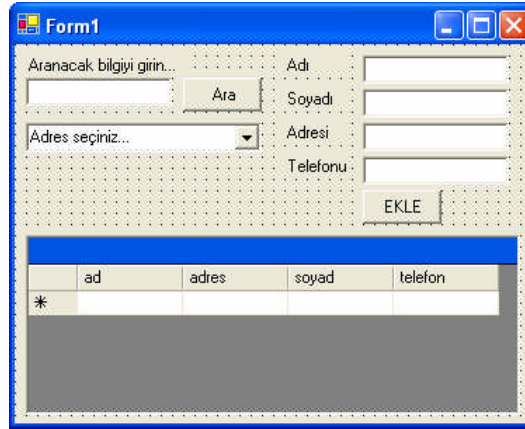


Resim 1.59: Kaydın bulunamaması

1.10.3. DataView'e Kayıt Ekleme

DataView'e yeni bir kayıt eklemek için DataView'in AddNew() metodunu kullanmak gerekir. Ekleme işlemi yapmadan önce DataView nesnesinin AllowNew özelliğinin True olmasına dikkat etmelisiniz.

DataView'e kayıt eklemek için formunuza bir komut düğmesi ekleyiniz.



Resim 1.60: Form'a EKLE düğmesinin yerleştirilmesi

EKLE düğmesine tıkladığımızda text kutularına girdiğimiz bilgiler DataView nesnesine eklenmiş olacaktır. Bunun için EKLE düğmesi üzerinde fareyle çift tıklayarak Click() metodunu aşağıdaki gibi düzenleyiniz.

```
private void button2_Click(object sender, System.EventArgs e)
{
    DataRowView yeni_kayit;
    yeni_kayit=dataView1.AddNew();
    yeni_kayit["ad"]=textBox1.Text;
    yeni_kayit["soyad"]=textBox2.Text;
    yeni_kayit["adres"]=textBox3.Text;
    yeni_kayit["telefon"]=textBox4.Text;
    yeni_kayit.EndEdit();
}
```

EKLE düğmesinin Click() metodunda DataRowView ile “yeni_kayit” adında bir değişken tanımlanmış ve AddNew() metoduyla DataView’in özellikleri bu değişkene aktarılmıştır.

```
DataRowView yeni_kayit;  
yeni_kayit=dataView1.AddNew();
```

Text kutularına girilen bilgiler ilgili alanlara aktarılmıştır.

```
yeni_kayit["ad"]=textBox1.Text;  
yeni_kayit["soyad"]=textBox2.Text;  
yeni_kayit["adres"]=textBox3.Text;  
yeni_kayit["telefon"]=textBox4.Text;
```

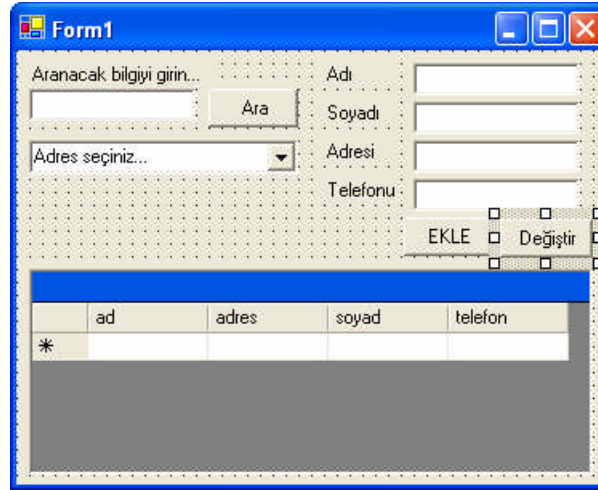
Ekleme işlemini bitirmek içinde EndEdit() metodu kullanılmıştır.

```
yeni_kayit.EndEdit();
```

1.10.4. DataView'deki Kaydı Değiştirmek

DataView nesnesinde bulunan kayıtları değiştirmek için ilk önce kayıtların değiştirilmeye hazır hale gelmeleri gerekmektedir. Değiştirme işlemine başlamak için BeginEdit() metodu kullanılır. Değiştirme işleminin ardından yine EndEdit() metoduyla değiştirme işlemi sonlandırılır. Kaydı değiştirebilmek için DataView nesnesinin AllowEdit özelliğinin True olmasına dikkat ediniz.

Değiştirme işlemi için formunuza bir komut düğmesi ekleyiniz ve Text özelliğine “Değiştir” yazınız.



Resim 1.61: “Değiştir” düğmesinin eklenmesi

Değiştirilecek kaydın kayıt numarasının bilinmesi gerekir. Arama işlemi yapılırken aktif kaydın kayıt numarası “kayit_numara” değişkeninde tutulmaktadır. Uygulamada, önce kayıt aranıp sonra değiştirme işlemi yapılacağından, “kayit_numara” değişkeninin global olarak tanımlanması gerekir. Yani, arama işleminin yapıldığı metodun içerisinde bulunun “kayit_numara” değişkeninin Click() metodunun dışına yazılması gereklidir.

```
int kayit_numara;  
private void button1_Click(  
{  
    string ara;  
    ara=textBox5.Text;  
    dataView1.Sort="ad";
```

Bu işlemin ardından “Değiştir” düğmesinin Click() metodunu aşağıdaki gibi düzenleyiniz.

```
private void button3_Click(object sender, System.EventArgs e)  
{  
    DataRowView kayit;  
    kayit=dataView1[kayit_numara];  
    kayit.BeginEdit();  
    kayit["ad"]=textBox1.Text;  
    kayit["soyad"]=textBox2.Text;  
    kayit["adres"]=textBox3.Text;  
    kayit["telefon"]=textBox4.Text;  
    kayit.EndEdit();  
}
```

Aktif kaydın numarası “kayit_numara” değişkeniyle DataRowView tipindeki “kayit” değişkenine aktarılmış ve değiştirme işlemi BeginEdit() metoduyla başlatılmıştır.

```
DataRowView kayit;  
kayit=dataView1[kayit_numara];  
kayit.BeginEdit();
```

Text kutularındaki değişiklikler ilgili alanlara aktarılıp değiştirme işlemi EndEdit() metoduyla sonlandırılmıştır.

```
kayit["ad"]=textBox1.Text;  
kayit["soyad"]=textBox2.Text;  
kayit["adres"]=textBox3.Text;  
kayit["telefon"]=textBox4.Text;  
kayit.EndEdit();
```

Kayıtların değiştirilmeye başlamadan önceki ve değiştirilme işlemi yapıldıktan sonraki ekran görüntüleri aşağıda gösterilmiştir.

