

# **C# Windows Veritabanı İşlemleri**

## **ADO.Net (Active Data Objects)**

- **SQL diline giriş (INSERT,UPDATE,SELECT,DELETE)**
- **ADO.Net bileşenleri**
- **SqlConnection, OleDbConnection,ODBCConnection**
- **SqlCommand,OldeDbCommand**
- **DataAdapter**
- **DataReader**
- **DataSet (DataTable)**
- **DataGridView**

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### ADO.NET Nedir?

ADO.NET (ActiveX Data Objects.NET), Microsoft tarafından bize sunulan, veritabanı ile uygulamalarımız arasında köprü görevini görmektedir. ADO.NET ile uygulama tarafımızda veritabanımıza bağlanabilir, verilerimizi buradan listeleyebilir, güncelleyebilir, veri ekleyebilir veya silebiliriz. ADO.NET ile SQL sorguları, komutlarını uygulama tarafımızda kullanabiliriz.

ADO.NET ortamını sadece SQL veri tabanı için değil, ACCESS, ORACLE gibi diğer veri tabanları içinde kullanabiliriz.



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### ADO.NET Nedir?

- SQL veri tabanı kullanacağımız için Visual Studio ortamında oluşturduğumuz .NET projesi için -bu bir windows projesi de olabilir WEB projeside olabilir- öncelikle ilgili namespace'leri eklememiz gerekir.
- **System.Data:** Bu isim alanının projeye eklenmesi şarttır. Bu alan içerisinde SQL, Access, Oracle gibi yapılar için ortak bileşenler bulunur.
- **System.Data.SqlClient:** SQL kullanarak oluşturacağımız bağlantılar için bu alanı projemize eklememiz diğer bir şarttır.

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### ADO.NET Nedir?

Veri Tabanıyla ilgili işlemlerimizde kullanacağımız bazı yapılar vardır.

- Connection – bağlantı nesnesi
- Command – sorgu çeşiti, sorgu cümlesi
- DataAdapter – veritabanı ile dataset arası köprü.(sql cümlesini tutar)
- DataSet – Bağlantısız veritabanı işlemlerinde kullanılan DataTable havuzu. Sorgu sonucu dönen tablo havuzu.
- DataReader – Sadece vt'den okuma yapılacaksa kullanılan yapı.



