**Polis**

**GBT Sorgulama**

**146206051**

**Mehmet UĞUZ**

**Görsel Programlama III**

**Hüseyin ELDEM**

**15/16**

**İçindekiler**

1. Genel Kullanım Amacı ve Bilgi…………………………………………. 3

2. Sistem Tasarımı ve Giriş……………………………………………....…. 4

3. Sistem Tasarımı……………………………………………………...…… 5

3.1 Veritabanı Tasarımı ve Detayları……………………….…...…... 6

3.2 Veritabanı İlişkileri..………………………………..……..……... 13

4. Form (Arayüz) Tasarımı……………………………………….………… 14

4.1 Giriş Arayüzü Tasarımı…………………………………..……… 14

4.2 Şahıs Sorgulama Arayüzü Tasarımı………………………...….... 15

4.3 Şahıs Değiştir/Ekle/Sil Arayüzü Tasarımı………………..……... 16

4.4 Şahısları Görüntüle Arayüzü Tasarımı……………………..…… 17

5. Kod Tasarımı………………………………………………………...…… 18

5.1 Giriş Ekranı………………………………………………………. 18

5.2 Şahıs Değiştir/Ekle/Sil Ekranı ………………………………...… 19

Son………………………………………………………………………..…. 28

**1. Genel Kullanım Amacı ve Bilgi:**

**(**G)enel (B)ilgi (T)oplama sistemi adı altında şahısların kimlik, adres ve suçların detaylı şekilde tutulduğu fiş kayıt sistemidir.

GBT sorgulama ile;

* Sisteme suç işlemiş, işlememiş veya ceza aldıysa cezasını çekmiş bütün şahıs bilgileri kayıt edilebilir.
* Sistemde bulunan bir şahısı, TC kimlik numarası, şahıs numarası, fiş numarası ile arama yapıp kayıtlı olan bütün bilgilerine ulaşılabilir.

Bu yapmış olunan otomasyon ile hızlı ve kolay şekilde sisteme şahıs ekleme, şahıslar üstünde bilgi düzenlemesi yapma ve şahısların detaylı bilgilerine ulaşıp gerekli işlemleri gerçekleştirme gibi işlemler için basit ve işlevsel bir arayüz ile tasarlanmıştır.

**2. Sistem Tasarımı ve Giriş:**

Bu proje öncelikle sistemdeki şahısların kimlik, adres ve fiş bilgilerini sorgulamak için tasarlanmıştır. Ek olarak sisteme şahıs ekleme, kayıtlı bir şahsı düzenleme ve silme gibi işlemleri de bünyesinde bulundurmaktadır. Proje, programlama dili olarak C#, veritabanı olarak ise Microsoft SQL Server 2014 kullanılmıştır.

Kullanıcıyı birçok doküman ve kolay olmayan kayıt-fiş ilişkilendirmesinden kurtarıp daha kolay bir arayüz ile basit bir kullanım için tasarlanmıştır.

Diğer otomasyonlardaki gibi kullanıcıyı başka bir yerde, işlemi başka bir yerde kayıt etmek veya aramak yerine bütün işlemleri tek bir sayfada birbirine ilişkili şekilde yapılabilmesine olanak sağlar.

Projenin adımları:

* Veritabanının mantıksal olarak tasarlanması
* Veritabanının fiziksel olarak MSSQL ile tasarlanması
* C# ile kullanıcı giriş formunun tasarlanması
* C# ile kayıtların eklenip, düzenlendiği ve silindiği formun tasarlanması
* Proje raporunun hazırlanması

**3. Veri Tasarımı:**

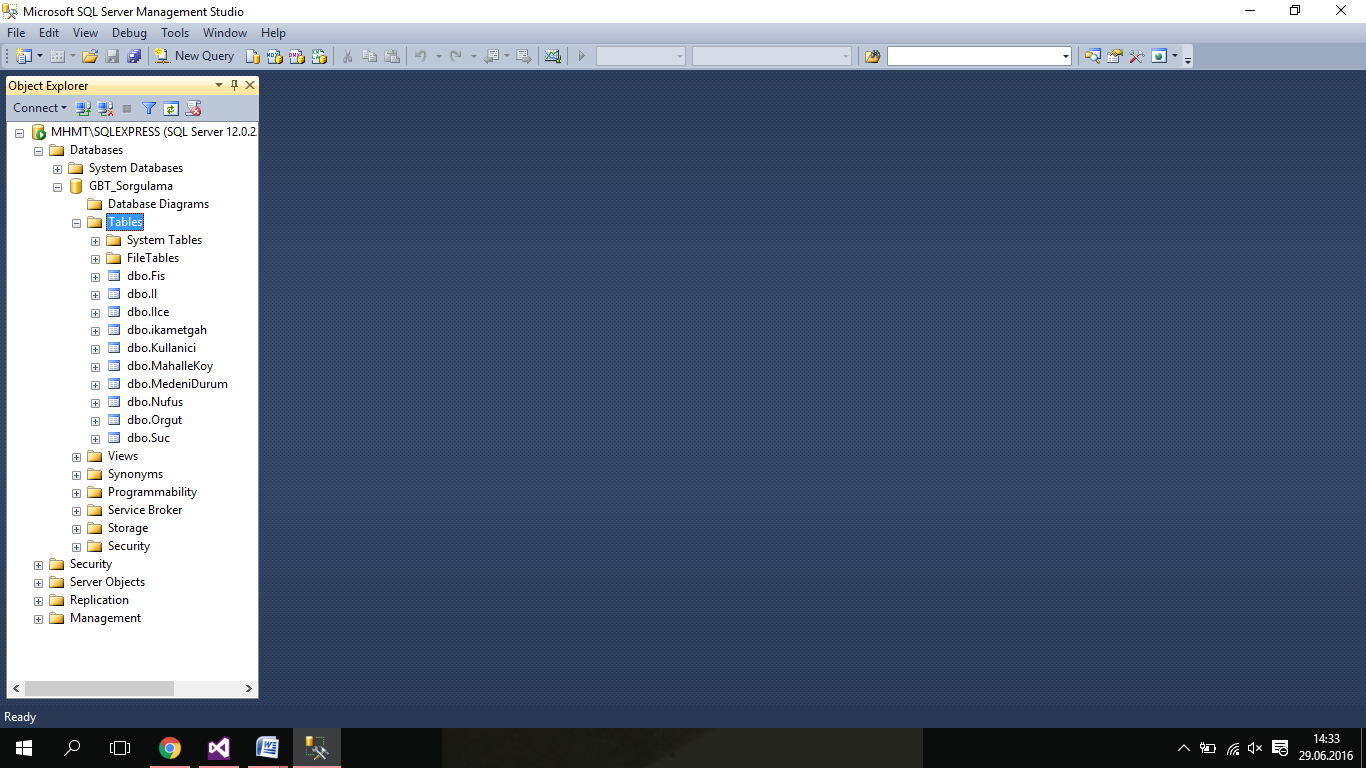
Sistem veri kaynağı olarak MSSQL’de yapılmış veritabanına kayıtlı verileri kullanmaktadır. Bu veritabanı ise 146206051\_MehmetUğuz\GBT\_Sorgulama\Veritabanı Sql klasörünün içinde bulunmaktadır.

Ek olarak;

Sisteme kayıtlı şahısların vesikalık fotoğrafları için ayrı bir dosya kullanılmaktır. Bu dosyada bulunan resimlerin url’leri veri tabanında kayıtlı bulunmaktadır. Bu resimler 146206051\_MehmetUğuz\GBT\_Sorgulama\SahisFoto klasörü içinde bulunmaktadır.

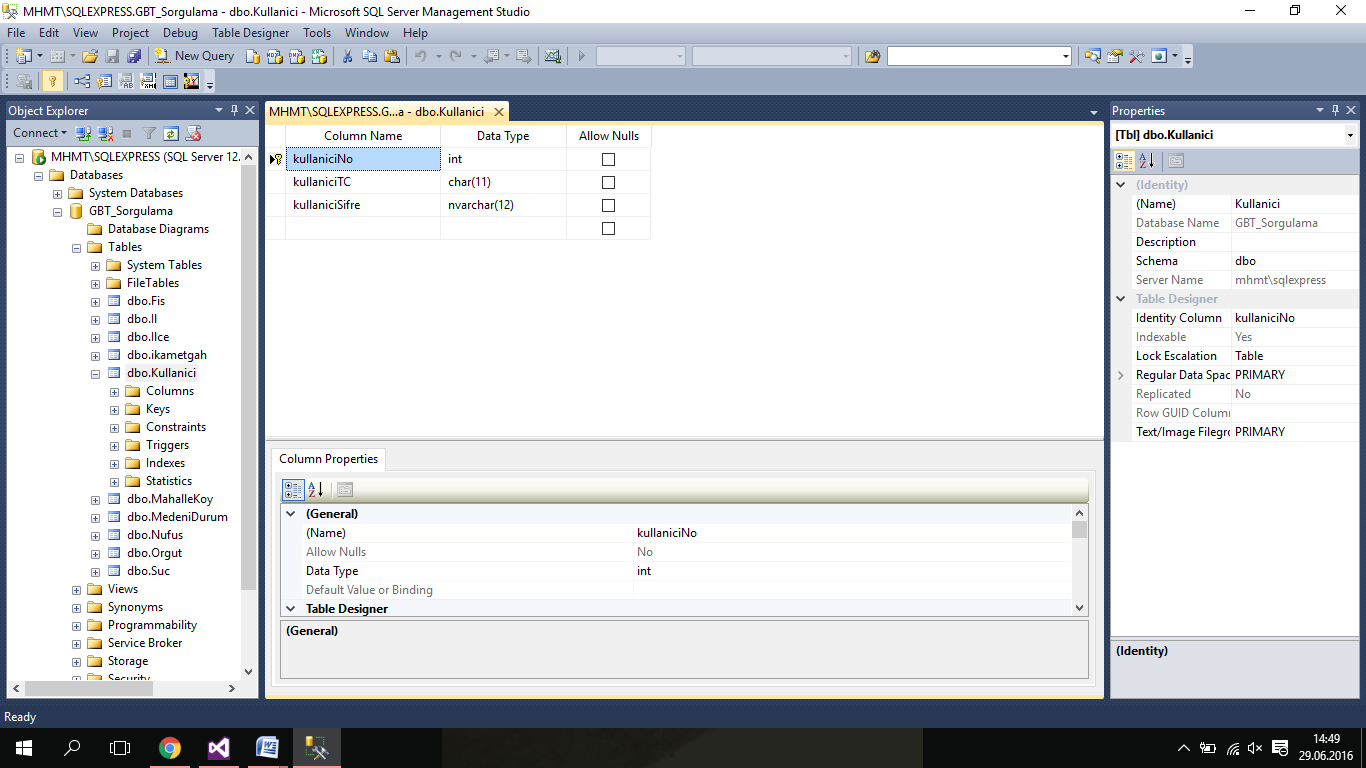
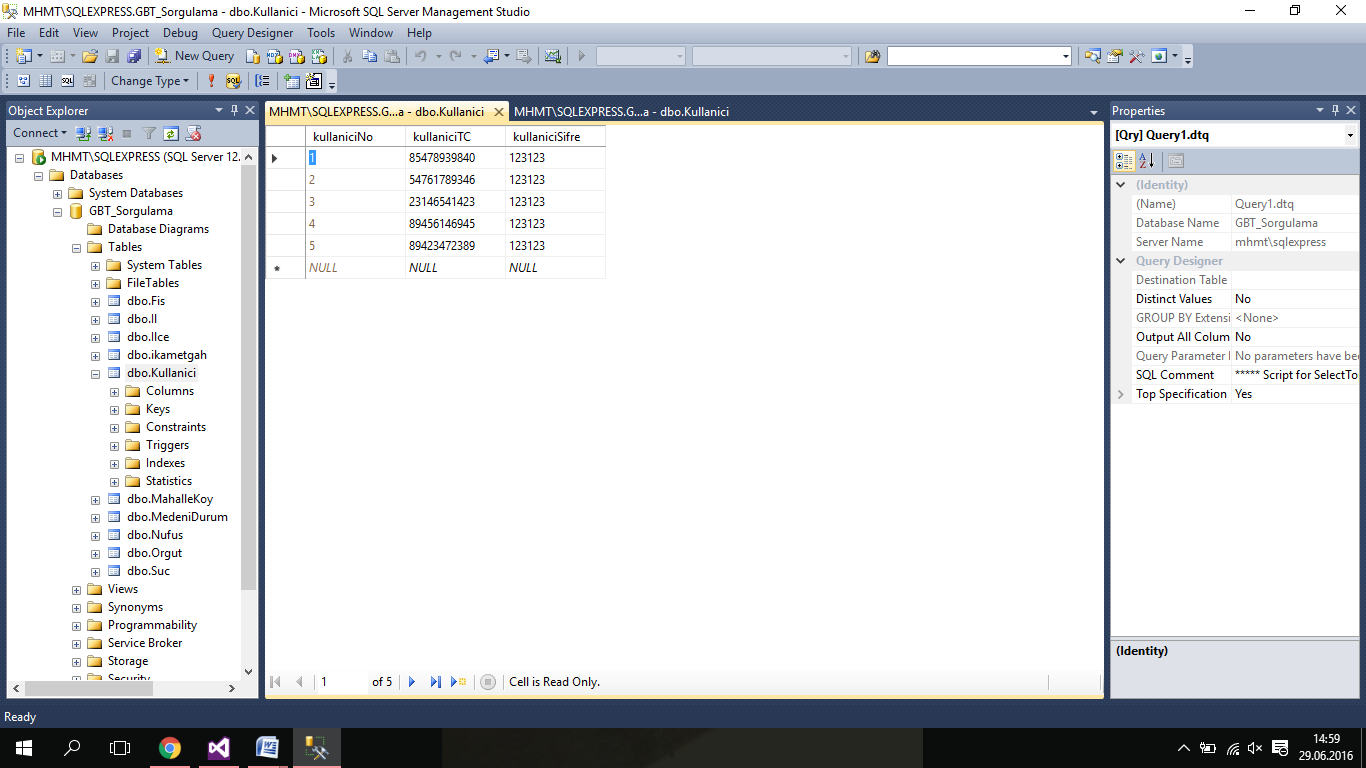
Tasarımda kullanılan bütün icon,resim,tema gibi itemler ise 146206051\_MehmetUğuz\GBT\_Sorgulama\Dosyalar içinde bulunmaktadır.

**3.1 Veritabanı Tasarımı ve Detayları:**

****Hazırlanan veritabanı içerisinde;

* Kullanici (Sisteme giriş yapacak kullanıcılar)’ın bilgilerinin tablosu
* Nufus (Ad,soyad vb.) bilgilerinin tablosu
* İkametgah (Sıra No,Adres vb.) bilgilerin tablosu
* Fis (Suç,Tahk. yapan makam vb.) bilgilerin tablosu
* Il,Ilce,MahalleKoy gerekli adres bilgilerinin tutulduğu “İkametgah” tablosu ile ilişkili tablolar
* MedeniDurum (evli,bekar vb.) bilgilerin tutulduğu “Nufus” tablosu ile ilişkili tablo
* Orgut,Suc gerekli bilgilerin tutulduğu “Fis” tablosu ile ilişkili tablolar

“Kullanici” Tablosu:

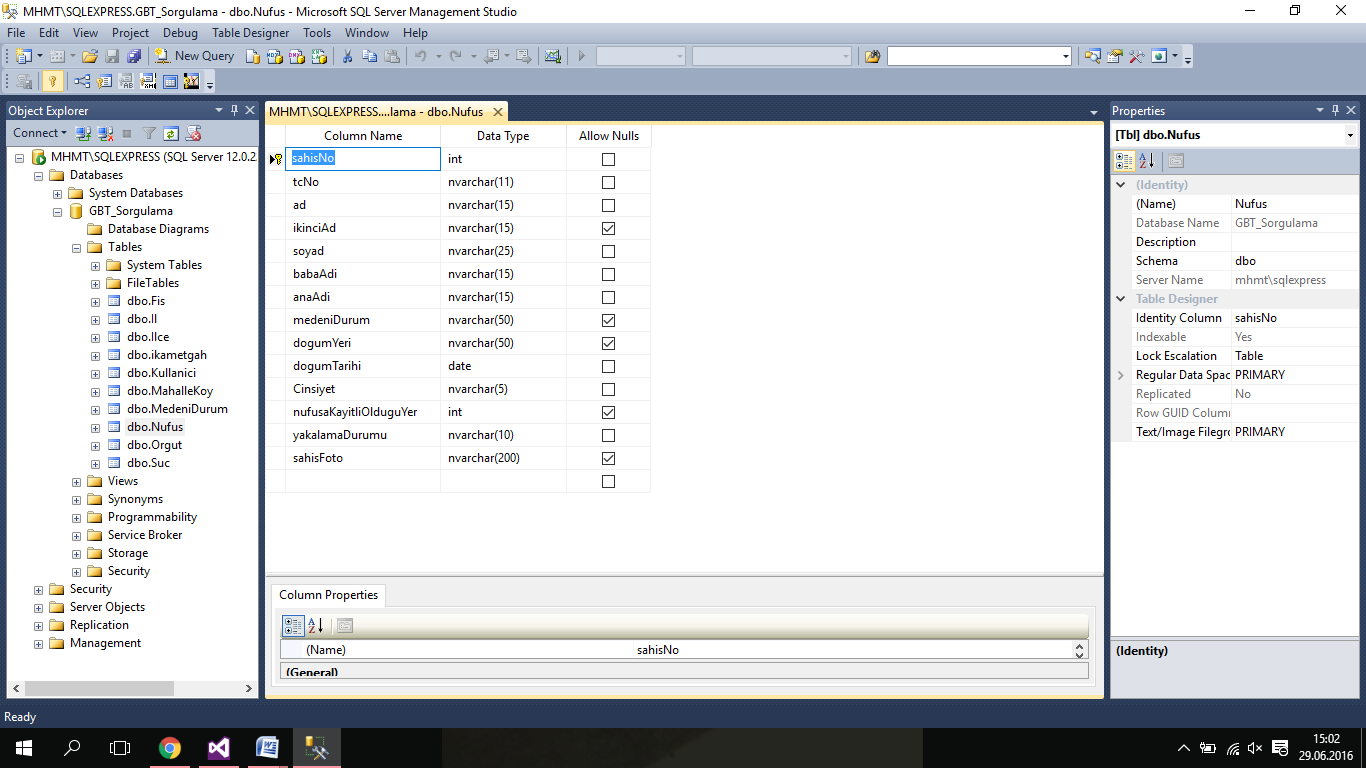
kullaniciNo, kullaniciTC, kullaniciSifre isminde 3 adet alan tanımlanmıştır.

kullaniciNo: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır.

kullaniciTC: Boş geçilemez. Veri tipi char ve 11 karakter ile sınırlıdır. Sisteme giriş yaparken kullanılan id benzeri öğe bu kayıttır.

kullaniciSifre: Boş geçilemez. Veri tipi nvarchar ve 12 karakter ile sınırlıdır .Sisteme giriş yaparken kullanılan şifre buradaki kayıtlardır.

