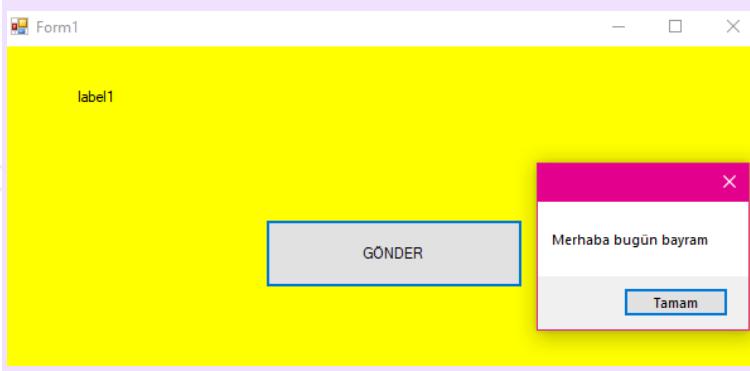


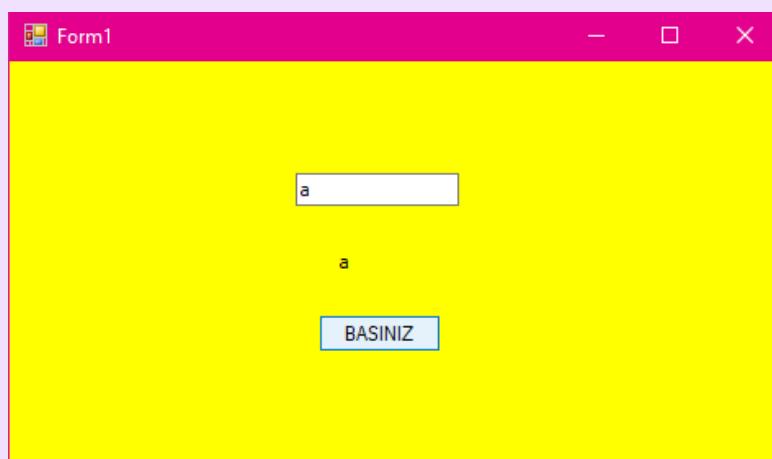
# C#

## DEĞİŞKENLER

```
namespace degiskenler
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        /* Metin içeren değişkenlerdir. "" çift tırnak arasına yazılır.*/
        private void button1_Click(object sender, EventArgs e)
        {
            string metin;
            metin = "Merhaba bugün bayram";
            MessageBox.Show(metin);
            label1.Text = "Merhaba Cengiz Bey. Nasılsınız?";
        }
    }
}
```

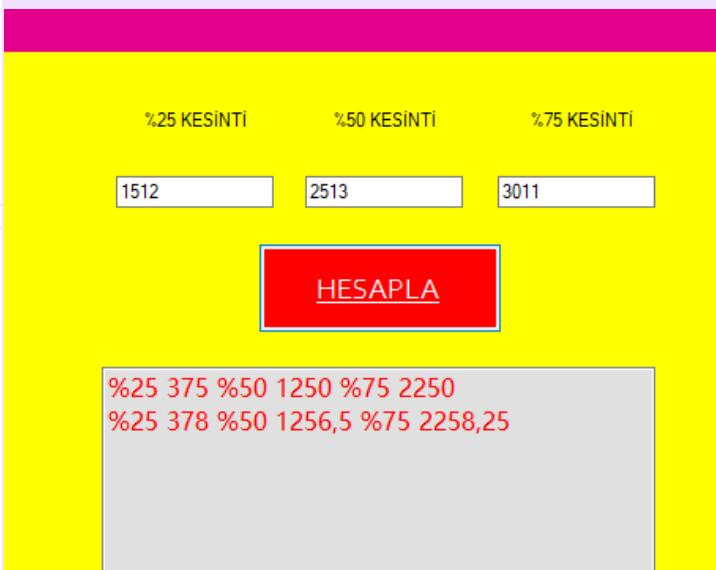


```
namespace degiskenler_char
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        /*
         Tek karakterli değişkenlerdir. Değişken yazılmırken
         * '' tek tırnak kullanılır.
        */
        private void button1_Click(object sender, EventArgs e)
        {
            char ders = Convert.ToChar(textBox1.Text);
            label1.Text = ders.ToString();
        }
    }
}
```



```
namespace degiskenlerson
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        /* double değişkenler virgülü değişkenlerdir*/
        private void button1_Click(object sender, EventArgs e)
        {
            double yirmibes, elli, yetmisbes;
            yirmibes = (Convert.ToDouble(textBox1.Text))*25/100;
            elli = (Convert.ToDouble(textBox2.Text))*50/100;
            yetmisbes = (Convert.ToDouble(textBox3.Text))*75/100;

            listBox1.Items.Add("%25 " + yirmibes + " " +"%50 "+
                elli + " " +"%75 " + yetmisbes);
        }
    }
}
```



# OPERATÖRLER

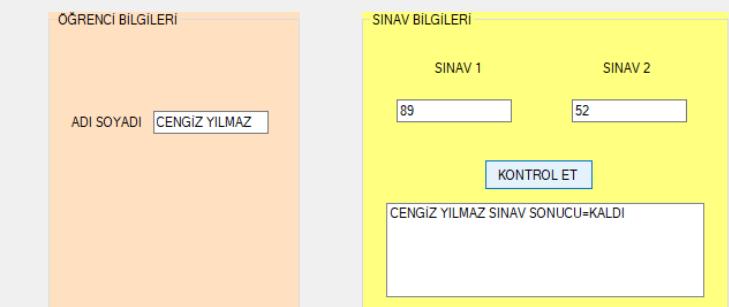
= -----> ATAMA  
== -----> EŞİT Mİ(SORGULAMA)  
+-\* / -----> ARİTMETİK OPERATÖRLER  
!= -----> EŞİT DEĞİLSE  
>= BÜYÜK EŞİT  
<= KÜÇÜK EŞİT  
&& -----> VE  
|| -----> VEYA

## IF, IF ELSE

```
private void button1_Click(object sender, EventArgs e)
{
    string ad;
    int birincisinav, ikincisinav, sinavsonunu;

    ad = textBox1.Text;
    birincisinav = Convert.ToInt16(textBox2.Text);
    ikincisinav = Convert.ToInt16(textBox3.Text);
    sinavsonunu = birincisinav * 30 / 100 + ikincisinav * 70 / 100;

    if(sinavsonunu>=70)
    {
        listBox1.Items.Add(ad + " " + "SINAV SONUCU= GEÇTİ");
    }
    else
    {
        listBox1.Items.Add(ad + " " + "SINAV SONUCU=KALDI");
    }
}
```



## SWITCH CASE

```
private void button1_Click(object sender, EventArgs e)
{
    int ay=Convert.ToInt16(textBox1.Text);
    switch(ay)
    {
        case 1: label1.Text = "ocak"; break;
        case 2: label1.Text = "şubat"; break;
        case 3: label1.Text = "mart"; break;
        default: label1.Text = "yanlış ay girdiniz"; break;
    }
}
```

```
string mevsim(textBox2.Text);

switch(mevsim)
{
    case "aralık": label2.Text = "kış"; break;
    case "ocak": label2.Text = "sonbahar"; break;
    case "şubat": label2.Text = "ilkbahar"; break;
    case "mart": label2.Text = "ilkbahar"; break;
    case "nisan": label2.Text = "ilkbahar"; break;
    case "mayıs": label2.Text = "ilkbahar"; break;
    case "haziran": label2.Text = "yaz"; break;
    case "temmuz": label2.Text = "yaz"; break;
    case "ağustos": label2.Text = "yaz"; break;
    case "eylül": label2.Text = "sonbahar"; break;
    case "ekim": label2.Text = "sonbahar"; break;
    case "kasım": label2.Text = "sonbahar"; break;
    default: label2.Text = "yanlış ay girdiniz"; break;
}
```

## COMBOBOX ÖZELLİKLERİ

```
private void Form1_Load(object sender, EventArgs e)
{
    combosehir.DropDownStyle = 150; /*yüksekliği ayarlar*/
    combosehir.DropDownWidth = 100; /*genişliği ayarlar*/
    combosehir.BackColor = System.Drawing.Color.Yellow; /*arkaplan rengi*/
    combosehir.ForeColor = System.Drawing.Color.Black; /*font rengi*/
    combosehir.DropDownStyle = ComboBoxStyle.DropDownList; /*listenin nasıl
                                                               * açılacağını belirtir*/
    combosehir.Sorted = true; /*alfabetik sıraya göre dizer*/
}
```







# KAPSÜLLEME

```
class futbolcu
{
    public string adi;
    public string takimi;
    public double boyu;
    public char mevkii;
    private int yil;
    private string millet;

    /* private yani sadece class içinde kullanılacak bir
    verinin gizliliğini koruyarak public yapılip erişime açılmasıdır.*/
    public int YIL
    {
        get { return yil; }
        set { yil = Math.Abs(value); } // mutlak değer alır.
    }

    public string MILLET
    {
        get { return millet; }
        set { millet = value.ToUpper(); } // değerin bütün harflerini büyük yapar.
    }
}
```

```
private void Form1_Load(object sender, EventArgs e)
{
    futbolcu ft = new futbolcu(); /* ft adında bir nesne oluşturulur
    ve değerler o nesneye atanır*/
    ft.adi = "Messi";
    ft.takimi = "Barcelona";
    ft.boyu = 1.70;
    ft.mevkii = 'O';
    ft.MILLET = "portekiz";
    ft.YIL = -1981;

    label1.Text = ft.adi;
    label2.Text = ft.takimi;
    label3.Text = ft.boyu.ToString("0.00");
    label4.Text = ft.mevkii.ToString();
    label5.Text = ft.MILLET;
    label6.Text = ft.YIL.ToString();
}
```

## KALITIM

```
namespace class_kalitim
{
    class genel
    {
        private string milleti;
        public string takimi;

        public string MILLETI
        {
            get { return milleti; }
            set { milleti = value; }
        }
    }

    namespace class_kalitim
    {
        /*class fiziksel:genel--- genel class ina yeni bir nesne
        oluşturmadan fiziksel classının nesnesini
        kullanarak oluşturduğumuz değerlerini farklı
        bir classtan fiziksel classın aktardık*/
        class fiziksel:genel
        {
            private string adi;
            public double boyu;
            public char mevkii;
            public string fiyat;

            public string ADI
            {
                get { return adi; }
                set { adi = value; }
            }
        }
    }
}
```

```
private void button1_Click(object sender, EventArgs e)
{
    fiziksel ft = new fiziksel();
    ft.ADI = "RONALDO";
    ft.boyu = 1.89;
    ft.mevkii = 'O';
    ft.fiyati = "10 milyon euro";
    ft.MILLETI = "PORTEKİZ";
    ft.takimi = "REAL MADRID";

    label1.Text = ft.ADI;
    label2.Text = ft.boyu.ToString();
    label3.Text = ft.mevkii.ToString();
    label4.Text = ft.fiyati;
    label5.Text = ft.takimi;
    string genel.takimi
}
```

RONALDO  
1,89  
O  
10 milyon euro

REAL MADRID



GÖSTER

## SINIF İÇİNDE DEĞER DONDURMEYEN METOT

```
İsminiz:cengiz  
cengiz
```

```
namespace sinifcindegeriyedegerdondurmeyemetotlar
{
    class kisibilgileri
    {
        public void bilgiler(string adsoyad)
        {
            Console.WriteLine(adsoyad);
            Console.Read();
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            kisibilgileri ks = new kisibilgileri();
            Console.Write("İsminiz:");
            string ads;
            ads=Console.ReadLine();/* console den girilen
                                         değeri ads ye atama*/
            ks.bilgiler(ads);
        }
    }
}
```

## SINIF İÇİNDE DEĞER DONDUREN METOT

```
class dortislem
{
    public int topla (int sy1, int sy2)
    {
        int sonuc=sy1+sy2;
        Console.WriteLine("Sonuc: "+sonuc);
        Console.Read();
        return sonuc;
    }

    class Program
    {
        static void Main(string[] args)
        {
            dortislem sl= new dortislem();
            {
                sl.topla(7, 8);
                Console.WriteLine("\n\n");/*consol ekranında
                                         boşluk bırakır*/
            }
        }
    }
}
```

## CONSTRUCTOR METOT(YAPICI METOTLAR)

```
class sinif
{
    public sinif(string adi) /* sınıfın adını metoda
                               verdiğinizde nesne türetip
                               metodu kullanabilirsiniz.
                               metodu nesneye kullanmaya gerek
                               kalmaz.*/
    {
        Console.WriteLine(adi);
        Console.Read();
    }
}
```

```
file:///C:/Users/PC/Desktop/1.cs
ADI:CENGİZ YILMAZ
CENGİZ YILMAZ
```

```
class Program
{
    static void Main(string[] args)
    {
        string ADI;
        Console.Write("ADI:");
        ADI = Console.ReadLine();
        sinif og = new sinif(ADI);

    }
}
```

## CONSTRUCTOR METOT 2(YAPICI METOTLAR)

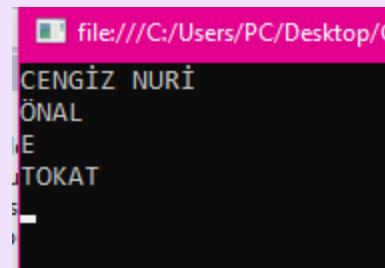
```
class dil
{
    string adi = "Cengiz";
    string soyadi = "";
    char cinsiyet = 'E';
    string memleketi = "";

    public string ADI
    {
        get { return adi; }
        set { adi = value.ToUpper(); }
    }
    public string SOYADI
    {
        get { return soyadi; }
        set { soyadi = value.ToUpper(); }
    }

    public char CINSIYET
    {
        get { return cinsiyet; }
        set { cinsiyet = value; }
    }

    public string MEMLEKET
    {
        get { return memleketi; }
        set { memleketi = value.ToUpper(); }
    }
}
```

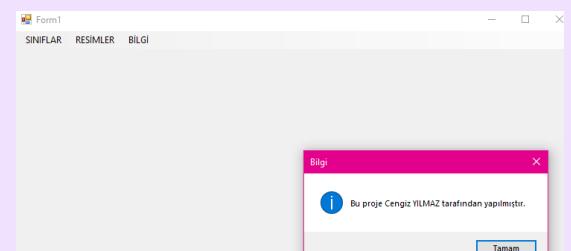
```
class Program
{
    static void Main(string[] args)
    {
        dil dl = new dil();
        dl.ADI = "CENGİZ NURİ";
        dl.SOYADI = "ÖNAL";
        dl.CINSIYET = 'E';
        dl.MEMLEKET = "TOKAT";
        Console.WriteLine(dl.ADI);
        Console.WriteLine(dl.SOYADI);
        Console.WriteLine(dl.CINSIYET);
        Console.WriteLine(dl.MEMLEKET);
        Console.Read();
    }
}
```



## MENU STRİP VE MESSAGEBOX KULLANIMI

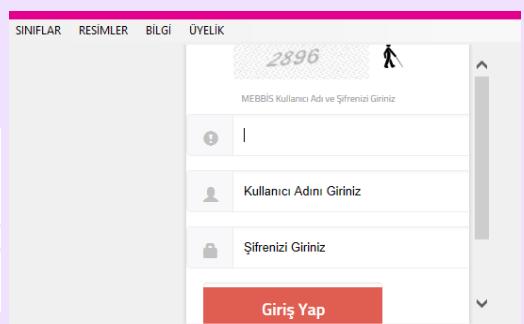
```
private void dOĞARESİMLERİToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Red;
}
private void hAKKIMIZDAToolStripMenuItem1_Click(object sender, EventArgs e)
{
    /*messagebox ta parametre sırası (icerik, başlık, button ne, ikon)dur.*/
    MessageBox.Show("Bu proje Cengiz YILMAZ tarafından yapılmıştır.",
                    "Bilgi",
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Information);
}

private void çIKIŞToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit()/*uygulamadan çıkış*/;
}
```



## WEB BROWSER KULLANIMI

```
private void gİRİŞToolStripMenuItem1_Click(object sender, EventArgs e)
{
    webBrowser1.Navigate("https://mebbis.meb.gov.tr/ssologinBIDB.aspx?id=10&url"
                         + "https://pbs.meb.gov.tr/Hesap/MebbisGiris");
}
```

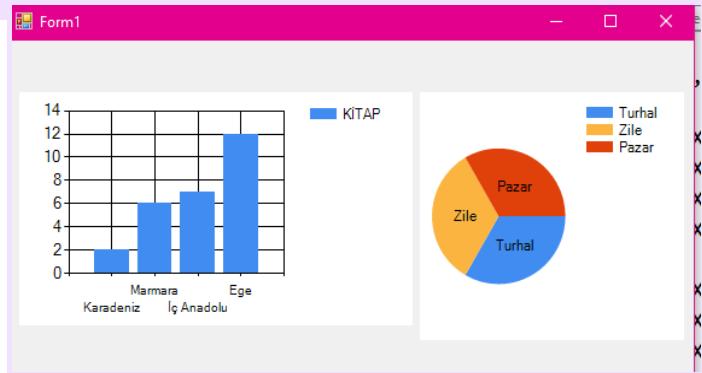


# CHART GRAFİK KULLANIMI

```
private void Form1_Load(object sender, EventArgs e)
{
    chart1.Series["KİTAP"].Points.AddXY("Karadeniz", 2);
    chart1.Series["KİTAP"].Points.AddXY("Marmara", 6);
    chart1.Series["KİTAP"].Points.AddXY("İç Anadolu", 7);
    chart1.Series["KİTAP"].Points.AddXY("Ege", 12);

    chart2.Series["GELİR"].Points.AddXY("Turhal", 120);
    chart2.Series["GELİR"].Points.AddXY("Zile", 120);
    chart2.Series["GELİR"].Points.AddXY("Pazar", 120);
```

```
private void cmbilce_TextChanged(object sender, EventArgs e)
{
    chart1.Series["OYLAR"].Points.Clear(); //combox her değiştiğinde önce temizle
    sonra grafiği doldur
```