Form1

Aranacak bilgiyi girin... Adı Ceylin
Ceylin Ara Soyadı TÜRK
Adres seçiniz... Adresi İzmir
Telefonu 3211234
EKLE Değiştir

ad	adres	soyad	telefon
▶ Aliye	ankara	TAYLAN	1112222333
Ceylin	İzmir	TÜRK	3211234
Yasemin	İstanbul	CAN	7654321
*			

Form1

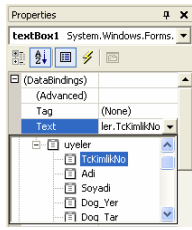
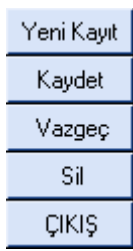
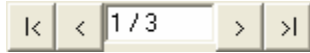
Aranacak bilgiyi girin... Adı Ceylin
Ceylin Ara Soyadı YILMAZ
Adres seçiniz... Adresi Kırşehir
Telefonu 3211234
EKLE Değiştir

ad	adres	soyad	telefon
▶ Aliye	ankara	TAYLAN	1112222333
Ceylin	Kırşehir	YILMAZ	3211234
Yasemin	İstanbul	CAN	7654321
*			

Resim 1.62: Değiştirme işlemi öncesi ve sonrasının ekran görüntüsü

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p><input type="checkbox"/> Bir yardımlaşma derneğinin üye bilgileri ile ilgili bir uygulama yapılacaktır. Bu programda üyelerin T.C. Kimlik Nu, Adı, Soyadı, Doğum Yeri, Doğum Tarihi, Adresi, Ev Telefonu, İş Telefonu, Cep Telefonu, Kan Grubu, Medeni Hali, Eğitim Durumu, Mesleği ve Çocuk Sayısı bilgileriyle ilgili işlemler yapılacaktır.</p>	<p><input type="checkbox"/> Microsoft Access programını kullanarak bir tablo oluşturunuz. Alan adlarını oluşturduktan sonra birincil anahtar olarak T.C.Kimlik Nu alanını seçiniz.</p> <p><input type="checkbox"/> Tabloya birkaç kayıt giriniz.</p> <p><input type="checkbox"/> Alan adları olarak aşağıda verilenleri kullanabilirsiniz.</p> <p><input type="checkbox"/> TCKimlikNu---Metin---11, Adi---Metin---15, Soyadi---Metin---20, Dog_Yer---Metin---15, Dog_Tar---Metin---10, Adres---Metin---50, Ev_Tel---Metin---13, Is_Tel---Metin---13, Cep_Tel---Metin---13, Kan_Grubu---Metin---8, Medeni_Hali---Metin---5, Egitim_Durumu---Metin---11, Meslek---Metin---20, Cocuk_Sayisi---Metin---1</p>
<p><input type="checkbox"/> Formunuza bir OleDbConnection nesnesi ekleyiniz ve veri tabanı ile bağlantısını yapınız.</p>	<p><input type="checkbox"/> Server Explorer penceresinden bu işlemi yapabilirsiniz.</p>
<p><input type="checkbox"/> Formunuza bir OleDbDataAdapter nesnesi ekleyiniz ve veri tabanından kullanacağınız alanları seçiniz.</p>	<p><input type="checkbox"/> Sihirbaz yardımıyla bu işlemi gerçekleştirebilirsiniz.</p>
<p><input type="checkbox"/> DataSet'i oluşturunuz.</p>	<p><input type="checkbox"/> OleDbDataAdapter'in özelliklerinden Generate DataSet linkine tıklayınız..</p>
<p><input type="checkbox"/> Formunuza 16 adet Label ve TextBox nesnesi ekleyiniz.</p>	<p><input type="checkbox"/> ToolBox'taki WindowsForms sekmesinden bu nesnelere seçip form üzerine yerleştirebilirsiniz. Ayrıca bir nesneyi kopyalayıp yapıştırarak zaman kazanabilirsiniz.</p>
<p><input type="checkbox"/> Label nesnelerinin Text özelliklerini programda kullanılacağı duruma uygun olarak değiştiriniz.</p>	<p><input type="checkbox"/> Label1 nesnesinin Text özelliğine "TC Kimlik Numarası" yazabilirsiniz.</p>

<p><input type="checkbox"/> TextBox nesnelerinin DataBindings bağlantılarını yapınız.</p>	<p><input type="checkbox"/> TextBox nesnesini seçin ve Properties penceresinden DataBindings özelliğinin Text seçeneğinden yapabilirsiniz.</p>  <p>Resim 1.63: DataBindings bağlantısı</p>
<p><input type="checkbox"/> Formunuza veri tabanına kayıt girmek ve mevcut kayıtlar üzerinde işlem yapmak için 5 adet Button nesnesi ekleyiniz.</p>	<p><input type="checkbox"/> Button nesnelerinin Text özelliklerine Yeni kayıt, Kaydet, Vazgeç, Sil, ÇIKIŞ yazabilirsiniz.</p>  <p>Resim 1.64: Button nesneleri</p>
<p><input type="checkbox"/> Formunuza veriler arasında dolaşabilmek için 4 adet Button ve 1 adet TextBox nesnesi yerleştiriniz.</p>	 <p>Resim 1.65: Kayıtlar arasında dolaşabilmek için gerekli düğmeler</p>
<p><input type="checkbox"/> Tüm verileri de form üzerinde görebilmek için bir DataGrid nesnesini formunuzun alt kısmına yerleştiriniz.</p>	<p><input type="checkbox"/> DataGrid nesnesini ToolBox'taki WindowsForms sekmesinden seçip form üzerine yerleştirebilirsiniz.</p>
<p><input type="checkbox"/> DataGrid nesnesinin DataSource'unu belirtiniz.</p>	<p><input type="checkbox"/> DataSource özelliğine tabloyu ekleyiniz.</p>
<p><input type="checkbox"/> Formun Load() metoduna OleDbConnection nesnesinin bağlantısını açan kodu yazınız.</p>	<p><input type="checkbox"/> Open() metodunu kullanabilirsiniz.</p>
<p><input type="checkbox"/> Yine Load() metoduna Dataset'teki tablonun OleDbDataAdapter'e doldurunuz.</p>	<p><input type="checkbox"/> OleDbDataAdapter1.Fill(dataSet11,"uyeler");</p>
<p><input type="checkbox"/> Veriler arasında dolaşabilmek</p>	<p><input type="checkbox"/> Bir önceki kayıt için pozisyonu bir azaltabilirsiniz.</p>