“Nufus” Tablosu:



sahisNo, tcNo, ad, ikinciAd, soyad, babaAdi, anaAdi, medeniDurum, dogumYeri, dogumTarihi, Cinsiyet, nufusaKayitliOlduğuYer, yakalamaDurumu, sahisFoto isminde 14 adet alan tanımlanmıştır.

sahisNo: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır. Şahıs arama parametrelerinden biridir.

tcNo: Boş geçilemez. Veri tipi nvarchar ve 11 karakter ile sınırlıdır. Şahısın kimliğindeki vatandaşlık numarasının olduğu kayıttır. Şahıs arama parametrelerinden biridir.

ad: Boş geçilemez. Veri tipi nvarchar ve 15 karakter ile sınırlıdır. Şahısın isminin olduğu kayıttır.

ikinciAd: Boş geçilebilir. Veri tipi nvarchar ve 15 karakter ile sınırlıdır. Şahısın ikinci isminin olduğu kayıttır.

soyad: Boş geçilemez. Veri tipi nvarchar ve 25 karakter ile sınırlıdır. Şahısın soyisminin olduğu kayıttır.

babaAdi: Boş geçilemez. Veri tipi nvarchar ve 15 karakter ile sınırlıdır. Şahısın babasının isminin olduğu kayıttır.

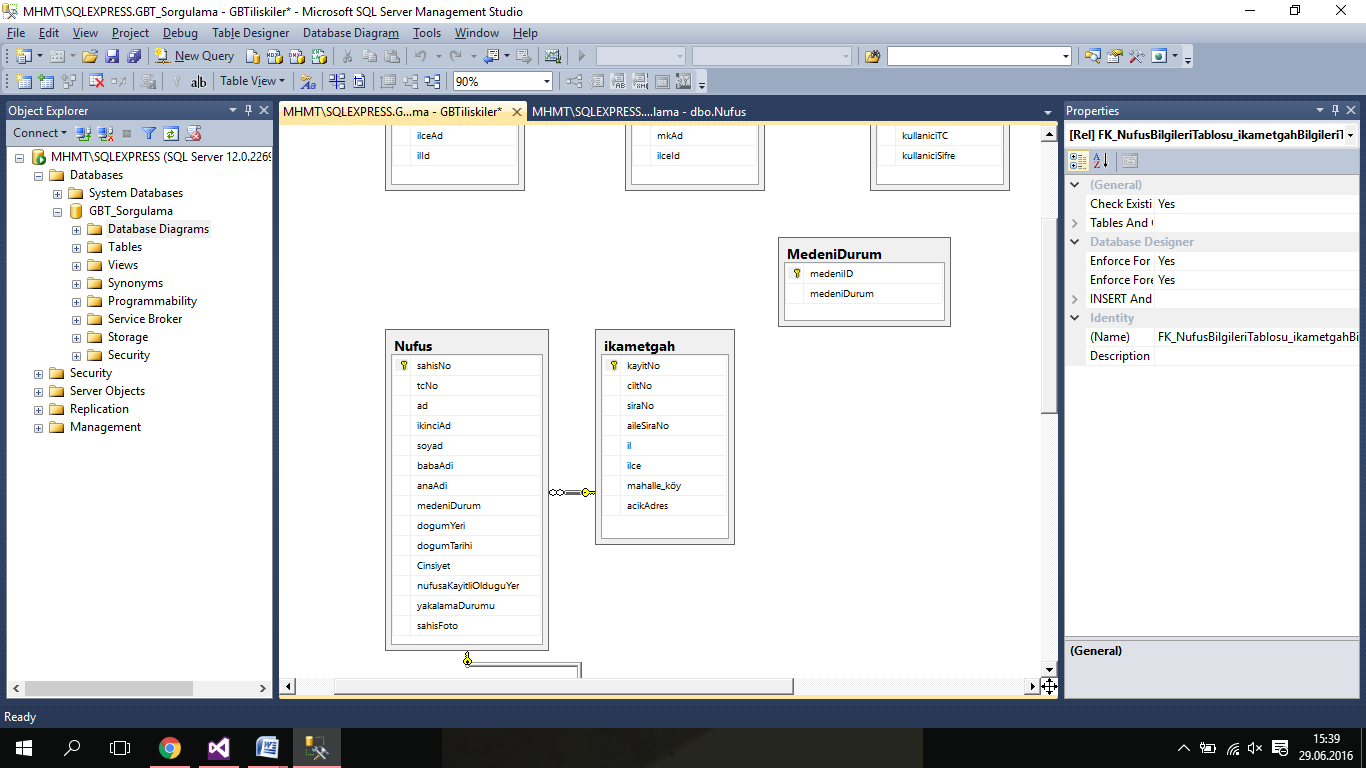
anaAdi: Boş geçilemez. Veri tipi nvarchar ve 15 karakter ile sınırlıdır. Şahısın annesinin isminin olduğu kayıttır.

medeniDurum: Boş geçilebilir. Veri tipi nvarchar ve 50 karakter ile sınırlıdır. Şahısın medeni durumunun bulunduğu kayıttır. Ekleme yapılırken “MedeniDurum” tablosundaki parametreler kullanılır.İlişkili değildir.

dogumYeri: Boş geçilebilir. Veri tipi nvarchar ve 50 karakter ile sınırlıdır. Şahısın doğum yerinin olduğu kayıttır. Ekleme yapılırken “Il” tablosundaki parametreler kullanılır. İlişkili değildir.

dogumTarihi: Boş geçilemez. Veri tipi date’dir. Şahısın doğum tarihinin gün/ay/yıl olarak kayıtlı olduğu alandır.

Cinsiyet: Boş geçilemez. Veri tipi nvarchar ve 5 karakter ile sınırlıdır. Şahısın cinsiyet durmunun olduğu kayıttır. Radio buttonların durumuna göre erkek/kadın bilgisi alır.

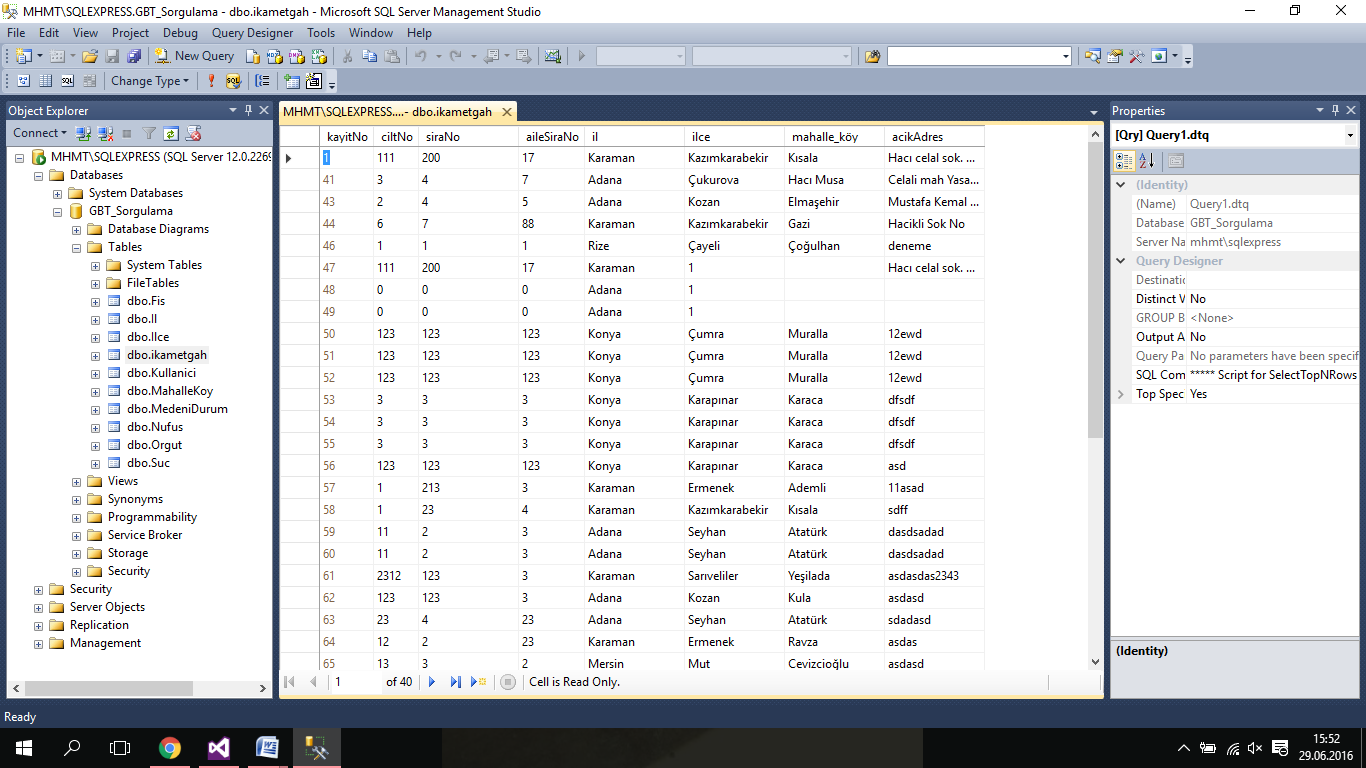
nufusaKayitliOlduğuYer: Boş geçilebilir. Veri tipi int(sayı)’dir. İnt olmasının sebebi ise ikametgah tablosu ile ilişkili olmasıdır. Nufus.nufusaKayitliOlduguyer =ikametgah.kayitNo (Şekil 1)

yakalamaDurumu: Boş geçilemez. Veri tipi nvarchar ve 10 karakter ile sınırlıdır. Şahısın yakalama durumunun olduğu kayıttır.

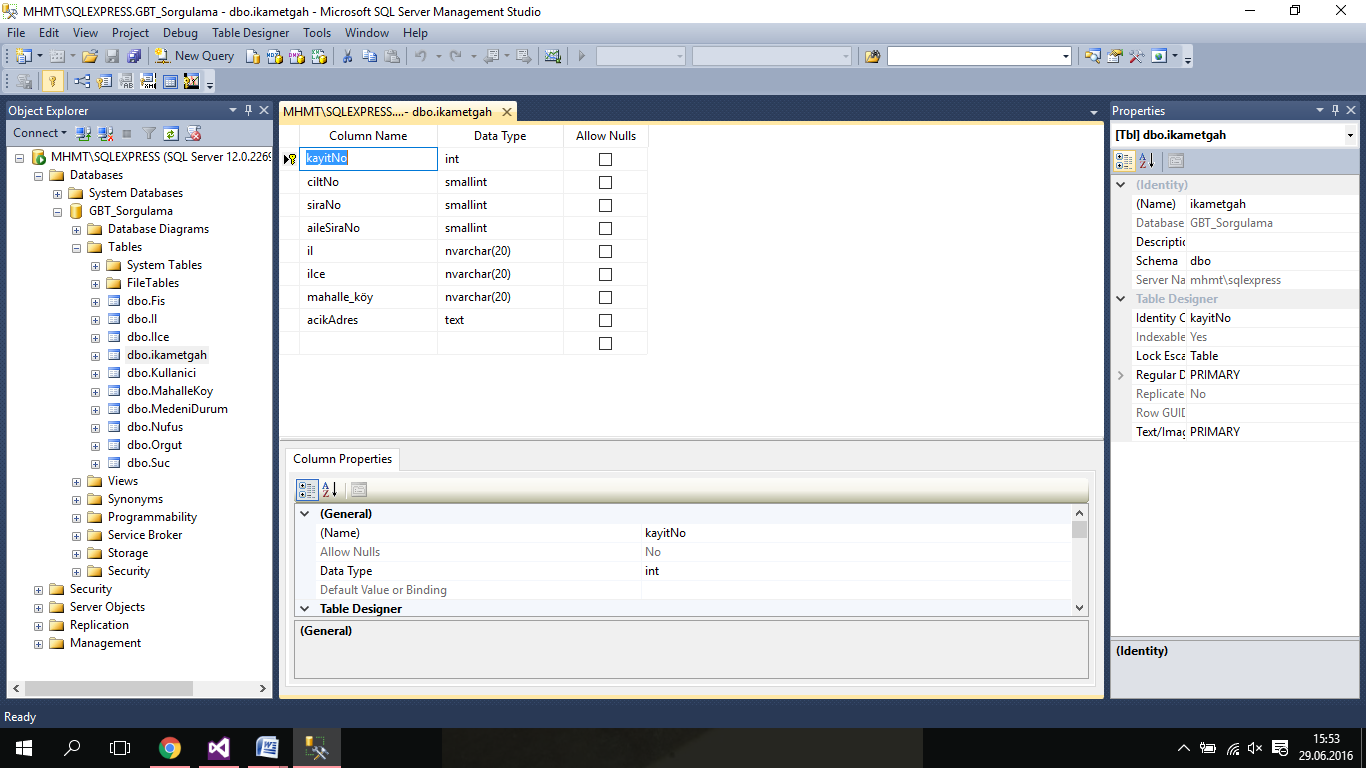
sahisFoto: Boş geçilebilir. Veri tipi nvarchar ve 200 karakter ile sınırlıdır. Şahısın resminin bulunduğu yerin(url) olduğu kayıttır.

Şekil 1

“ikametgah” Tablosu:



kayitNo, ciltNo, siraNo, aileSiraNo, il, ilce, mahalle\_köy, acikAdres isimli 8 adet alan tanımlanmıştır.

 kayitNo: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır. Yukarıdaki şekil 1’de gösterildiği gibi “Nufus” tablosu ile ilişkilidir.

ciltNo: Boş geçilemez. Veri tipi smallint’dir.Çok fazla bir sayısal değer tutmayacağı için veritabanında az yer kaplayan smallint seçilmiştir. Şahısın kimlikteki cilt numarasının olduğu kayıttır.

siraNo: Boş geçilemez. Veri tipi smallint’dir.Çok fazla bir sayısal değer tutmayacağı için veritabanında az yer kaplayan smallint seçilmiştir. Şahısın kimlikteki sira numarasının olduğu kayıttır.

aileSiraNo: Boş geçilemez. Veri tipi smallint’dir.Çok fazla bir sayısal değer tutmayacağı için veritabanında az yer kaplayan smallint seçilmiştir. Şahısın kimlikteki aile sıra numarasının olduğu kayıttır.

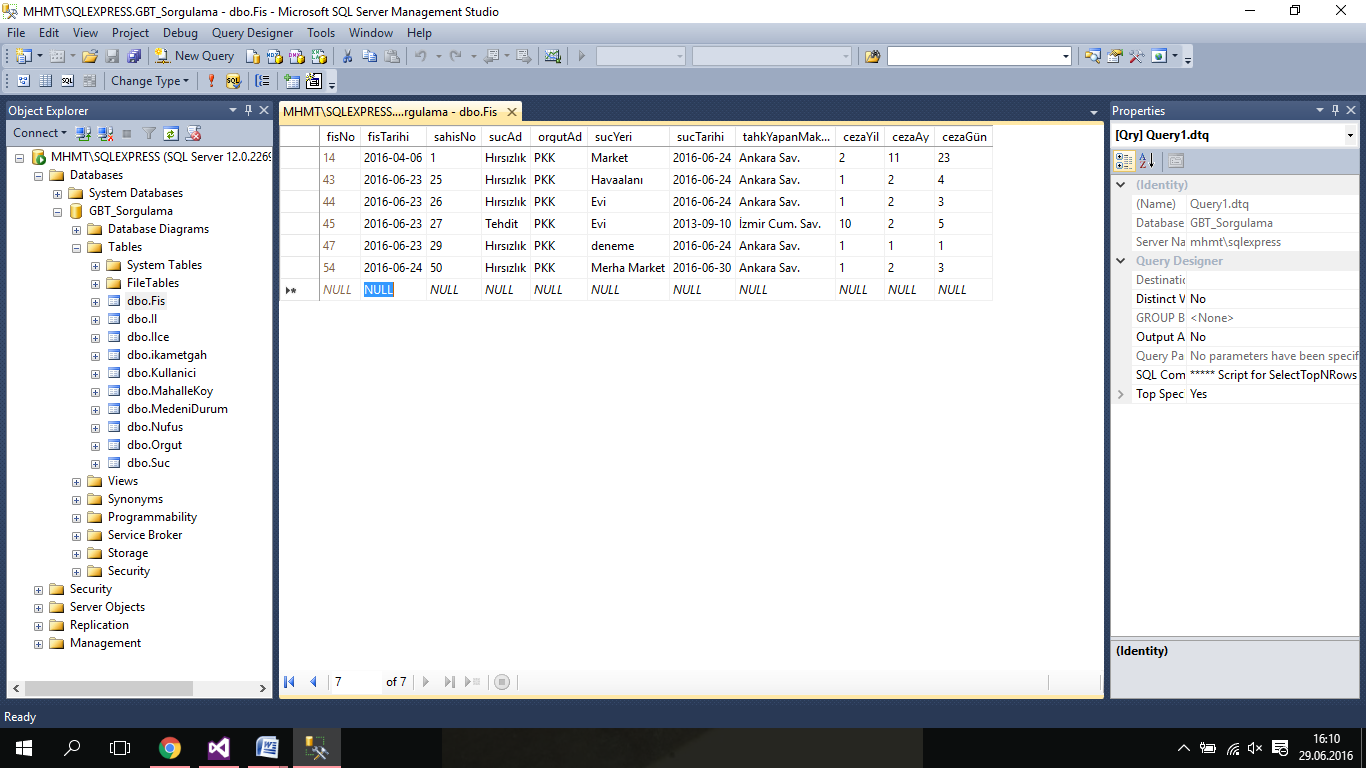
il: : Boş geçilemez. Veri tipi nvarchar ve 20 karakter ile sınırlıdır. Ekleme yapılırken “Il” tablosundaki parametreler kullanılır. İlişkili değildir. Şahısın il bilgisinin olduğu kayıttır.