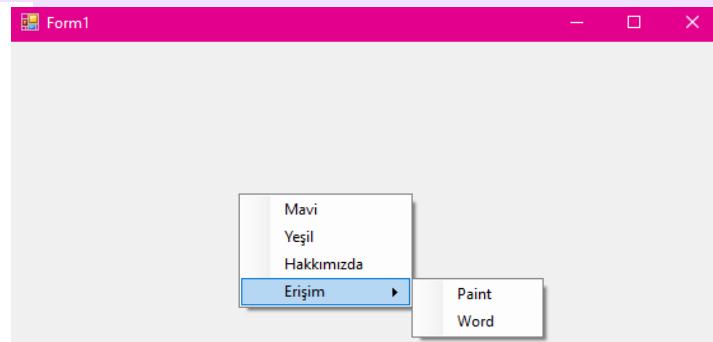


2-system.diagnostics.process.start("mspaint.exe")-Program açma

```
private void yeşilToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Yellow;
}

private void hakkımızdaToolStripMenuItem_Click(object sender, EventArgs e)
{
    MessageBox.Show("Bu proje Cengiz Yılmaz tarafından geliştirilmiştir",
        "Projemiz", MessageBoxButtons.OK, MessageBoxIcon.Information);
}

private void paintToolStripMenuItem_Click(object sender, EventArgs e)
{
    System.Diagnostics.Process.Start("mspaint.exe"); /* bilgisayarda kayıtlı
        bir programı açar*/
}
```



# RANDOM(rastgele) KULLANIMI

```
private void button1_Click(object sender, EventArgs e)
{
    Random rnd = new Random(); /*random sınıfında nesne oluştur*/
    int sy1, sy2, sy3;
    sy1 = rnd.Next(1, 10); /* 1-10 arasında 10 dahil değil değer atar*/
    sy2 = rnd.Next(1, 10);
    sy3 = rnd.Next(1, 10);
    label1.Text = sy1.ToString();
    label2.Text = sy2.ToString();
    label3.Text = sy3.ToString();
}
```

4            2            8

button1

# CAPTCHA KULLANIMI

```
public void islem()
{
    Random rnd = new Random();
    char[] sembol = { '*', '+', '/', '$', '!', '?' };
    string[] bykhrf = { "A", "C", "Ç", "K", "Z" };
    string[] kckhrf = { "f", "l", "m", "n", "x" };
    int sy1 = rnd.Next(1, 10);
    int sy2 = rnd.Next(1, 10);

    int sb = rnd.Next(1, sembol.Length);
    int bh = rnd.Next(1, bykhrf.Length);
    int kh = rnd.Next(1, kckhrf.Length);

    label1.Text = sembol[sb].ToString() + bykhrf[bh].ToString() +
    kckhrf[kh].ToString() + sy1.ToString() + sy2.ToString();
}
```

/Zn46

DEĞİŞTİR

GİRİŞ

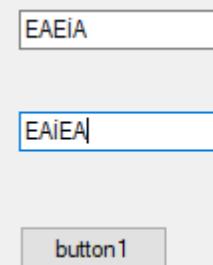
## FORMLAR ARASI GEÇİŞ

```
Form2 frm=new Form2();
frm.Show();      /* frm nesnesini bağlı form sınıfını göster
                   */
this.Hide(); /*bu formu gizle*/
```

## FORMLAR ARASI GEÇİŞ-VERİ AKTARIMI

```
public Form1()
{
    InitializeComponent();

private void button1_Click(object sender, EventArgs e)
{
    Form2 frm = new Form2();
    frm.adi = textBox1.Text;
    frm.mesaj = textBox2.Text;
    frm.Show();
    this.Hide();
}
```



```
public Form2()
{
    InitializeComponent();
}
public string adi;
public string mesaj;

private void Form2_Load(object sender, EventArgs e)
{
    label1.Text = adi;
    label2.Text = mesaj;    }
```



## POINT KULLANIMI(YER)

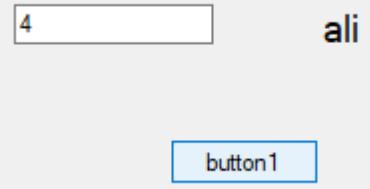
```
Point konum = new Point();

private void timer1_Tick(object sender, EventArgs e)
{
    konum.X = pictureBox1.Location.X;
    konum.Y = pictureBox1.Location.Y;
    konum.X = konum.X + 10;
    pictureBox1.Location = konum;
```

## ENUM YAPISI

```
enum ogrenciler { x, ahmet, mehmet, hüseyin, ali }

private void button1_Click(object sender, EventArgs e)
{
    int numara = Convert.ToInt16(textBox1.Text);
    ogrenciler o; /* ogrenciler enumu içinden o değişkeni*/
    o = (ogrenciler)numara; /*o değişkeni ogrenciler enumundaki
                           numaranıncı parametre*/
    label1.Text = o.ToString();/*o yu göster*/
}
```



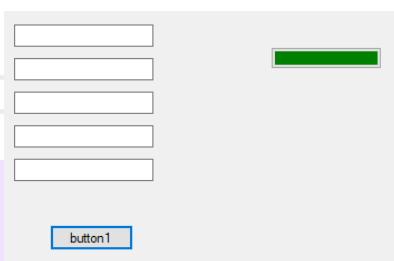
## MATEMATİK FONKSİYONLARI

```
private void button1_Click(object sender, EventArgs e)
{
    double sayı = Convert.ToDouble(textBox1.Text);
    double sayı2 = Convert.ToDouble(textBox2.Text);
    label1.Text = Math.Abs(sayı).ToString();
    label1.Text = Math.Ceiling(sayı).ToString();
    label1.Text = Math.Floor(sayı).ToString();
    label1.Text = Math.Max(sayı,sayı2).ToString();
    label1.Text = Math.Pow(sayı,2).ToString();
    label1.Text = Math.Sqrt(sayı).ToString();
}

/*
 *ABS=mutlak değer,
 *ceiling=yukarı yuvarla
 *floor=aşağı yuvarla
 *max, min=iki sayı arasındaki en büyü en küçük değer
 *pow= üs alma. birinci parametre sayı ikinci üs değeri
 *sqrt=karekök alır.
 */
```

## DİNAMİK ARAÇLAR

```
private void button1_Click(object sender, EventArgs e)
{
    Button btn = new Button();
    Point btnknm = new Point(250, 50);
    btn.Location = btnknm;
    btn.BackColor = Color.Green;
    btn.Name = "button2";
    btn.Height = 20;
    btn.Width = 100;
    this.Controls.Add(btn);
}
```



```
for(int i=1;i<=5;i++)
{
    TextBox txt = new TextBox();
    Point txtknm = new Point(20, i*30);
    txt.Location=txtknm;
    txt.Width = 125;
    txt.Height = 20;
    txt.Name="textBox"+i;
    this.Controls.Add(txt);
}
```

# SQL

## VERİ TİPLERİ

- \* bit -----> ikili sonuçları tutar. 0 ve 1 ya da true veya false
- \* char() -----> Sabit uzunluklu verileri tutar. Or. telefon numarası ya da tc.
- decimal(18,0) -----> Ondalıklı sayıları tutar. Double gibi.
- decimal(x,y) x virgülden önceki ve virgülden sonraki toplam basamak sayısı, y virgülden sonraki basamak sayısı. Ör. decimal(5,2)----->100,00**
- \* smalldatetime -----> 1700 ila 2070 arasındaki zaman aralığını kapsar
- smallint -----> +- 32 bin toint16 gibi
- text -----> çok uzun metin değerleri içindir
- int -----> tam sayılar için kullanılır.
- \* tinyint -----> 0-256 arasındaki sayılar
- \* varchar -----> değişken uzunluklu veriler için kullanılır. örneği ad soyad gibi.

## DML(DATA MANIPULATION)

Select(sec)  
Insert(ekle)  
Delete(sil)  
Update(güncelle)

\* ----> Tümünü, all  
from ----> den dan  
Querry

The screenshot shows a SQL query window with the following code:

```
SQLQuery1.sql - M...(MUDYARD\PC (55))* X MUDYARD\SQLEXPRESS...ani - dbo.Table_1
select * from dbo.Table_1/*dbo.Table_1 tablosundaki
bütün değerleri seç */
```

The results pane displays a table with 4 rows and 7 columns:

	perid	perad	persoyad	persehir	pemaas	permedeni	pemeslek
1	1	Cengiz	Yilmaz	Tokat	5000	1	öğretmen
2	2	Ayşenur	Yilmaz	Malatya	6000	1	öğretmen
3	3	Selim	Yilmaz	Tokat	8000	0	doktor
4	4	Ali	Ruhi	Bitlis	4000	0	ışçi

where ---> if yerine  
and ---> ve  
or --->veya

The screenshot shows a SQL query window with the following code:

```
SQLQuery1.sql - M...(MUDYARD\PC (53))* X MUDYARD\SQLEXPRESS...ani - dbo.Table_1
select * from dbo.Table_1
where
permedeni=0 and persehir='Tokat' or
pemaas>=6000
```

The results pane displays a table with 2 rows and 7 columns:

	perid	perad	persoyad	persehir	pemaas	permedeni	pemeslek
1	2	Ayşenur	Yılmaz	Malatya	6000	1	öğretmen
2	3	Selim	Yılmaz	Tokat	8000	0	doktor

## SQL BAĞLANTI KURMA

### İNSERT VERİ GİRME

```
using System.Data.SqlClient; /*sql kütüphanesini ekle*/\n\nnamespace personelkayit\n{\n    public partial class Form1 : Form\n    {\n        public Form1()\n        {\n            InitializeComponent();\n        }\n\n        SqlConnection baglanti = new SqlConnection(\n            "Data Source=MUDYARD\\SQLEXPRESS;Initial Catalog=Personelveritabani;Integrated Security=True");\n        /*sqlconnection sınıfında baglanti nesnesi oluştur ve parantez içine baglanti adresini yaz\n        .adreste / var onu // yap ve - nin iki yanını boş bırakma*/\n    }\n}
```

```
private void btnliste_Click(object sender, EventArgs e)\n{\n    this.table_1TableAdapter1.Fill(this.personelveritabaniDataSet2.Table_1);\n}\n\nprivate void btnkaydet_Click(object sender, EventArgs e)\n{\n    baglanti.Open(); /*baglantiya gir*/\n    SqlCommand komut = new SqlCommand(\n        /*table_1 tablosu içinden perad ve persoyad sütunlarından p1 ve p2 parametresi oluştur\n        * */\n        "Insert into Table_1 (perad, persoyad) values(@p1,@p2)", baglanti);\n    /*sqlCommand sınıfından komut nesnesi oluştur. parantez içini yaz*/\n\n    komut.Parameters.AddWithValue("@p1", TxtAd.Text);\n    /*komut nesnesinden p1 parametresine değer ekle*/\n    komut.Parameters.AddWithValue("@p2", TxtSoyad.Text);\n    /*komut nesnesinden p2 parametresine değer ekle*/\n    komut.ExecuteNonQuery();\n    /*executenonQuery komutuyla ekle, sil ve kaydet için çalıştırırız*/\n    baglanti.Close();\n    MessageBox.Show("Personel eklendi.");\n}
```

perid	perad	persoyad	persehir	permaas	permemedi	permeslek	...
4	Ali	Ruhi	Bitlis	4000	<input type="checkbox"/>	<input type="checkbox"/>	İşçi
5	aeae	aeae			<input type="checkbox"/>	<input type="checkbox"/>	
6	EAEA	EAEA			<input type="checkbox"/>	<input type="checkbox"/>	

### VERİLERİ ÇEKME

```
private void /*datagridin event özelliklerinden celldoubleclick\n          olduğunda seçilir.*/\n    dataGridView1_CellDoubleClick(object sender, DataGridViewCellEventArgs e)\n    {\n        int secilen = dataGridView1.SelectedCells[0].RowIndex;\n        /*datagridview1 in seçilen hücresinin 0. indexi*/\n\n        msktextid.Text = dataGridView1.Rows[secilen].Cells[0].Value.ToString();\n        /*datagridin seçilen sırasının 0. hücresinin değerini msktextid ye ata*/\n        TxtAd.Text = dataGridView1.Rows[secilen].Cells[1].Value.ToString();\n        TxtSoyad.Text = dataGridView1.Rows[secilen].Cells[2].Value.ToString();\n        combosehir.Text = dataGridView1.Rows[secilen].Cells[3].Value.ToString();\n        mskmaas.Text = dataGridView1.Rows[secilen].Cells[4].Value.ToString();\n        label8.Text = dataGridView1.Rows[secilen].Cells[5].Value.ToString();\n\n    }\n\n    private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)\n    {\n        mskid.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();\n        txtkulupad.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();\n        // e datagridin cursor olan yani farenin üzerinde olduğu yerdir. e.RowIndex üzerinde\n        //olunan yerin sıra indeksi\n    }
```

## KAYIT SILME

```
private void btnsil_Click(object sender, EventArgs e)
{
    baglanti.Open();
    SqlCommand sil = new SqlCommand("Delete From Table_1 Where perid=@k1", baglanti);
    sil.Parameters.AddWithValue("@k1", msktextid.Text);
    sil.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("Kayıt silindi");
}
```

## KAYIT GUNCELLEME

```
baglanti.Open();
SqlCommand komutguncelle = new SqlCommand("Update Table_1 Set perad=@a1, persoyad=@a2, persehir=@a3, permaas=@a4, permedeni=@a5, permeslek=@a6 where perid=@a7", baglanti);
komutguncelle.Parameters.AddWithValue("@a1", TxtAd.Text);
komutguncelle.Parameters.AddWithValue("@a2", TxtSoyad.Text);
komutguncelle.Parameters.AddWithValue("@a3", combosehir.Text);
komutguncelle.Parameters.AddWithValue("@a4", mskmaas.Text);
komutguncelle.Parameters.AddWithValue("@a5", label8.Text);
komutguncelle.Parameters.AddWithValue("@a6", txtmeslek.Text);
komutguncelle.Parameters.AddWithValue("@a7", msktextid.Text);
komutguncelle.ExecuteNonQuery();
baglanti.Close();
MessageBox.Show("Kayıt güncellendi.");
```

## INSERT(EKLEME)

```
baglanti.open();
sqlcommand komut= new sqlcommand("insert into Table_1 (peid, perad, persoyad) values (@p1, @p2, @p3), baglanti);
komut.parametres.addwithvalue("@p1, textid.text");
komut.parametres.addwithvalue("@p2", textad.text);
komut.parametres.addwithvalue("@p3", textsoyad.text);
komut.executenonequery();
baglanti.close();
```

## DELETE

```
baglanti.open();
sqlcommand kmtsil= new sqlcommand("delete from table_1 where perid=@p1",baglanti);
kmtsil.parameters.Addwithvalue("@p1", textid.text);
kmtsil.executenonequery();
baglanti.close();
```

## UPDATE

```
baglanti.open();
sqlcommand kmtguncel= new sqlcommand("update Table_1 set perad=@a2, persoyad=@a3 where perid=@a1", baglanti);
kmtguncel.parameters.addwithvalue("@a2", textad.text);
kmtguncel.parameters.addwithvalue("@a3", textsoya.text);
kmtguncel.parameters.addwithvalue("@a1", textid.text);
kmtguncel.executenonequerry();
baglanti.close();
```

# SELECT İŞLEMLERİ

*count= sayıç, sayıyor  
sum= topla,  
avg=ortalama al  
distinct= tekrarlıları ayır*

*select count(\*) from table\_1 -----> table\_1 deki bütün kayıtları say  
select count(distinct(persehir)) from table\_1 -----> persehirdeki bütün şehirleri tekrar etmeden say  
select sum(permaas) from table\_1 -----> permaas sütunundaki bütün maaşları topla  
select avg(permaas) from table\_1 -----> permaas sütunundaki değerlerin ortalamasını al*

```
//toplam sayi
baglanti.Open();
SqlCommand komut1 = new SqlCommand("select count(*) from Table_1", baglanti);
SqlDataReader r1 = komut1.ExecuteReader();
/*data oku sınıfından nesne oluşturup onu komut1 nesnesindeki okuma komutuna
lar.*/
while (r1.Read())/*r1 nesnesi okudu müddetçe*/
{
    lblpersonelsayisi.Text = r1[0].ToString();
}
baglanti.Close();
```

```
//evliistatistik
baglanti.Open();
SqlCommand komut2 = new SqlCommand("select count(*) from Table_1 where permedeni=1", baglanti);
SqlDataReader b1 = komut2.ExecuteReader();
while (b1.Read())
{
    lblevlisayisi.Text = b1[0].ToString();
}
baglanti.Close();
```

```
//ortalamaaas
baglanti.Open();
SqlCommand komut6 = new SqlCommand("select avg(permaas) from table_1", baglanti);
SqlDataReader rd6 = komut6.ExecuteReader();
while (rd6.Read())
{
    lblortmaas.Text = rd6[0].ToString();
}
baglanti.Close();
```

```
//sehirsayisi
baglanti.Open();
SqlCommand komut4 = new SqlCommand("select count(distinct(persehir)) from table_1", baglanti);
SqlDataReader rd4 = komut4.ExecuteReader();
while (rd4.Read())
{
    lblsehirsayisi.Text = rd4[0].ToString();
}
baglanti.Close();
```

```
// toplam maas
baglanti.Open();
SqlCommand komut5 = new SqlCommand("select sum(permaas) from table_1", baglanti);
SqlDataReader rd5 = komut5.ExecuteReader();
while (rd5.Read())
{
    lbltoplammaas.Text = rd5[0].ToString();
}
baglanti.Close();
```