<p> için kullanılacak düğmelerin kodlarını düğmelerin üzerine çift tıklayarak yazınız.</p>	<p><input type="checkbox"/> Bir sonraki kayıt için pozisyonu bir artırabilirsiniz.</p> <p><input type="checkbox"/> İlk kayıt için pozisyonu sıfırlayabilirsiniz.</p> <p><input type="checkbox"/> Son kayıt için <code>this.BindingContext[dataSet11,"uyeler"].Position=this.dataSet11.Tables["uyeler"].Rows.Count-1;</code></p>
<p><input type="checkbox"/> Aktif kaydın ve tablodaki toplam kaydın gösterilmesi için <code>goster()</code> adından bir metod oluşturun ve metodun içerisine ilgili kodları yazınız.</p>	<p><input type="checkbox"/> <code>void goster()</code></p> <pre>{ }</pre>
<p><input type="checkbox"/> Bu metodun kayıtlar arasında dolaşmak için yazılan metodların içine ve <code>Load()</code> metoduna yerleştiriniz.</p>	<p><input type="checkbox"/> <code>goster();</code></p>
<p><input type="checkbox"/> Yeni Kayıt düğmesine tıkladığında imlecin birinci text kutusuna yönelmesini sağlayınız ve ilgili metodu kullanınız.</p>	<p><input type="checkbox"/> <code>Focus()</code> metodunu kullanabilirsiniz. <code>AddNew()</code> metodunu unutmayınız.</p>
<p><input type="checkbox"/> Kaydet düğmesine tıkladığında bilgilerin DataSet'te güncellenmesi sağlayınız. Kayıt yapıldığını bir mesaj kutusuyla kullanıcıya bildiriniz.</p>	<p><input type="checkbox"/> <code>Update()</code> metodunu ve <code>MessageBox()</code> metodunu kullanabilirsiniz.</p>
<p><input type="checkbox"/> Vazgeç düğmesine tıkladığında girilen bilgilerin DataSet'e girilmesini engelleyen kodu yazınız.</p>	<p><input type="checkbox"/> <code>CancelCurrentEdit()</code> metodunu kullanabilirsiniz.</p>
<p><input type="checkbox"/> Sil düğmesine tıkladığında aktif kaydın silinmesini kullanıcının mesaj kutusundan vereceği mesaja göre sağlayınız.</p>	<p><input type="checkbox"/> <code>RemoveAt()</code> metonu kullanabilirsiniz.</p>
<p><input type="checkbox"/> Çıkış düğmesine tıkladığında projenin sonlanması için gerekli kodu yazınız.</p>	<p><input type="checkbox"/> <code>Close()</code> metodunu kullanabilirsiniz.</p>

Projenin ekran görüntüleri ve kodları aşağıda verilmiştir.

Form1

TcKimlik No: 12345678901 Ev Telefonu: 3125555555 Medeni Hali: Bekar
Adı: Alican İş Telefonu: 3125555555 Eğitim Durumu: Lise
Soyadı: CAN Cep Telefonu: 5955555555 Mesleği: Öğretmen
Doğum Yeri: Ankara Kan Grubu: A Rh + Çocuk Sayısı: 1
Doğum Tarihi: 01.01.1980
Adresi: Atatürk Bulv.
Semt: Çankaya İl: Ankara

Yeni Kayıt
Kaydet
Vazgeç
Sil
ÇIKIŞ

1 / 2

TcKimlikNo	Adi	Soyadi	Dog_Yer	Dog_Tar	Adres	Semt	il	Ev_Tel
12345678901	Alican	CAN	Ankara	01.01.1980	Atatürk Bulv.	Çankaya	Ankara	3125555555
987654321	Veli	CANCAN	İstanbul	01.01.1979	Mehmetçik S	Kadıköy	İzmir	216444444

Resim 1.66: Proje başlangıç ekranı

Proje Yöneticisi

Kayıt Yapıldı...

Tamam

Resim 1.67: Kayıt yapıldı mesaj penceresi

Uyarı

Bu Kayıtı Silmek İstediyinize Emin misiniz?