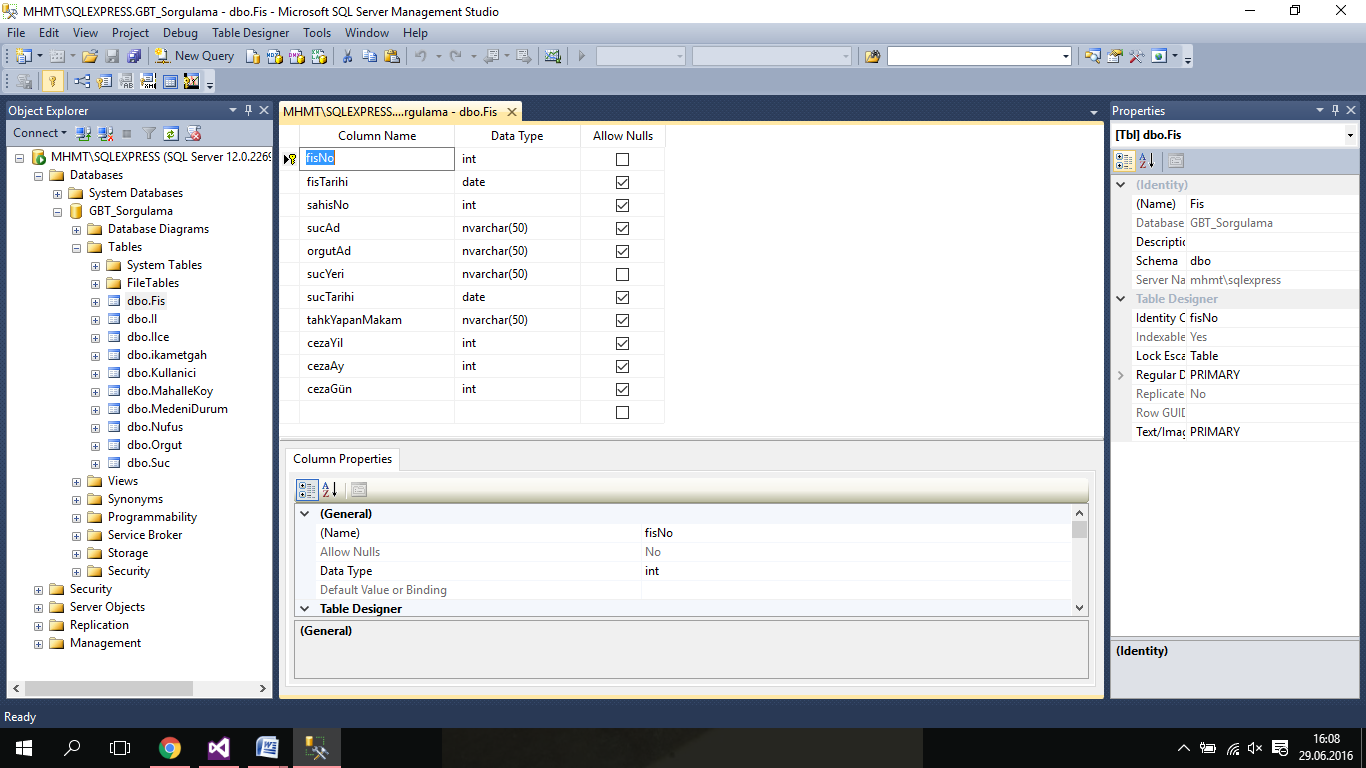
ilce: : Boş geçilemez. Veri tipi nvarchar ve 20 karakter ile sınırlıdır. Ekleme yapılırken “Ilce” tablosundaki parametreler kullanılır. İlişkili değildir. Şahısın ilce bilgisinin olduğu kayıttır.

Mahalle\_köy: : Boş geçilemez. Veri tipi nvarchar ve 20 karakter ile sınırlıdır. Ekleme yapılırken “mahalleKoy” tablosundaki parametreler kullanılır. İlişkili değildir. Şahısın mahalle veya köy bilgisinin olduğu kayıttır.

acikAdres: : Boş geçilemez. Veri tipi text’dir. Açık adres genel, uzun bir metin olacağından bir değişken tipi seçilmiştir. Şahısın açık adres bilgisinin olduğu kayıttır.

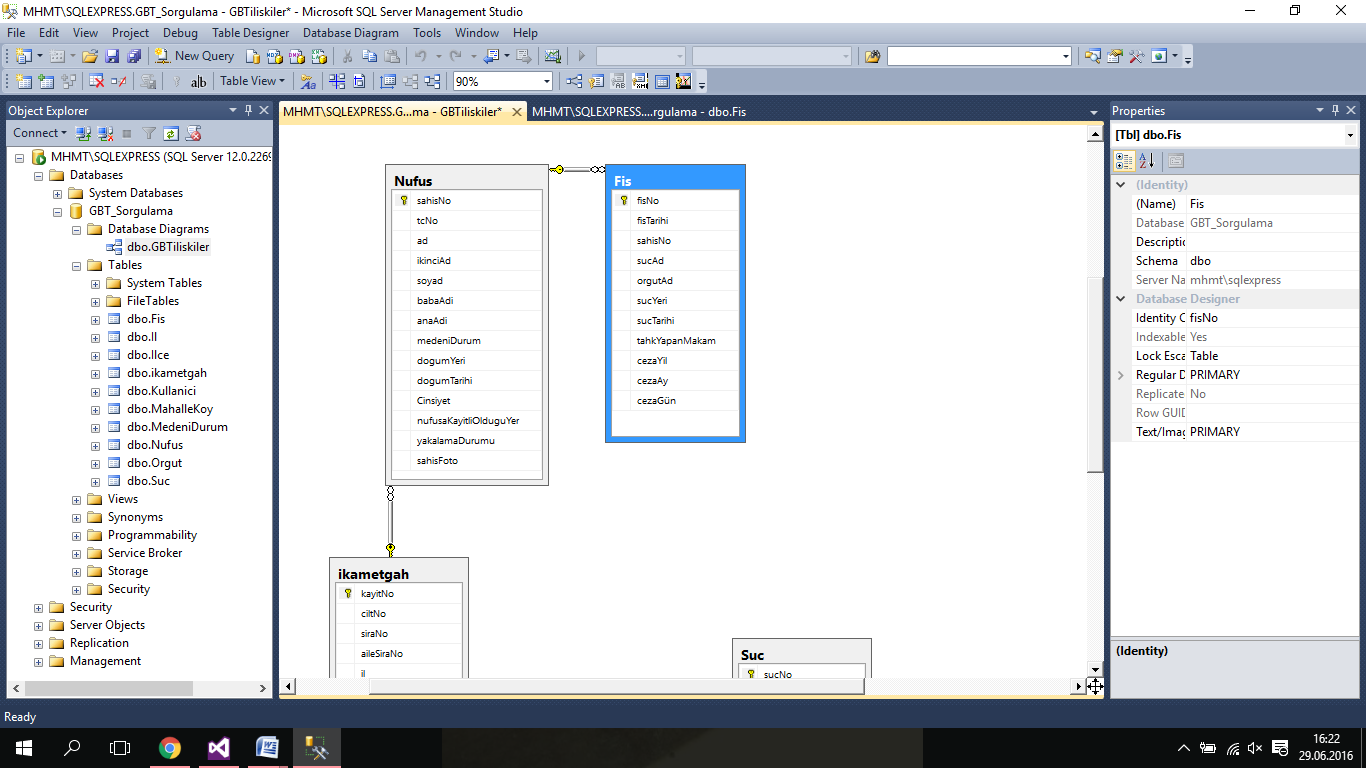
“Fis” Tablosu:



fisNo, fisTarihi, sahisNo, sucAd, orgutAd, sucYeri, sucTarihi, tahkYapanMakam, cezaYil, cezaAy, CezaGün isimli 11 adet alan tanımlanmıştır.

fisNo: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır. Şahıs arama parametrelerinden biridir.

fisTarihi: Boş geçilebilir. Veri tipi date’dir. Şahısın fişinin sisteme kayıt edildiği tarihtir. Sistemde gün/ay/yıl olarak kayıtlıdır. Otomasyondan tarih değerini date.now() ile otomatik alır.

sahisNo: Boş geçilebilir. Veri tipi int(sayı)’dir. İnt olmasının sebebi ise Nufus tablosu ile ilişkili olmasıdır. Fis.sahisNo=Nufus.sahisNo (Şekil 2)

sucAd: Boş geçilebilir. Veri tipi nvarchar ve 50 karakter ile sınırlıdır. Ekleme yapılırken “Suc” tablosundaki parametreler kullanılır. İlişkili değildir. Şahısın suç bilgisinin olduğu kayıttır.

sucAd: Boş geçilebilir. Veri tipi nvarchar ve 50 karakter ile sınırlıdır. Ekleme yapılırken “Suc” tablosundaki parametreler kullanılır. İlişkili değildir. Şahısın suç bilgisinin olduğu kayıttır.

Şekil 2

orgutAd: Boş geçilebilir. Veri tipi nvarchar ve 50 karakter ile sınırlıdır. Ekleme yapılırken “Orgut” tablosundaki parametreler kullanılır. İlişkili değildir. Şahısın Orgut bilgisinin olduğu kayıttır.

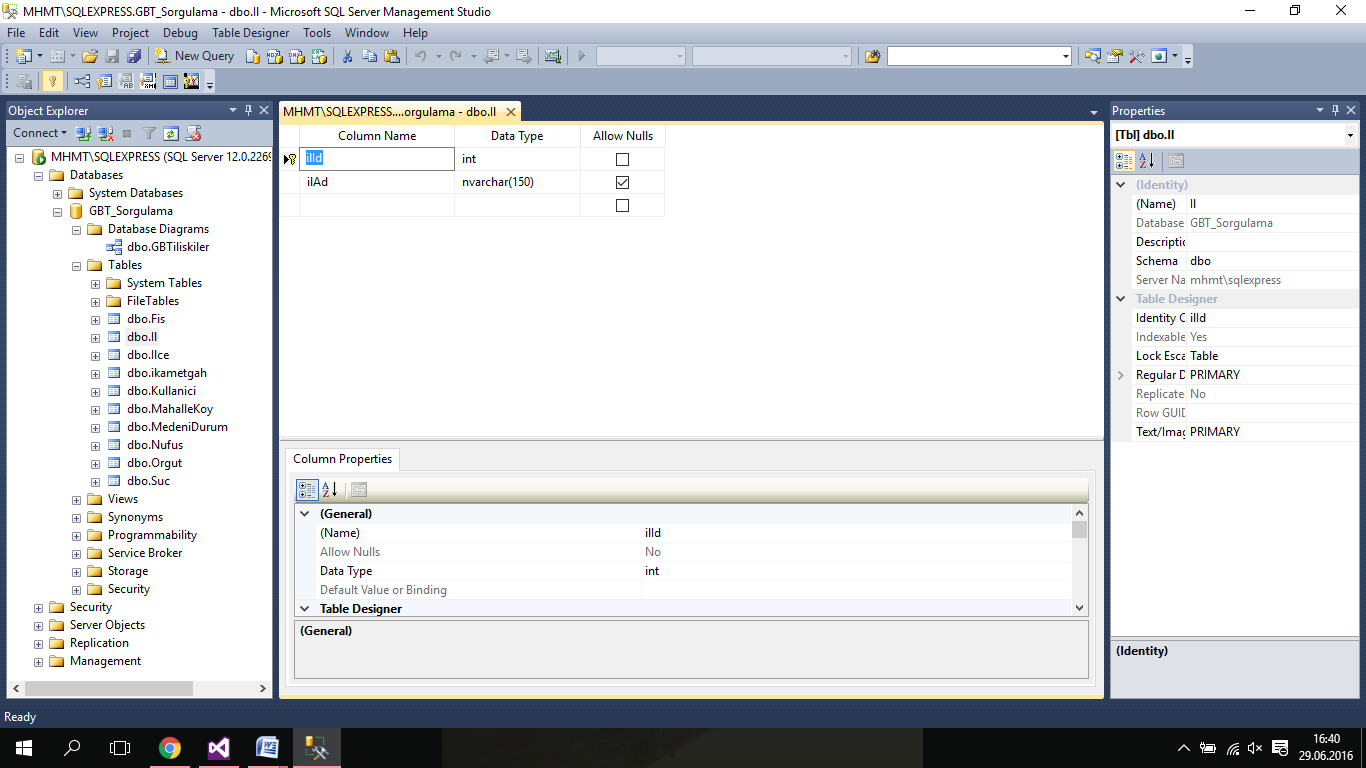
sucYeri: Boş geçilemez. Veri tipi nvarchar ve 50 karakter ile sınırlıdır. Şahısın suç işlediği yerin özel bilgisinin olduğu kayıttır.

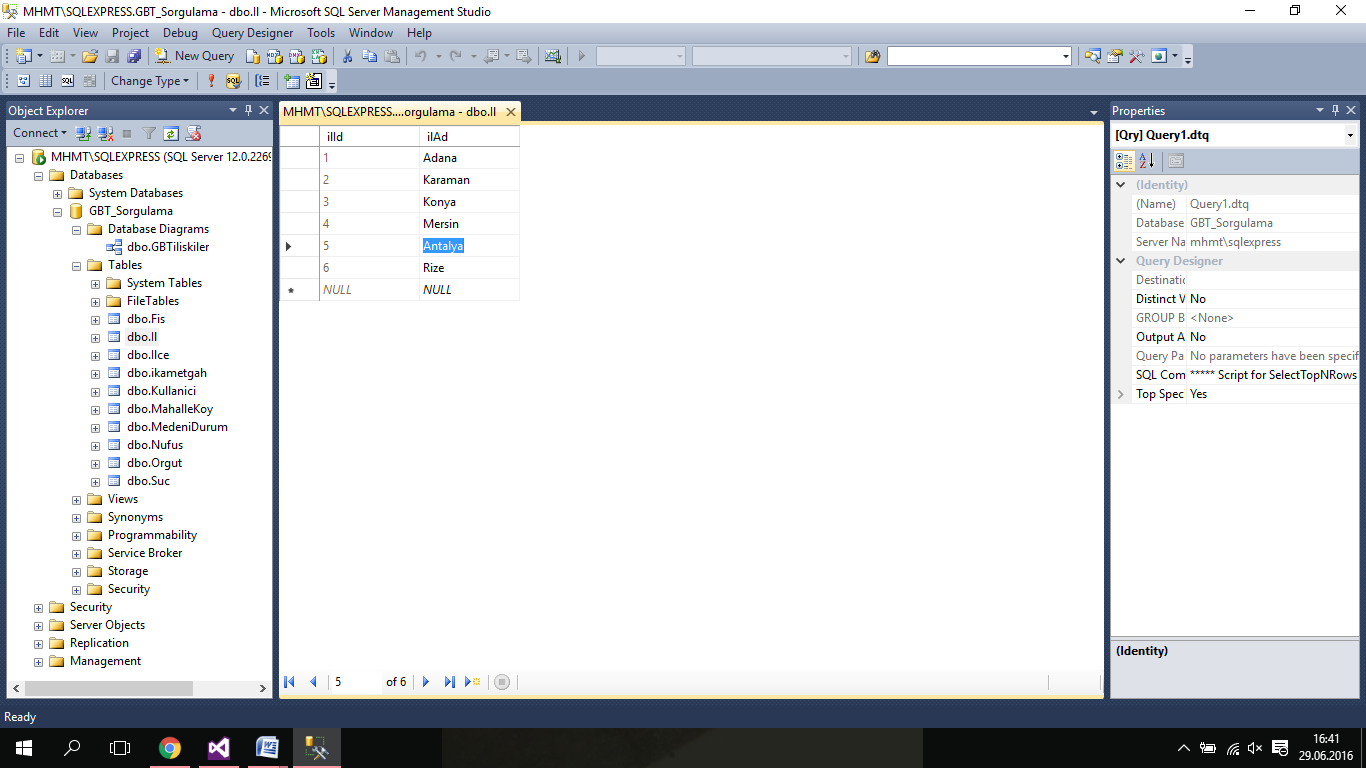
tahkYapanMakam: Boş geçilebilir. Veri tipi nvarchar ve 50 karakter ile sınırlıdır. Şahısın cezasını onaylayan makamın bilgisinin olduğu kayıttır.

cezaYil: Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır. Şahsın aldığı cezanın yıl bölümünü tutar.

cezaAy: Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır. Şahsın aldığı cezanın ay bölümünü tutar.

cezaGün: Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır. Şahsın aldığı cezanın gün bölümünü tutar.

