**Veri Tabanı Yönetim Sistemleri Dönem Projesi**

**Ahmet Şimşek**

**Bilgisayar Mühendisliği (İ.Ö.)**

**06.01.2024**

**Proje Sunum İçeriği**

Çalışan Projemden Ekran Çıktılarım = **sayfa 3-9**

Projem İçin Oluşturduğum Tablolar ve İlişkileri = **sayfa 10-12**

Stored Procedure(SP) Örneklerim = **sayfa 13-18**

Trigger Örneklerim = **sayfa 19-23**

Fonksiyon Örneklerim = **sayfa 24-26**

**Çalışan Projemden Ekran Çıktılarım**

Projem 3 giriş ekranından oluşmaktadır. Bunlar, Kullanıcı giriş paneli, Personel Giriş paneli ve Admin giriş panelidir. Her bir panelin kendine ait fazla veya az işlemleri bulunmaktadır. Sırasıyla projemin çıktılarına şu şekildedir;

**1.ANA LOGİN PANEL**

**metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu**

İlk olarak projem ilk açıldığında ana login ekranım açılıyor. Giriş panelimde Kullanıcı hesabına giriş yap, Hesap oluştur, Personel girişi, Admin girişi butonlarım beni karşılıyor. İlk olarak kullanıcı web kısmı ekran çıktılarını inceleyeceğiz.

**2.KULLANICI ARAYÜZÜ**

**metin, yazılım, multimedya yazılımı, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

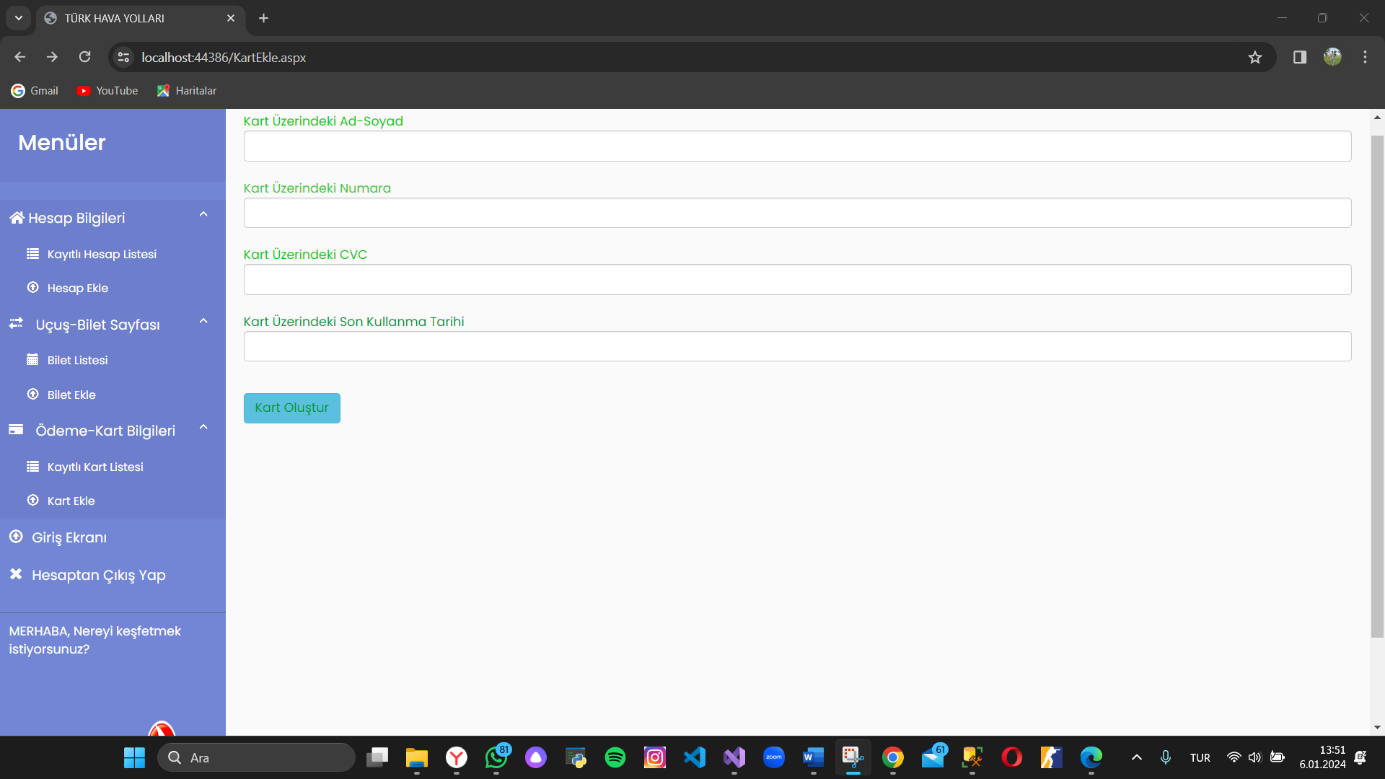
Kullanıcı T.C. numarasını ve şifresini giren kullanıcının hesaba giriş yapınca üyelik bilgileri sayfası bizi karşılıyor. Bu ekranda kullanıcı sol paneldeki bilet oluşturma kısmından;