Toplam Personel Sayısı:	7
Evli Personel Sayısı:	3
Bekar Personel Sayısı:	4
Toplam Şehir Sayısı:	5
Toplam Maaş:	34922
Ortalama Maaş:	4988

## SQL SORGULAMASINDA GROUP BY KULLANIMI

```
- select permeslek, avg(permaas) from Table_1 group by permeslek
  select permedeni, count(*) from Table_1 where permedeni=1 group by permedeni
  select persehir, count(*) from Table_1 where persehir='Tokat' group by persehir
  select permeslek, sum(permaas) from Table_1 group by permeslek
```

permeslek	(No column name)
avukat	5000
doktor	5211
hemşire	4500
mühendis	8000
öğretmen	3500

permedeni	(No column name)
1	3

persehir	(No column name)
TOKAT	3

permeslek	(No column name)
avukat	5000
doktor	10422
hemşire	4500
mühendis	8000
öğretmen	7000

```

baglanti.Open();
SqlCommand komutgrf2evli = new SqlCommand("select permedeni, count(*) from table_1 where permedeni=1 group by permedeni");
SqlDataReader grf2ev = komutgrf2evli.ExecuteReader();
while (grf2ev.Read())
{
    chart2.Series["Evli-Bekar"].Points.AddXY("Evli", grf2ev[1]);
}
baglanti.Close();
baglanti.Open();
SqlCommand komutgrf2bekar = new SqlCommand("select permedeni, count(*) from table_1 where permedeni=0 group by permedeni");
SqlDataReader grf2be = komutgrf2bekar.ExecuteReader();
while (grf2be.Read())
{
    chart2.Series["Evli-Bekar"].Points.AddXY("Bekar", grf2be[1]);
}
baglanti.Close();

```

```

baglanti.Close();
baglanti.Open();
SqlCommand komutgrf2bekar = new SqlCommand("select permedeni, count(*) from table_1 where permedeni=0 group by permedeni");
SqlDataReader grf2be = komutgrf2bekar.ExecuteReader();
while (grf2be.Read())
{
    chart2.Series["Evli-Bekar"].Points.AddXY("Bekar", grf2be[1]);
}
baglanti.Close();

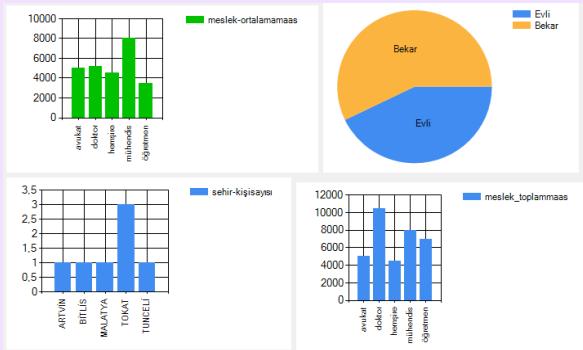
//sehirkisisayisi
baglanti.Open();
SqlCommand komutgrf3 = new SqlCommand("select persehir, count(*) from table_1 group by persehir", baglanti);
SqlDataReader grf3 = komutgrf3.ExecuteReader();
while (grf3.Read())
{
    chart3.Series["sehir-kisisayisi"].Points.AddXY(grf3[0], grf3[1]);
}
baglanti.Close();

```

```

//meslektoplammaas
baglanti.Open();
SqlCommand komutgrf4 = new SqlCommand("select permeslek, sum(permaas) from table_1 group by permeslek", baglanti);
SqlDataReader grf4 = komutgrf4.ExecuteReader();
while (grf4.Read())
{
    chart4.Series["meslek_toplammaas"].Points.AddXY(grf4[0], grf4[1]);
}
baglanti.Close();

```



## KULLANICI ADMIN GİRİŞ ŞİFRE OLUŞTURMA

```

baglanti.Open();
SqlCommand kullanicigiris = new SqlCommand("select * from table_2 where kullaniciadi=@p1 and kullanicişifre=@p2", baglanti);
kullanicigiris.Parameters.AddWithValue("@p1", txtkullaniciadi.Text);
kullanicigiris.Parameters.AddWithValue("@p2", txtsifre.Text);
SqlDataReader giris = kullanicigiris.ExecuteReader();
if (giris.Read())
{
    frmannaform frмана = new frmannaform();
    frмана.Show();
    this.Hide();
}

else
{
    MessageBox.Show("Yanlış kullanıcı adı ve şifre girdiniz");
    txtkullaniciadi.Text = "";
    txtsifre.Text = "";
    txtkullaniciadi.Focus();
}
baglanti.Close();

```

## DATATABLE VE DATAADAPTER İLE TABLOYA ŞARTLI VE ŞARTSIZ VERİ AKTARMA

```

DataTable dt1 = new DataTable();
SqlCommand komut1 = new SqlCommand("select * from Tbl_Randevular where HastaTC=" + hastanintc, bgl.baglanti());
SqlDataAdapter da1 = new SqlDataAdapter();
da1.SelectCommand = komut1;
da1.Fill(dt1);
dataGridViewlisteleme.DataSource = dt1;
bgl.baglanti().Close();
//sqlcomman kullanarak şartlı ifade yazma

DataTable dt2=new DataTable();
SqlCommand komut3=new SqlCommand("select * from Tbl_Randevular where RandevuBrans=@rnbr1 and RandevuDoktor=@rnd2 and RandevuDurum='True'", bgl.baglanti());
komut3.Parameters.AddWithValue("@rnbr1", cmbbrans.Text);
komut3.Parameters.AddWithValue("@rnd2", cmbdoktor.Text);
SqlDataAdapter da2=new SqlDataAdapter();
da2.SelectCommand=komut3;
da2.Fill(dt2);
dataGridView2.DataSource=dt2;

//sadece dataadapter kullanarak şartlı ifade yazma
DataTable dt3 = new DataTable();
SqlDataAdapter da3 = new SqlDataAdapter("select * from Tbl_Randevular where RandevuBrans='"+cmbbrans.Text+"' and RandevuDoktor='"+cmbdoktor.Text+"' and RandevuDurum='True'", bgl.baglanti());
da3.Fill(dt3);
dataGridView2.DataSource = dt3;

```

## DATABASE DA YAZDIKÇA ARANAN VERİ OLUŞTURMA

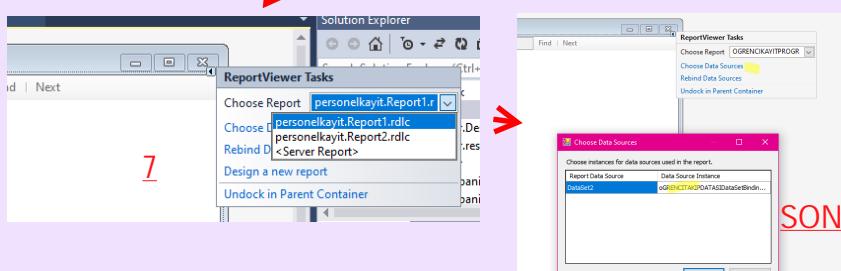
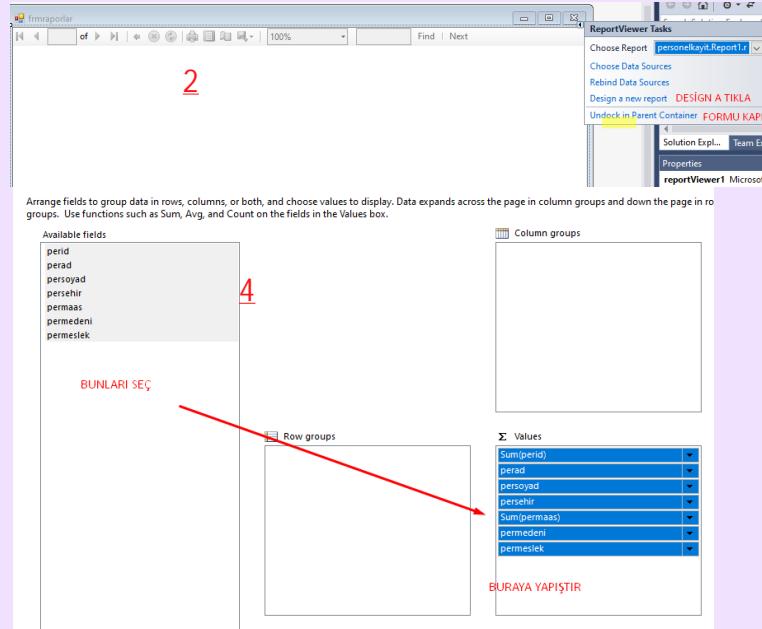
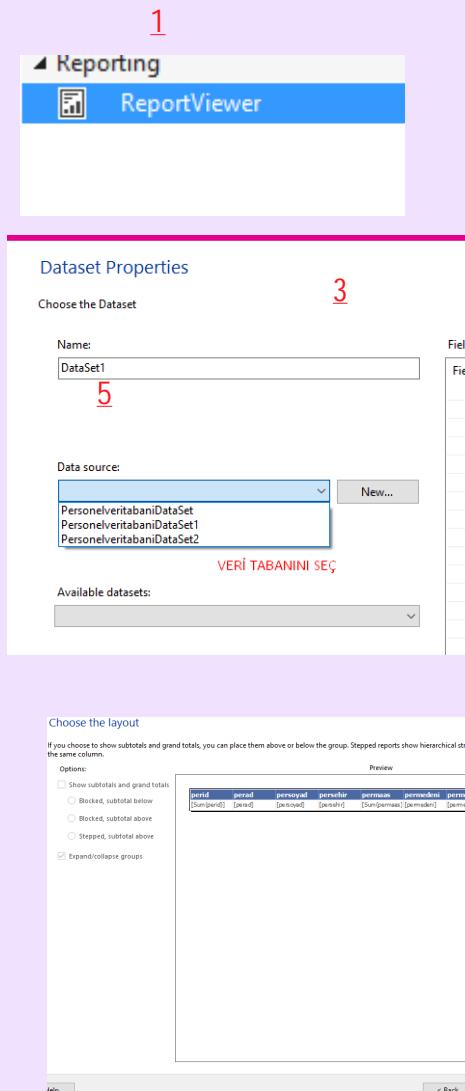
```

private void txtbul_TextChanged(object sender, EventArgs e)
{
    baglanti.Open();
    DataTable dt3 = new DataTable();
    OleDbCommand komut6 = new OleDbCommand("Select * from Tbl_Kitaplik where Kitapad like '%" + txtbul.Text + "%'", baglanti);
    OleDbDataAdapter da3 = new OleDbDataAdapter(komut6);
    da3.Fill(dt3);
    datagridview1.DataSource = dt3;
    baglanti.Close();
}

```

("Select \* from Tbl\_Kitaplik where Kitapad like '%"+txtbul.Text+"%', baglanti);  
 ("Select \* from Tbl\_Kitaplik where Kitapad like '%a%', baglanti);  
 //ortada a geçenleri ara//  
 ("Select \* from Tbl\_Kitaplik where Kitapad like 'a%', baglanti);  
 //başta a geçenleri ara//  
 "Select \* from Tbl\_Kitaplik where Kitapad like '%%a%', baglanti);  
 //sonda a geçenleri ara//

## RAPORLAMA



SON

## DOSYA İŞLEMLERİ

```

 OpenFileDialog opdialog = new OpenFileDialog();
 opdialog.Title = "Lütfen bir resim seçiniz";
 opdialog.Filter = "|Png dosyası|*.png|jpg dosyası|*.jpg|";
 opdialog.ShowDialog();

 /* Bilgisayar içindeki bir dosyayı açmanızı yarar.*/

 SaveFileDialog svdialog = new SaveFileDialog();
 svdialog.Title = "Lütfen bir pdf dosyası seçiniz";
 svdialog.Filter = "|Pdf dosyası|*.pdf|Excel dosyası|*.xls|";
 svdialog.ShowDialog();

 /* Bilgisayar içindeki bir dosyayı servere kaydetmenize yarar.*/

 FolderBrowserDialog flddialog = new FolderBrowserDialog();
 flddialog.ShowDialog();

 /*bilgisayar içindeki dosyalarda gezinmeye yarar*/

```

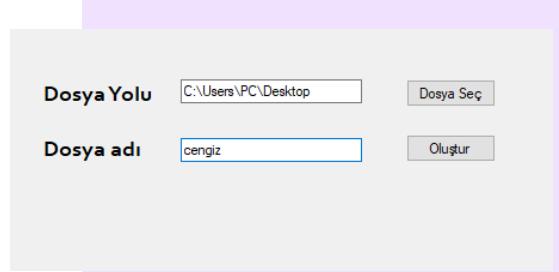
## METİN BELGESİ OLUŞTURMA

### stream.writer

```

using System.Windows.Forms;
using System.IO;
/*10 sistem kütüphanesini ekle*/
string dosyaadi, dosyayolu;
StreamWriter st;
/*yazma oluşturma aktarımı*/
private void btnsec_Click(object sender, EventArgs e)
{
    if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)
        /*folderbrowserdialog1 açılıp okey tuşuna basılırsa*/
    {
        dosyayolu = folderBrowserDialog1.SelectedPath.ToString();
        /*dosyayolu değişkenini folderbrowserdialog1 den gelen seçim yolunun
         metin değerine eşitle*/
        textBox1.Text = dosyayolu;
    }
}
private void btnolustur_Click(object sender, EventArgs e)
{
    dosyaadi=textBox2.Text;
    st = File.CreateText(dosyayolu + "//" + dosyaadi + ".docx");
    /*yazmaakımı şudur: dosyayolu belirtilmiş yere dosyaadında bir docx
     * uzantılı belge oluştur*/
    st.Close();/*yazmayı kapat*/
}

```



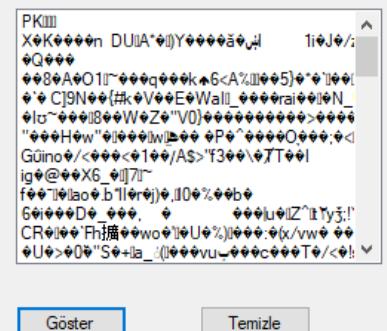
## METİN BELGESİ OKUMA

### streamreader

```

using System.Windows.Forms;
using System.IO;
/*10 sistem kütüphanesini ekle*/
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        StreamReader str = new StreamReader(openFileDialog1.FileName);
        string satir=str.ReadLine();
        while(satir!=null)
        {
            listBox1.Items.Add(satir);
            satir=str.ReadLine();
        }
    }
}

```

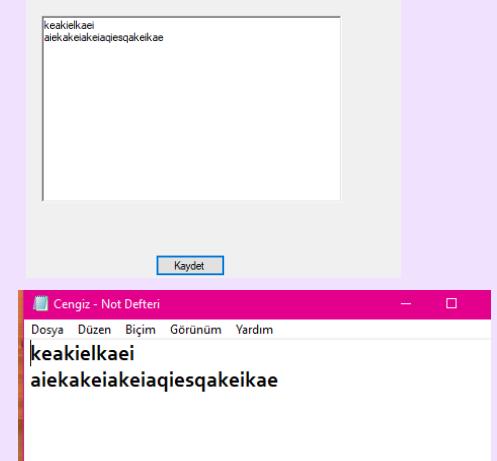


## METİN BELGESİ KAYDETME

```

using System.Windows.Forms;
using System.IO;
/*10 sistem kütüphanesini ekle*/
private void button1_Click(object sender, EventArgs e)
{
    saveFileDialog1.Filter = "Metin Belgesi|*.txt";
    saveFileDialog1.Title = "Metin Belgesi Oluştur";
    saveFileDialog1.ShowDialog();
    StreamWriter str = new StreamWriter(saveFileDialog1.FileName);
    str.WriteLine(richTextBox1.Text);
    str.Close();
    MessageBox.Show("Kaydedildi.");
}

```

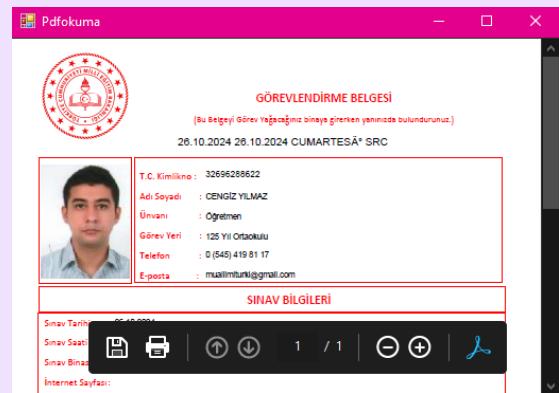


## PDF AÇMA OKUMA

Toolbox

ado| Adobe PDF Reader

```
private void Pdfokuma_Load(object sender, EventArgs e)
{
    OpenFileDialog ac = new OpenFileDialog();
    if (ac.ShowDialog() == DialogResult.OK)
    {
        axAcroPDF1.LoadFile(ac.FileName);
    }
}
```