“Il” Tablosu:

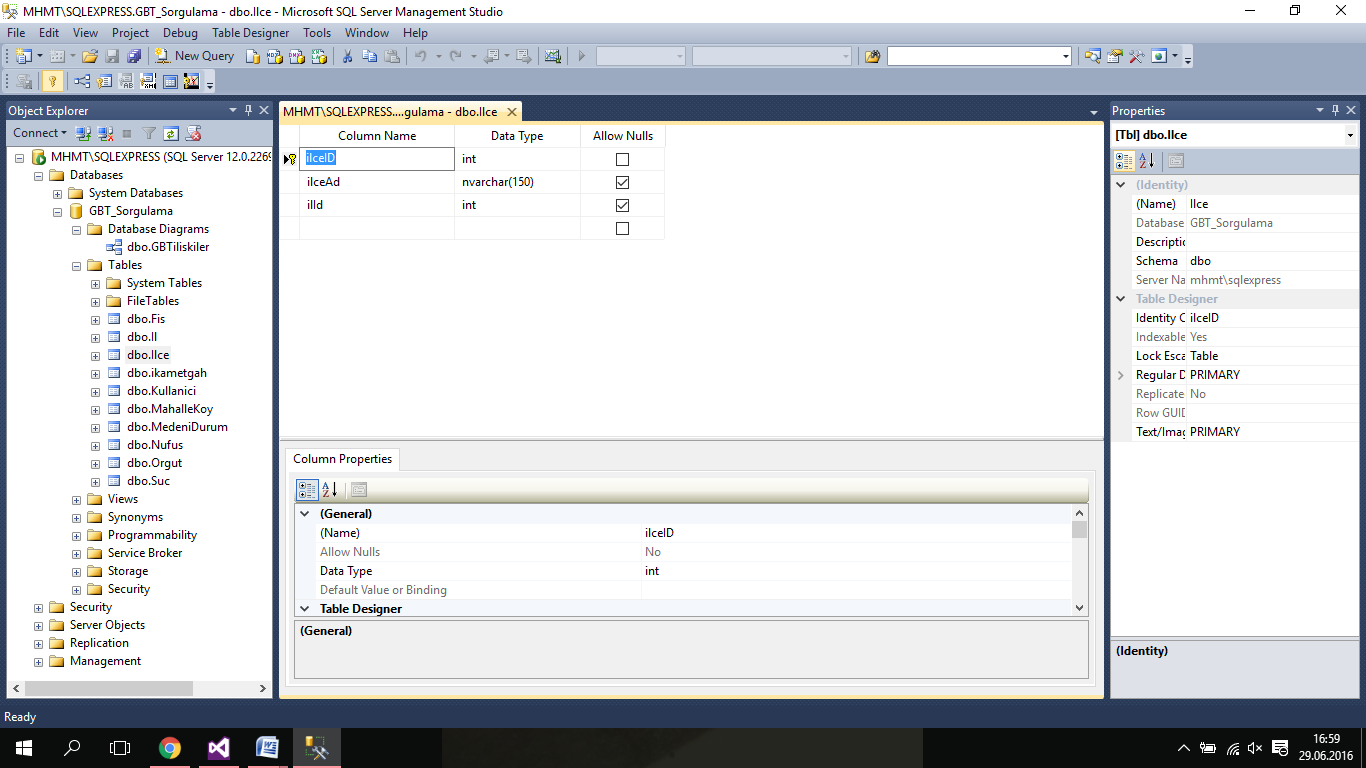
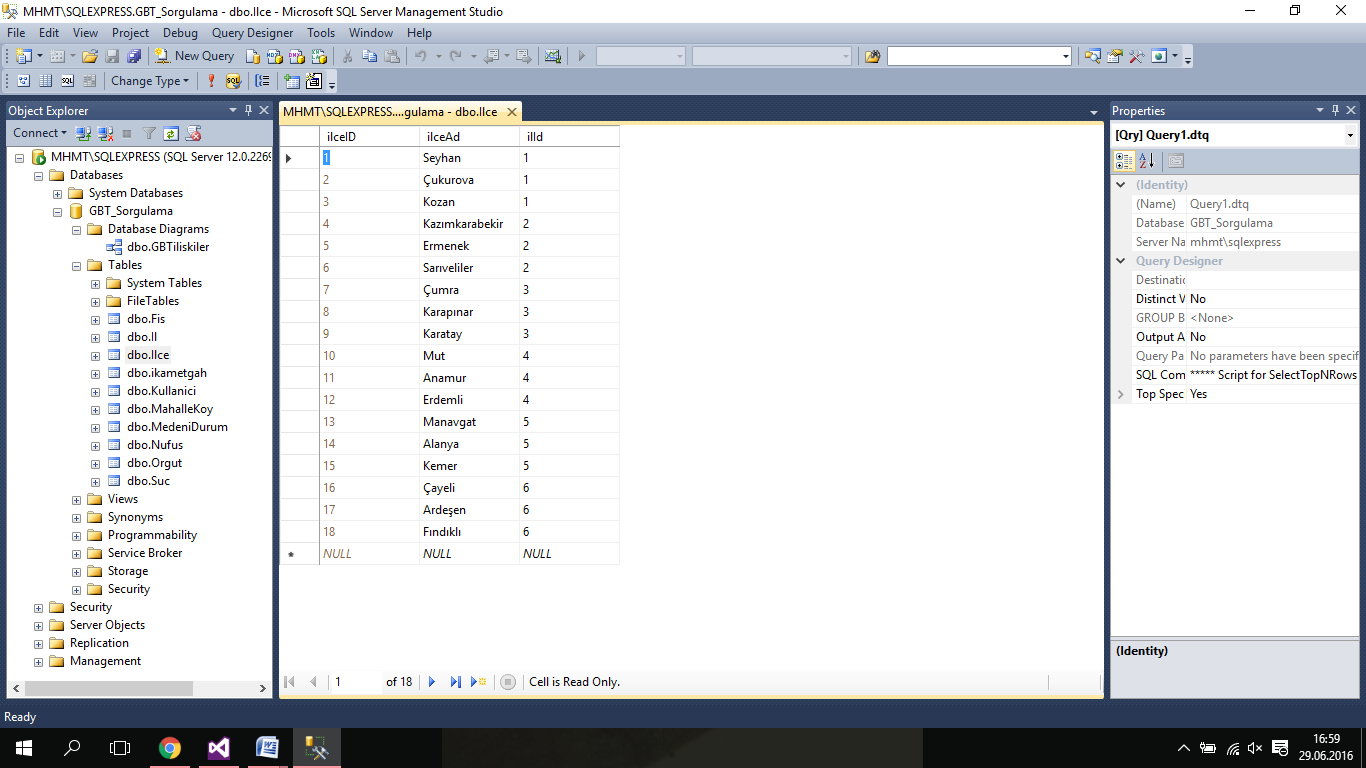


ilId ve ilAd isimli 2 adet alan bulunmaktadır.

ilId: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır.

ilAd: Boş geçilebilir. Veri tipi nvarchar ve 150 karakter ile sınırlıdır. Diğer tablolara eklenen iller bu tablodaki kayıtlardandır.

“Ilce” Tablosu:

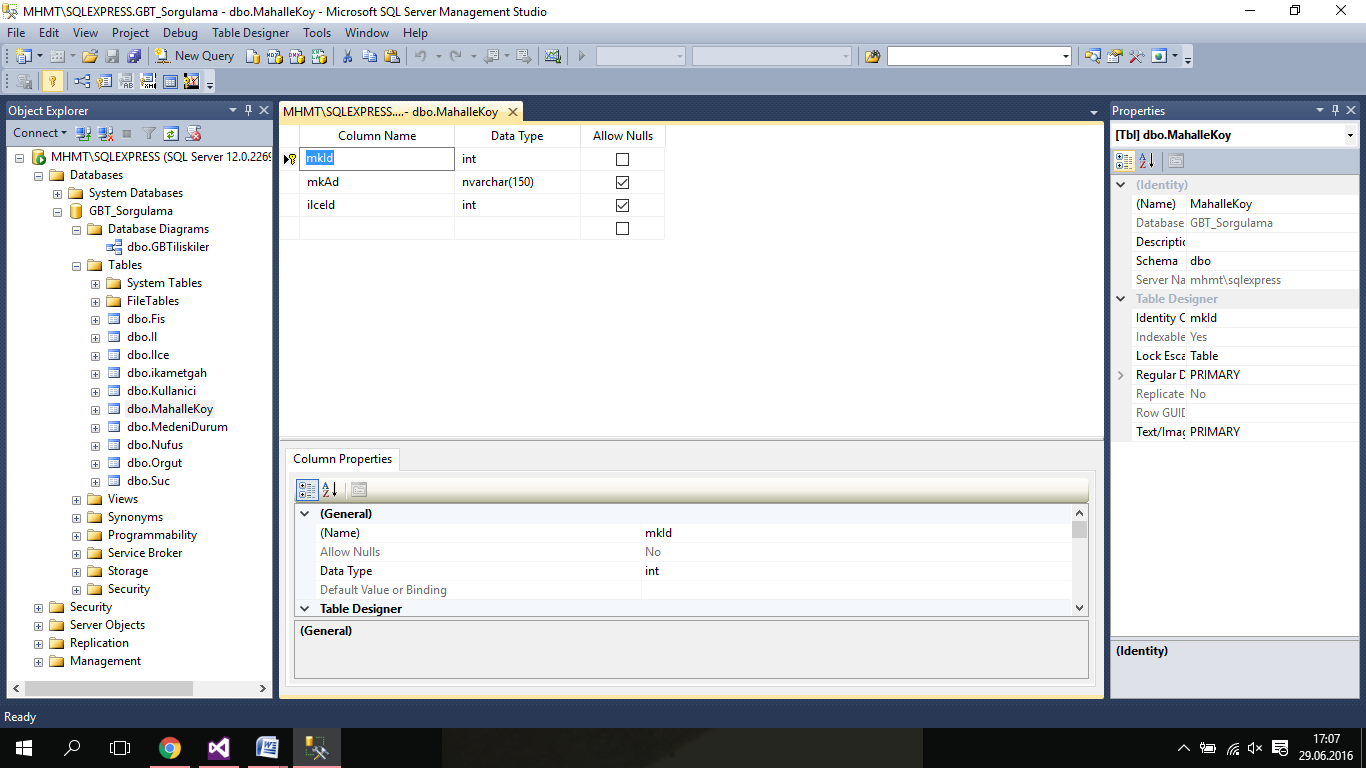
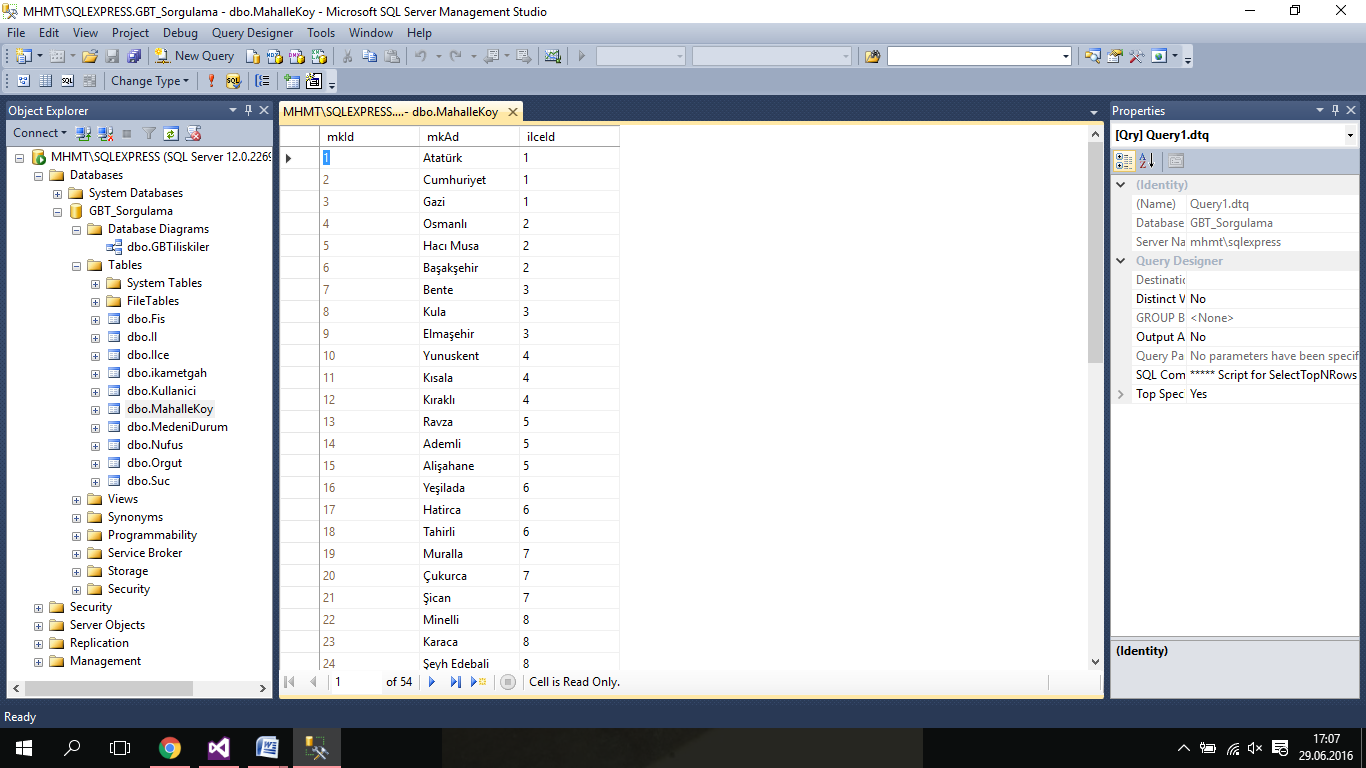


ilceID, ilceAd ve ilId isimli 3 adet alan bulunmaktadır.

ilceID: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır.

ilceAd: Boş geçilebilir. Veri tipi nvarchar ve 150 karakter ile sınırlıdır. Diğer tablolara eklenen ilceler bu tablodaki kayıtlardandır.

ilId: Boş geçilebilir. Veri tipi int(sayı)’dir. İnt olmasının sebebi ise Il tablosu ile ilişkili olmasıdır. Fiziksel ilişki yoktur.

“MahalleKoy” Tablosu:

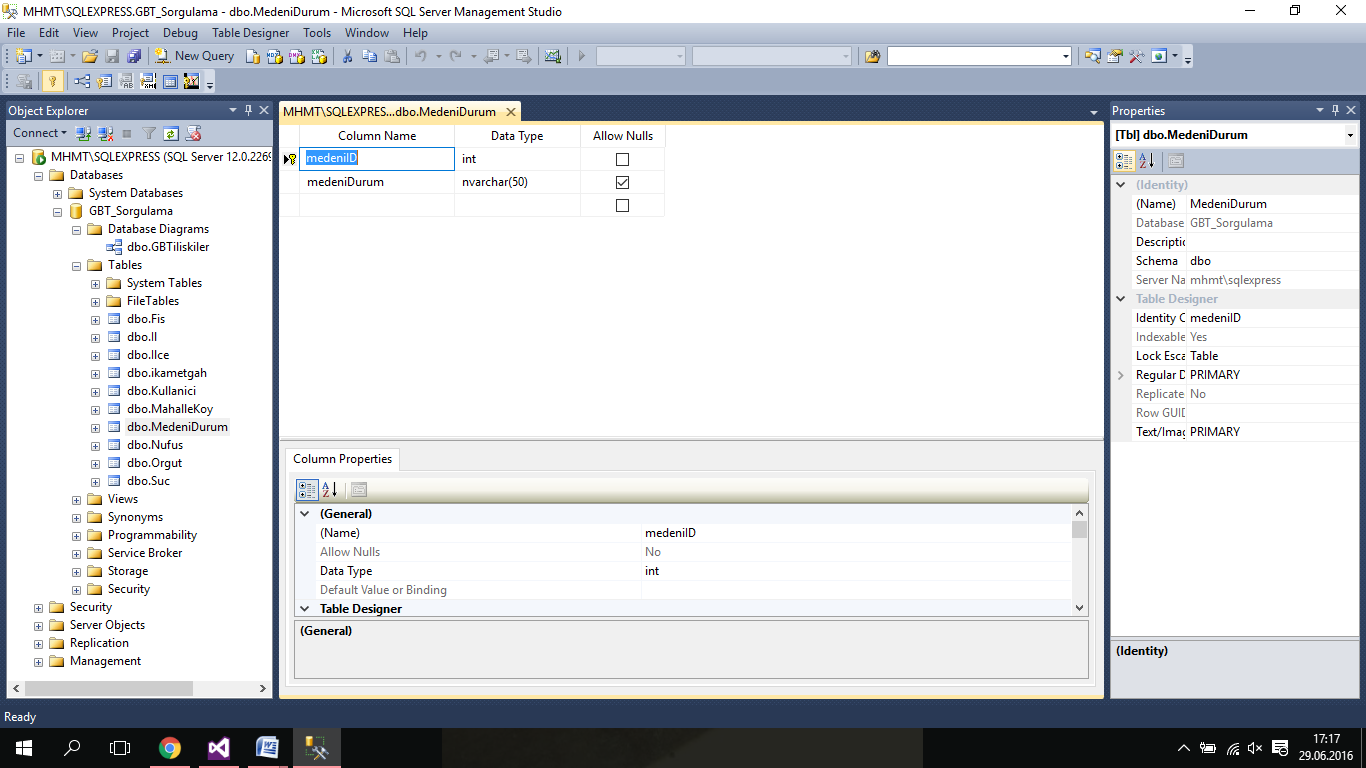
mkId, mkAd ve ilceId isimli 3 adet alan bulunmaktadır.

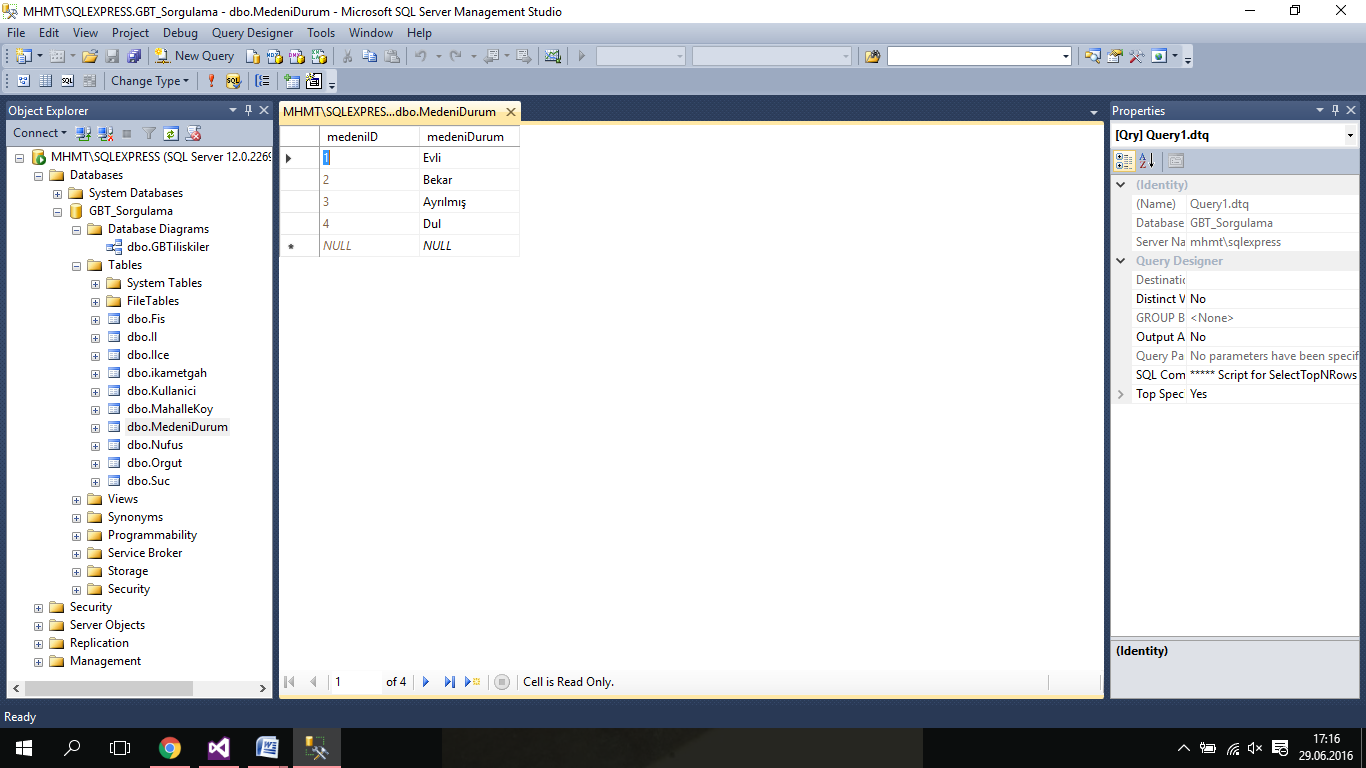
mkId: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır.

mkAd: Boş geçilebilir. Veri tipi nvarchar ve 150 karakter ile sınırlıdır. Diğer tablolara eklenen iller bu tablodaki kayıtlardandır.

ilceId: Boş geçilebilir. Veri tipi int(sayı)’dir. İnt olmasının sebebi ise Ilce tablosu ile ilişkili olmasıdır. Fiziksel ilişki yoktur.