Evet Hayır

Resim 1.68: Kayıt silme mesaj penceresi

□ Program Kodları

```
private void Form1_Load(object sender, System.EventArgs e)
{
    OleDbConnection1.Open();
    OleDbDataAdapter1.Fill(dataSet11,"uyeler");
    goster();
}

int kayno,topkay;
void goster()
{
    kayno=this.BindingContext[dataSet11,"uyeler"].Position+1;
    topkay=this.dataSet11.Tables["uyeler"].Rows.Count;
    textBox12.Text=(kayno).ToString()+ " / " + (topkay).ToString();
}

int pozisyon;
private void button1_Click(object sender, System.EventArgs e)
{
    pozisyon=this.BindingContext[dataSet11,"uyeler"].Position;
    this.BindingContext[dataSet11,"uyeler"].AddNew();
}

private void button2_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"uyeler"].EndCurrentEdit();
    OleDbDataAdapter1.Update(dataSet11);
    MessageBox.Show("Kayıt Yapıldı...", "Proje Yöneticisi");
}

private void button3_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"uyeler"].CancelCurrentEdit();
    this.BindingContext[dataSet11,"uyeler"].Position=pozisyon;
    button7.Focus();
}

private void button5_Click(object sender, System.EventArgs e)
{
    int Kay_no;
    Kay_no=this.BindingContext[dataSet11,"uyeler"].Position;
    DialogResult c;
    c=MessageBox.Show("Bu Kayıtı Silmek İstediginize Emin misiniz?", "Uyarı",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if(c==DialogResult.Yes)
```

```

    {
        this.BindingContext[dataSet11,"uyeler"].RemoveAt(Kay_no);
        this.BindingContext[dataSet11,"uyeler"].Position-=1;
        this.BindingContext[dataSet11,"uyeler"].EndCurrentEdit();
        OleDbDataAdapter1.Update(dataSet11);
    }
}

private void button7_Click(object sender, System.EventArgs e)
{
    Close();
}

private void button9_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"uyeler"].Position-=1;
    goster();
}

private void button10_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"uyeler"].Position=
        this.BindingContext[dataSet11,"uyeler"].Position+1;
    goster();
}

private void button8_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"uyeler"].Position=0;
    goster();
}

private void button11_Click(object sender, System.EventArgs e)
{
    this.BindingContext[dataSet11,"uyeler"].Position=
        this.dataSet11.Tables["uyeler"].Rows.Count-1;
    goster();
}

```

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru/yanlış seçenekli sorularda uygun harfleri yuvarlak içine alınız. Seçenekli sorularda ise uygun şıkkı işaretleyiniz.

1. Veri tabanı işlemleri için kullanılan .NET sınıflarının tümüne denir.
2. Aşağıdaki seçeneklerden hangisi için .NET'te veri sağlayıcısı yoktur?
A) Oecd
B) OleDb
C) Oracle
D) SQL Server
3. Access veri tabanına bağlantı yapabilmek için kullanılan bağlantı sınıfı'dır.
4. Gerçekleşen bağlantıyı açmak içinmetodu, kapatmak içinmetodu kullanılır.
5. Veri tabanına bağlantının simgesel görünümü Solution Explorer penceresinden görülebilir. (D/Y)
6. SQL deyimlerinde kriter belirleme komutuyla gerçekleştirilir.
7. Oluşturulan Access tablosunda birincil anahtar yoksa ve SQL deyimleri oluşturulmaz.
8. DataAdapter ile bilgiler DataSet'e aktarılırken Open() metodundan yararlanır.(D/Y)
9. Bir text kutusunu DataTable'daki bir alana bağlamak için kullanılan özellik aşağıdakilerden hangisidir?
A) Tag
B) Text
C) DataBindings
D) DataView
10. Command nesnesinin CommandType özelliğine TableDirect seçeneği aktarılırsa kod yazılmaya gerek yoktur.(D/Y)

Değerlendirme

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konulara geri dönerek tekrar inceleyiniz. Tüm sorulara doğru cevap verdiğinizde diğer öğrenme faaliyetine geçiniz.

ÖĞRENME FAALİYETİ- 2

AMAÇ

Gerekli ortam sağlandığında, XML ile veri tabanı destekli uygulamalar geliştirebileceksiniz.

ARAŞTIRMA

- XML dosyaların hangi programlama dillerinde kullanıldığını araştırınız.

2. XML DOKÜMANLARI

2.1. XML Nedir

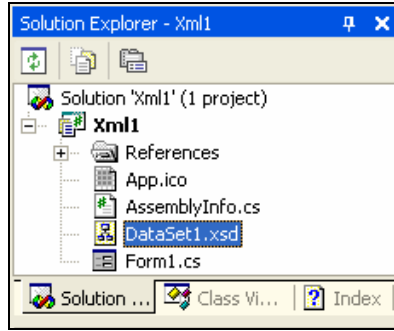
Kişilerin kendi sistemlerini oluşturabilecekleri, kendi etiketlerini tanımlayarak çok daha rahat ve etkin programlama yapabilecekleri ve bu belirlenen etiketleri kendi yapıları içerisinde standardize edebilecekleri esnek, genişleyebilir ve kolay uygulanabilir bir meta dildir.

Çok farklı tipteki verileri orijinal formatlarında tek bir havuzda tutabilen XML (genişletilebilir işaretleme dili), bilgiye hızlı, kolay ve ortamdaki bağımsız olarak erişebilme imkânı sunar. Günlük yaşantımızda kullanmakta olduğumuz verilerin %80'ini oluşturan ve "unstructured" olma özellikleri nedeniyle kendi buldukları medya dışında veri özelliklerini koruyamayan (kelime işlem, elektronik tablo çıktıları, PDF dokümanları, ses, resim vb.) farklı tipteki verilerin, oryantasyona gerek duymadan hiyerarşik bir yapıda kullanılabilmelerine olanak vermekte ve bu verilerin hızlı bir şekilde sorgulanabilmelerini sağlamaktadır. Öncelikle veri transferinin kolaylaşmasını ve verinin içerik bilgisiyle saklanabilmesini hedefleyen XML, içerik ve sunum bilgilerini birbirinden ayırır. Bu özelliği ile de HTML'den farklılaşır. XML için aşağıdakiler söylenebilir.

- XML hem bir teknolojidir hem de bir dildir.
- XML dil olarak markup dil'leri (bir belgedeki verileri işaretlemeye yarayan diller) oluşturmaya yarar.
- XML verileri tanımlamak için kullanılan bir teknolojidir.
- XML verileri tanımlama açısından bir standart oluşturmak için oluşturulmuştur.
- XML verileri standart bir şekilde tanımladığından web'te veya herhangi iki program arasında veri alışverişini kolaylaştırmaktadır.