## TRY CATCH KULLANIMI

```
private void button1_Click(object sender, EventArgs e)
{
    // try kısmı şartlar sağlandığında bunu yap der.
    //catch kısımı programda logaritmik bir boşluk(hata) varsa bunu yap der ve
    // programın patlamasını engeller.
    //exception hatanın detayını bize söyler
    //finally try veya catch ten hangisi çalışırsa çalışın sonunda çalışan kısımdır.
    try
    {
        int sayi1 = Convert.ToInt16(maskedTextBox1.Text);
        int sayi2 = Convert.ToInt16(maskedTextBox2.Text);
        label3.Text = (sayi1 + sayi2).ToString();
    }
    catch (Exception Hata)
    {
        MessageBox.Show(Hata.ToString());
    }
    finally
    {
        MessageBox.Show("Program çalıştı.");
    }
}
```

## DİALOG RESULTU KULLANMA(messagebox ta verilen yanıta göre işlem yapırma)

```
DialogResult result2 = MessageBox.Show(txtad.Text + " adlı mezun bilgileri güncellenecek onaylıyor musunuz?", "Bilgi", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
if (result2 == DialogResult.Yes)
{
    try
    {
        komut4.ExecuteNonQuery();
        MessageBox.Show("Kayıt güncellendi", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception hata)
    {
        MessageBox.Show(hata.ToString());
    }
}
```

# WORD İŞLEMLERİ(belge oluşturma)

```
using word = Microsoft.Office.Interop.Word;    office word kütüphanesini ekliyoruz
using System.Reflection;      reflection sınıfını ekliyoruz(reflection sistemdeki programların uyumlu çalışmasını sağlar)
```

```
object omissing = System.Reflection.Missing.Value;    reflection sınıfından omissing adında bir obje türetiyoruz
object dokumansonu = "\\endofdoc";      dökümanın sonuna yazıyı eklemek için dokumansonu adında bir obje türetiyoruz.
```

```
word.Application olustur = new word.Application();   word kütüphanesinden oluştur adında bir uygulama oluşturuyoruz.
word.Document icerik;   word kütüphanesinden içerik adında bir döküman oluşturuyoruz.
olustur.Visible = true;  uygulamayı görünür kıliyoruz.
icerik = olustur.Documents.Add(ref omissing);  dökümanı uygulamaya ekliyoruz. omissing reflectionı kullanıyor.
```

```
word.Paragraph paragraf1;  word kütüphanesinden bir paragraf oluşturuyoruz.
paragraf1 = icerik.Content.Paragraphs.Add(ref omissing);  paragrafi içerik dokümanın içeriğine ekliyoruz.
paragraf1.Range.Text = "  paragrafin text ini belirliyoruz                                DİPLOMA BELGESİ";
paragraf1.Range.Font.Size = 16;  fontu
paragraf1.Range.Font.Name = "Times New Roman";
paragraf1.Range.Font.Bold = 1;
paragraf1.Format.SpaceAfter = 10;  paragraftan sonra alt satırda ne kadar boşluk olacağını belirliyoruz.
paragraf1.Range.InsertParagraphAfter();  paragrafi en sona ekliyoruz.
```

```
word.Paragraph paragraf2;
object hedef = icerik.Bookmarks.get_Item(ref dokumansonu).Range;  hedef adında obje oluşturup bunu içerik nesnesinin sonuna boşluk bırakır
paragraf2 = icerik.Content.Paragraphs.Add(ref hedef);  paragraf2 yi içerik dokumanına hedef objesini kullanarak ekliyoruz.
paragraf2.Range.Text = "Sayı:737891/198.04/" + belgesayi.ToString();
paragraf2.Range.Font.Size = 12;
paragraf2.Range.Font.Name = "Times New Roman";
paragraf2.Format.SpaceAfter = 15;
paragraf2.Range.InsertParagraphAfter();
```

T  
A  
B  
L  
O  
L  
U  
Ş  
T  
U  
R  
M  
A

```
word.Table tablo;
word.Range wrng = icerik.Bookmarks.get_Item(ref dokumansonu).Range;
tablo = icerik.Tables.Add(wrng, 16, 2, ref omissing, ref omissing);
tablo.Range.ParagraphFormat.SpaceAfter = 5;
tablo.Range.Font.Size = 12;
tablo.Range.Font.Name = "Times New Roman";
int r, c;
for(r=1; r<=16; r++)
{
    for (c = 1; c <= 2; c++)
    {
        }
    tablo.Cell(1, 1).Range.Text = "T.C. KİMLİK NUMARASI:";
    tablo.Cell(1, 2).Range.Text = msktc.Text;
    tablo.Cell(2, 1).Range.Text = "ADI SOYADI:";
```

```
tablo.Cell(2, 2).Range.Text = txtad.Text + " " + txtsoyad.Text;
tablo.Cell(3, 1).Range.Text = "BABA ADI:";
```

```
tablo.Cell(3, 2).Range.Text = txtbaba.Text;
tablo.Cell(4, 1).Range.Text = "ANNE ADI:";
```

```
tablo.Cell(4, 2).Range.Text = txtanne.Text;
tablo.Cell(5, 1).Range.Text = "DOĞUM YERİ VE YILI:";
```

```
tablo.Cell(5, 2).Range.Text = txtdogumyeri.Text + " " + txtdogumtarihi.Text;
tablo.Cell(6, 1).Range.Text = "MEZUN OLDUĞU OKUL:";
```

```
tablo.Cell(6, 2).Range.Text = "125. YIL ORTAOKULU (GÖZTEPE ORTAOKULU)"
```

```
tablo.Cell(7, 1).Range.Text = "OKULUN BULUNDUĞU İL/İLÇE:";
```

```
tablo.Cell(7, 2).Range.Text = "SULTANGAZİ/İSTANBUL";
```

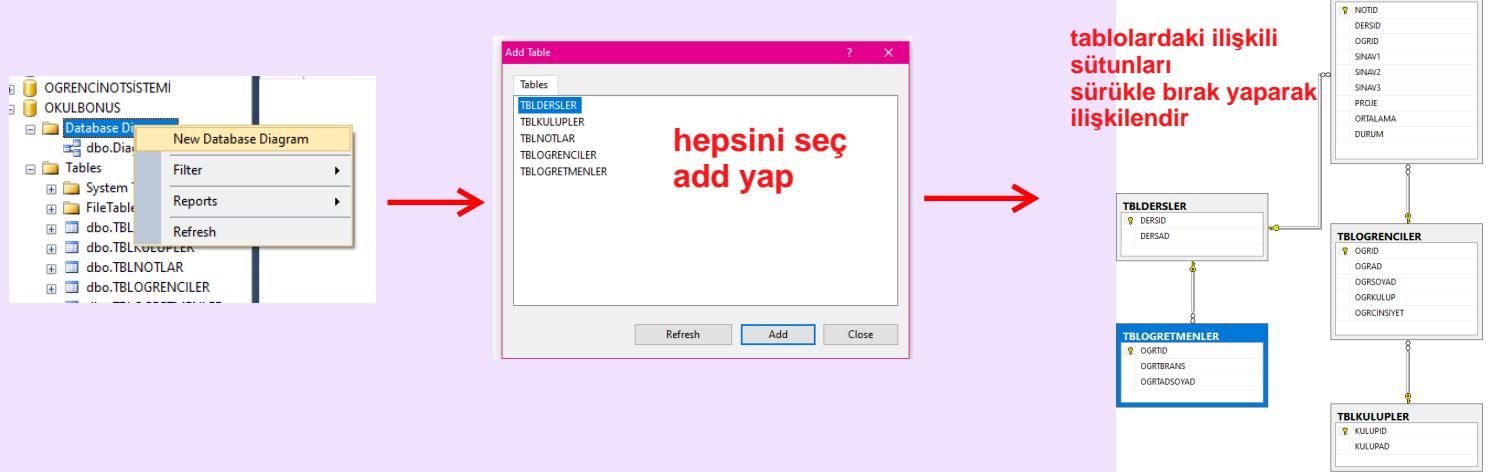
```
word.Paragraph paragraf8;
hedef = icerik.Bookmarks.get_Item(ref dokumansonu).Range;
paragraf8 = icerik.Content.Paragraphs.Add(ref hedef);
paragraf8.Range.Text = " ";
paragraf8.Range.Font.Size = 12;
paragraf8.Range.Font.Name = "Times New Roman";
paragraf8.Format.SpaceAfter = 15;
paragraf8.Range.InsertParagraphAfter();
```

```
tablo.Cell(16, 1).Range.Text = "BELGENİN DÜZENLENME TARİHİ:";
```

```
tablo.Cell(16, 2).Range.Text = dtpbelgeduzentarih.Value.ToString("d");
```

# SQL İLİŞKİLİ TABLOLAR

## sql diyagram oluşturma



Başa tablodan datagridde diyagramı kullanarak sütun ekleme ve gösterme-INNER JOIN

```
MUDYARD\SQLExpr...NUS - Diagram_1*      MUDYARD\SQLEXP...o.TBLOGRENCILER      SQLQuery1.sql - M... (MUDYARD\PC (55))*  X  MUDYARD\SQLExpr...- dbo.TBLNOTLAR
select DERSAD, NOTID, SINAV1, SINAV2, SINAV3, ORTALAMA FROM TBLNOTLAR
INNER JOIN TBLDERSLER ON TBLNOTLAR.DERSID=TBLDERSLER.DERSID WHERE OGRID=1
```

	DERSAD	NOTID	SINAV1	SINAV2	SINAV3	ORTALAMA
1	VERİ YÖNETİMİ	1	55	55	NULL	NULL

```
SqlCommand komut1 = new SqlCommand("select DERSAD, NOTID, SINAV1, SINAV2, SINAV3, PROJE, ORTALAMA from TBLNOTLAR INNER
JOIN TBLDERSLER ON TBLNOTLAR.DERSID=TBLDERSLER.DERSID where OGRID=@P1", bgl);
komut1.Parameters.AddWithValue("@P1", label1.Text);
DataTable dt1 = new DataTable();
SqlDataAdapter da1 = new SqlDataAdapter(komut1);
da1.Fill(dt1);
dataGridView1.DataSource = dt1;
```

CENGİZ YILMAZ							
	DERSAD	NOTID	SINAV1	SINAV2	SINAV3	PROJE	ORTALAMA
▶	VERİ YÖNETİMİ	1	55	55			
*							

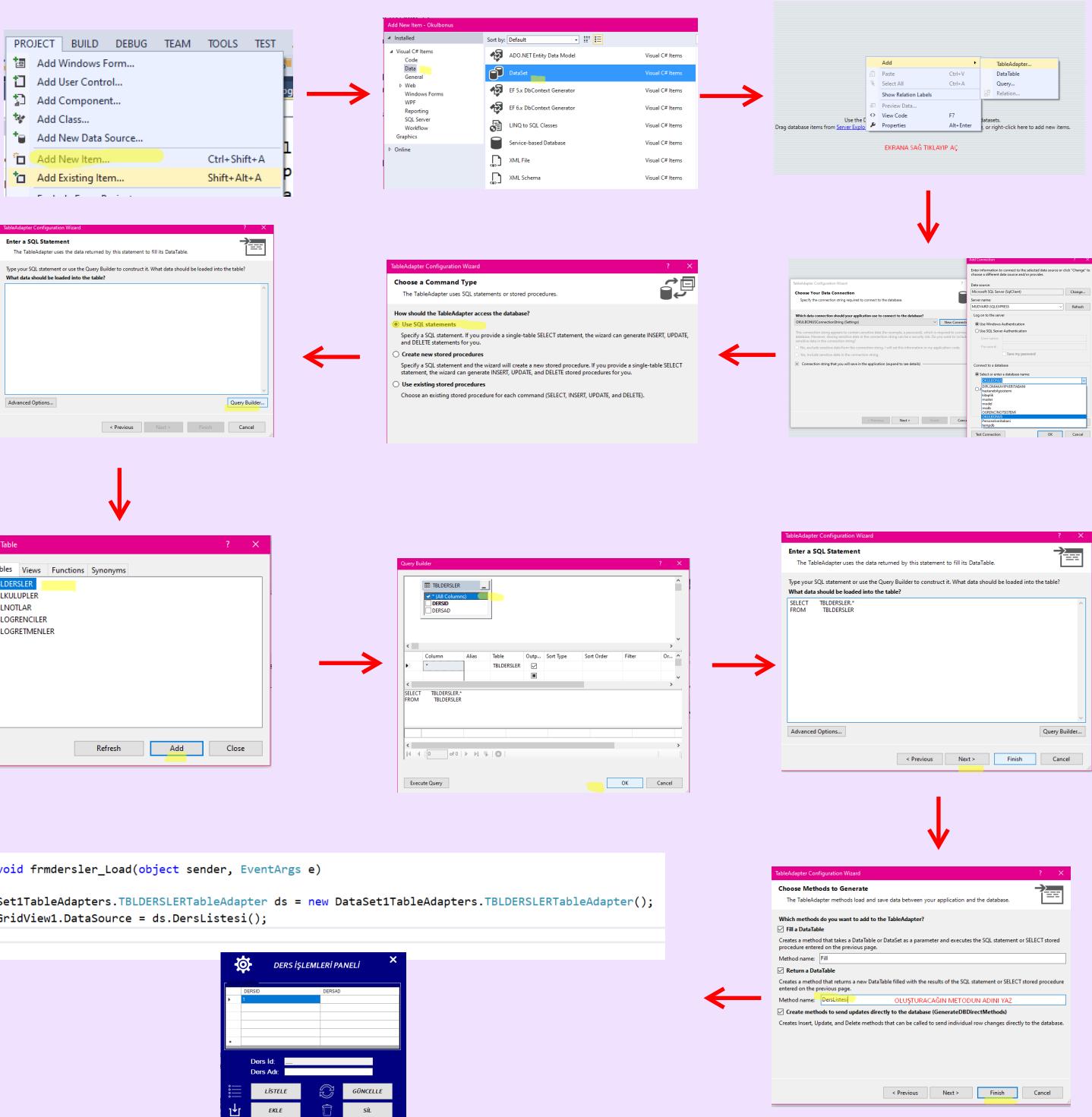
  

NOTID	DERSID	OGRID	SINAV1	SINAV2	SINAV3	PROJE	ORTALAMA	DURUM
1	4	1	55	55	NULL	NULL	NULL	NULL
2	5	2	55	89	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

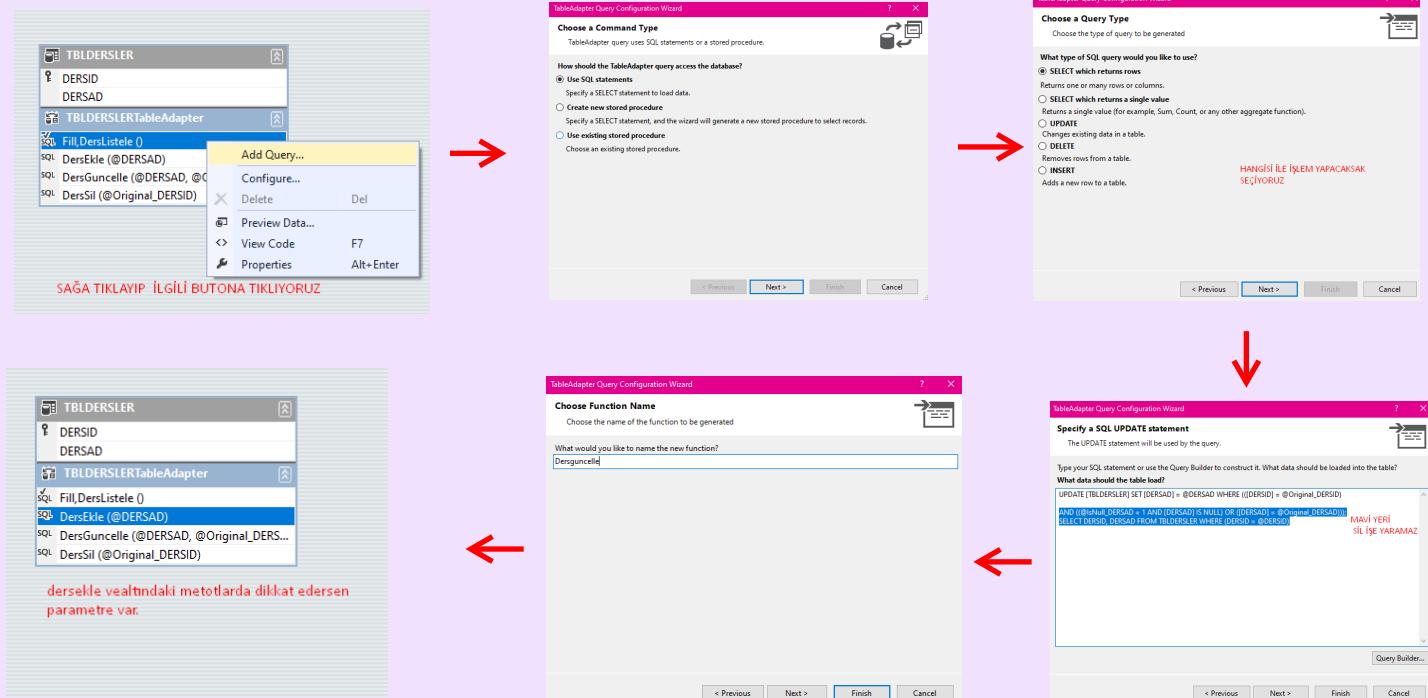
# DATASET KULLANIMI

## DATASET İLE LİSTELEME YAPMA

Dataset kullanılarak hazır metod oluşturulur. Aşağıda select metodу oluşturulup hazır metod kullanılmıştır.