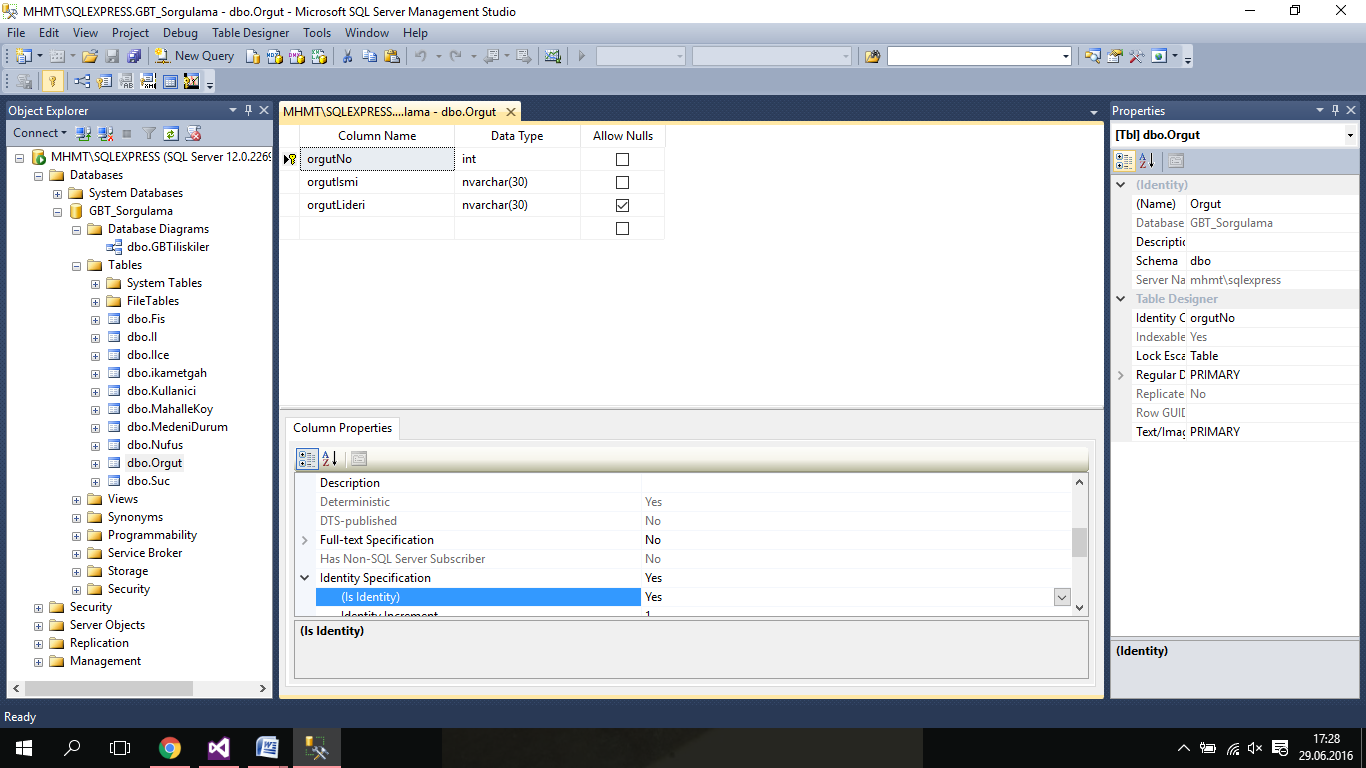
“MedeniDurum” Tablosu:

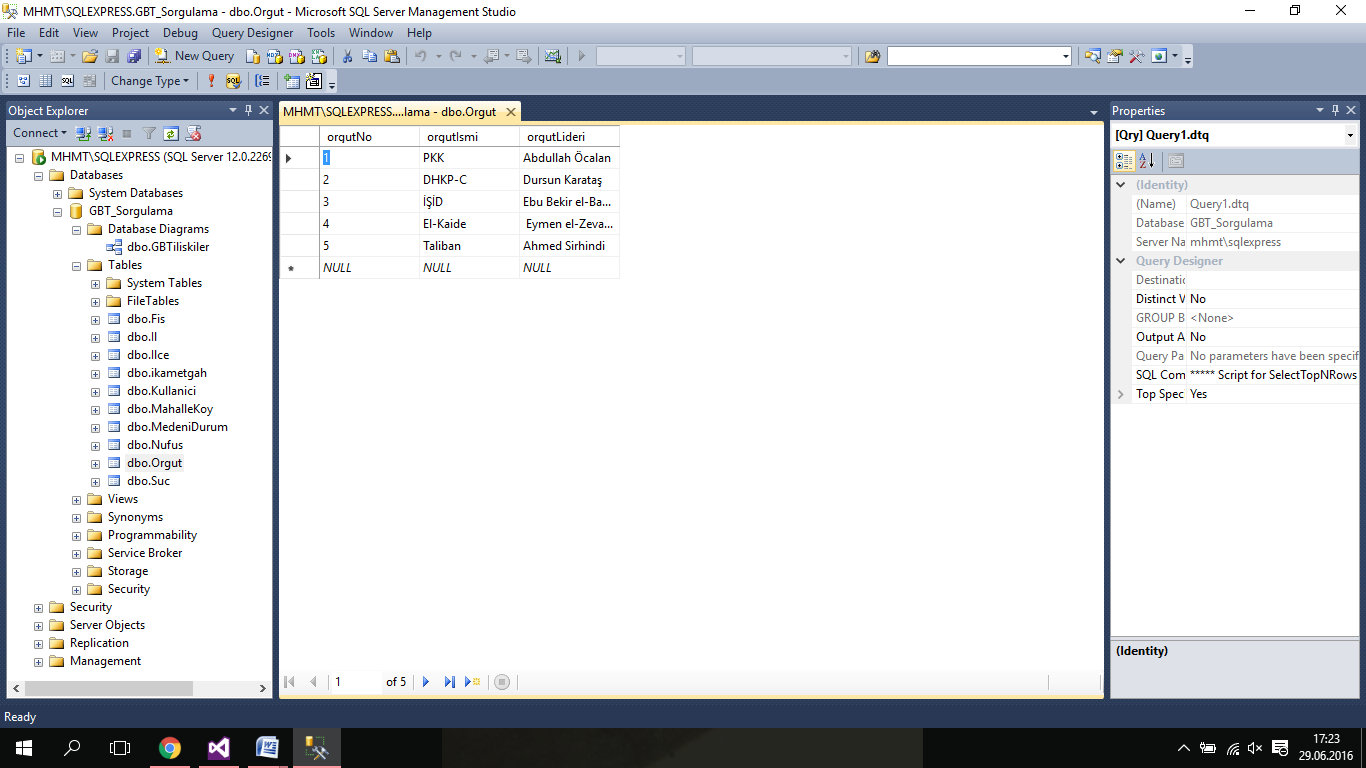
medeniID ve medeniDurum isimli 2 adet alan bulunmaktadır.

medeniID: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır

medeniDurum: Boş geçilebilir. Veri tipi nvarchar ve 50 karakter ile sınırlıdır. Nufus tablosuna eklenen medeni durum bu tablodaki kayıtlardandır.

“Orgut” Tablosu:

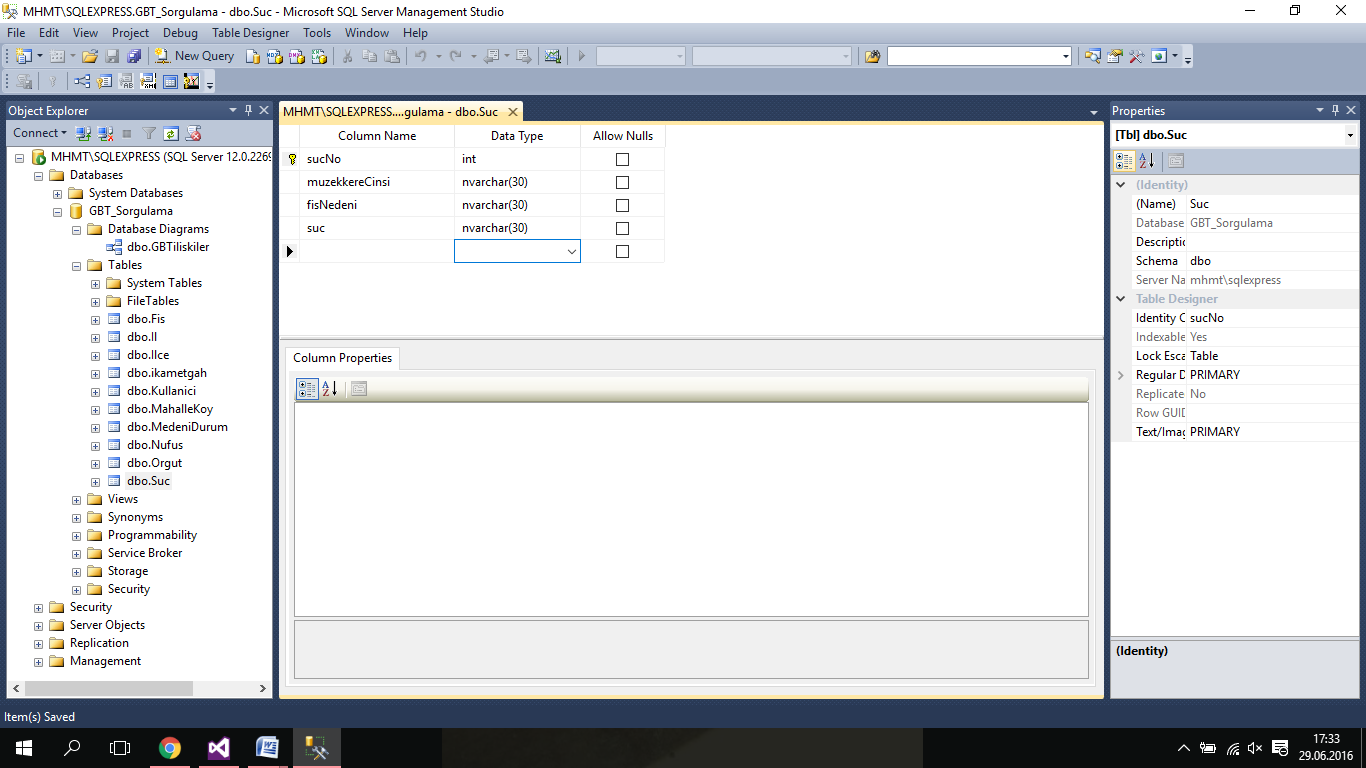
orgutNo, orgutIsmi ve orgutLideri isimli 3 adet alan bulunmaktadır.

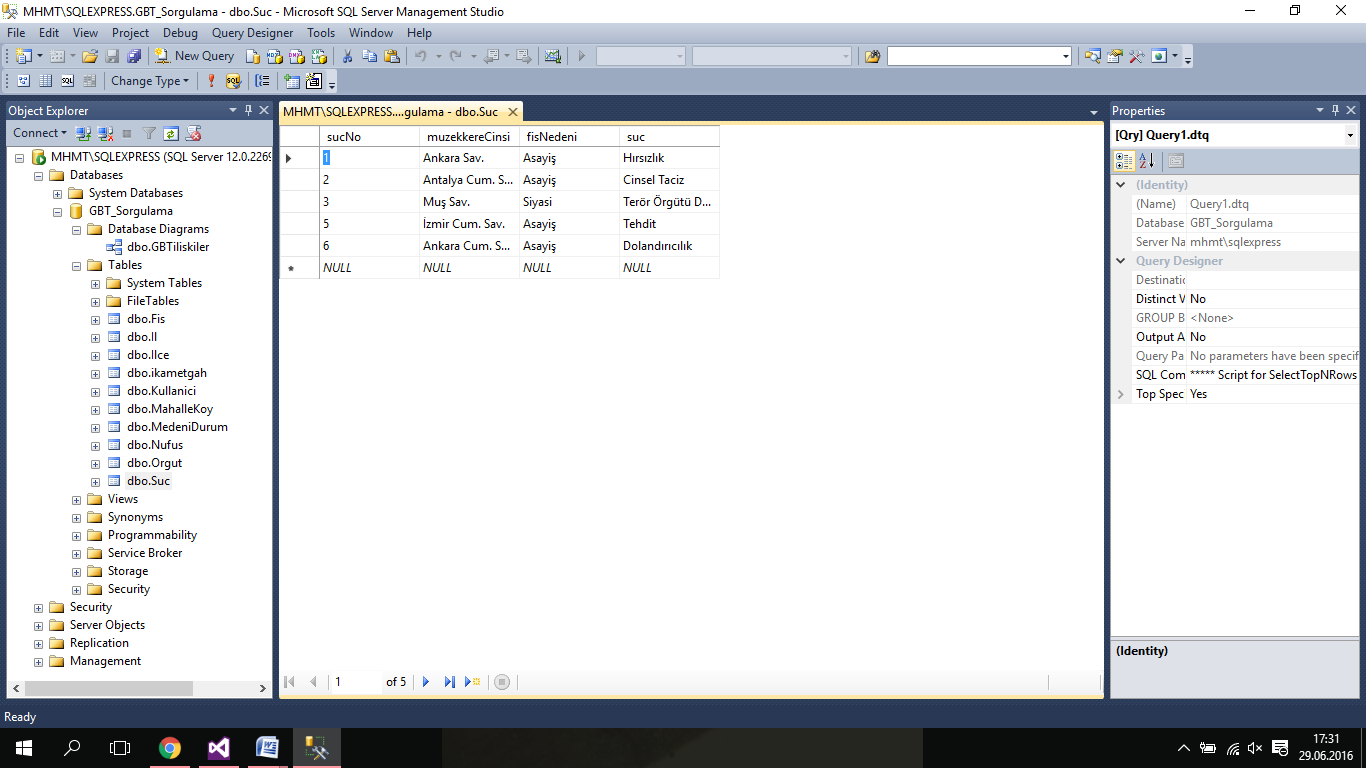
 orgutNo: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır.

orgutIsmi: Boş geçilemez. Veri tipi nvarchar ve 30 karakter ile sınırlıdır.

orgutLideri: Boş geçilemez. Veri tipi nvarchar ve 30 karakter ile sınırlıdır. (Bu alan otomasyonda görüntülenmez.)

“Suc” Tablosu:

sucNo, muzekkereCinsi, fisNedeni ve suc isimli 4 adet alan bulunmaktadır.

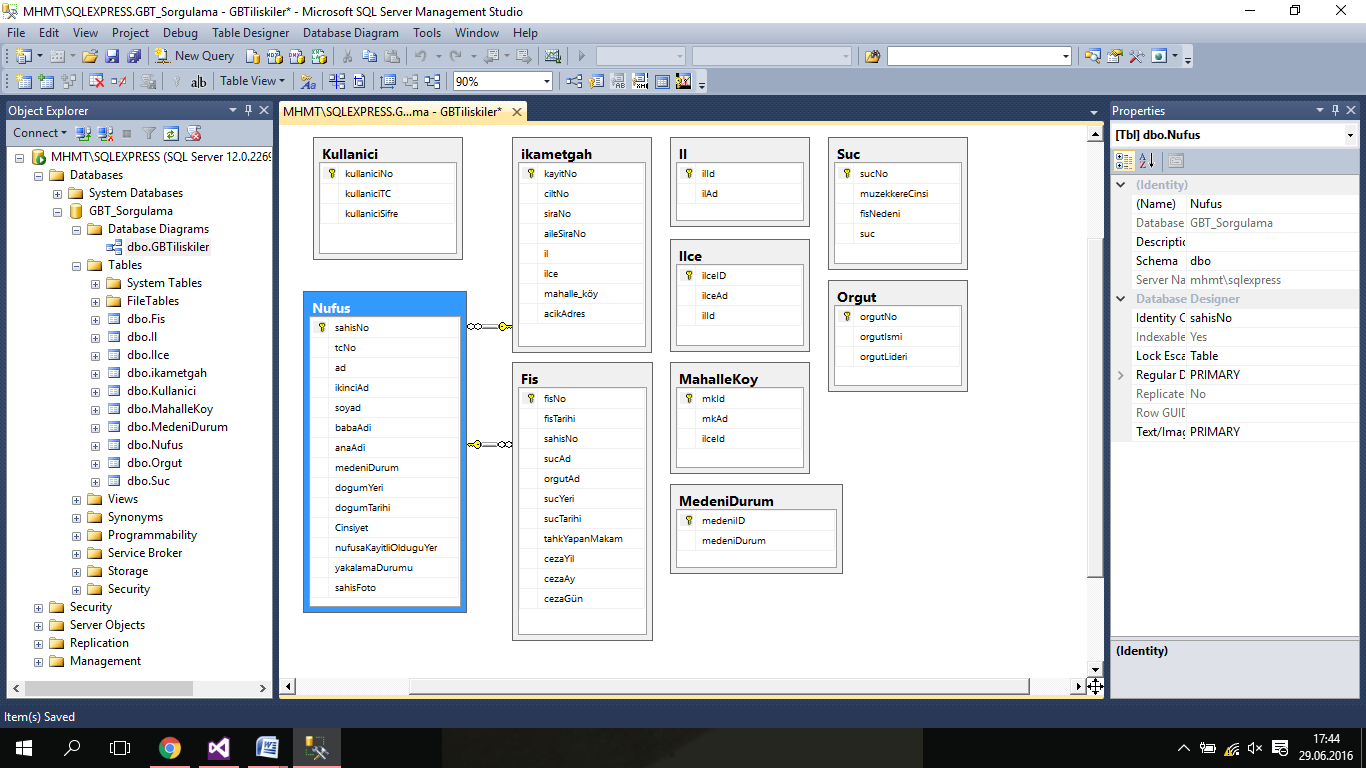
sucNo: Birincil anahtar olarak ayarlanmıştır. Otomatik artan sayıdır. Boş geçilemez. Veri tipi ise int(sayı) olarak ayarlanmıştır.

muzekkereCinsi: Boş geçilemez. Veri tipi nvarchar ve 30 karakter ile sınırlıdır.

fisNedeni: Boş geçilemez. Veri tipi nvarchar ve 30 karakter ile sınırlıdır.

suc: Boş geçilemez. Veri tipi nvarchar ve 30 karakter ile sınırlıdır.