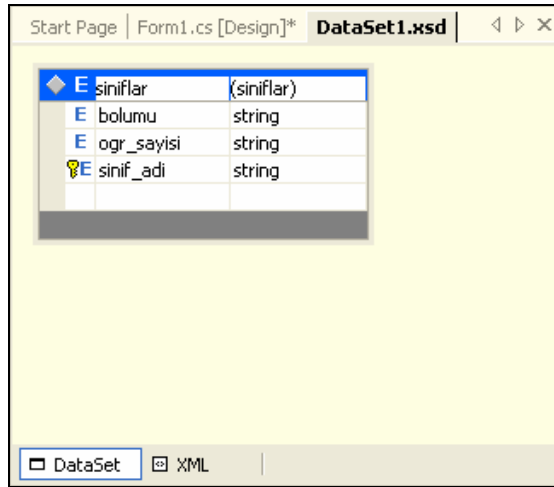
2.2. DataSet ve DataTable Nesneleri

Veri tabanıyla işlem yaparken kullandığımız çeşitli nesnelere vardır. Şu ana kadar anlatılan veri tabanı işlemlerinde Microsoft Access veri tabanı kullandığımız için OleDbConnection, OleDbDataAdapter ve DataSet nesnelerini sıkça kullandık. DataSet'i oluşturduktan sonra Solution Explorer penceresine dikkat ederseniz **xsd** uzantılı bir dosya otomatik olarak oluşturulur.



Resim 2.1: Solution Explorer penceresi

Bu dosya, DataSet hakkında bilgiler içermektedir ve XML dosyasının içeriğini görüntüler. Bu dosyayı açmak için fareyle üzerinde çift tıklattın veya fare sağ tuşuyla açılan menüden Open komutunu verin.



Resim 2.2: DataSet içeriği

Dosya açıldığı zaman iki seçenek karşınıza gelir. Birincisi DataSet, ikincisi de XML seçeneğidir. DataSet seçeneği, DataSet'teki DataTable nesnesinin yapısını göstermektedir. XML seçeneğindeyse, DataTable nesnesinin XML kodlarını görmek mümkündür.

```
Start Page | Form1.cs [Design]* DataSet1.xsd
<?xml version="1.0" standalone="yes"?>
<xs:schema id="DataSet1" targetNamespace="http://www.tempuri.org/DataS
  <xs:element name="DataSet1" msdata:IsDataSet="true" msdata:Locale="t
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="siniflar">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="bolumu" type="xs:string" minOccurs="0"
              <xs:element name="ogr_sayisi" type="xs:string" minOccurs
              <xs:element name="sinif_adi" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
    <xs:unique name="Constraint1" msdata:PrimaryKey="true">
      <xs:selector xpath="//mstns:siniflar" />
      <xs:field xpath="mstns:sinif_adi" />
    </xs:unique>
  </xs:element>
</xs:schema>
```

Resim 2.3: DataSet nesnesinin XML kodları

Bu XML kodları, DataTable nesnesinin kayıt yapısını ve içeriği hakkında bilgileri içermektedir.

2.3. DataTable İçeriğini XML'e Kaydetmek

Veri tabanında oluşturduğumuz tablonun içeriğini XML dosyasına kaydedebilirsiniz. Bunun için aşağıdaki adımları takip edebilirsiniz.

- Formunuza bir OleDbConnection, OleDbDataAdapter ekleyin ve DataSet'i oluşturunuz.
- OleDbConnection ile bağlantı kurduğunuz veri tabanındaki tablo içeriğini DataSet'e aktarınız.

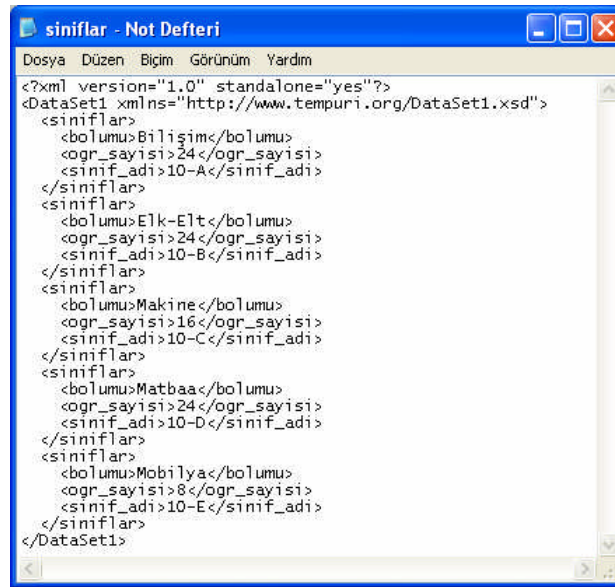
```
private void Form1_Load(object sender, System.EventArgs e)
{
    OleDbDataAdapter1.Fill(dataSet11,"siniflar");
}
```

- Form'a bir komut düğmesi (button) ekleyerek Text özelliğine "XML'ye Yazdır" yazınız.
- Eklediğiniz düğmenin Click() metoduna tablo içeriğinin XML dosyaya yazdırılması için DataSet'in WriteXML() metodunu kullanınız.

```
private void button1_Click(object sender, System.EventArgs e)
{
    dataSet11.WriteXml("C://siniflar.xml", XmlWriteMode.IgnoreSchema);
}
```

WriteXml() metodu iki parametreye sahiptir. Birinci parametrede XML dosyasının adı, ikinci parametrede de XML dosyaya yazma modu belirtilir. İkinci parametre belirtilmediği takdirde dosyanın modu IgnoreSchema olarak varsayılır. IgnoreSchema özelliğiyle sadece DataTable'daki kayıtlar dosyaya yazdırılacaktır.

- Projenizi çalıştırın ve düğmeye tıklatınız.
- Oluşturulan XML dosyasının içeriğini görmek için XML dosyasını açın. XML dosyasının nereye kaydedildiğine dikkat ediniz.



Resim 2.4: Hazırlanan XML dosyasının içeriği

Böylece tablo içeriği XML dosyasına yazdırılmış olur. Tablo içeriğiyle beraber tablonun şemasını da bu dosyaya yazdırmak isterseniz XML dosyanın modunu WriteSchema olarak değiştirmeniz gerekir.