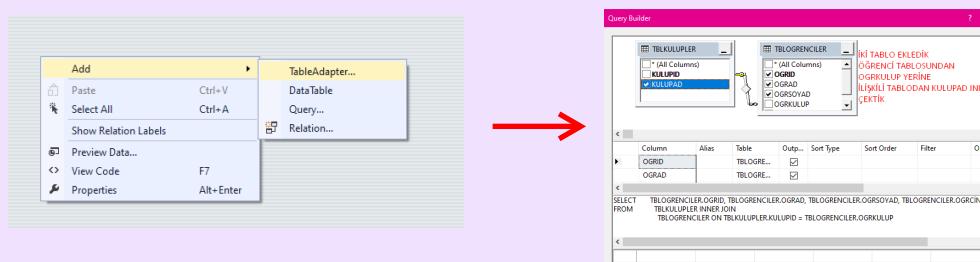
## DATASET İLE INSERT, UPDATE VE DELETE SORGULARI OLUŞTURMA



```
DataSet1TableAdapters.TBLDERSLERTableAdapter ds = new DataSet1TableAdapters.TBLDERSLERTableAdapter();
```

```
private void btnselect_Click(object sender, EventArgs e)
{
    dataGridView2.DataSource = ds.DersListele(); //LİSTELEME
}
private void btninsert_Click(object sender, EventArgs e)
{
    ds.DersEkle(txtdersad.Text); //parametreye bilgiyi nereden alacağını yazıyoruz.
    dataGridView2.DataSource = ds.DersListele();
}
private void btnupdate_Click(object sender, EventArgs e)
{
    ds.DersGuncelle(txtdersad.Text, byte.Parse(mskid.Text)); // iki tane parametre var. ilki
    string, ikincisi byte türünde. mskid byte değeri olduğu için
    //byte çeviriyoruz.
```

## DATASET İLE DİYAGRAMI KULLANARAK İLİŞKİ İKİ TABLOYU KULLANARAK VERİ GÖRÜNTÜLEME

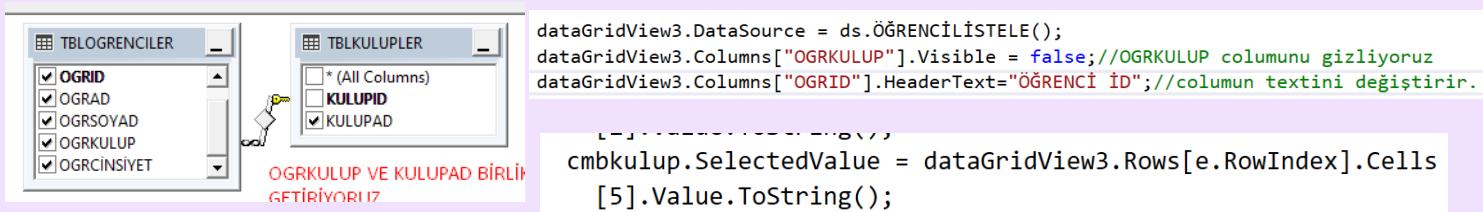


## DISPLAYMEMBER VALUEMEMBER KULLANIMI

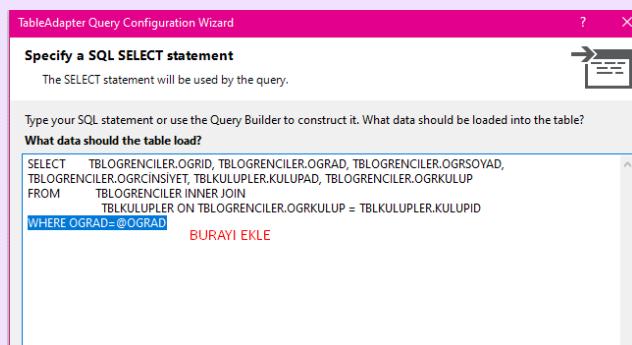
```
dataGridView3.DataSource = ds.ÖĞRENCİLİSTELE();
bg1.Open();
SqlCommand komut = new SqlCommand("select * from TBKULUPLER", bg1);
SqlDataAdapter da = new SqlDataAdapter(komut);
DataTable dt = new DataTable();
da.Fill(dt);

cmbkulup.DisplayMember = "KULUPAD"; //comboboxta görünen öğe adı
cmbkulup.ValueMember = "KULUPID"; // comboboxta arkaplanda dönen değer tablodaki hangi sütündan
cmbkulup.DataSource = dt;
bg1.Close();
```

## DATAGRİDE TIKLANDIĞINDA COMBOXIN SELECTEDVALUE DEĞERİNİ İLİŞKİLİ TABLODAN GETİRME



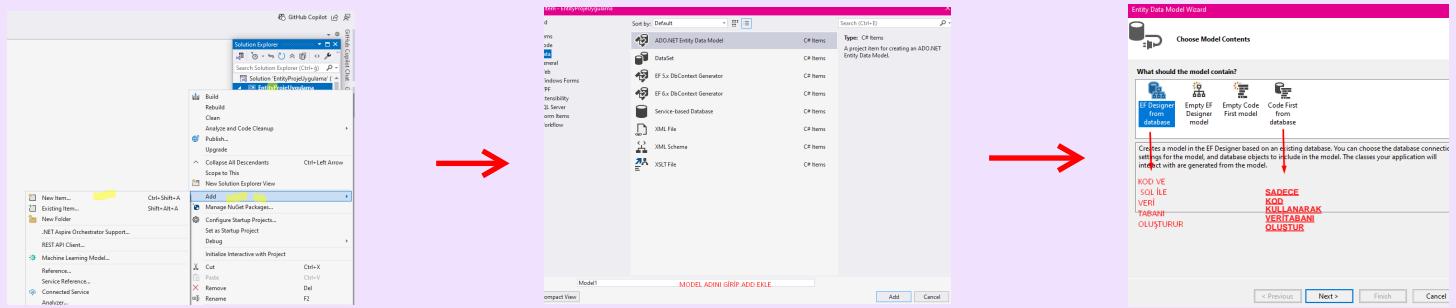
## DATAGRİDDE ÖĞRENCİ ARA-ARA BUTONU İLE



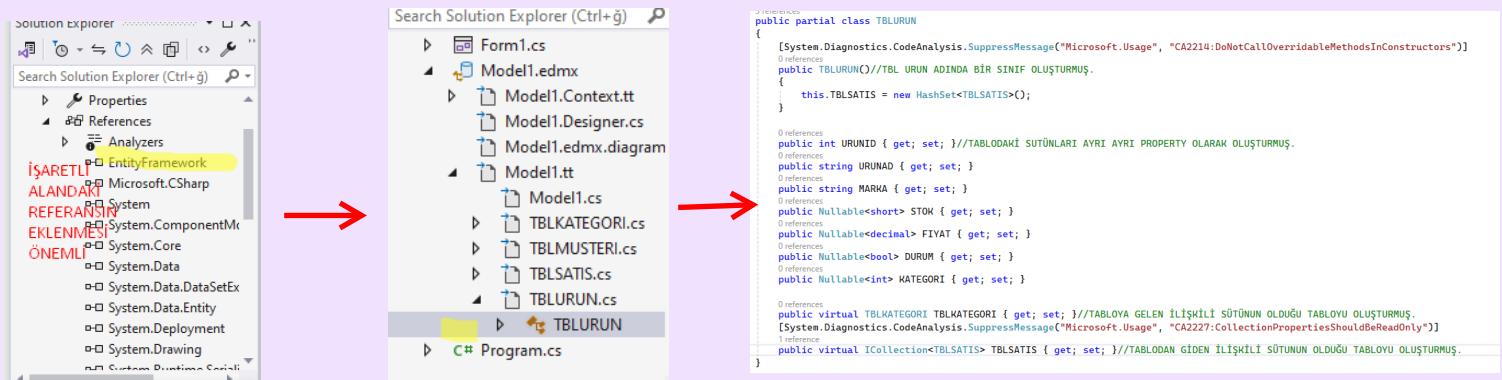
```
private void btnogrenciara_Click(object sender, EventArgs e)
{
    dataGridView3.DataSource=ds.OgrenciGet(txtara.Text);
}
```

# ENTITY FRAMEWORK

## ENTITY MODELİNİN OLUŞTURULMASI



## ENTITY MODELİNİ TANIYALIM :-)



## ENTITY TOLİST METODU(LİSTELEME)

```
private void button1_Click(object sender, EventArgs e)
{
    var kategoriler = db.TBLKATEGORI.ToList(); // kategoriler adında bir
    // değişken oluşturup bu değişkeni db nesnesi aracılığıyla kategoriler
    // tablosuna bağlayıp tolist metoduyla listeledik.
    dataGridView1.DataSource = kategoriler; //datagridin kaynağını
    // kategoriler değişkenine atadık
}
```

## ENTITY ADD METODU(KAYDETME)

```
private void btnkaydet_Click(object sender, EventArgs e)
{
    TBLKATEGORI t = new TBLKATEGORI(); //tblkategori
    // tablosundan bir nesne türetiyoruz.
    t.AD = textBox2.Text; //nesne aracılığıyla kategorideki
    // sütuna ulaştık ve sutünün nereden bilgi alacağını atadık
    db.TBLKATEGORI.Add(t); //tabloya eklemeye çalıştık(t)
    db.SaveChanges(); //yaptığımız değişikliği save ettik.
    //MessageBox.Show("Eklendi");
}
```

## ENTITY REMOVE METODU(SİLME)

```

private void button3_Click(object sender, EventArgs e)
{
    int x = Convert.ToInt16(textBox1.Text); // x
    // değişkenini textbox1 e atıyoruz.
    var ktgrd = db.TBLKATEGORI.Find(x); // ilgili
    tablodan x değişkenine ait bir sütun seçici
    değişken oluşturuyoruz.
    db.TBLKATEGORI.Remove(ktgrd); // ktgrd sutünunu
    remove ediyoruz.
    db.SaveChanges();
    MessageBox.Show(x + " numaralı kategori silindi.");
}

```

## ENTITY GÜNCELLEME

```

private void button4_Click(object sender, EventArgs e)
{
    int a = Convert.ToInt32(textBox1.Text);
    var ktgrgn = db.TBLKATEGORI.Find(a);
    ktgrgn.AD = textBox2.Text; // ktgrgn row una ait AD sütununu atıyoruz.
    db.SaveChanges();
    MessageBox.Show(a + " numaralı kategori güncellendi.");
}

```

## ENTITY VERİLER COMBOXA GETİRME VEYA DATAGRİDDE SÜTUN LİSTELEMЕ

### SARTSIZ GETİRME

```

var kategoriler = (from x in db.TBLKATEGORİLER TABLOSUN
                    // DAN x adında bir değişken oluşturduk. from
                    // döngü gibi sürekli değişken oluşturmak
                    // işlevinde
                    select new
                    {
                        x.ID, // x'i ID ye ata
                        x.AD // x'i AD a ata
                    })
                    .ToList(); // listeleyin
CMBKATEGORİ.DisplayMember = "AD";
CMBKATEGORİ.ValueMember = "ID";
CMBKATEGORİ.DataSource = kategoriler;

```

### SARTLI GETİRME

```

int a = Convert.ToInt16(cmbbrans.SelectedValue);
var bransi = (from x in db.Tbl_doktorlar where x.Doktorbrans == a
              select new
              {
                  adsoyad = x.Doktorad + " " + x.Doktorsoyad // ad soyad değişkeni
                  // oluşturup bunu şekildeki gibi atıyoruz.
              }).ToList();
cmbdoktor.DisplayMember = "adsoyad"; // görünen yüzeye yukarıdaki değişkenin adını
// yazıyoruz.
cmbdoktor.ValueMember = "Doktorid";
cmbdoktor.DataSource = bransi;

```

```

dataGridView1.DataSource = (from x in db.TBLURUN
                           select new
                           {
                               x.URUNID,
                               x.URUNAD,
                               x.MARKA,
                               x.FİYAT,
                               x.DURUM,
                               x.TBLKATEGORİ.AD // tabloda x aracılığıyla başka
                               tablodaki sütunu da gösterebiliriz.
                           })
                           .ToList();

```

```

int a = Convert.ToInt16(cmbbrans.SelectedValue);
dataGridView2.DataSource = (from x in db.Tbl_doktorlar
                           where x.Doktorbrans == a
                           select new
                           {
                               x.Doktorid,
                               x.Doktorad,
                               x.Doktorsoyad,
                               x.Doktortc
                           })
                           .ToList();

```

## LINQ METOTLAR

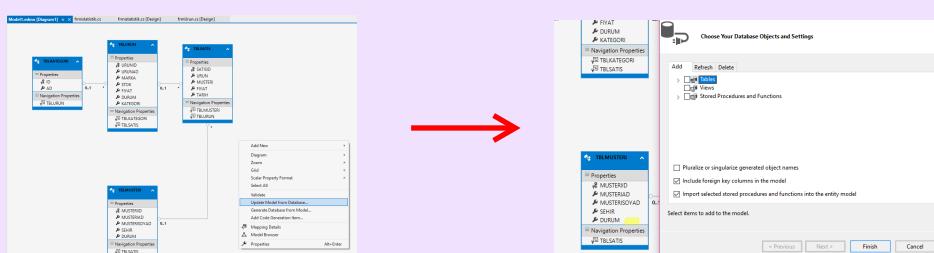
```
label2.Text = db.TBLKATEGORI.Count().ToString(); //TBLKATEGORİLER İÇİNDEKİ BÜTÜN  
    DATALARIN SAYISINI VERİR  
label3.Text = db.TBLURUN.Count().ToString();  
label5.Text = db.TBLMUSTERI.Count(x => x.DURUM == true).ToString(); //x adında bir  
    değişken oluşturup bu değişkeni tablodaki durumu true olanlara atayıp sayısını verir.  
label7.Text = db.TBLMUSTERI.Count(x => x.DURUM == false).ToString();  
label13.Text = db.TBLURUN.Sum(x => x.STOK).ToString(); //TBLURUN tablosundan x adında  
    bir değişken oluşturup bu değişkeni tablodaki STOK sütununa atayıp toplamını verir.  
label21.Text = db.TBLSATIS.Sum(x => x.FIYAT)+" TL";  
label11.Text = (from x in db.TBLURUN orderby x.FIYAT descending select  
    x.URUNAD).FirstOrDefault();  
label9.Text = (from x in db.TBLURUN orderby x.FIYAT ascending select  
    x.URUNAD).FirstOrDefault();  
// TBLURUN tablosundan x adında bir değişken oluştur, x i fiyat sutünuna bağlayarak  
    sırala, x i urunadına ata ve ascending(a dan z ye) sıralayıp seç.  
label15.Text = db.TBLURUN.Count(x => x.KATEGORI == 1).ToString();  
label23.Text = (from x in db.TBLMUSTERI select x.SEHIR).Distinct().Count().ToString();  
//TBLMUSTERİ tablosundan x adında bir değişken oluştur, x i şehire bağla ve seç,distinct  
    (tekrarsız) şekilde countu al.  
label19.Text = db.MARKAGETIR().FirstOrDefault();
```

### SUM(COUNT METODU)

```
private void frmistatistik_Load(object sender, EventArgs e)  
{  
    label2.Text = db.TBLKATEGORI.Count().ToString(); //TBLKATEGORİLER İÇİNDEKİ BÜTÜN  
        DATALARIN SAYISINI VERİR  
    label3.Text = db.TBLURUN.Count().ToString();  
}
```



### UPDATE FROM MODEL DATABASE(tablodaki değişikliği model1 entity de güncellenme)



### SUM VE ŞARTLI SORGU OLUŞTURMA

```
label5.Text = db.TBLMUSTERI.Count(x => x.DURUM == true).ToString(); //x adında bir  
    değişken oluşturup bu değişkeni tablodaki durumu true olanlara atayıp sayısını verir.  
label7.Text = db.TBLMUSTERI.Count(x => x.DURUM == false).ToString();  
label13.Text = db.TBLURUN.Sum(x => x.STOK).ToString(); //TBLURUN tablosundan x adında  
    bir değişken oluşturup bu değişkeni tablodaki STOK sütununa atayıp toplamını verir.
```



### EN YÜKSEK VE EN DÜŞÜK FİYATLI ÜRÜN SORGULAMA

```
label11.Text = (from x in db.TBLURUN orderby x.FIYAT descending select x.URUNAD).FirstOrDefault();  
    TBL.URUN TABLOSUNDAN  
    X ADINDA BİR DEĞİŞKEN  
    OLÜSTUR  
X i fiyatata ve fiyat  
    ları sırala  
    x'i urunadı sütünuna bağla ve  
    descending(z den a ya) sırala ve  
    seç  
    sıranın en üstünü  
    al.
```

```
label9.Text = (from x in db.TBLURUN orderby x.FIYAT ascending select x.URUNAD).FirstOrDefault();  
    // TBLURUN tablosundan x adında bir değişken oluştur, x i fiyat sutünuna  
    bağlayarak sırala, x i urunadına ata ve ascending(a dan z ye) sıralayıp seç.
```

## DİSTINCT COUNT SORGULAMA

```
label23.Text = (from x in db.TBLMUSTERI select x.SEHIR).Distinct().Count().ToString();
//TBLMUSTERI tablosundan x adında bir değişken oluştur, x i şehire bağlı ve seç,distinct(tekarsız) şekilde countu al.
```

## SQL DE PROCEDURE OLUŞTURMA

`CREATE PROCEDURE MARKAGETIR` markagetir adında bir procedure oluştur.