**3.2 Veritabanı İlişkileri:**

****

1

2

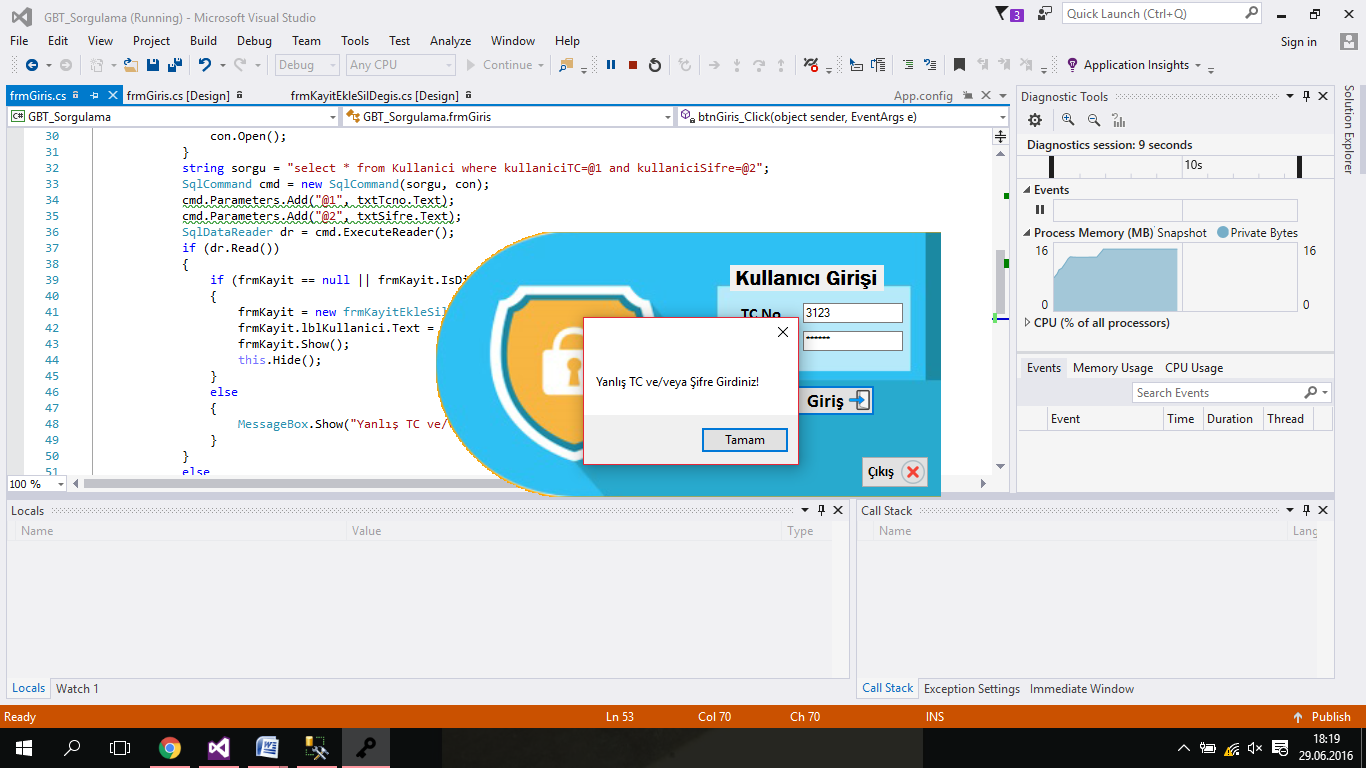
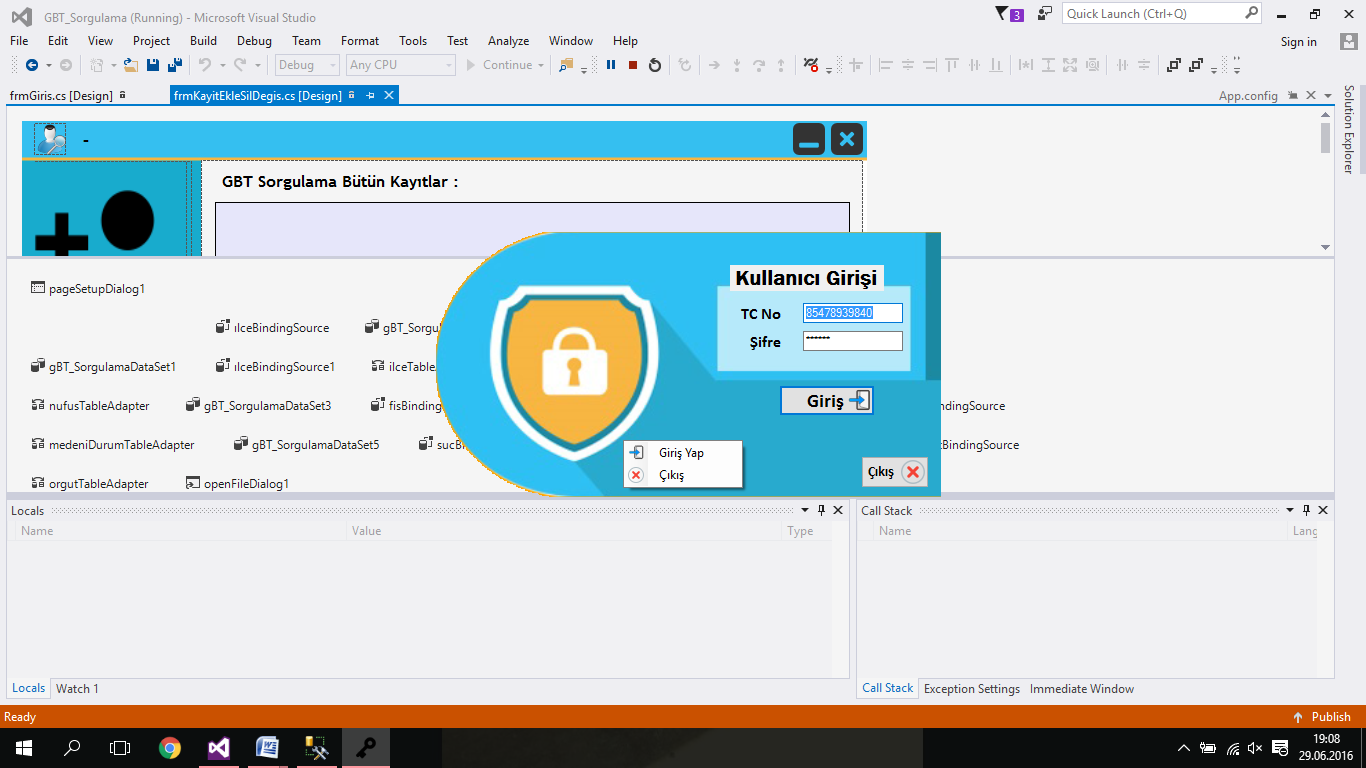
**4. Form (Arayüz) Tasarımı:**

2 adet form(frmGiris ve frmKayitEkleSilDegis) ve bu formların içerisinde birçok üst üste panel yapısı ile tasarlanmış bir arayüzdür.

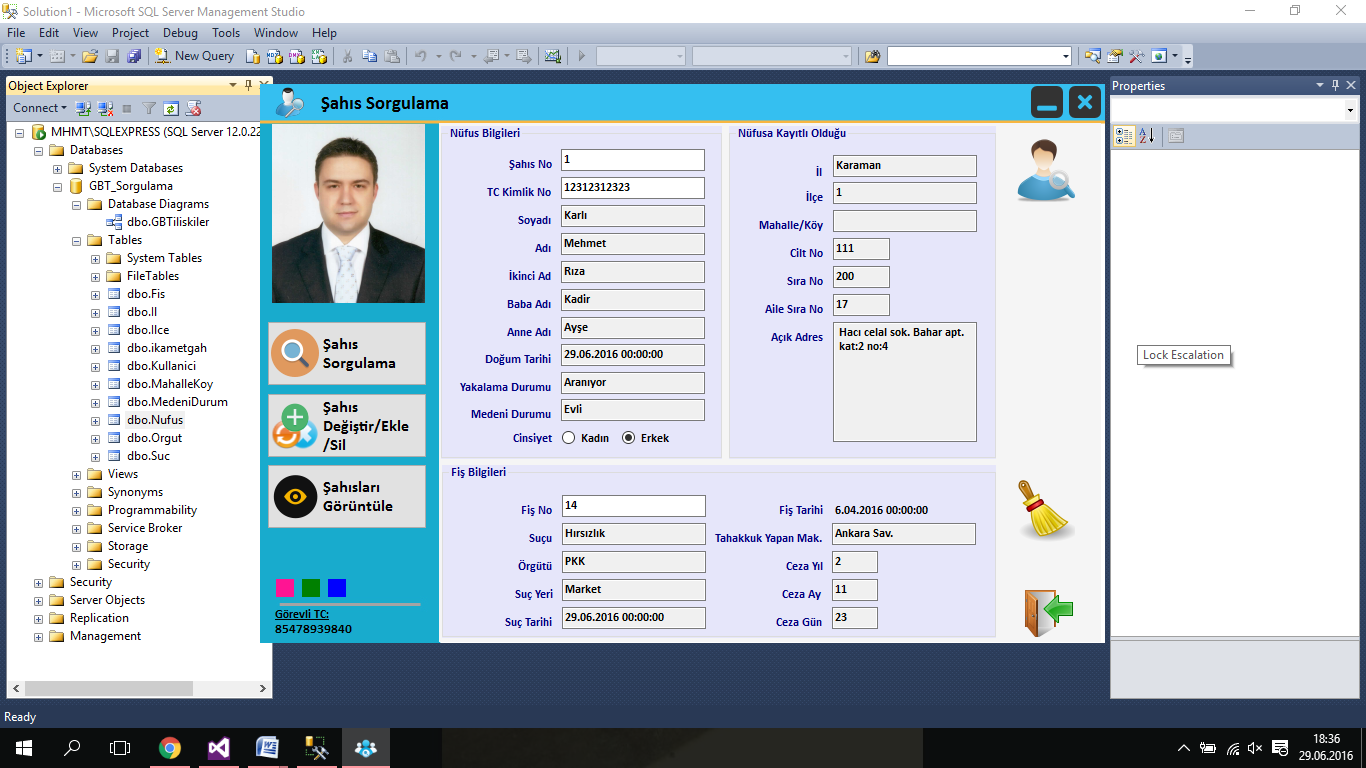
**4.1 Giriş Arayüzü Tasarımı:**

****

Resimde görüldüğü gibi oldukça şık ve bir o kadarda kullanışlı bir arayüz karşımıza geliyor. Öncelikle görsel olarak detaylara bakalım:

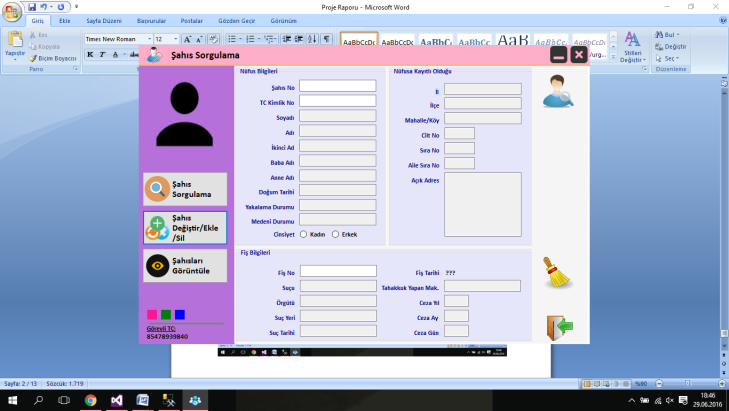
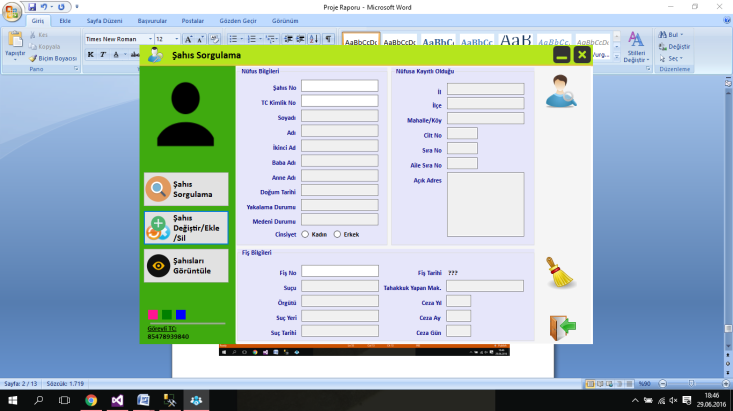
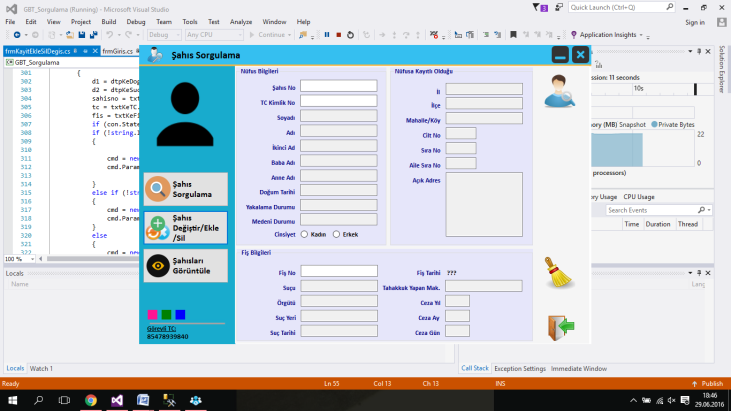
* Arkası transparan olan bir arkaplan resmi kullanıldı.
* Windowsun kullandığı pencere sistemi kapatılarak daha özgün bir tasarım uygulandı.
* Mavi ve turkuaz gibi uyumlu renk geçişleri ve turuncu gibi yenilikçi bir renk ile görsel arayüz canlandırıldı.
* Basit okuması kolay ve anlamayı kolaylaştırıcı ikon resimleri ile butonlar anlaşılır hale getirildi.
* Klavyeden giriş yapılan alanların isimleri yanlarında belirtilerek olası yanlış giriş girişimlerinin önüne geçildi.
* Şifre gibi bir özel bilginin yazıldığı bölüm gizlilik ve güvenlik nedeniyle “\*” karakteri ile gizlendi.
* TC yada Şifre yanlış girildiğinde ise böyle bir uyarı karşımıza çıkmaktadır.
* Giriş ekranının herhangi yerine tıklanarak ile de giriş-çıkış yapmak mümkündür.

**4.2 Şahıs Sorgulama Arayüzü Tasarımı:**

****

Resimde Şahıs Sorgulama arayüzünü görmekteyiz. Bu ekran program ilk açıldığında veya “Şahıs Sorgulama” butonuna tıklandığında gelen ekrandır.

* Bu arayüzde de windowsun pencere yapısının dışına çıkılıp birkaç buton yardımı ile çıkış, minimize özellikleri tanımlanmıştır. Özgün tasarım yapısı burada da uygulanmıştır.
* Sağ tarafta Arama butonu onun altında ise Temizleme ve Çıkış butonu yer almaktadır.
* Sağ en altta ise sisteme giriş yapan kullanıcının TC kimlik numarası yer almaktadır.
* Sağ altta tekrar “Pembe, Yeşil, Mavi” renkte butonlar ile farklı renk temaları uygulanmıştır.



* Arama butonu ile öncelik sırasına göre şahıs no, tc no veya fiş no ile arama yapıp kayıtlı bilgileri görüntüleyebilirsiniz.
* Temizle butonu ile de gelen kayıtların arayüzden temizlenmesi sağlanabilir.

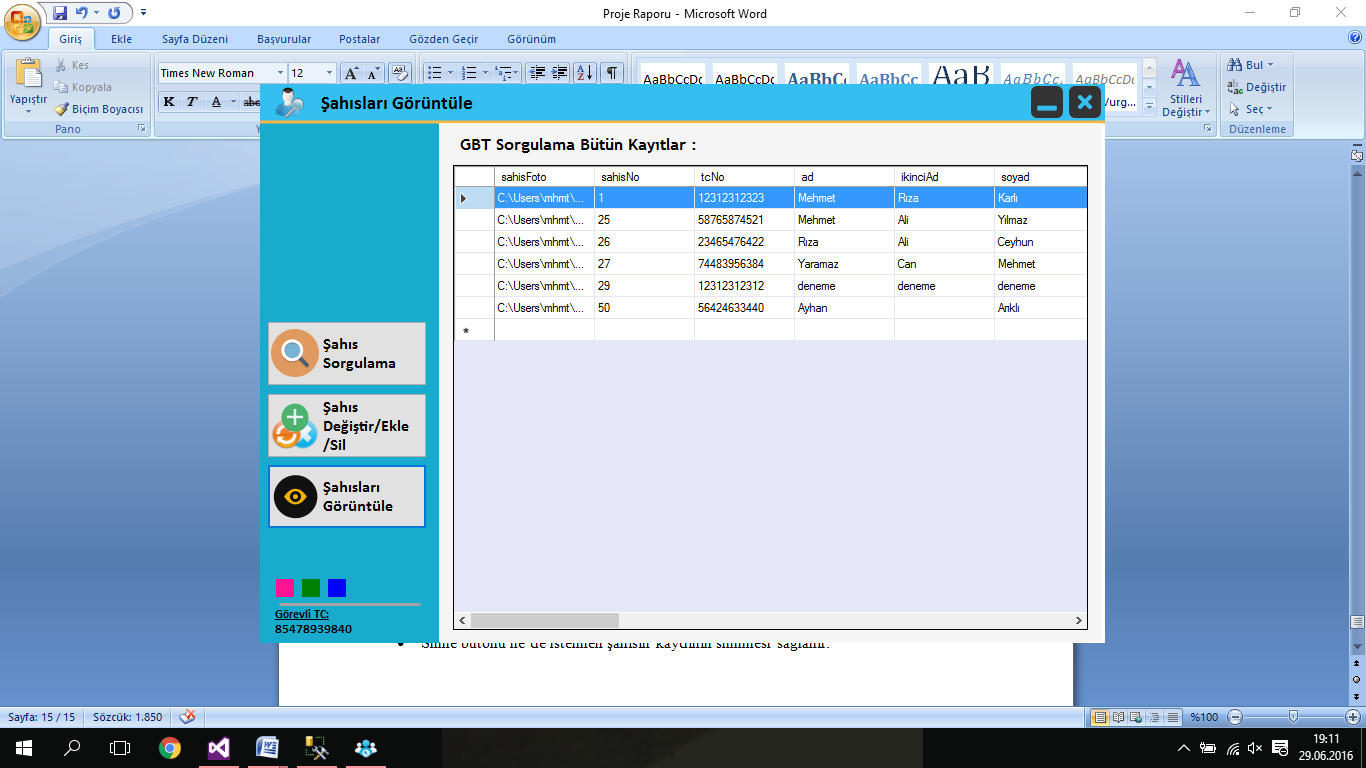
**4.3 Şahıs Değiştir/Ekle/Sil Arayüzü Tasarımı:**



Resimde Şahıs ekleme, düzenleme, silme arayüzünü görmekteyiz. Bu ekran “Şahıs Değiştir/Ekle/Sil” butonuna tıklandığında gelen ekrandır.