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

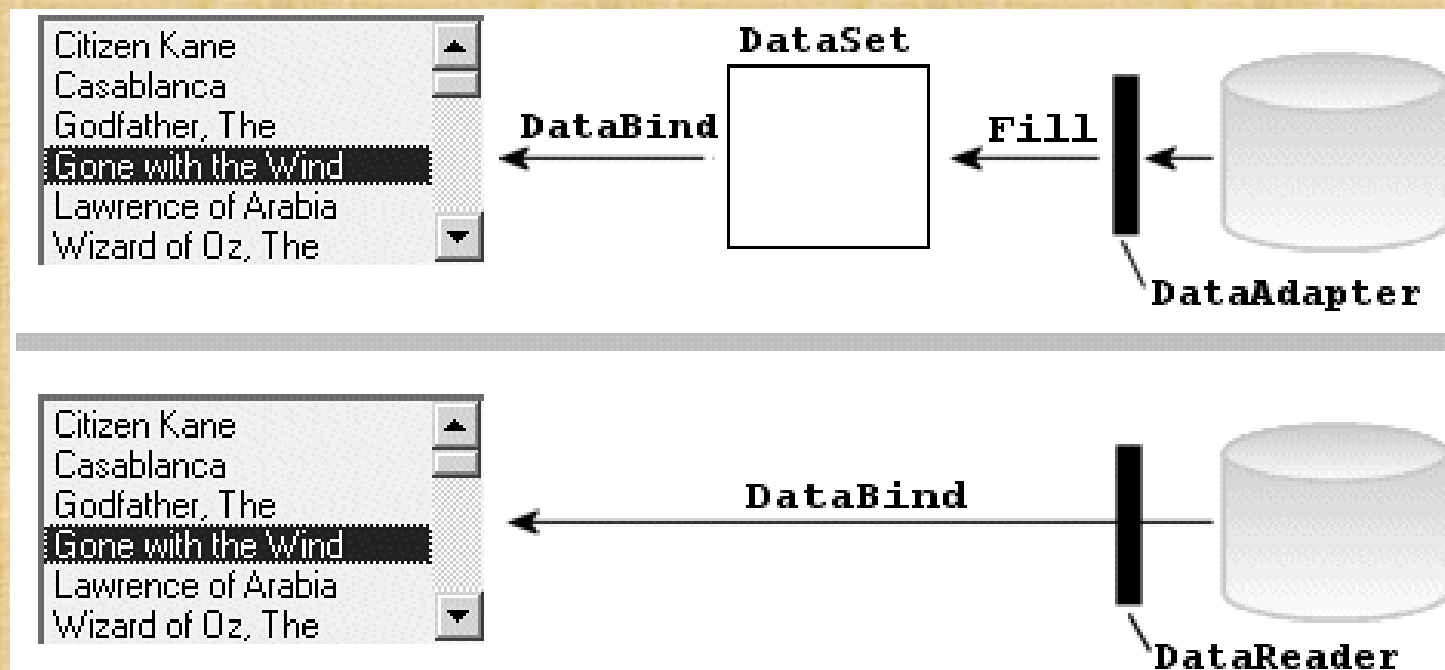
### ADO.NET Veri Sağlayıcıları

Herbir veri sağlayıcısı aşağıdaki nesneleri içerir.

- **XxxConnection**
  - Bağlantı kurmak için kullanılır.
- **XxxCommand**
  - Veritabanına sorgu yollamak için kullanılır.
- **XxxDataReader**
  - Çevrimiçi bağlantı ile sadece veri okuma.
- **XxxDataAdapter**
  - Çevrimdışı bağlantılarda veri işleme nesnesi.

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### SqlConnection()

- Veritabanına bağlanmak için kullanılan sınıftır. Bu yapının bazı metot ve özellikleri vardır.
- **ConnectionString**: Hangi veritabanına bağlanacağımızı belirten kelimelere, connectionstring denir.
- **Data Source**: Bağlanacağımız bilgisayarın adı yada ip numarasını belirtebiliriz.
- **State** : Bağlantımızın açık olup olmadığını bu property yardımıyla öğrenebiliriz ayrıca StateChanged event'i ile bağlantı durumu değiştiğinde bir işlem yaptırabilirsiniz.



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### SqlConnection()

- **User ID(uid)**: Bağlanacağımız veritabanına hangi kullanıcı adı ile gireceğimizi belirtebiliriz.
- **Password (pwd)**: user id'mize ait şifremizi belirtebiliriz.
- **ChangeDatabase(EVENT)** : Bu metot sayesinde, bir veritabanına bağlı iken, bir başka veritabanında işlem yapmamız gerektiğinde kullanabiliriz.

**Örnek sqlconnection nesnesi ve state örnek satırı:**

```
Baglan=SqlConnection("Data Source=.\sqlexpress; Initial  
Catalog=elginkan;Integrated Security=true")  
if (baglan.State == ConnectionState.Closed)
```



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### SqlCommand()

- SqlCommand nesnesi veritabanı işlemlerini yapmamızı sağlayan bir araçtır.
- CommandText : Bu property'de, sorgunuz SQL sorgusu ise sorgunun kendisi, stored procedure ise stored procedure adı yada text SQL cümlesi yada tablo ismini yazabiliriz.
- Parameters : Sorgularımıza parametre olarak değer yollamamızı sağlayan özelliktir.

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### SqlCommand()

- **ExecuteReader** : Sorgumuzdan geriye birden çok satır dönme ihtimali yüksekse kullanılacak bir property'dir. Geriye SqlDataReader döndürmektedir.
- **ExecuteScalar** : Sorgudan dönen sonucun ilk kolonunun ve ilk satırının değerini alır. Kısacası geriye tek hücrelik bir sonuç döndüreğinizde kullanabileceğiniz bir property'dir. Veritabanı update edildimi vb işlemlerde yada `Select kullanıcıAdi From Tablo where kullanıcıAdi = @kullanıcıAdi and kullanıcıSifre = @kullanıcıSifre` şeklinde yaparak, üye giriş kontrolünü daha performanslı hale getirebilirsiniz.
- **ExecuteNonQuery** : Çalıştırdığımız sorgu geriye bir değer döndürmeyecekse (insert, update, delete vb..) kullanılacak bir property'dir.