Sadece şema bilgilerini dosyaya yazdırılmak istendiğinde WriteXmlSchema() metodu kullanılır.

- Bu işlemin sonucunu görmek için ikinci bir komut düğmesini formunuza ekleyin ve Click() metodu aşağıdaki gibi düzenleyiniz.

```
private void button2_Click(object sender, System.EventArgs e)
{
    dataSet11.WriteXmlSchema("C://XMLsema.xml");
}
```

Projenizi çalıştırıp düğmeye tıkladığınızda “XMLsema.xml” dosyası belirtilen konumda oluşturulmuş olacaktır.

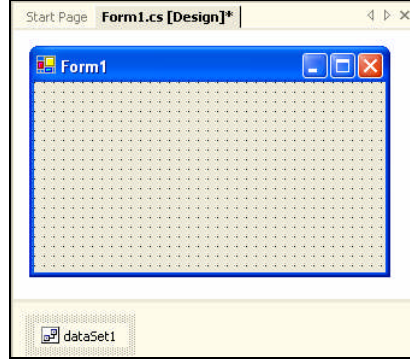


Resim 2.5: XML dosyasının şema görünümü

2.4. XML Dosyayı DataSet’e Aktarmak

Bir XML dosyanın içeriğini projenize ekleyeceğiniz bir DataSet’e aktarabilirsiniz. Aktarma işlemi için diğer uygulamalarda olduğu gibi OleDbConnection ve OleDbDataAdapter nesnelere ihtiyaç yoktur.

Yeni bir WindowsApplication oluşturup ToolBox’taki Data sekmesinden DataSet nesnesini formunuza ekleyiniz.



Resim 2.6:DataSet nesnesinin forma eklenmesi

XML dosyadan verileri okuyabilmek içinde form üzerine bir düğme yerleştirin ve Text özelliğine “XML’den oku” yazınız.

XML dosyadan okunan bilgilerin ekranda görüntülenebilmesi için de bir DataGridView nesnesini formunuza ekleyiniz.



Resim 2.7: Düğmenin (button) ve DataGridView’in forma eklenmesi

Eklediğiniz düğmenin üzerinde çift tıklayarak Click() metodunu aşağıdaki gibi düzenleyiniz.

```
private void button1_Click(object sender, System.EventArgs e)
{
    dataSet1.ReadXml("C://siniflar.xml", XmlReadMode.Auto);
    dataGridView1.DataSource=dataSet1.Tables[0];
}
```

DataSet’e XML dosyadan verileri okuyabilmek için ReadXml() metodu kullanılır. ReadXml() kendisine iki parametre alır. Birinci parametrede XML dosyanın adı, ikinci parametredeyse XML dosyası okuma modu seçilir. DataGridView nesnesinin DataSource özelliğine DataSet’in DataTable numarası aktarılır.

XML dosyadan şema bilgilerini (ReadSchema) okuyabileceğiniz gibi sadece bilgileri de (IgnoreSchema) okuyabilirsiniz.

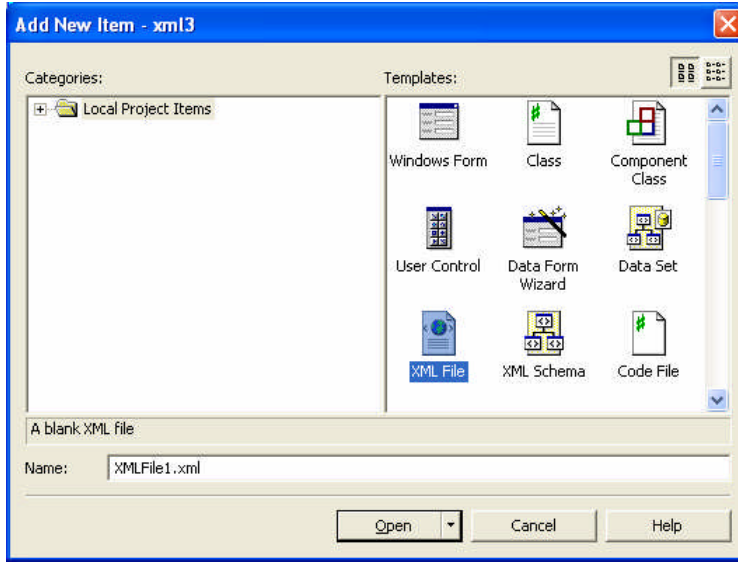
XML dosyaya sadece şema bilgilerini (WriteXmlSchema) yazdırdıysanız ve okuma için ReadXml() metodunu kullandıysanız DataTable nesnesinin içeriği boş olarak görüntülenir. ReadXml() metodu yerine ReadXmlSchema() metodunu kullanmanız gerekir.

2.5. Projenize Xml Dosya Ekleme

Projenize Xml dosya eklemek için Solution Explorer penceresinde fare sağ tuşuyla açılan menüden Add seçeneğinden Add New Item komutunu vermeniz gerekir.

Templates'ler içerisinde XML File seçeneğini seçerek Name alanına Xml dosyanız için bir isim veriniz. Eğer isim vermediğiniz takdirde dosyanın varsayılan adı XMLFile1.xml olacaktır.

Sonra, Open düğmesine tıklatın. Böylece XML dosya projenize eklenmiş olacaktır.



Resim 2.8: XMLFile'in projeye eklenmesi

2.6. XML Dosyanın Yapısını Belirlemek

Yeni oluşturduğunuz XML dosyanın içeriği ilk başta boş olarak karşınıza gelecektir. Kayıt yapısını sizin oluşturmanız gerekecektir. Bir XML dosyanın kayıt yapısını oluşturabilmek demek tablonun kayıt yapısını tanımlamak demektir. Örnek olarak XML bir dosya tanımlayalım.

```
<?xml version="1.0" encoding="utf-8" ?>
<dersler>
  <ders_adi></ders_adi>
  <dersin_kredisi></dersin_kredisi>
  <ders_saati></ders_saati>
</dersler>
```

Tablonun adı "dersler" olarak belirtilmiştir. Tablonun kayıt yapısında da ilgili alanlar vardır.