* Şahıs sorgulama ekranında olduğu gibi burada da sorgulama işlemi yapılabilmektedir.
* Farklı olarak ekleme, düzenleme, silme butonları mevcuttur.
* Ekle butonu ile uygun yerlere klavyeden girilenleri veritabanına kaydedebilirsiniz.
* Düzenle butonu ile aranıp gelen kaydın bilgileri üzerinde değişiklik yapabilirsiniz.
* Silme butonu ile de istenilen şahısın kaydının silinmesi sağlanır.

**4.4 Şahısları Görüntüle Arayüzü Tasarımı:**



Resimde Şahısları Görüntüleme arayüzünü görmekteyiz. Bu ekran “Şahısları Görüntüle” butonuna tıklandığında gelen ekrandır.

* Bu basit ekranda ise sistemdeki kayıtların bütün bilgileri görüntülenmektedir.

Ek;

* Ekranlar arası geçişlerde gösterilen kayıtlar temizlenmektedir.

**5. Kod Tasarımı**

**5.1 Giriş Ekranı (frmGiris)**

frmKayitEkleSilDegis frmKayit; //Giriş işleminden sonra açmak için tanımlandı.

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS; Initial Catalog=GBT\_Sorgulama; Integrated Security=TRUE;"); // Veritabanı bağlantı cümlecisi

Giriş Butonu(btnGiris): Öncelikle bağlantının kapalı olup olmadığına bakıyor, kapalıysa açıyor. Sonra işlem yapılacak sql sorgusunu string bir değişkene atıyoruz. Daha sonra sorguyu veritabanına gönderip geriye bir deger alınıyorsa tekrar bir koşul cümlesine giriyor. Burada ise form kapalı veya yoksa yenisini oluştur ve göstermesi isteniyor. Ek olarak girilen kullanıcı TC’sini diğer forma gönderiyor ve bu formu gizliyor.Eğer şifre yanlış ise mbox ile yanlış olduğuna dair ileti veriyor.

private void btnGiris\_Click(object sender, EventArgs e)

{

if (con.State == ConnectionState.Closed)

{

con.Open();

}

string sorgu = "select \* from Kullanici where kullaniciTC=@1 and kullaniciSifre=@2";

SqlCommand cmd = new SqlCommand(sorgu, con);

cmd.Parameters.Add("@1", txtTcno.Text);

cmd.Parameters.Add("@2", txtSifre.Text);

SqlDataReader dr = cmd.ExecuteReader();

if (dr.Read())

{

if (frmKayit == null || frmKayit.IsDisposed)

{

frmKayit = new frmKayitEkleSilDegis();

frmKayit.lblKullanici.Text = txtTcno.Text;

frmKayit.Show();

this.Hide();

}

else

{

MessageBox.Show("Yanlış TC ve/veya Şifre Girdiniz!");

}

}

else

{

MessageBox.Show("Yanlış TC ve/veya Şifre Girdiniz!");

con.Close();

}

}

Çıkış Butonu(btnCikis): Tıklama ile uygulamayı kapatır.

private void btnCikis\_Click(object sender, EventArgs e)

{

Application.Exit();

}

Ek Özellikler: Tc kimlik no girilen bölüme harf girişini engelleme ve form arkaplan rengini ayarlama.

public void txtTcno\_KeyPress(object sender, KeyPressEventArgs e)

{

e.Handled = !char.IsDigit(e.KeyChar) && !char.IsControl(e.KeyChar);

}

private void frmGiris\_Load(object sender, EventArgs e)

{

BackColor = Color.Orange;

TransparencyKey = Color.Orange;

FormBorderStyle = FormBorderStyle.None;

}

**5.2 Şahıs Değiştir/Ekle/Sil Ekranı (frmKayitEkleSil)**

SqlConnection con = new SqlConnection(@"Data Source=.\SQLEXPRESS; Initial Catalog=GBT\_Sorgulama; Integrated Security=TRUE;"); // Veritabanı bağlantı cümlecisi

SqlDataAdapter da;

SqlCommand cmd; //Gerekli global değişkenler

string sahisno, tc, fis, d1, d2;

DialogResult dia, dr;

Formun ilk açılışı(Form Load): Gerekli panelleri gösterir, gereksizleri gizler. Başlığı değiştirir.

private void frmKayitEkleSilDegis\_Load(object sender, EventArgs e)

{

pnlSahisSorgu.Show();

pnlKayitGor.Hide();

pnlSahisResim.Show();

pnlKayitEkle.Hide();

pnlSahisKeResim.Hide();

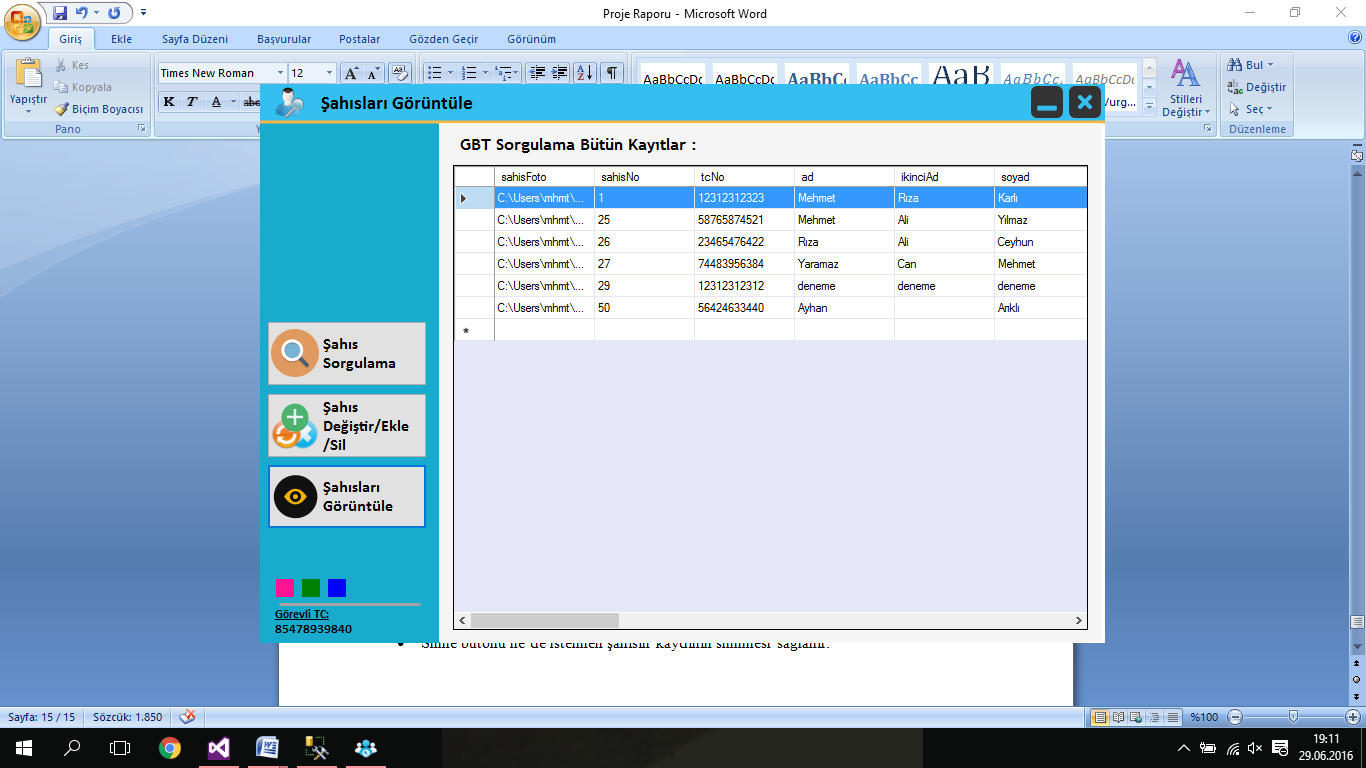
lblTitle.Text = "Şahıs Sorgulama";

}

Şahıs Değiştir/Ekle/Sil Butonu(btnKayitEkleSil): Temizleme methodunu çagırır. Gerekli panelleri gösterir, gereksizleri gizler. Picturebox resmini değiştirir. Başlığı değiştirir.

private void btnKayitEkleSil\_Click(object sender, EventArgs e)

{

 textclear(this);

pnlKayitGor.Hide();

pnlSahisKeResim.Show();

pnlSahisSorgu.Hide();

pnlSahisResim.Hide();

pnlKayitEkle.Show();

pBoxKeResim.Image = Image.FromFile(Application.StartupPath + "\\add\_user.png");

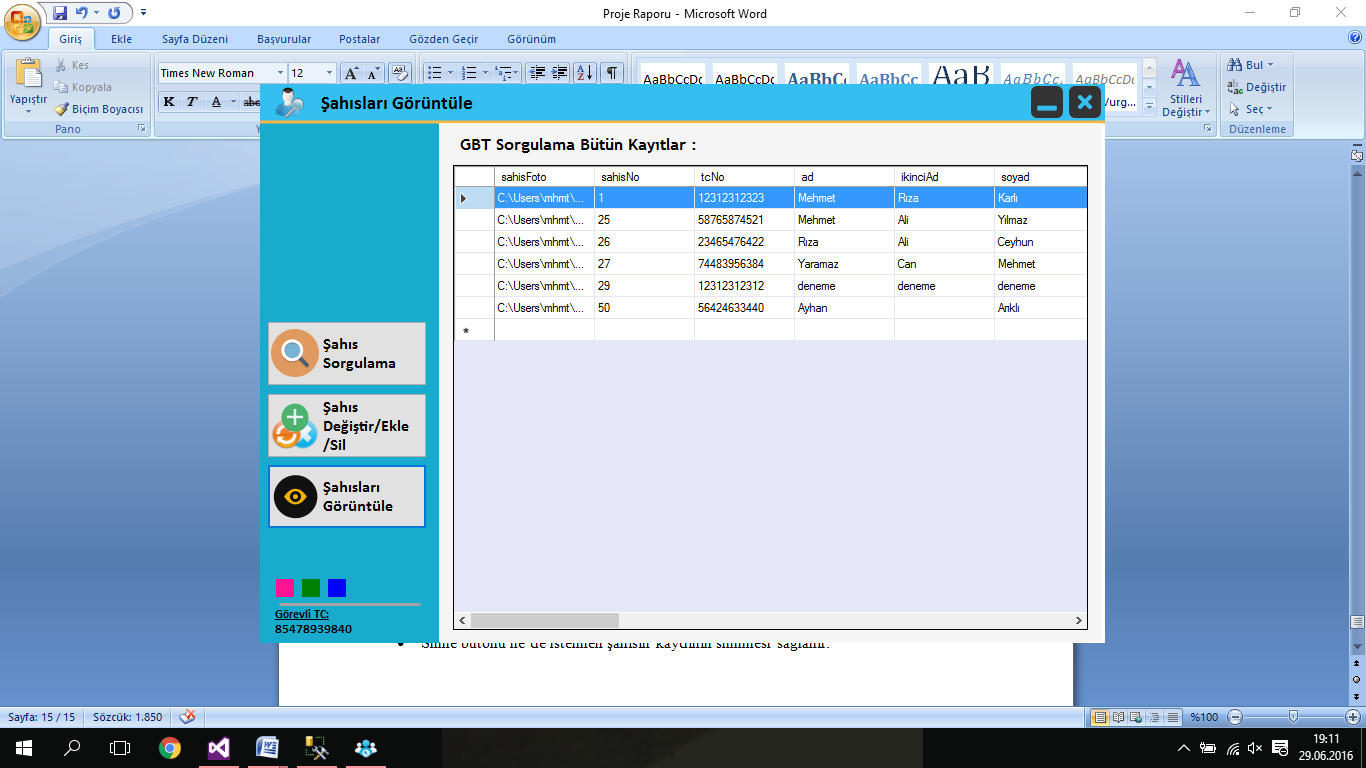
lblTitle.Text = "Şahıs Değiştir/Ekle/Sil";

}

Şahıs Sorgulama(btnSahis): Temizleme methodunu çagırır. Gerekli panelleri gösterir, gereksizleri gizler. Picturebox resmini değiştirir. Başlığı değiştirir.

private void btnSahis\_Click(object sender, EventArgs e)

{

 textclear(this);

pnlKayitGor.Hide();

pnlSahisResim.Show();

pnlKayitEkle.Hide();

pnlSahisKeResim.Hide();

pnlSahisSorgu.Show();

pBoxSahisResim.Image = Image.FromFile(Application.StartupPath + "\\user.png");

lblTitle.Text = "Şahıs Sorgulama";

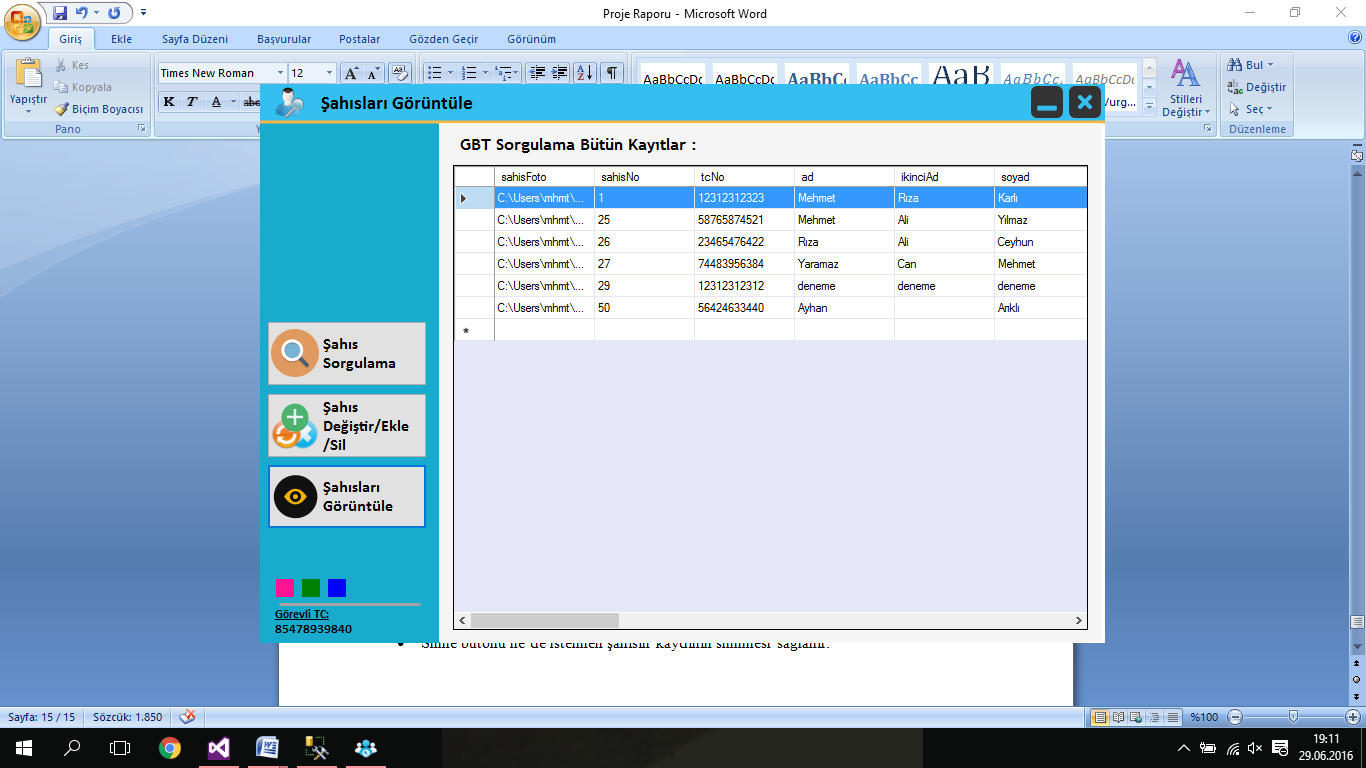
}

Şahısları Görüntüleme(btnSahis): Temizleme methodunu çagırır. Gerekli panelleri gösterir, panel içerisinde bütün kayıtları gösteren methodu çagırır. Gereksiz panelleri gizler. Picturebox resmini boşaltır. Başlığı değiştirir.

private void btnGörüntüle\_Click(object sender, EventArgs e)

{

textclear(this);

 tumKayit();

pnlSahisKeResim.Hide();

pnlSahisSorgu.Hide();

pnlSahisResim.Hide();

pnlKayitEkle.Hide();

pnlKayitGor.Show();

pBoxSahisResim.ImageLocation = null;

pBoxKeResim.ImageLocation = null;

lblTitle.Text = "Şahısları Görüntüle";

}

Pencere Çıkış Butonu(btnExit): Sağ üstteki kapatma butonu programı kapatır.

private void button1\_Click(object sender, EventArgs e)

{

Application.Exit();

}

Pencere Küçültme Butonu(btnMini): Sağ üstteki minimize butonu programı küçültür.

private void button3\_Click(object sender, EventArgs e)

{

this.WindowState = FormWindowState.Minimized;

}

Temizleme Butonu(btnSahTemizle) ve Temizleme(textclear) Methodu:

private void btnSahTemizle\_Click(object sender, EventArgs e)

{

textclear(this);

}

private void textclear(Control ctl)

{

foreach (Control item in ctl.Controls)

{

if (item is TextBox)

{

((TextBox)item).Clear();

}

if (item.Controls.Count > 0)

{

textclear(item);

}

}

}

Tema butonları(pboxPink, pboxGreen, pictureBox4): Tıklandığında arkaplan renkleri değişir.

private void pboxPink\_Click(object sender, EventArgs e)

{

this.BackgroundImage = Image.FromFile("Arkaplanlar\\arkaplanPembe.jpg");

}

private void pboxGreen\_Click(object sender, EventArgs e)

{

this.BackgroundImage = Image.FromFile("Arkaplanlar\\arkaplanYesil.jpg");

}

private void pictureBox4\_Click(object sender, EventArgs e)

{

this.BackgroundImage = Image.FromFile("Arkaplanlar\\arkaplanMavi.jpg");

}

Combobox doldurma işlemi(il(cbKeil) ve ilce(cbKeilce)): İstenilen id’ye göre ad’nı gösteren işlem.

private void cbKeil\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

da = new SqlDataAdapter("Select \* From Ilce where ilId=@p1", con);

da.SelectCommand.Parameters.Add("@p1", cbKeil.SelectedValue);

DataTable dt = new DataTable();

if (con.State == ConnectionState.Closed) { con.Open(); }

da.Fill(dt);

cbKeilce.DataSource = dt;

cbKeilce.DisplayMember = "ilceAd";

cbKeilce.ValueMember = "ilceID";

con.Close();

}

catch (Exception)