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### SqlCommand kayıt örneği

```
if (baglan.State == ConnectionState.Closed)
    baglan.Open();

komut = new SqlCommand();
komut.CommandType = CommandType.Text;
komut.CommandText = "insert into ogrTablo(ogrNo,
ogrAdiSoyadi,sifre) values('" + ogrTxt.Text + "', '" +
adTxt.Text + "', '" + adTxt.Text + "')";
komut.Connection = baglan;
komut.ExecuteNonQuery();
baglan.Close();
Listele();
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

```
public void Listele()
{
    if (baglanti.State == ConnectionState.Closed)
        baglanti.Open();
    komut = new SqlCommand("select * from personel",baglanti);
    da = new SqlDataAdapter(komut);
    ds = new DataSet();
    da.Fill(ds, "tbl");
    dataGridView1.DataSource = ds.Tables["tbl"];
    baglanti.Close();
}
```



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### SqlCommand Parametrelili ile kaydet örneği

```
if (baglan.State == ConnectionState.Closed)
    baglan.Open();
komut=new SqlCommand("insert into ogrTablo values(@ogrNo,@ogrAdiSoyadi,
@eposta,@sifre,@dtarih,@vize,@dersId)",baglan);
    komut.Parameters.AddWithValue("@ogrNo", ogrTxt.Text);
    komut.Parameters.AddWithValue("@ogrAdiSoyadi", adTxt.Text);
try
{
    komut.ExecuteNonQuery();
}
catch(Exception ee)
{
    MessageBox.Show("hata"+ee.Message);}
    baglan.Close();
Listele();
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### **DataReader()**

SqlDataReader, sadece okunabilir olarak kullanılmaktadır. Satır satır okuma işlemi yapılmaktadır. SqlDataReader kullanımı boyunca veritabanı bağlantısı açık olacaktır. Çünkü SqlDataReader veritabanı ile bağlantılı olarak çalışmaktadır. Okuma işlemi sona erdiğinde SqlDataReader bağlantısının kapatılması gerekmektedir.



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### **DataReader() örneği-Form Arama işlemi**

```
SqlDataReader dr;  
  
if (baglan.State==ConnectionState.Closed)  
  
    baglan.Open();  
  
komut = new SqlCommand();  
  
komut.Connection = baglan;  
  
komut.CommandText = "select * from ogrTablo where  
ogrNo='" + araTxt.Text + "'";
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### DataReader() örneği (devam)

```
dr=komut.ExecuteReader();  
while (dr.Read())  
{  
    ogrTxt.Text=dr["ogrNo"].ToString();  
    adTxt.Text = dr["ogrAdiSoyadi"].ToString();  
    emailTxt.Text = dr.GetString(2);  
    dtarihTxt.Text = dr.GetString(2);  
}  
dr.Close();  
baglan.Close();
```



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### DataReader() - Listbox MultiColumn Uygulaması

```
komut = new SqlCommand("select * from personel",  
baglanti);  
dr = komut.ExecuteReader();  
while (dr.Read())  
{  
//comboBox1.Items.Add(dr["persadsoyad"].ToString());  
listBox1.Items.AddRange(new object[]  
{dr["sicilno"].ToString()+"\t"+dr["persadsoyad"].ToString()  
});  
}
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### DataSet() ve DataTable()

- Dataset Veritabanınızdan bağlantısız olan çalışan bir nesnedir. Bir kez veritabanına bağlandıktan sonra veriyi alır ve bağlantıyı keser.
- Bu veriler dataset'e bir kez aktardıktan sonra veriler sunucunun hafızasında saklanacağından artık database ile ilgimiz kalmamakta. Sunucunun hafızasında bulunan bu verilerle görüntüleme, silme, güncelleme gibi işlemlerimizi rahatlıkla yapabiliriz. **Dataset** içerisinde bir veya birden fazla **DataTable** kullanma şansına sahibiz. ***DataTable'ler Datasetlerin bir alt sınıfı gibi ele alınabilir.***