Bu şekilde oluşturulan bir XML dosyaya kayıt da girebilirsiniz. XMLFile alt kısmında bulunana Data seçeneğine tıklatın.



Böylece oluşturulan alanlar size gösterilecektir.

Data:			
Data for dersler			
	ders_adi	dersin_kredisi	ders_saati
▶			
*			

Resim 2.9: Alanların gösterimi

Alanlara kayıt girebilirsiniz.

Data:			
Data for dersler			
	ders_adi	dersin_kredisi	ders_saati
	Matematik	3	3
	Fizik	2	2
✎	Atölye	15	15
*			

Resim 2.10: Alanlara bilgilerin girilmesi

Girilen bilgilerin XML dosyada nasıl görüntülendiğini görmek için de XML seçeneğine tıklatın.



XML dosyanın içeriğini değiştiğini göreceksiniz.


```
<?xml version="1.0" encoding="utf-8" ?>
<NewDataSet>
  <dersler>
    <ders_adi>Matematik</ders_adi>
    <dersin_kredisi>3</dersin_kredisi>
    <ders_saati>3</ders_saati>
  </dersler>
  <dersler>
    <ders_adi>Fizik</ders_adi>
    <dersin_kredisi>2</dersin_kredisi>
    <ders_saati>2</ders_saati>
  </dersler>
  <dersler>
    <ders_adi>Atölye</ders_adi>
    <dersin_kredisi>15</dersin_kredisi>
    <ders_saati>15</ders_saati>
  </dersler>
</NewDataSet>
```

Girdiğiniz bilgiler şema içerisine yerleştirilmiştir.

2.7. XSLT (Extensible StyleSheet Language Transformation)

Xml (eXtensible Markup Language - genişletilebilir işaretleme dili), farklı platformlar arasında kolayca bilgi taşınmasına izin veren, veri kümelerini kendi kurallarımızla oluşturmamızı sağlayan önemli standartlardan birisidir. Lakin zaman zaman Xml dokümanlarını okumak çok kolay olmamaktadır. En azından oluşturulan Xml dokümanlarında tutulan içeriği, son kullanıcıya farklı şekillerde göstermek ihtiyacını duyabiliriz.

İşte bu noktada, var olan Xml verisinin farklı bir formata dönüştürülebilmesi gerekmektedir. XSLT (eXtensible Stylesheet Language Transformation) standardı tam bu noktada devreye girmektedir. Xslt herhangi bir Xml içeriğini farklı bir Xml, Html, Csv (Comma Separated Values) veya Text formatına dönüştürme işlemi ile ilgili materyalleri sağlayan bir işaretleme dilidir.

Bu sayede verileri sakladığımız bir Xml içeriğini son kullanıcıya daha uygun formatlarda sunabilme imkanını kazanmış oluruz. Bu özellikle WEB tabanlı sistemlerde son derece önemlidir. Dönüştürme işlemi sırasında sıradan bir tarayıcı kullanılabileceği gibi, .Net Framework içerisinde yer alan tiplerden de (types) faydalanılabilir.

2.8. XmlDataDocument Nesnesi

Bu nesne, bir XML dosyasının DataSet'ini otomatik olarak oluşturur. Böylece XML dosya bir DataSet nesnesi haline gelir ve istenildiği gibi kullanılabilir. XmlDataDocument nesnesini kullanabilmek için System isim uzayına XML'in eklenmesi gereklidir.

Using System.Xml;

Uygulama olması açısından XmlDocument'in kullanımı bir örnekle açıklayalım. Formunuza bir DataGridView nesnesi ekleyiniz ve formun Load() metodunu aşağıdaki gibi düzenleyiniz.

```
private void Form1_Load(object sender, System.EventArgs e)
{
    XmlDocument belge=new XmlDocument();
    belge.DataSet.ReadXml("C://siniflar.xml");
    dataGridView1.DataSource=belge.DataSet;
}
```

XmlDataDocument sınıfından “belge” adlı bir nesne oluşturulmaktadır.

```
XmlDataDocument belge=new XmlDocument();
```

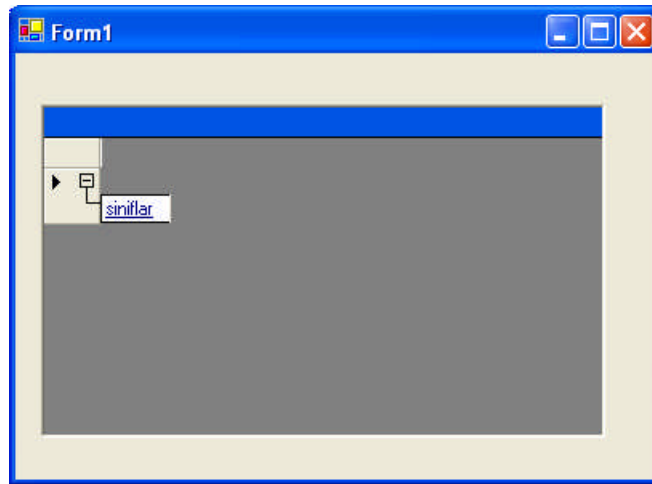
Bu nesneyle beraber DataSet'teki bilgiler ReadXml() metodu kullanılarak okunmaktadır.

```
belge.DataSet.ReadXml("C://siniflar.xml");
```

Son olarak DataGridView'in DataSource özelliğine XmlDocument sınıfından “belge” ile DataSet aktarılmaktadır.

```
dataGridView1.DataSource=belge.DataSet;
```

Projenizi çalıştırdığınızda ekran görüntüsü aşağıdaki gibi olacaktır.



Resim 2.11: Projenin çalıştırılması

DataGridView nesnesinde DataTable'ı oluşturan tablonun adı görüntülenecektir. Bu tablo ismine tıkladığınızda verileri DataGridView'de görebilirsiniz.