AS

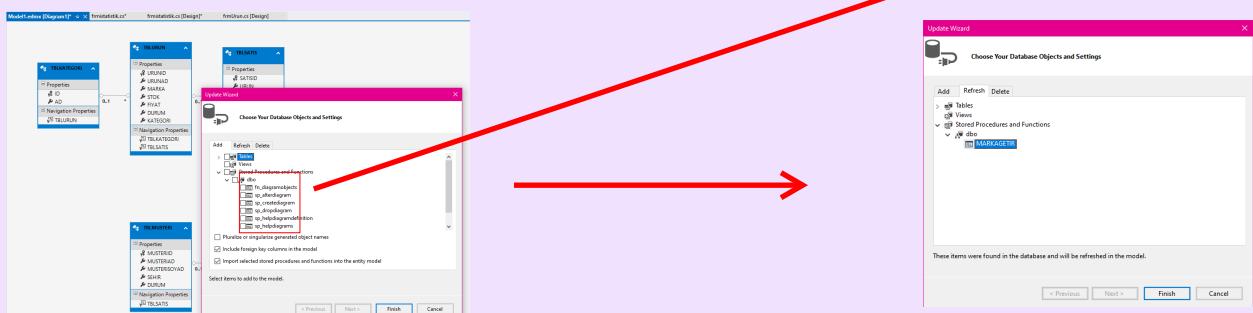
`SELECT TOP 1 MARKA, COUNT(*) FROM TBLURUN GROUP BY MARKA` not:daha sonra count silinir sadece marka adı kalır.

en üst teki 1 TBLURUN tablosundaki markaların sayılarını göster ORDER BY COUNT(\*) DESC sayıları z den a ya sırala

MARKA	(No column name)
ARÇELİK	2

## PROCEDURE Ü PROJELYE DAHİL ETME

model1.edmx üzerinde sağ tıklayıp update model from database denir daha sonra bu alandan procedure seçiliğin eklenir. Daha sonra refresh yapılır.



```
label19.Text = db.MARKAGETIR().FirstOrDefault();
```

## SARTLI KULLANICI ADMIN İN GİRİŞİ OLUŞTURMA

```
DBEntityUrunkategoriEntities1 db = new DBEntityUrunkategoriEntities1();
var soru = from x in db.TBLADMINGIRIS where
x.KULLANICIADI == textBox1.Text && x.KULLANICISIFRE ==
maskedTextBox1.Text select x;
//TBLADMINGIRIS tablosundan x değişkeni oluştur, x
seçildiğinde
if (soru.Any())//sorgu bir sonuca ulaşırsa
{
    frmAnaform fr = new frmAnaform();
    fr.Show();
}
else
{
    MessageBox.Show("Hatalı giriş");
}
```

## SARTLI BİLGİ GETİRME

```
lbltc.Text = tc.ToString();

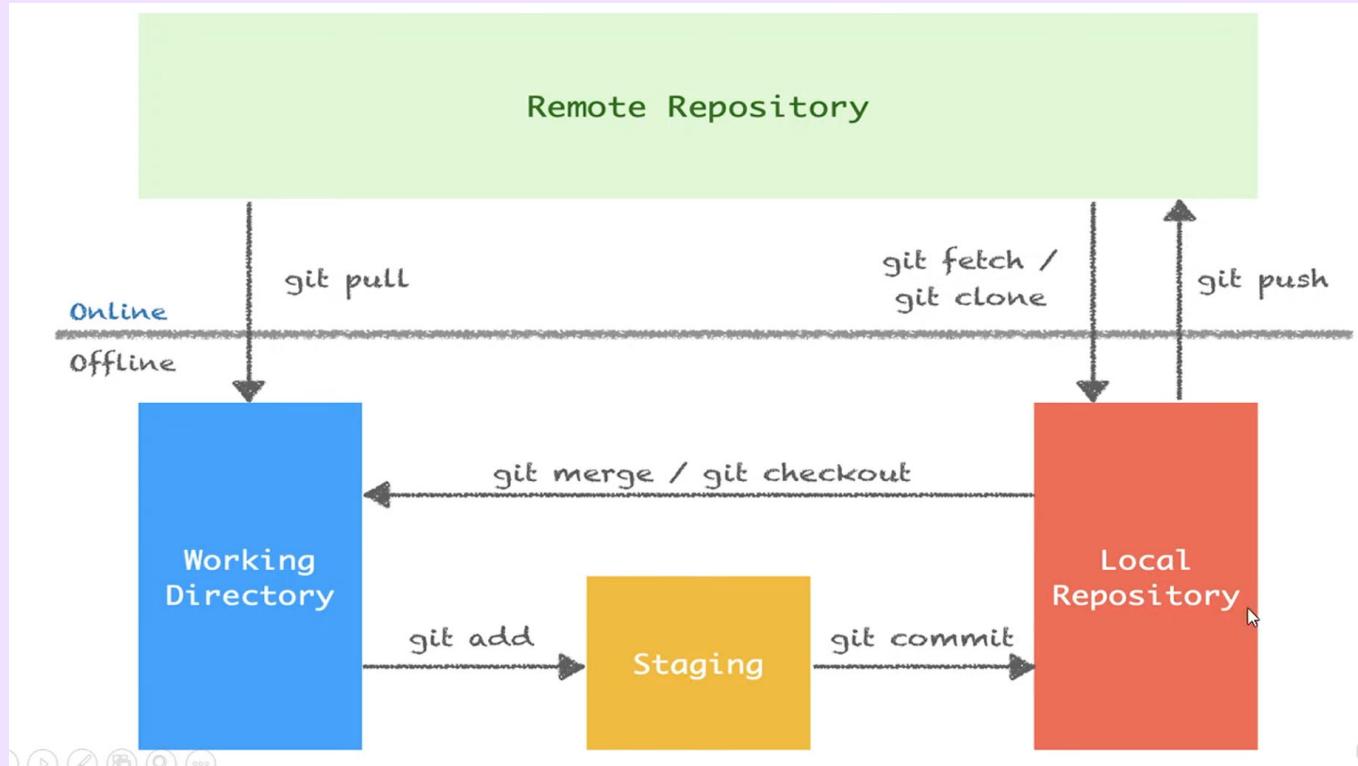
Tbl_sekreter T = new Tbl_sekreter();
var id = (from x in db.Tbl_sekreter where x.SekreterTC==tc select
x.Sekreterid).FirstOrDefault();

var ktgr = db.Tbl_sekreter.Find(T.Sekreterid=id);
lbladsoyadsekreter.Text = (ktgr.SekreterAd + " " + ktgr.SekreterSoyad).ToString();
```

# GİT && GİTHUB

## GİT KULLANIMI

### GİT ŞEMASI



[cd dosya yolundaki klasörleri yazarak ulaşırız.](#)

[mkdir klasör ismini yazıyoruz](#)

[touch dosya ismini uzantısıyla birlikte yazıyoruz.](#)

[git config --global user.name "kullanıcıadiyazılır".](#)  
[git config list](#)

```

PC@MudYard MINGW64 /c
$ pwd dosya yolunu verir
/c

PC@MudYard MINGW64 /c
$ cd users c nin içinde users klasörüne girmeni sağlar

PC@MudYard MINGW64 /c/users
$ c PC
bash: c: command not found

PC@MudYard MINGW64 /c/users
$ cd pc
PC@MudYard MINGW64 /c/users/pc
$ cd desktop
● desktop klasörüne bulana kadar dosya yolunu takip ederek
indik.

PC@MudYard MINGW64 /c/users/pc/desktop
$ pwd
/c/users/pc/desktop

PC@MudYard MINGW64 /c/users/pc/desktop
$ mkdir app yukarıdaki dosya yolunda klasör oluşturmamızı sağlar

PC@MudYard MINGW64 /c/users/pc/desktop
$ cd app

PC@MudYard MINGW64 /c/users/pc/desktop/app
$ touch text.txt yukarıdaki dosya yolunda dosya oluşturmamızı sağlar

PC@MudYard MINGW64 /c/users/pc/desktop/app
$ 
```

```

PC@MudYard MINGW64 /c/Users
$ git config --global user.name "cngzylmz89" git programında config(düzenle) global alanda
kullanıcı adını "" yap

PC@MudYard MINGW64 /c/Users
$ git config list git düzenele liste
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=schannel
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=cngzylmz89
user.email=muallimiturki@gmail.com
pull.rebase=false

PC@MudYard MINGW64 /c/Users
$ | 
```

[pwd , ls, git init, ls -a](#)

```
PC@MudYard MINGW64 /c/users/pc/desktop/app
$ pwd
/c/users/pc/desktop/app

PC@MudYard MINGW64 /c/users/pc/desktop/app
$ ls yukardaki dosya yolundaki klasörde hangi dosyaların olduğunu verir
text.txt

PC@MudYard MINGW64 /c/users/pc/desktop/app
$ git init yukardaki klasörde git ile ilgili işlemleri saklayacağımız gizli bir klasör oluşturur
Initialized empty Git repository in C:/Users/PC/Desktop/app/.git/

PC@MudYard MINGW64 /c/users/pc/desktop/app (master)
$ ls -a gizli veya açık bütün dosyaları gösterir
./ ../ .git/ text.txt
```

[git status](#)

```
PC@MudYard MINGW64 /c/users/pc/desktop/app (master)
$ git status git te ne durumda olduğumuzun özétini verir
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    text.txt

nothing added to commit but untracked files present (use "git add" to track)
```

[git add text.txt](#)

[git add .](#)

```
PC@MudYard MINGW64 /c/users/pc/desktop/app/Hastaneprojesikopya (master)
$ git add .

PC@MudYard MINGW64 /c/users/pc/desktop/app/Hastaneprojesikopya (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .vs/hastaneprojesi/FileContentIndex/3e6c2b85-cc5f-4094-9629-9f75e1f5d485.vsidx
    new file:   .vs/hastaneprojesi/FileContentIndex/455b6c32-fb03-498e-bc3c-68655d8067a1.vsidx
    new file:   .vs/hastaneprojesi/FileContentIndex/ab32eea5-5d75-4ee4-ba5c-b0411c23836c.vsidx
    new file:   .vs/hastaneprojesi/FileContentIndex/dcac7dca-a3be-493a-8b79-71adc3706efd.vsidx
    new file:   .vs/hastaneprojesi/FileContentIndex/e460dabf-0152-4ba1-8e3a-01bd0b2bbdb3.vsidx
    new file:   .vs/hastaneprojesi/v17/.suo
```

[git commit -m "commit1\(komutdışı satır:commitin ismini veriyoruz.\)"](#)

```
PC@MudYard MINGW64 /c/users/pc/desktop/app/Hastaneprojesikopya (master)
$ git commit -m "commit1"
[master (root-commit) 0358cbe] commit1
148 files changed, 8815 insertions(+)
create mode 100644 Hastaneprojesikopya/.vs/hastaneprojesi/FileContentIndex/3e6c2b85-cc5f-4094-9629-9f75e1f5d485.vsidx
create mode 100644 Hastaneprojesikopya/.vs/hastaneprojesi/FileContentIndex/455b6c32-fb03-498e-bc3c-68655d8067a1.vsidx
create mode 100644 Hastaneprojesikopya/.vs/hastaneprojesi/FileContentIndex/ab32eea5-5d75-4ee4-ba5c-b0411c23836c.vsidx
create mode 100644 Hastaneprojesikopya/.vs/hastaneprojesi/FileContentIndex/dcac7dca-a3be-493a-8b79-71adc3706efd.vsidx
create mode 100644 Hastaneprojesikopya/.vs/hastaneprojesi/FileContentIndex/e460dabf-0152-4ba1-8e3a-01bd0b2bbdb3.vsidx
create mode 100644 Hastaneprojesikopya/.vs/hastaneprojesi/v17/.suo
create mode 100644 Hastaneprojesikopya/.vs/hastaneprojesi/v17/DocumentLayout.backup.json
create mode 100644 Hastaneprojesikopya/.vs/hastaneprojesi/v17/DocumentLayout.json
create mode 100644 Hastaneprojesikopya/Hastaneprojesikrokisi.pdf
create mode 100644 Hastaneprojesikopya/hastaneprojesi.sln
create mode 100644 Hastaneprojesikopya/hastaneprojesi.v12.suo
```

## git log

```
PC@MudYard MINGW64 /C/users/PC/desktop/App (master)
$ git log yapılan commitleri gösterir
commit 935a79bd88e6f757fd029dd1815d93e17147fd05 (HEAD -> master)
Author: cngzylmz89 <muallimiturki@gmail.com>
Date: Mon May 12 12:27:42 2025 +0300

deneme4 silindi

commit 99dcb819efdc5e63bc2a816572e268764d454105
Author: cngzylmz89 <muallimiturki@gmail.com>
Date: Mon May 12 12:23:32 2025 +0300

deneme4 eklendi

commit 0358cbe6dfd2b30c5149103f535dbc6a073acb64
Author: cngzylmz89 <muallimiturki@gmail.com>
Date: Fri May 9 13:50:54 2025 +0300

commit1
```

**git checkout commitin idisi yazılır -- .**

```
PC@MudYard MINGW64 /C/users/PC/desktop/App (master)
$ git checkout 99dcb819efdc5e63bc2a816572e268764d454105 -- .
yapılan değişikliğin id numarası yazarak geri getirebiliriz.
```

## git diff

```
$ git diff dosya içinde yapılan değişikliği gösterir
diff --git a/deneme3.txt b/deneme3.txt
index e69de29..fb2917f 100644
--- a/deneme3.txt
+++ b/deneme3.txt
@@ -0,0 +1 @@
+cengiz yılmaz dosyanın içine bu yazılmış
\ No newline at end of file
```

**git diff dosyanın ismi uzantısıyla yazılır**

```
PC@MudYard MINGW64 /C/users/PC/desktop/App (master)
$ git diff deneme3.txt dosya adını vererek de değişikliği görebiliriz.
diff --git a/deneme3.txt b/deneme3.txt
index e69de29..fb2917f 100644
--- a/deneme3.txt
+++ b/deneme3.txt
@@ -0,0 +1 @@
+cengiz yılmaz
\ No newline at end of file
```

**git checkout -- dosya ismi uzantısıyla birlikte yazılır**

```
PC@MudYard MINGW64 ~/desktop/app (master) klasördeki kaydedilmiş değişikliği getirir
$ git checkout -- deneme3.txt
```

**git rm dosya ismini uzantısıyla yaz**

```
PC@MudYard MINGW64 ~/desktop/app/test (master)
$ git rm test12.txt dosyayı siler
rm 'test/test12.txt'
```

**git rm -r klasörismi/**

```
PC@MudYard MINGW64 ~/desktop/app (master)
$ git rm -r test/ test adlı klasörü içindekiyle birlikte siler
rm 'test/test13.txt'
```

[git mv dosyaadı dosyaadı](#)

```
PC@MudYard MINGW64 /c/users/pc/desktop/app (master)
$ git mv deneme4.txt deneme5.txt deneme4.txt iyi deneme5.txt olarak değiştir.
fatal: Unable to create 'C:/Users/PC/Desktop/app/.git/index.lock': File exists.
```

[git mv dosyaadı klasöradi/](#)

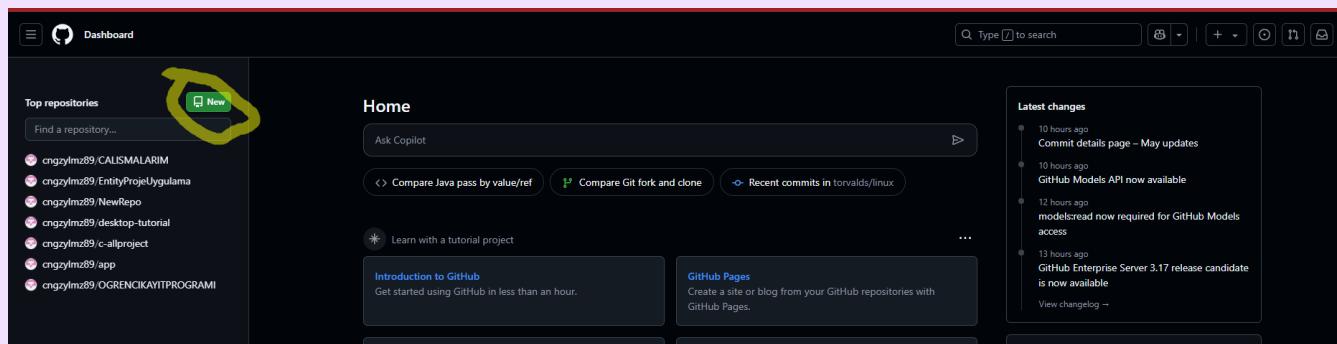
```
PC@MudYard MINGW64 /c/users/pc/desktop/app (master)
$ git mv deneme3.txt dosya/ deneme3.txt iyi dosya klasörüne atar
fatal: unable to move 'C:/Users/PC/Desktop/app/deneme3.txt' to 'C:/Users/PC/Desktop/app/dosya/deneme3.txt': File exists.
```

[git mv dosyaadı klasöradi/dosyaadı](#)

```
PC@MudYard MINGW64 /c/users/pc/desktop/app (master)
$ git mv deneme3.txt dosya/deneme5.txt| deneme3 ün adını değiştirir dosya klasörüne
```

# GİTHUB

## GİTHUB REPOSITORY OLUŞTURMA



[git remote add origin bağlantı adresini yaz](#)

-----> uzak bağlantı komutu

[git push -u origin master](#)

[git pull](#)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (\*).