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

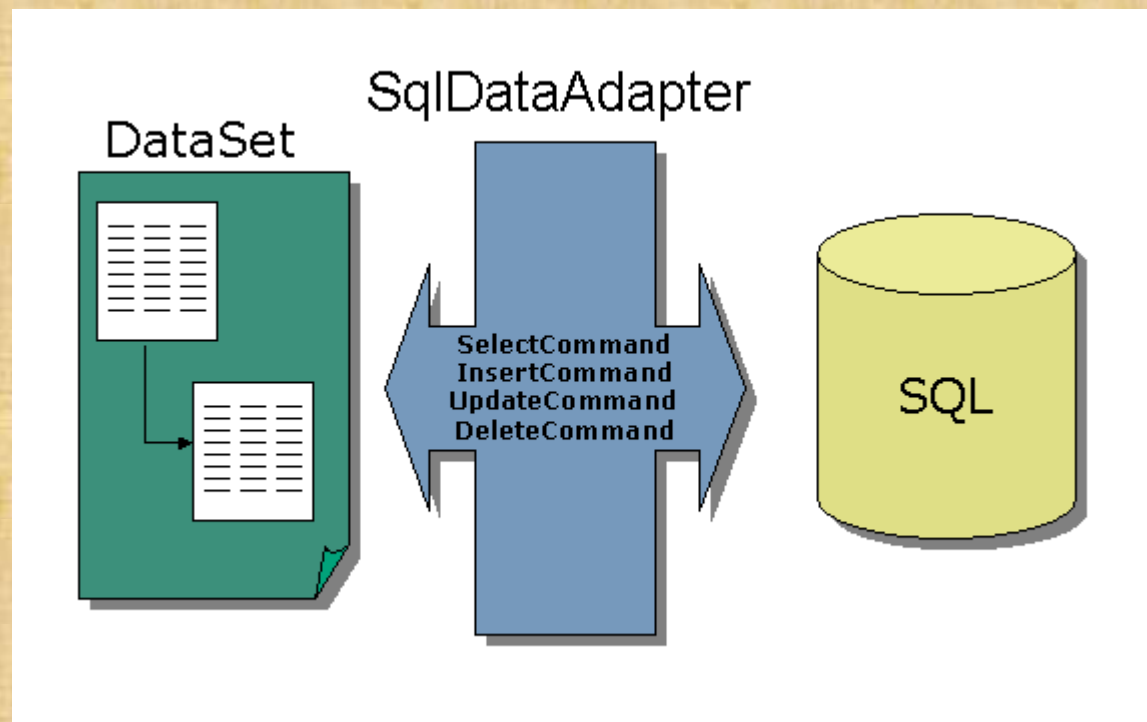
### DataAdapter()

- DataAdapter nesnesi Connected ve Disconnected bağlantı yapısı ile veri arasında köprü vazifesi görür. Databaseinizden çektiğiniz veriyi DataSetinize doldurmak yada DataSetinizde güncellenerek cachelenmiş verilerinizi veri kaynağınızda da güncellemek için istediğiniz zaman DataAdapter nesnesini kullanabilirsiniz.
- Çevrimdisi (offline, bağlantısız) bağlantı sağlar ve serverdan clienta (select), clienttan servera (update, delete, insert) komutlarını çok rahat ve toplu halde iletilmesini sağlayan metotlara sahiptir.
- Biz DataAdapter sınıfını, DataTable veya DataSetler ile beraber kullanırız.
- Bir DataAdapter nesnesi, mutlaka select sorgusu içermelidir. Select sorgusu ile ortaya çıkan veri kümesi ise DataTable veya DataSet nesnelere aktarılır.

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### DataAdapter()





# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

**DataSet() DataTable() DataAdapter() Örnl:Listele**

```
if (baglan.State == ConnectionState.Closed)
    baglan.Open();
    komut = new SqlCommand("select * from ogrTablo", baglan);
    da = new SqlDataAdapter(komut);
    dt = new DataTable();
    ds = new DataSet();
    da.Fill(ds,"tbl");
    da.Fill(dt);
    dataGridView1.DataSource = dt;
    dataGridView1.DataSource = ds.Tables["tbl"];
    dataGridView1.ClearSelection();
    dataGridView1.Rows[5].Cells[0].Selected = true;
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- *DataGridView (Rows, CurrentRow, NAV button)*