DataSet1:		
bolumu	ogr_sayisi	sinif_adi
Bilisim	24	10-A
Elk-Elt	24	10-B
Makine	16	10-C
Matbaa	24	10-D
Mobilya	8	10-E
*		

Resim 2.12: Verilerin DataGrid'de gösterilmesi

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<input type="checkbox"/> Yeni bir WindowsAppllication oluşturunuz.	<input type="checkbox"/> File menüsünden New komutunu verebilirsiniz.
<input type="checkbox"/> Projenize bir XML dosya ekleyiniz.	<input type="checkbox"/> Solution Explorer penceresinde fare sağ tuşuyla açılan pencereden Add seçeneğinden Add New Item() komutundan XML dosyayı seçebilirsiniz.
<input type="checkbox"/> XML dosyanın yapısını belirleyiniz.	<input type="checkbox"/> Aşağıdaki gibi belirleyebilirsiniz. <Okul> <Ogr_No></Ogr_No> <Adi></Adi> <Soyadi></Soyadi> <Sinifi></Sinifi> </Okul>
<input type="checkbox"/> Data seçeneğinden alanlara birkaç kayıt giriniz.	<input type="checkbox"/> XML dosyanın alt kısmındaki Data seçeneğine tıklayabilirsiniz.
<input type="checkbox"/> Forma bir Button nesnesi ekleyerek Click() metoduna XML dosyanın yapısını ve içeriğini farklı bir dizinde farklı bir XML dosya adı belirterek yazdıran kodları yazınız.	<input type="checkbox"/> Bilgisayarınızın C: root'unu belirtebilirsiniz.
<input type="checkbox"/> Forma bir DataGrid ekleyiniz.	<input type="checkbox"/> Nesneyi ToolBox'taki WindowsForms sekmesinden seçebilirsiniz.
<input type="checkbox"/> Bir XmlDataDocument nesnesi aracılığıyla yeni oluşturduğunuz XML dosyanın DataSet'ini bu nesneye okutunuz.	<input type="checkbox"/> System isim uzayına XML'i eklemeyi unutmayınız.
<input type="checkbox"/> Okunan DataSet'i DataGrid'in DataSource'una aktarınız.	<input type="checkbox"/> dataGrid1.DataSource=belge.DataSet;
<input type="checkbox"/> Projenizi çalıştırınız.	<input type="checkbox"/> F5 tuşuna basabilirsiniz.

ÖLÇME VE DEĞERLENDİRME

Aşağıdaki soruları dikkatlice okuyarak doğru/yanlış seçenekli sorularda uygun harfleri yuvarlak içine alınız. Boşlukları uygun şekilde doldurunuz.

1. XML herhangi iki programlama dili arasında veri alış verişini kolaylaştırır.(D/Y)
2. DataSet oluşturulduktan sonra uzantılı bir dosya otomatik olarak oluşturulur.
3. Tablo içeriğinin XML dosyaya yazdırılması içinmetodu kullanılır.
4. IgnoreSchema() özelliğiyle DataTable'ın şeması ve kayıtları dosyaya yazdırılır.(D/Y)
5. XML dosyasının DataSet'ini otomatik olarak oluşturmak için nesnesi kullanılır.

Değerlendirme

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Doğru cevap sayınızı belirleyerek kendinizi değerlendiriniz. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konulara geri dönerek tekrar inceleyiniz. Tüm sorulara doğru cevap verdiyseniz diğer öğrenme faaliyetine geçiniz.

MODÜL DEĞERLENDİRME

PERFORMANS TESTİ (YETERLİK ÖLÇME)

Modül ile kazandığınız yeterliği aşağıdaki kriterlere göre değerlendiriniz.

Değerlendirme Kriterleri	Evet	Hayır
1. Forma OleDbConnection nesnesi eklediniz mi?		
2. Veri tabanıyla bağlantı kurdunuz mu?		
3. OleDbDataAdapter nesnesi eklediniz mi?		
4. SQL komutlarını tanımladınız mı?		
5. DataSet oluşturduğunuz mu?		
6. Databindings bağlantılarını yaptınız mı?		
7. Forma DataGrid nesnesi yerleştirdiniz mi?		
8. DataGrid nesnesinin DataSource özelliğini belirttiniz mi?		
9. OleDbConnection nesnesinin bağlantısını açtınız mı?		
10. Fill metodunu kullandınız mı?		
11. Veriler arasında dolaştınız mı?		
12. Metod oluşturduğunuz mu?		
13. Update metodunu kullandınız mı?		
14. CancelCurrentEdit metodunu kullandınız mı?		
15. RemoveAt metodunu kullandınız mı?		
16. Close metodunu kullandınız mı?		
17. Projeye XML dosya eklediniz mi?		
18. XML dosya yapısını belirlediniz mi?		
19. System isim uzayına XML'i eklediniz mi?		
20. DataSet'i DataGrid'in DataSource'una aktardınız mı?		

Değerlendirme

Yaptığınız değerlendirme sonucunda eksikleriniz varsa öğrenme faaliyetlerini tekrarlayınız.

Modülü tamamladınız, tebrik ederiz. Öğretmeniniz size çeşitli ölçme araçları uygulayacaktır, öğretmeninizle iletişime geçiniz.

CEVAP ANAHTARLARI

Öğrenme Faaliyeti-1'in Cevap Anahtarı

1	ADO.NET
2	A
3	OleDbConnection
4	Open()-Close()
5	Yanlış
6	Where
7	Update ve Delete
8	Yanlış
9	C
10	Doğru

Öğrenme Faaliyeti-2'nin Cevap Anahtarı

1	Doğru
2	Xsd
3	WriteXml()
4	Yanlış
5	XmlDataDocument

KAYNAKÇA

- YANIK Memik, **Microsoft Visual C#.NET**, Seçkin Yayıncılık, Ankara, 2004.
- ZENGİN Abdullah, **C# 2005**, Nirvana Yayınları, Ankara, 2006.
- ZEKİ Yasemin, **Adım Adım C++ Uygulamaları**, Nirvana Yayınları, Ankara, 2006.
- AKPINAR Erdal, Ders Notları
- www.csharpnedir.com.
- www.msakademik.net