Owner \* [cngzylmz89](#) / Repository name \* [repository iye isim ver](#)

Great repository names are short and memorable. Need inspiration? How about [reimagined-octo-guacamole](#)?

Description (optional) [apklärma yapabilirsin](#)

Public Anyone on the internet can see this repository. You choose who can commit.

Private You choose who can see and commit to this repository.

Initialize this repository with:

Add a README file This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore [.gitignore template: None](#)

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license [License: None](#)

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

You are creating a public repository in your personal account.

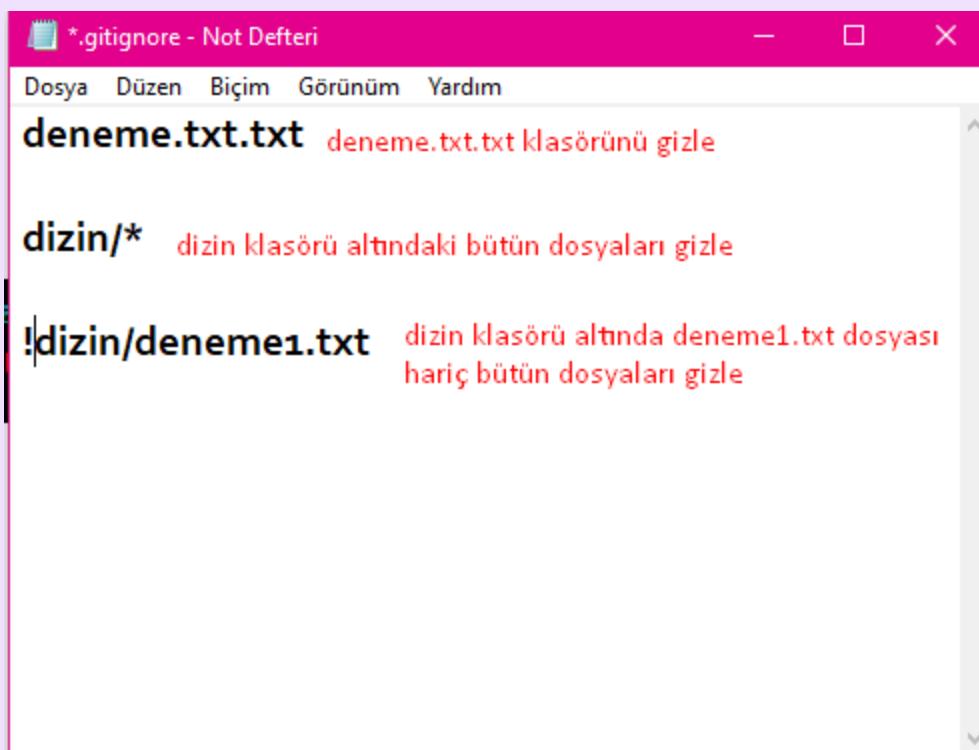
[Create repository](#) [oluştur](#)

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ git remote add origin https://github.com/cngzylmz89/c#allproject.git
uzak bağlantı oluştur
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ git remote
origin uzak bağlantının adını teyit et
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ git status
On branch master bekleyen commit var mı bak
nothing to commit, working tree clean
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ git push -u origin master dosyaları push et
Enumerating objects: 196, done.
Counting objects: 100% (196/196), done.
Delta compression using up to 4 threads
Compressing objects: 100% (184/184), done.
Writing objects: 100% (196/196), 11.10 MiB | 1.90 MiB/s, done.
```

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ git pull yapılan değişikliği local e pull eder
Already up-to-date
```

touch .gitignore

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ touch .gitignore .gitignore adında bir txt dosyası oluşturur
```



## BRANCHLER

git branch

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (dev)
$ git branch
* dev    oluşturulmuş branchleri gösterir
  master
```

git branch --all

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (dev)
$ git branch --all  bütün brancleri gösterir
  all
* dev
  master
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
```

git fetch

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (dev)
$ git fetch  oluşturulan branchleri locale çeker
From https://github.com/cngzylmz89/c-allproject
 * [new branch]      dev        -> origin/dev
```

git checkout branchadı

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ git checkout dev  master branchten dev branch e geçer
Switched to branch 'dev'
```

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (dev)
$ git push -u origin dev sondaki branch adına push yapılır
```

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (dev)
$ git pull origin dev sondaki branch adına pull yapılır
```

### git merge branchadı

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ git merge dev dev branch indeki farklılıklar mastera aktarılır
```

### git fetch -p

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ git fetch -p github ta silinen branchleri locale taşıriz.
From https://github.com/cngzylmz89/c-allproject
 * [deleted]          (none)    -> origin/dev
 * [deleted]          (none)    -> origin/test
```

### git branch -D branchadı

```
PC@MudYard MINGW64 /c/users/pc/desktop/c#allproject (master)
$ git branch -D test githubta silinen ve locale taşıenan branch localde de silinir
Deleted branch test (was 66e3046).
```

# OBJECT ORIENTED PROGRAMMING (OOP) NESNE YÖNELİMLİ PROGRAMLAMA

## OOP MANTIĞI



### OOP Avantajları

- Proje yönetimini kolaylaştırır, zamandan tasarruf etmemizi sağlar.
- Esnek bir yapı sunar. Class'lar genişletilebilir, yeni özellikler eklenebilir.
- Her class bağımsız bir yapıya sahiptir. Oluşturulan class'lar farklı projelere entegre edilebilir.

## OOP Avantajları

- Miras alma işlemini destekler. Herhangi bir class, başka bir class miras alınarak geliştirilebilir.
- Örneğin; **araç** class'ı oluşturulup, bu class miras alınarak **otomobil**, **otobüs**, **kamyon** gibi class'lar yazılabilir.

## OOP Avantajları

- Bir nesneye ait bilgiler gizlenebilir ve hangi durumlarda bu bilgilere erişim ve düzenleme olanağı sağlanacağı belirlenir. (Bilgi gizliliği)
- Hata ayıklama işlemini kolaylaştırır.
- Ekip çalışmasına olanak sağlar. Geliştiriciler birbirinden bağımsız olarak class geliştirebilir.

## KATMANLAR ----->

-----ENTITY LAYER-----

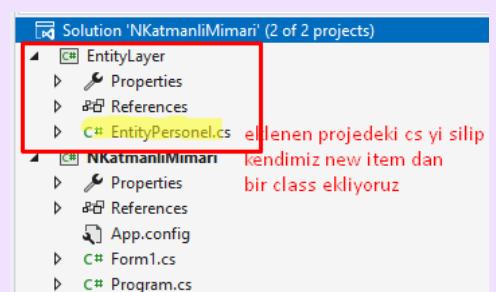
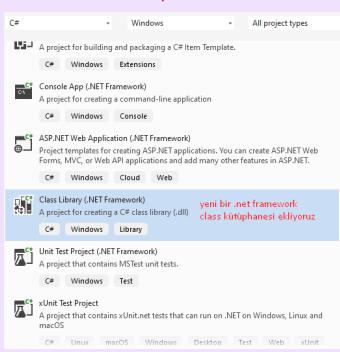
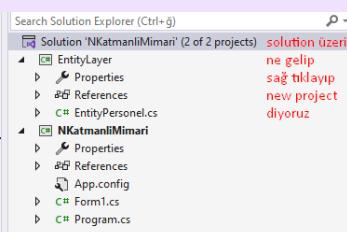
-----DATA ACCESS LAYER-----

-----LOGIC LAYER-----

-----PRESENTATION LAYER-----

## ENTITY LAYER

Birinci aşama entity katmanının oluşturulmasıdır.



## DATA ACCESS LAYER

```
using EntityLayer; //EntityLayer sınıfını kutuphane olarak ekliyoruz
using System.Data; //18. satırda geçen ConnectionState sınıfını görebilmek için data kütüphanesini ekliyoruz

namespace DataAccessLayer
{
    public class DALPersonel
    {
        public static List<EntityPersonel> Personellistesi ()//EntityPersonel sınıfı kullanmak üzere listele metodu oluşturduk ve ismini Personellistesi koyduk
        {
            List<EntityPersonel> degerler = new List<EntityPersonel>(); //değer döndüren bir metod olduğu için oluşturduğumuz metod sınıfı ve sınıfın bir DEGERLER adında bir nesne türetiyoruz
            SqlCommand komut = new SqlCommand("select * from TBLBILGI", Baglanti.bgl); //sqldeki tabloyla bağlantıyı kuruyoruz ve komutumuzu oluşturuyoruz.
            if (komut.Connection.State != ConnectionState.Open) //komut bağlantı durumu açık değilse
            {
                komut.Connection.Open(); //komutun bağlantısını aç
            }

            SqlDataReader dr = komut.ExecuteReader(); //tablodaki değerleri okuyacağız

            while (dr.Read())
            {
                EntityPersonel ent = new EntityPersonel(); //EntityPersonel sınıfından nesne türetip bu nesne aracılığıyla aynı sınıfta oluşturduğumuz değişkenleri sqldeki tablonun sütunlarıyla ilişkilendiriyoruz.
                ent.ID1 = int.Parse(dr["ID"].ToString());
                ent.Ad1 = dr["AD"].ToString();
                ent.Soyad1 = dr["SOYAD"].ToString();
                ent.Sehir1 = dr["SEHIR"].ToString();
                ent.Gorev = dr["GOREV"].ToString();
                ent.Maas1 = short.Parse(dr["MAAS"].ToString());
                degerler.Add(ent); //metottan oluşturduğumuz nesneye ent den gelen değerleri aktarıyoruz.
            }
            dr.Close(); //bağlılığı kapatıyoruz
            return degerler; //değerler den gelenleri döndürüyoruz.
        }
    }
}
```

### 1- METOT VEYA PROCEDURE OLUŞTURDUK

2-METOTTAN NESNE TÜRETTİK

3-SQL KOMUTUMUZU OLUSTURDUK

4-SQL DEN GELEN BİLGİLERİ OLUŞTURDUĞUMUZ DEĞİŞKENLERE SINIF

NESNE ARACILIĞIYLA AKTARDIK.

### 5-METODUMUZU BİTİRDİK.

## LOGIC LAYER

```
using EntityLayer; //reference kütüphanesini ekliyoruz
using DataAccessLayer; //reference kütüphanesini ekliyoruz

namespace LogicLayer
{
    public class LogicPersonel
    {
        public static List<EntityPersonel> LLpersonellistesi() //Entitypersonel sınıfından list metodу oluşturuyoruz
        {
            return DALPersonel.Personellistesi(); //metotta DALPERSONEL sınıfında oluşturduğumuz Personellistesi metodunu çağırıyoruz.
        }
    }
}
```

# PRENSENTATION LAYER

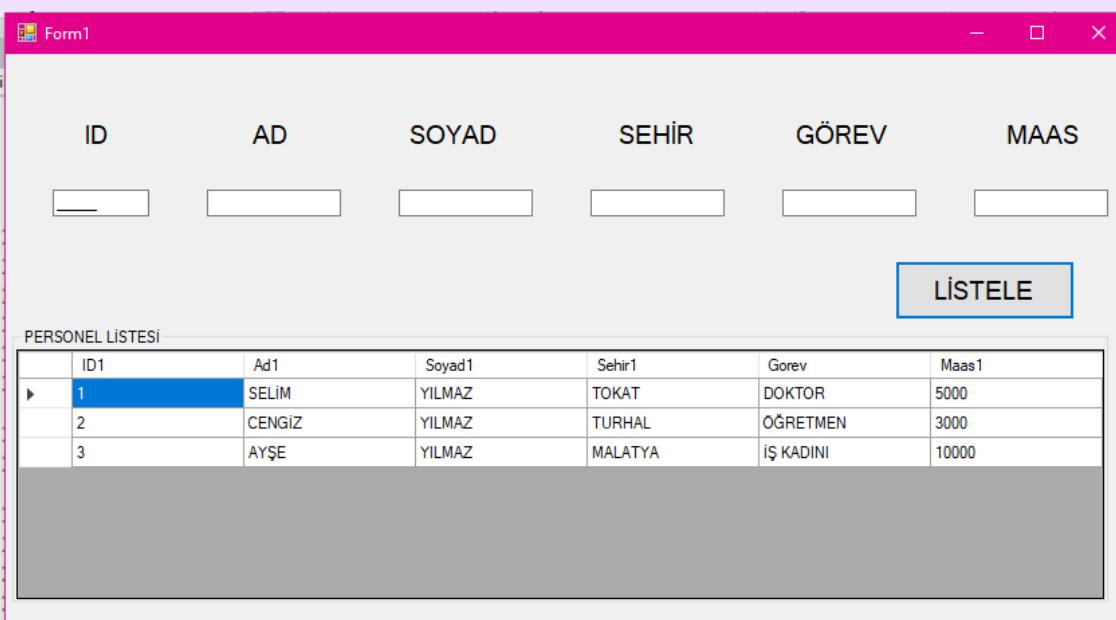
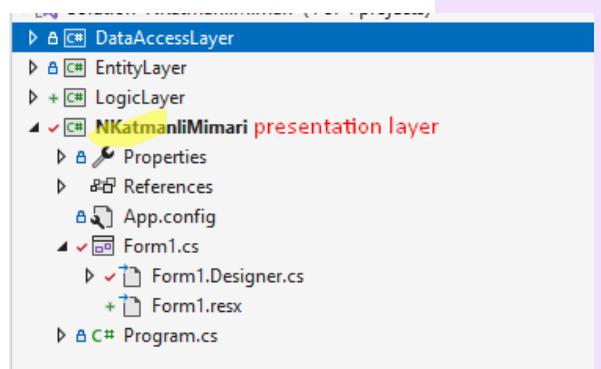
```

using EntityLayer;//referance 1 ekle
using DataAccessLayer;//reference 1 ekle
using LogicLayer;//reference 1 ekle

namespace NKatmanliMimari
{
    3 references
    public partial class Form1: Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            List<EntityPersonel> Perslist = LogicPersonel.LLpersonellistesi(); // List<EntityPersonel> den Perslist
            // adında bir değişken türet. LogicPersonel sınıfından oluşturduğumuz LLpersonellistesi metoduna ata.
            dataGridView1.DataSource = Perslist; //dataGridView1 in kaynağını Perslist değişkeni olarak ata.
        }
    }
}

```



## DEĞİŞKENLERİN OLUŞTURULMASI VE LİSTELEME

### ENTITY LAYER

değişkenleri oluştur

### DATAACCESS LAYER

Değişkenleri kullanarak metodu oluştur

### LOGİC LAYER

metodu doğrula metoda ata

### PRESENTATION LAYER

değişken oluşturup logic layerdaki metoda ata

```

public class EntityPersonel //veri tabanı之外クラスに変更
{
    //Bu class bittiğinde silin de olsa da klasının adını saklamak istenir.
    //bu classın içindeki değişkenlerin adlarını saklamak istenir.

    private int id; //id
    private string Ad; //Ad
    private string Soyad; //Soyad
    private string Sehir; //Sehir
    private string Gorev; //Gorev
    private short Maas; //Maas

    public int Id { get => id; set => id = value; }
    public string Ad { get => Ad; set => Ad = value; }
    public string Soyad { get => Soyad; set => Soyad = value; }
    public string Sehir { get => Sehir; set => Sehir = value; }
    public string Gorev { get => Gorev; set => Gorev = value; }
    public short Maas { get => Maas; set => Maas = value; }
}

```

```

using EntityLayer;//reference 1 ekle
using DataAccessLayer;//reference 1 ekle
using LogicLayer;//reference 1 ekle

namespace NKatmanliMimari
{
    public class EntityPersonel
    {
        //Bu class bittiğinde silin de olsa da klasının adını saklamak istenir.
        //bu classın içindeki değişkenlerin adlarını saklamak istenir.

        private int id; //id
        private string Ad; //Ad
        private string Soyad; //Soyad
        private string Sehir; //Sehir
        private string Gorev; //Gorev
        private short Maas; //Maas

        public int Id { get => id; set => id = value; }
        public string Ad { get => Ad; set => Ad = value; }
        public string Soyad { get => Soyad; set => Soyad = value; }
        public string Sehir { get => Sehir; set => Sehir = value; }
        public string Gorev { get => Gorev; set => Gorev = value; }
        public short Maas { get => Maas; set => Maas = value; }
    }
}

```

```

using EntityLayer;//reference 1 ekle
using DataAccessLayer;//reference 1 ekle
using LogicLayer;//reference 1 ekle

namespace NKatmanliMimari
{
    3 references
    public partial class Form1: Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            List<EntityPersonel> Perslist = LogicPersonel.LLpersonellistesi(); // EntityPersonel sınıfından list metodu oluşturuyoruz
            dataGridView1.DataSource = Perslist; //dataGridView1 in kaynağını Perslist değişkeni olarak ata.
        }
    }
}

```

## INSERT-EKLEME

**ENTITY LAYER**

**DATAACCESS LAYER**  
 -int türünde bir metot oluşturduk  
 -EntityClass sınıfından p adında nesne ile değişkenlere ulaşık.  
 -sonra sql kodlarını yazdık.

**LOGİC LAYER**  
metodu düzenley ve metoda ata

**PRESENTATION LAYER**  
nesne ile propertylere ulaş ve metotta çalıştır.