```
void Baglan()
{
    try
    {
        conn = new SqlConnection("Data Source=.\sqlexpress;Initial
Catalog=tmpUrunler;Integrated Security=true");
        //conn = new SqlConnection("Data Source=(localdb)\\MSSQLLocalDB;Attach
Datafile=c:\\deneme.mdf;Initial Catalog=ogr;Integrated Security=true");
        conn.Open();
        button1.Text = "Bağlandı..";
    }
    catch (Exception ee)
    {
        MessageBox.Show(ee.Message);
    }
}
```



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- ***DataGridView (Rows, CurrentRow, NAV button)***

```
void Listele() //basit Listele!!
{
    Baglan();
    komut = new SqlCommand("select projeId,projeAd,projeMaliyet,
projeBasTar,projeBitTar from Urunler",conn);
    da = new SqlDataAdapter(komut);
    ds = new DataSet();
    da.Fill(ds,"tumListe");
    dataGridView1.DataSource = ds.Tables["tumListe"];
}
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- *DataGridView (Rows, CurrentRow, NAV button)*

```
//CELL CLICK: veri çekme
private void dataGridView1_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    secilensatir = dataGridView1.SelectedCells[0].RowIndex;
    int projeID =(int)
dataGridView1.Rows[secilensatir].Cells[0].Value;
    string projeAdi =
dataGridView1.Rows[secilensatir].Cells[1].Value.ToString();
    projeITxt.Text = projeID.ToString();
    projeAdiTxt.Text= projeAdi; //diğer alanlarda yazılmalı
}
```



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- ***DataGridView (Rows, CurrentRow, NAV button)***

//CELL ENTER: veri çekme

```
private void dataGridView1_CellEnter(object sender,
DataGridViewCellEventArgs e)
{
    dataGridView1.Rows[aktifSatir].Selected = false;
    aktifSatir = dataGridView1.CurrentRow.Index;
    dataGridView1.Rows[aktifSatir].Selected = true;
    txtdoldur();
}

void txtdoldur()
{
    projeAdiTxt.Text = dataGridView1.Rows[aktifSatir].Cells[1].Value.ToString();
    projeNoTxt.Text = dataGridView1.Rows[aktifSatir].Cells[0].Value.ToString();
    projeBasTarTxt.Text = dataGridView1.Rows[aktifSatir].Cells[2].Value.ToString();
    projeBitTarTxt.Text = dataGridView1.Rows[aktifSatir].Cells[3].Value.ToString();
    projeMaliyetTxt.Text = dataGridView1.Rows[aktifSatir].Cells["projeMaliyet"].Value.ToString();
}
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- ***DataGridView (Rows, CurrentRow, NAV button)***

```
//Yukarı-Aşağı hareket ettirmek
private void dataGridView1_CellEnter(object sender,
DataGridViewCellEventArgs e)
{
    int secilen = dataGridView1.CurrentRow.Index;
    dataGridView1.CurrentRow.Selected = true;
    prjIDTxt.Text =
dataGridView1.Rows[secilen].Cells[0].Value.ToString();
    prjAdiTxt.Text =
dataGridView1.Rows[secilen].Cells[1].Value.ToString();
    prjMalTxt.Text =
dataGridView1.Rows[secilen].Cells[2].Value.ToString();
    prjBasTarTxt.Text =
dataGridView1.Rows[secilen].Cells[3].Value.ToString();
}
```



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- *DataGridView (Rows, CurrentRow, NAV button)*

//NAV: Geri butonu

```
private void button3_Click(object sender, EventArgs e)
{
    int onceki;
    int maxKayit = dataGridView1.Rows.Count;
    if (dataGridView1.CurrentRow.Index < maxKayit &&
        dataGridView1.CurrentRow.Index > 0)
    {
        onceki = dataGridView1.CurrentRow.Index - 1;
        dataGridView1.CurrentCell =
            dataGridView1.Rows[onceki].Cells[dataGridView1.CurrentCell.ColumnIndex];
        secilensatir = dataGridView1.CurrentRow.Index;
        dataGridView1_CellClick(dataGridView1, null);
    }
}
```

Alternatif:

```
BindingSource.MoveNext();
BindingSource.MovePrevious();
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- *DataGridView (Rows, CurrentRow, NAV button)*

```
//NAV :ileri
private void button4_Click(object sender, EventArgs e)
{
    int onceki;
    int maxKayit = dataGridView1.Rows.Count;
    if (dataGridView1.CurrentRow.Index < maxKayit-1)
    {
        onceki = dataGridView1.CurrentRow.Index + 1;
        dataGridView1.CurrentCell =
dataGridView1.Rows[onceki].Cells[dataGridView1.CurrentCell.ColumnIndex];
        secilensatir = dataGridView1.CurrentRow.Index;
        dataGridView1_CellClick(dataGridView1, null);
    }
}
```



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- ***DataGridView (Rows, CurrentRow, NAV button)***

```
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if(degisim)
    {
        Listele(null, null);
        degisim = false;
    }
    int secilen = dataGridView1.CurrentRow.Index;
    dataGridView1.CurrentRow.Selected = true;
    prjIDTxt.Text = dataGridView1.Rows[secilen].Cells[0].Value.ToString();
    prjAdiTxt.Text = dataGridView1.Rows[secilen].Cells[1].Value.ToString();
    prjMalTxt.Text = dataGridView1.Rows[secilen].Cells[2].Value.ToString();
    prjBasTarTxt.Text = dataGridView1.Rows[secilen].Cells[3].Value.ToString();
}
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- ***DataGridView (Rows, CurrentRow, NAV button)***

```
private void dataGridView1_CellEndEdit(object sender, DataGridViewCellEventArgs e)
{
    Baglan();
    string prjAd = dataGridView1.CurrentRow.Cells[1].Value.ToString();
    string prjBasTar= dataGridView1.CurrentRow.Cells[3].Value.ToString();
    int projeMal =
int.Parse(dataGridView1.CurrentRow.Cells[2].Value.ToString());
    int prjId =
int.Parse(dataGridView1.CurrentRow.Cells[0].Value.ToString());
    komut = new SqlCommand("update Urunler set projeAd='" + prjAd +
'',projeMaliyet="+projeMal+",projeBasTar='"+prjBasTar+" where projeId=" +prjId,
conn);
    komut.ExecuteNonQuery();
    degisim = true;
}
```



```

void Listele(string prm,string deger) //Advanced Listele()
{
    ds = new DataSet();
    try
    {
        if (prm == null)
        {
            komut = new SqlCommand("select projeId,projeAd,projeMaliyet,projeBasTar
Urunler", conn);
            da = new SqlDataAdapter(komut);
            da.Fill(ds, "prjListe");
            dataGridView1.DataSource = ds.Tables["prjListe"];
        }
        else
        {
            switch (prm)
            {
                case "projeId": komut = new SqlCommand("select projeId,projeAd,projeMaliyet,projeBasT
from Urunler where " + prm + " like '%" + int.Parse(deger) + "%'", conn); break;

                case "projeAd": komut = new SqlCommand("select projeId,projeAd,projeMaliyet,projeBasT
from Urunler where " + prm + " like '%" + deger + "%'", conn); break;
            }
            da = new SqlDataAdapter(komut);
            da.Fill(ds, "prjArama");
            dataGridView1.DataSource = ds.Tables["prjArama"];
        }
    }
    catch (Exception ee)
    {
        MessageBox.Show("HATA!!!" + ee.Message);
    }
    aramaTxt.Enabled = true; comboBox1.Enabled = true;
}

```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- **Arama**

```
private void arama_TextChanged(object sender, EventArgs e)
{
    if (aramaCmb.SelectedIndex > -1)
    {
        switch (aramaCmb.SelectedIndex)
        {
            case 0: komut = new SqlCommand("select * from projeTakip where projNo 1
'" + arama.Text + "'", bag); break;
            case 1: komut = new SqlCommand("select * from projeTakip where projAdi
like '" + arama.Text + "'", bag); break;
        }
        da = new SqlDataAdapter(komut);
        ds = new DataSet();
        da.Fill(ds, "aramaListe");
        dataGridView1.DataSource = ds.Tables["aramaListe"];
        dataGridView1.Refresh();
        //Listele();
    }
    else
        MessageBox.Show("Arama için bir kriter seçiniz...");
}
```



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- ***DataGridView (Auto update, CurrentRow etc.)***

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        SqlConnection baglanti;
        SqlCommand komut;
        SqlDataAdapter da;
        DataSet ds;
        bool degisim = false;
        public Form1()
        {
            baglanti = new SqlConnection("Data
Source=.\sqlexpress;Initial
Catalog=elginkan;Integrated Security=true");
            InitializeComponent();
        }
    }
}
```


# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- *Birden fazla Textbox için tek Event atama*

```
private void Form1_Load(object sender, EventArgs e)
{
    aramaCmb.Items.Add("projNo");
    aramaCmb.Items.Add("projAdi");
    this.Text = "Proje Takip-" + DateTime.Now.ToString();
    foreach (Control ctrl in this.Controls)
    {
        if (ctrl.Name.Contains("Txt") && ctrl is TextBox)
        {
            TextBox tb = (TextBox)ctrl;
            tb.TextChanged += new EventHandler(tb_TextChanged);
        }
    }
}

void tb_TextChanged(object sender, EventArgs e)
{
    guncelleBtn.Enabled = true;
}
```





# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- ***DataGridView (Auto update, CurrentRow etc.)***

```
private void Form1_Load(object sender, EventArgs e)
{
    baglanti.Open();
    Listele();
}

void Listele()
{
    if (baglanti.State==ConnectionState.Closed)
        baglanti.Open();
    da=new SqlDataAdapter("Select * from ogrTablo",baglanti);
    ds = new DataSet();
    da.Fill(ds);
    dataGridView1.DataSource = ds.Tables[0];
}

//listele grid'i dolduruyor.
```

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- *DataGridView (Auto update, CurrentRow etc.)*

```
private void dataGridView1_CellClick(object sender,
DataGridViewCellEventArgs e)
{
    if (degisim == true)
    {
        da = new SqlDataAdapter("Select * from ogrTablo", baglanti);
        ds = new DataSet();
        da.Fill(ds);
        dataGridView1.DataSource = ds.Tables[0];
    }
    int secilenSatir = dataGridView1.SelectedCells[0].RowIndex;
    string ogrAdi =
dataGridView1.Rows[secilenSatir].Cells[1].Value.ToString();
    textBox1.Text = ogrAdi;
}
```

*//'degisim' bool değeri true ise değişiklik var ve datagrid güncelleniyor.*



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- *DataGridView (Auto update, CurrentRow etc.)*

```
private void dataGridView1_CellEndEdit(object sender,
DataGridViewCellEventArgs e)
{
    if (baglanti.State==ConnectionState.Closed)
        baglanti.Open();
    komut = new SqlCommand("update ogrTablo set ogrAdiSoyadi='" +
dataGridView1.CurrentRow.Cells[1].Value.ToString() + "'" where ogrNo='" +
dataGridView1.CurrentRow.Cells[0].Value.ToString() + "'", baglanti);
    komut.ExecuteNonQuery();
    degisim = true;
}
}
```

*//Cell edit modda olduğu için datagrid datasource'unu ds(dataset) ile besleyemiyoruz,  
«**SetCurrentCellAddressCore**» hatası veriyor*

*//bu yüzden '**degisim**' degiskeni tanımlandı ve kişi edit işi bitip başka bir hücreyi  
tıklayınca //(CellClick event'te) '**degisim**' degiskenine bakarak true ise ds gridview'in  
source'unu //besleyecek güncelleme olacak*

# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

### Kod ile Data Grid Nesnesi

```
dataGridView1 = new DataGridView();
dataGridView1.Size = new Size(650, 300);
this.Size = dataGridView1.Size;
dataGridView1.ColumnCount = 2;
dataGridView1.Columns[0].Name = "sicilno";
dataGridView1.Columns[1].Name = "persadsoyad";
object[] sirasira={"1","ali"};
dataGridView1.Rows.Add(sira);
DataGridViewButtonColumn btn = new DataGridViewButtonColumn();
dataGridView1.Columns.Add(btn);
this.Controls.Add(dataGridView1);
```



# C# Windows Veritabanı İşlemleri

## ADO.Net (Active Data Objects)

- ***Combobox'a Data Çekmek***

```
SqlDataReader dr;  
komut = new SqlCommand("select bolum from  
ogrenci",baglanti);  
dr=komut.ExecuteReader();  
while (dr.Read())  
{  
    Combobox1.Items.Add(dr["bolum"]);  
}
```