{

}

}

private void cbKeilce\_SelectedIndexChanged(object sender, EventArgs e)

{

try

{

da = new SqlDataAdapter("Select \* From MahalleKoy where ilceId=@p1", con);

da.SelectCommand.Parameters.Add("@p1", cbKeilce.SelectedValue);

DataTable dt = new DataTable();

if (con.State == ConnectionState.Closed) { con.Open(); }

da.Fill(dt);

cbKeMahalle.DataSource = dt;

cbKeMahalle.DisplayMember = "mkAd";

cbKeMahalle.ValueMember = "mkId";

con.Close();

}

catch (Exception)

{

}

}

Şahıs ekleme butonu(btnEkle): insert into sql komutu ile gerekli alanlara gerekli value değerlerini programdaki istenilen itemlerden alıp kayıt etme işlemi yapılmaktadır.Aynı anda 3 tabloya birden kayıt ekleme işlemi yapılmaktadır. İlişki kayıtlarda basit bir işlem ile ilişki sağlanmaktadır. Bu işlem ilk tabloya eklenen kaydın max(…) ve …[0] işlemleri ile son değerini alıp bir sonraki tablonun gerekli yerine atıyoruz.

private void btnKeEkle\_Click(object sender, EventArgs e)

{

if (con.State == ConnectionState.Closed) { con.Open(); }

SqlCommand cmd = new SqlCommand("insert into Nufus(tcNo,ad,ikinciAd,soyad,babaAdi,anaAdi,dogumYeri,dogumTarihi,Cinsiyet,medeniDurum,nufusaKayitliOlduguYer,yakalamaDurumu,sahisFoto) values (@p1,@p2,@p3,@p4,@p5,@p6,@p7,@p8,@p9,@p10,@p11,@p12,@p13)", con);

SqlCommand cmd1 = new SqlCommand("insert into ikametgah(ciltNo,siraNo,aileSiraNo,il,ilce,mahalle\_köy,acikAdres) values(@p1,@p2,@p3,@p4,@p5,@p6,@p7)", con);

SqlCommand cmd2 = new SqlCommand("insert into Fis(fisTarihi,sucAd,orgutAd,sucYeri,sucTarihi,tahkYapanMakam,cezaYil,cezaAy,cezaGün,sahisNo) values(@p1,@p2,@p3,@p4,@p5,@p6,@p7,@p8,@p9,@p10)", con);

try

{

cmd1.Parameters.AddWithValue("@p1", txtKeCilt.Text);

cmd1.Parameters.AddWithValue("@p2", txtKeSira.Text);

cmd1.Parameters.AddWithValue("@p3", txtKeAileSira.Text);

cmd1.Parameters.AddWithValue("@p4", cbKeil.Text);

cmd1.Parameters.AddWithValue("@p5", cbKeilce.Text);

cmd1.Parameters.AddWithValue("@p6", cbKeMahalle.Text);

cmd1.Parameters.AddWithValue("@p7", txtKeAcikAdres.Text);

try

{

cmd1.ExecuteNonQuery();

SqlCommand idgetir = new SqlCommand("select max(kayitNo) as id from ikametgah", con);

SqlDataReader oku = idgetir.ExecuteReader();

oku.Read();

string gelenmaxid = oku[0].ToString();

oku.Close();

cmd.Parameters.AddWithValue("@p1", txtKeTC.Text);

cmd.Parameters.AddWithValue("@p2", txtKEAd.Text);

cmd.Parameters.AddWithValue("@p3", txtKeİkinciAd.Text);

cmd.Parameters.AddWithValue("@p4", txtKeSoyad.Text);

cmd.Parameters.AddWithValue("@p5", txtKeBabaAd.Text);

cmd.Parameters.AddWithValue("@p6", txtKeAnneAd.Text);

cmd.Parameters.AddWithValue("@p7", cbKeil.Text);

cmd.Parameters.AddWithValue("@p8", dtpKeDogum.Value);

cmd.Parameters.AddWithValue("@p9", radioKeErkek.Checked ? "Erkek" : "Kadın");

cmd.Parameters.AddWithValue("@p10", cbKeMedeni.Text);

cmd.Parameters.AddWithValue("@p11", gelenmaxid);

cmd.Parameters.AddWithValue("@p12", cbKeDurumu.Text);

cmd.Parameters.AddWithValue("@p13", pBoxKeResim.Tag);

}

catch (Exception)

{

MessageBox.Show("Nüfusa kayıtlı olduğu yerin bilgilerini kontrol ediniz.!!");

}

try

{

cmd.ExecuteNonQuery();

SqlCommand idgetir2 = new SqlCommand("select max(sahisNo) as id from Nufus", con);

SqlDataReader oku2 = idgetir2.ExecuteReader();

oku2.Read();

string gelenmaxid2 = oku2[0].ToString();

oku2.Close();

cmd2.Parameters.AddWithValue("@p1", DateTime.Now);

cmd2.Parameters.AddWithValue("@p2", cbKeSuc.Text);

cmd2.Parameters.AddWithValue("@p3", cbKeOrgut.Text);

cmd2.Parameters.AddWithValue("@p4", txtKeSucYeri.Text);

cmd2.Parameters.AddWithValue("@p5", dtpKeSucTarih.Value);

cmd2.Parameters.AddWithValue("@p6", cbKeTahakkuk.Text);

cmd2.Parameters.AddWithValue("@p7", txtKeCezaYil.Text);

cmd2.Parameters.AddWithValue("@p8", txtKeCezaAy.Text);

cmd2.Parameters.AddWithValue("@p9", txtKeCezaGun.Text);

cmd2.Parameters.AddWithValue("@p10", gelenmaxid2);

}

catch (Exception)

{

MessageBox.Show("Nüfus bilgilerini kontrol ediniz.!!");

}

try

{

cmd2.ExecuteNonQuery();

}

catch (Exception)

{

MessageBox.Show("Fiş bilgilerini kontrol ediniz.!!");

}

dr = MessageBox.Show("Ekleme Yapılsın mı?", "Uyarı!", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dr == DialogResult.Yes)

MessageBox.Show("Ekleme işlemi başarıyla tamamlandı.");

else

MessageBox.Show("Ekleme işlemi iptal edildi.");

}

catch (Exception)

{

MessageBox.Show("Kayıt işlemi yapılırken hata oluştu.!!");

}

con.Close();

}

Şahıs arama butonu(btnAra): Gösterilen tarihin formatı ayarlandı. Gerekli değişkenler tutuldu. İf else yapısı ile farklı arama seçenekleri üretildi.Çağırılan kayıtlarla gerekli itemler dolduruldu. (Butondan 2 tane olduğu için 2.si burada yoktur. Sadece kullanılan öğeler farklıdır.Kod yapısı aynıdır.)

private void button1\_Click\_1(object sender, EventArgs e)

{

d1 = dtpKeDogum.Value.ToString("yyyy - MM - dd");

d2 = dtpKeSucTarih.Value.ToString("yyyy - MM - dd");

sahisno = txtKeSah.Text;

tc = txtKeTC.Text;

fis = txtKeFis.Text;

if (con.State == ConnectionState.Closed) { con.Open(); }

if (!string.IsNullOrEmpty(txtKeSah.Text))

{

cmd = new SqlCommand("select \* from Nufus,ikametgah,fis where Nufus.sahisNo=@id and Nufus.nufusaKayitliOlduguYer=ikametgah.kayitNo and fis.sahisNo=Nufus.sahisNo", con);

cmd.Parameters.AddWithValue("@id", sahisno);

}

else if (!string.IsNullOrEmpty(txtKeTC.Text))

{

cmd = new SqlCommand("select \* from Nufus,ikametgah,fis where Nufus.tcNo=@id2 and Nufus.nufusaKayitliOlduguYer=ikametgah.kayitNo and fis.sahisNo=Nufus.sahisNo", con);

cmd.Parameters.AddWithValue("@id2", tc);

}

else

{

cmd = new SqlCommand("select \* from Nufus,ikametgah,fis where Fis.fisNo=@id3 and Nufus.nufusaKayitliOlduguYer=ikametgah.kayitNo and fis.sahisNo=Nufus.sahisNo", con);

cmd.Parameters.AddWithValue("@id3", fis);

}

SqlDataReader drs = cmd.ExecuteReader();

if (drs.Read())

{

txtSahSahisNo.Text = drs["sahisNo"].ToString();

txtKEAd.Text = drs["ad"].ToString();

txtKeTC.Text = drs["tcNo"].ToString();

txtKeİkinciAd.Text = drs["ikinciAd"].ToString();

txtKeSoyad.Text = drs["soyad"].ToString();

txtKeBabaAd.Text = drs["babaAdi"].ToString();

txtKeAnneAd.Text = drs["anaAdi"].ToString();

d1 = drs["dogumTarihi"].ToString();

radioKeErkek.Checked = drs["Cinsiyet"].ToString() == "Erkek" ? true : false;

cbKeDurumu.Text = drs["yakalamaDurumu"].ToString();

txtSahMedeni.Text = drs["medeniDurum"].ToString();

pBoxKeResim.ImageLocation = drs["sahisFoto"].ToString();

pBoxSahisResim.Tag = drs["sahisFoto"].ToString();

cbKeil.Text = drs["il"].ToString();

cbKeilce.Text = drs["ilce"].ToString();

cbKeMahalle.Text = drs["mahalle\_köy"].ToString();

txtKeCilt.Text = drs["ciltNo"].ToString();

txtKeSira.Text = drs["siraNo"].ToString();

txtKeAileSira.Text = drs["aileSiraNo"].ToString();

txtKeAcikAdres.Text = drs["acikAdres"].ToString();

label59.Tag = drs["kayitNo"].ToString();

txtKeFis.Text = drs["fisNo"].ToString();

lblFisTarihi.Text = drs["fisTarihi"].ToString();

cbKeSuc.SelectedText = drs["sucAd"].ToString();

cbKeOrgut.SelectedText = drs["orgutAd"].ToString();

d2 = drs["sucTarihi"].ToString();

txtKeSucYeri.SelectedText = drs["sucYeri"].ToString();

cbKeTahakkuk.SelectedText = drs["tahkYapanMakam"].ToString();

txtKeCezaYil.Text = drs["cezaYil"].ToString();

txtKeCezaAy.Text = drs["cezaAy"].ToString();

txtKeCezaGun.Text = drs["cezaGün"].ToString();

}

else

{

MessageBox.Show("Sistemde böyle bir şahıs yoktur.");

}

con.Close();

}

Şahıs silme butonu(btnKeSil):3 adet if else yapısı ile hangi parametrenin baz alınıp ona göre hangi ilişkili kayıtın silinecigine bakılıyor. Veri tabanı içerinde kayıtlar cascade olarak birbirine ilişkili olduğu için biri silinirken diğer hepsi siliniyor.

private void btnKeSil\_Click(object sender, EventArgs e)

{

sahisno = txtKeSah.Text;

tc = txtKeTC.Text;

fis = txtKeFis.Text;

try

{

if (con.State == ConnectionState.Closed) { con.Open(); }

if (!string.IsNullOrEmpty(txtKeSah.Text))

{

cmd = new SqlCommand("select \* from Nufus,ikametgah,fis where Nufus.sahisNo=@id and Nufus.nufusaKayitliOlduguYer=ikametgah.kayitNo and fis.sahisNo=Nufus.sahisNo", con);

cmd.Parameters.AddWithValue("@id", sahisno);

}

else if (!string.IsNullOrEmpty(txtKeTC.Text))

{

cmd = new SqlCommand("select \* from Nufus,ikametgah,fis where Nufus.tcNo=@id2 and Nufus.nufusaKayitliOlduguYer=ikametgah.kayitNo and fis.sahisNo=Nufus.sahisNo", con);

cmd.Parameters.AddWithValue("@id2", tc);

}

else

{

cmd = new SqlCommand("select \* from Nufus,ikametgah,fis where Fis.fisNo=@id3 and Nufus.nufusaKayitliOlduguYer=ikametgah.kayitNo and fis.sahisNo=Nufus.sahisNo", con);

cmd.Parameters.AddWithValue("@id3", fis);

}

SqlDataReader drsil = cmd.ExecuteReader();

if (drsil.Read())

{

dia = MessageBox.Show(txtKeSah.Text + " şahıs numaralı kayıt silinsin mi?", "Uyarı!", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

drsil.Close();

if (dia == DialogResult.Yes)

{

string sil = "DELETE from Nufus where sahisNo=@p1";

SqlCommand silkmt = new SqlCommand(sil, con);

silkmt.Parameters.AddWithValue("@p1", txtKeSah.Text);

silkmt.ExecuteNonQuery();

MessageBox.Show(txtKeSah.Text + " 'şahıs nolu kayıt silindi...");

textclear(this);

}

else

{

MessageBox.Show("Kayıt silme iptal edildi.");

}

}

else

{

MessageBox.Show("Böyle bir kayıt bulunmamaktadır.");

}

}

catch (Exception)

{

MessageBox.Show("Böyle bir kayıt bulunmamaktadır.");

}

con.Close();

}

Düzenleme butonu(btnKeDegistir): update set sql komutu ile değiştirilecek alanlar parametreleri ile tanımlanır ve addwithvalue kullanarak yeni değer güncellenir. Aynı ekleme işleminde olduğu gibi tablolara ekleme yapılırken ilk tablonun ilişkilendirilmiş kaydı bir yerde tutulur(label.tag vb) buradan tekrar çağırarak ilişkili kayıt ekleme tamamlanır.

private void btnKeDegistir\_Click(object sender, EventArgs e)

{

if (con.State == ConnectionState.Closed) { con.Open(); }

SqlCommand cmd = new SqlCommand("update Nufus set tcNo=@p1,ad=@p2,ikinciAd=@p3,soyad=@p4,babaAdi=@p5,anaAdi=@p6,medeniDurum=@p7,dogumTarihi=@p9,Cinsiyet=@p10,yakalamaDurumu=@p11,sahisFoto=@p12 where sahisNo=@sNo", con);

SqlCommand cmd1 = new SqlCommand("update ikametgah set ciltNo=@p1,siraNo=@p2,aileSiraNo=@p3,il=@p4,ilce=@p5,mahalle\_köy=@p6,acikAdres=@p7 where kayitNo=@kNo", con);

SqlCommand cmd2 = new SqlCommand("update Fis set sucAd=@p2,orgutAd=@p3,sucYeri=@p4,sucTarihi=@p5,tahkYapanMakam=@p6,cezaYil=@p7,cezaAy=@p8,cezaGün=@p9 where fisNo=@fNo", con);

dr = MessageBox.Show("Güncelleme Yapılsın mı?", "Uyarı!", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dr == DialogResult.Yes)

{

try

{

cmd.Parameters.AddWithValue("@p1", txtKeTC.Text);

cmd.Parameters.AddWithValue("@p2", txtKEAd.Text);

cmd.Parameters.AddWithValue("@p3", txtKeİkinciAd.Text);

cmd.Parameters.AddWithValue("@p4", txtKeSoyad.Text);

cmd.Parameters.AddWithValue("@p5", txtKeBabaAd.Text);

cmd.Parameters.AddWithValue("@p6", txtKeAnneAd.Text);

cmd.Parameters.AddWithValue("@p7", cbKeDurumu.Text);

cmd.Parameters.AddWithValue("@p9", dtpKeDogum.Value);

cmd.Parameters.AddWithValue("@p10", radioKeErkek.Checked ? "Erkek" : "Kadın");

cmd.Parameters.AddWithValue("@p11", cbKeDurumu.Text);

cmd.Parameters.AddWithValue("@p12", pBoxSahisResim.Tag);

cmd.Parameters.AddWithValue("@sNo", txtSahSahisNo.Text);

try

{

cmd.ExecuteNonQuery();

cmd1.Parameters.AddWithValue("@p1", txtKeCilt.Text);

cmd1.Parameters.AddWithValue("@p2", txtKeSira.Text);

cmd1.Parameters.AddWithValue("@p3", txtKeAileSira.Text);

cmd1.Parameters.AddWithValue("@p4", cbKeil.Text);

cmd1.Parameters.AddWithValue("@p5", cbKeilce.Text);

cmd1.Parameters.AddWithValue("@p6", cbKeMahalle.Text);

cmd1.Parameters.AddWithValue("@p7", txtKeAcikAdres.Text);

cmd1.Parameters.AddWithValue("@kNo",Convert.ToInt32(label59.Tag));

}

catch (Exception)

{

MessageBox.Show("Nufus Bilgilerini kontrol ediniz.");

}

try

{

cmd1.ExecuteNonQuery();

cmd2.Parameters.AddWithValue("@p2", cbKeSuc.Text);

cmd2.Parameters.AddWithValue("@p3", cbKeOrgut.Text);

cmd2.Parameters.AddWithValue("@p4", txtKeSucYeri.Text);

cmd2.Parameters.AddWithValue("@p5", dtpKeSucTarih.Value);

cmd2.Parameters.AddWithValue("@p6", cbKeTahakkuk.Text);

cmd2.Parameters.AddWithValue("@p7", txtKeCezaYil.Text);

cmd2.Parameters.AddWithValue("@p8", txtKeCezaAy.Text);

cmd2.Parameters.AddWithValue("@p9", txtKeCezaGun.Text);

cmd2.Parameters.AddWithValue("@fNo", txtKeFis.Text);

}

catch (Exception)

{

MessageBox.Show("Nufusa Kayıtlı Olduğu Yer Bilgilerini kontrol ediniz.");

}

try

{

cmd2.ExecuteNonQuery();

}

catch (Exception)

{

MessageBox.Show("Fiş Bilgilerini kontrol ediniz.");

}

MessageBox.Show("Güncelleme Başarılı");

}

catch (Exception)

{

MessageBox.Show("Güncelleme yapılırken hata oluştu.!!");

}

}

else

{

MessageBox.Show("Güncelleme İptal Edildi");

}

con.Close();

Resim Ekleme: Veritabanına resim eklerken kullanılan semce penceresi oluşturuluyor(openfiledialog). Daha sonra belirli formatlarda resim eklenebilmesi için resim formatı ayarlanıyor.Eklenen dosyanın yolu değişkene atanıyor. Bu değişkeni de gerekli yerlere atıyoruz.

*http://www.ahmetcansever.com/c-2/c-veritabanina-resim-yolu-ekleme-ve-pictureboxta-gosterme/ yardım alınmıştır.*

private void pBoxKeResim\_Click(object sender, EventArgs e)

{

OpenFileDialog dosya = new OpenFileDialog();

dosya.Filter = "Resim Dosyası |\*.jpg;\*.nef;\*.png | Tüm Dosyalar |\*.\*";

dosya.ShowDialog();

string dosyayolu = dosya.FileName;

pBoxKeResim.Tag = dosyayolu;

pBoxKeResim.ImageLocation = dosyayolu;

}