```
public static int Personellekle(EntityClass p)//farklı olarak bu sefer int türünden metot türümüzde parametre olarak EntityClass sınıfından p adında bir nesne kullanılmıştır.
{
    SqlCommand komut2 = new SqlCommand("insert into TBLBILGI (Ad, SOYAD, SEHIR, GOREV, MAAS) VALUES (@p1, @p2, @p3, @p4, @p5)", Baglanti.bgl);
    if (komut2.Connection.State == ConnectionState.Open)
    {
        komut2.Connection.Open();
    }

    komut2.Parameters.AddWithValue("@p1", p.Ad); //p il EntityClass taki dolgulama ile oluşmaktadır.
    komut2.Parameters.AddWithValue("@p2", p.Soyad);
    komut2.Parameters.AddWithValue("@p3", p.Sehir);
    komut2.Parameters.AddWithValue("@p4", p.Gorev);
    komut2.Parameters.AddWithValue("@p5", short.Parse(p.Maas.ToString()));

    return komut2.ExecuteNonQuery();
}
```

```
public static int LLPersonellekle(EntityClass p)//Dalpersonel sınıfında oluşturduğumuz türden bir metodu oluşturuyoruz.
//parametremiz ise belirtik metodu p nesnesiyi döndürüyor.
if(p.Ad=="" & p.Soyad=="")&p.Maas>=4000&p.Gorev.Length>=3)
{
    return DALPersonel.Personellekle(p);
}
else
{
    return -1;
}
```

```
private void btninsert_Click(object sender, EventArgs e)
{
    EntityClass ent = new EntityClass(); //EntityClass tan ent adında bir nesne oluşturuyoruz ent ile değişkenlere ulaşıp propertyler atıyoruz.
    ent.Id = txtid.Text;
    ent.Soyad = txtsoyad.Text;
    ent.Maas = short.Parse(txtmaas.Text);
    ent.Gorev = txtgorev.Text;
    ent.Sehir = txtsehir.Text;

    LogicPersonel.LLPersonellekle(ent); //LogicPersonel sınıfındaki LLPersonellekle metodunu ent ile çalıştırıyoruz
}
```

## DELETE-SİLME

**ENTITY LAYER**

**DATAACCESS LAYER**  
 -bool türünde bir metot oluşturduk  
 -int türünde parametre oluşturduk.  
 -sonra sql kodlarını yazdık.

**LOGİC LAYER**  
metodu düzenley ve metoda ata

**PRESENTATION LAYER**  
nesne ile propertylere ulaş ve metotta çalıştır.

```
public static bool Personelsil(int p)/bool türünde metot tanımlı ve parametresini int türünde ver
{
    SqlCommand komut3 = new SqlCommand("delete from TBLBILGI where ID=@p1",
    Baglanti.bgl); //silme komutunu yaz
    komut3.Parameters.AddWithValue("@p1", p);
    return komut3.ExecuteNonQuery() > 0?sonuçla döndürmek için bool olmalıdır
    dan büyük olduğunda yanlış olduğunda yani true olduğunda silinir diyoruz.
}
```

```
public static bool LLPersonelsil(int per)
{
    if (per > 0)
    {
        return DALPersonel.PersonelSil(per);
    }
    else
    {
        return false;
    }
}
```

```
private void btnsil_Click(object sender, EventArgs e)
{
    EntityClass ent = new EntityClass();
    ent.Id = int.Parse(mskperid.Text);
    LogicPersonel.LLPersonelsil(ent.Id);
}
```

## UPDATE-GÜNCELLEME

**ENTITY LAYER**

**DATAACCESS LAYER**  
 -bool türünde bir metot oluşturduk  
 -EntityClass sınıfından p adında nesne ile değişkenlere ulaşık..  
 -sonra sql kodlarını yazdık.

**LOGİC LAYER**  
metodu düzenley ve metoda ata

**PRESENTATION LAYER**  
nesne ile propertylere ulaş ve metotta çalıştır.

```
public static bool PersonelGuncelle(EntityClass p)/bool türünde metot oluştur
    parametresini EntityClass sınıfından p nesnesine atı
{
    SqlCommand komut5 = new SqlCommand("update TBLBILGI set Ad=@p1, SOYAD=@p2,
    SEHIR=@p3, GOREV=@p4, MAAS=@p5 where ID=@p6", Baglanti.bgl);

    komut5.Parameters.AddWithValue("@p1", p.Ad);
    komut5.Parameters.AddWithValue("@p2", p.Soyad);
    komut5.Parameters.AddWithValue("@p3", p.Sehir);
    komut5.Parameters.AddWithValue("@p4", p.Gorev);
    komut5.Parameters.AddWithValue("@p5", p.Maas);
    komut5.Parameters.AddWithValue("@p6", p.Id);

    if (komut5.Connection.State != ConnectionState.Open)
    {
        komut5.Connection.Open();
    }

    return komut5.ExecuteNonQuery() > 0;
}
```

```
public static bool LLPersonelguncelle(EntityClass per)
{
    if (per.Ad != "")
    {
        return DALPersonel.PersonelGuncelle(per);
    }
    else
    {
        class DataAccessLayer.DALPersonel
    }
    return false;
}
```

```
private void btnguncelle_Click(object sender, EventArgs e)
{
    EntityClass ent = new EntityClass();
    ent.Id = int.Parse(mskperid.Text);
    ent.Ad = txtad.Text;
    ent.Soyad = txtsoyad.Text;
    ent.Maas = short.Parse(txtmaas.Text);
    ent.Sehir = txtsehir.Text;
    ent.Gorev = txtgorev.Text;

    LogicPersonel.LLPersonelguncelle(ent);
}
```

# XML

(XML (Extensible Markup Language ya da Türkçesile Genişletilebilir İşaretleme Dili), hem insanlar hem bilgi işlem sistemleri tarafından kolayca okunabilecek dokümanlar oluşturmaya yarayan bir işaretleme dilidir)

```
using System.Xml; // XML kütüphanesini ekle  
using System.Data.SqlClient;
```

```
string adres = "https://www.tcmb.gov.tr/kurlar/today.xml"; //xml şeklinde ~  
verileri paylaşan sitenin adresini yaz.  
var xmldosya = new XmlDocument(); //xml dökümanını bir değişkene bağlıyoruz  
xmldosya.Load(adres); //dökümanı adrese bağlıyorak başlatıyoruz.  
  
string dolaralis = xmldosya.SelectSingleNode("Tarih_Date/Currency  
[@Kod='USD']/BanknoteBuying").InnerXml; //SelectSingleNode metodıyla ~  
sitedeki bilgileri ayıklıyorak değişkene bağlıyoruz.  
lbldolaralis.Text = dolaralis;
```

<https://www.tcmb.gov.tr/kurlar/today.xml>

yukarıdaki sitenin sayfa kaynağına gidip aşağıdaki kodlara ulaşıyoruz

```
<Tarih Date Tarih="27.06.2025" Date="06/27/2025" Bulten_No="2025/119" >  
  <Currency CrossOrder="0" Kod="USD" CurrencyCode="USD">  
    <Unit>1</Unit>  
    <Isim>ABD DOLARI</Isim>  
    <CurrencyName>US DOLLAR</CurrencyName>  
    <ForexBuying>39.7424</ForexBuying>  
    <ForexSelling>39.8140</ForexSelling>  
    <BanknoteBuying>39.7145</BanknoteBuying>  
    <BanknoteSelling>39.8737</BanknoteSelling>  
    <CrossRateUSD/>  
    <CrossRateOther/>
```

## KISA BİLGİ

```
private void txtkur_TextChanged(object sender, EventArgs e)  
{  
    txtkur.Text = txtkur.Text.Replace(".", ","); //ilk tırnaktaki sembol ~  
    // yerine ikinci tırnaktaki sembolle değiştirir.  
}
```

## İKİ BOYUTLU NESNELERİ HAREKET ETTİRME

```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.NumPad2 && pictureBox1.Top<388)
    {
        pictureBox1.Top += 5;//pictureboxın üstünü 5 er artırır.
    }

    if (e.KeyCode == Keys.NumPad8 && pictureBox1.Top>0)
    {
        pictureBox1.Top -= 5;
    }

    if(e.KeyCode==Keys.NumPad4&& pictureBox1.Left > 0)
    {
        pictureBox1.Left -= 5;//pictureboxın solunu 5 er azaltır.
    }

    if(e.KeyCode==Keys.NumPad6&& pictureBox1.Left <= 723)
    {
        pictureBox1.Left += 5;
    }
}
```

## SQL VERİLERİNİ ŞİFRELEME VE ŞİFREYİ ÇÖZME

### SİFRELEME

```
public string sifre(string ad)
{
    //parametrelili metot oluşturduk ve ad değişkeni oluşturduk
    string turad=((ad.ToLower()).Replace("ş", "s")).Replace("ö", "o").Replace
        ("ç", "c");//lower küçük harfe çevir, replace değiştirerek yaz
    byte[] addizi = ASCIIEncoding.ASCII.GetBytes(turad);
    //byte türünde addizi adında bir dizi oluşturduk. bu dizinin elemanlarını ad
    //değişkeninden gelen string bytelara ayırdık ve her bir byte 1
    //ASCII metoduna göre şifreleyip addizi dizisine attık.
    string adsifre = Convert.ToString(addizi);
    //adsifre stringi oluşturup bu stringin addizisini tobase64 yöntemiyle şifreledik.
    return adsifre;
}
```

### SİFRE ÇÖZME

```
public string sifrecoz(string adcozum)
{
    //parametrelili metot oluşturup adcozum adında string türünde bir parametre ~
    //oluşturduk
    byte[] adcozumdizisi = Convert.FromBase64String(adcozum);
    //byte türünde dizi oluşturup bu dizinin elemanlarını adcozum stringini şifresini ~
    //çözerek dizi elemanlarına atadık.
    string adverisi = ASCIIEncoding.ASCII.GetString(adcozumdizisi);
    //dizi elemanlarını ASCII sembollerine göre çözümleyip adverisine atadık.
    return adverisi;
}
```

## DATAGRIDDE ŞİFREYİ ÇÖZEREK GöSTERME(VERİTABANI ŞİRELEME PROJESİ)

```
void listele()
{
    SqlDataAdapter da = new SqlDataAdapter("select AD,SOYAD,MAIL,SIFRE,HESAPNO FROM
    TBLBANKAHESABI", baglanti());//sorguyu yazdık
    DataSet ds = new DataSet(); //dataset sınıfından ds adında nesne ürettik
    da.Fill(ds); //ds yi da ya atadık.
    foreach (DataRow row in ds.Tables[0].Rows) //foreach döngüsü

        //DataRow sınıfından row adında bir değişken oluşturduk ve in yöntemiyle ds
        //nin içindeki 0inci tablodaki sıraları dizi yapıp her bir dizi elemanına ~
        //atadık.
    {
        row["AD"] = sifrecoz(row["AD"] as string);
        //sifrecoz metoduyla AD sütunundaki bütün rowları string ederek şözdük ve ~
        //tekrar AD sutunundaki rowlara atadık.
        row["SOYAD"] = sifrecoz(row["SOYAD"] as string);
        row["MAIL"] = sifrecoz(row["MAIL"] as string);
        row["SIFRE"] = sifrecoz(row["SIFRE"] as string);
        row["HESAPNO"] = sifrecoz(row["HESAPNO"] as string);
    }
    dataGridView1.DataSource = ds.Tables[0];
}
```

```
SqlDataAdapter da = new SqlDataAdapter("SELECT OGRID,
    OGRAD,OGRSOYAD,FOTOGRAF ,SINIF,TELEFON,MAIL FROM TBLOGRENCI", bgl.baglanti());
DataSet ds = new DataSet();
da.Fill(ds);
foreach (DataRow row in ds.Tables[0].Rows)
{
    row["OGRID"] = row["OGRID"];
    row["OGRAD"] = row["OGRAD"];
    row["OGRSOYAD"] = row["OGRSOYAD"];
    row["FOTOGRAF"] = row["FOTOGRAF"];
    row["SINIF"] = row["SINIF"];
    row["TELEFON"] = sifrecoz(row["TELEFON"] as string);
    row["MAIL"] = row["MAIL"];
}
dataGridView1.DataSource = ds.Tables[0];
button3.Text = "ETÜT LİSTELE";
```

### ETÜT MERKEZİ PROJESİ

## DATAGRİDDE CELL İN DURUMUNA GÖRE ROW UN ÖZELLİKLERİNİ DEĞİŞTİRMЕ

```
void listeleetut()
{
    SqlDataAdapter da = new SqlDataAdapter("SELECT * FROM ETUT", bgl.baglanti());
    DataTable dt= new DataTable();
    da.Fill(dt);
    dataGridView1.DataSource = dt;

    DataGridViewCellStyle stil = new DataGridViewCellStyle();
    //datagridviewcellstyle sınıfından stil adında bir nesne tanımlıyoruz.
    for(int i=0; i < dataGridView1.Rows.Count -1; i++) //döngü oluşturuyoruz
    {
        if (Convert.ToBoolean(dataGridView1.Rows[i].Cells[9].Value) == true)
            //tablonun iinci sırasının 9uncu hücresinin değerini boolean değişkenine ~
            //çevirip sorguluyoruz.
        {
            stil.BackColor = Color.Yellow;
            dataGridView1.Rows[i].DefaultCellStyle = stil;
        }
    }
    dataGridView1.Columns[1].Visible = false;
    dataGridView1.Columns[3].Visible = false;
    dataGridView1.Columns[5].Visible = false;
}
```

## DATAREADER İLE KAYIT SORGULAYARAK INSERT YAPMA(ETUT MERKEZİ PROJESİ)

```
string okuders;
1 reference
private void btndersekle_Click(object sender, EventArgs e)
{
    if (txtdersadi.Text != "")
    {
        SqlCommand komutders = new SqlCommand("select BRANSAD FROM TBLBRANS WHERE BRANSAD=@A1", bgl.baglanti());
        komutders.Parameters.AddWithValue("@A1", txtdersadi.Text.ToUpper());
        SqlDataReader rd = komutders.ExecuteReader();
        if (rd.Read() == true)
        {
            MessageBox.Show("Bu ders zaten var. Başka bir ders ekleyiniz.");
            okuders = "true";
        }
        else
        {
            okuders = "false";
        }
        bgl.baglanti().Close();
        if (okuders=="false")
        {

            SqlCommand komutdersekle = new SqlCommand("INSERT INTO TBLBRANS (BRANSAD) VALUES(@P1)", bgl.baglanti());
            komutdersekle.Parameters.AddWithValue("@P1", txtdersadi.Text.ToUpper());
            komutdersekle.ExecuteNonQuery();
            MessageBox.Show("Ders eklendi");
            bgl.baglanti().Close();
        }
    }
}
```

## OPENFILEDIALOG İLE PICTUREBOX A RESİM GETİRME(ETUT MERKEZİ PROJESİ)

```
using System.IO;

private void btnfotografyukle_Click(object sender, EventArgs e)
{
    openFileDialog1.ShowDialog();
    pictureBox1.ImageLocation = openFileDialog1.FileName;
}

else if (button3.Text=="ETUT LISTELE")
{
    txtogrenciad.Text= dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
    txtogrencisoyad.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
    pictureBox1.ImageLocation = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();
    txtogrencisinif.Text = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();
    mskogrencitelefon.Text = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();
    txtogrencimail.Text = dataGridView1.Rows[e.RowIndex].Cells[6].Value.ToString();
}
```

## BITMAP İLE RESİM ÜZERİNE YAZI YAZMA UYGULAMASI

### ResimUzerineYaziYaz uygulaması

```

string resim;
1 reference
private void btnresimsec_Click(object sender, EventArgs e)
{
    openFileDialog1.ShowDialog();
    resim=openFileDialog1.FileName;
}
Color renk;
1 reference
private void btnrenksec_Click(object sender, EventArgs e)
{
    colorDialog1.ShowDialog();
    renk = colorDialog1.Color;
}
Bitmap bmp;
1 reference
private void btnyazdir_Click(object sender, EventArgs e)
{
    bmp=new Bitmap(resim);
    Graphics graphics = Graphics.FromImage(bmp);
    graphics.DrawString(txtmetin.Text,new Font("JOKERMAN",Convert.ToInt16(txtboyut.Text), FontStyle.Bold),new SolidBrush(renk),20,30);
    pictureBox1.Image = bmp;
}

1 reference
private void btnkaydet_Click(object sender, EventArgs e)
{
    saveFileDialog1.Filter = "RESİM|.PNG";
    saveFileDialog1.ShowDialog();
    bmp.Save(saveFileDialog1.FileName);
}

```

## EXCELL İLE KAYIT ALMA

```

namespace Excellilekayit
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        OleDbConnection bgl = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Users\CENGİZ\OneDrive\OGRENCIYAT.xlsx;Extended Properties='Excel 12.0 Xml;HDR=YES'");
        //BAGLANTI ADRESİNİ https://www.connectionstrings.com/excel/ adresinden alıyoruz. Datasource kısmını https://www.connectionstrings.com/excel/ dosyamızın dosya yolundan alıp yapıştırıyoruz. Properties kısmını tek tıkla olacak.

        3 references
        void listele()
        {
            OleDbDataAdapter da = new OleDbDataAdapter("select * from [SAYFA1$]", bgl);
            //oledb ile aynı özelliklere sahip. SADECE tablo ismini sayfa adını yukarıdaki yazıyoruz.
            DataTable dt = new DataTable();
            da.Fill(dt);
            dataGridView1.DataSource = dt;
        }
    }
}

```

<https://www.connectionstrings.com/excel/>

## FORMU SİMGİ DURUMUNA KÜÇÜLTME

```
private void pictureBox3_Click(object sender, EventArgs e)
{
    this.WindowState = FormWindowState.Minimized;
}
```

## DATAGRID SCROLLBAR POSITION

```
dataGridView1.FirstDisplayedScrollingRowIndex = sira;
//datagridin scroll görünür sırasını sira değişkeninden alır.
```















































































