* 1-Bilet Satın Alma Sayfası

**metin, yazılım, multimedya yazılımı, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

* 2-Kart Bilgisi Ekleme Sayfası

****

Bilet oluşturulup kart bilgilerini girerek biletin onaylanmasını sağlayabilir kullanıcı. Hesaptan çıkış yap diyerek Giriş paneline yönlenebiliyoruz.

**3.HESAP OLUŞTUR**

**metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Ana login ekranından HESAP OLUŞTUR butonuna basarak hesap oluşturma işlemi sağlayıp üye T.C. numara ve şifrenizle oluşturduğunuz kullanıcı hesabınıza giriş yapabilirsiniz.

**4.PERSONEL LOGİN PANEL**

**metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Eğer personel iseniz ana giriş panelinden PERSONEL GİRİŞİ BUTONUNA basınız. Personel giriş panelinde personel numaranızı ve personel şifrenizi girip hesabınıza giriş yapınız.

**5.PERSONEL WEB ARAYÜZ**

**metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Hesaba giriş yapan personelin ana sayfası bu şekildedir ve yapabileceği işlemler soldaki panelde mevcuttur. Bu işlemlerden;

* 1-KAYITLI HESAP BİLGİLERİ SEÇİLİRSE

**metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Personel kayıtlı hesap bilgilerine basarsa sistemde kayıtlı hesapları görebilecek ve bu hesapları sil-güncelle yetkisiyle değiştirebilecektir. Buna ek hesap ekleme yetkisine de sahiptir.

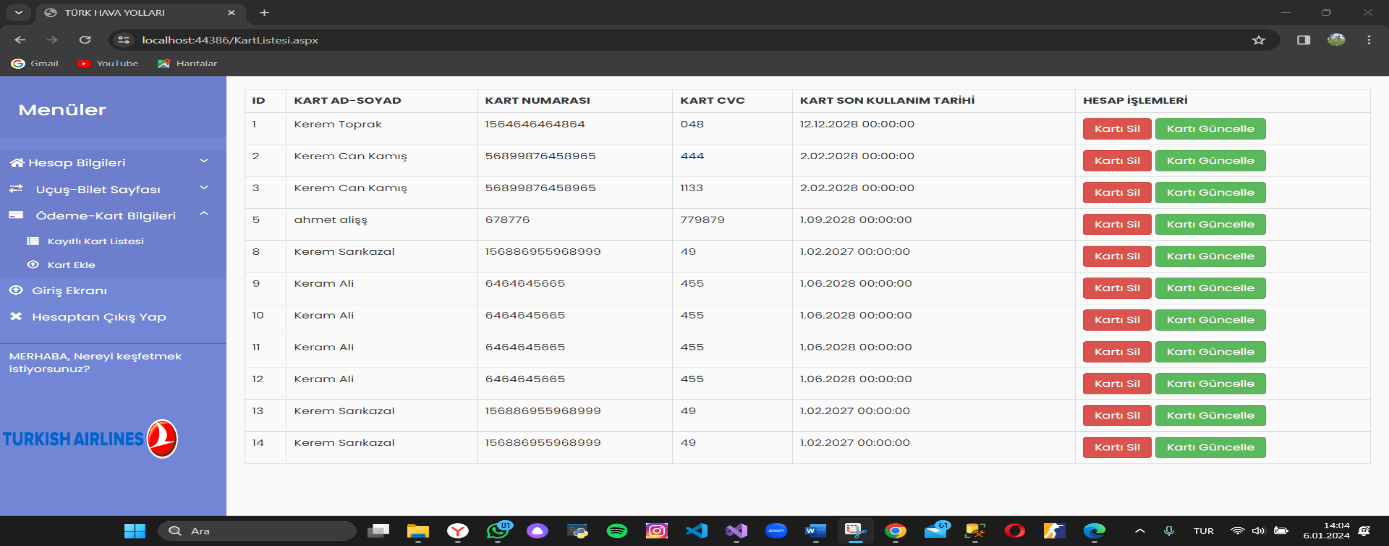
* 2-KAYITLI BİLET SAYFASI SEÇİLİRSE

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Personel kayıtlı bilet sayfasına basarsa sistemde kayıtlı biletleri görebilecek ve bu biletleri sil-güncelle yetkisiyle değiştirebilecektir. Buna ek bilet ekleme yetkisine de sahiptir.

* 3-KAYITLI KART LİSTESİ SEÇİLİRSE

****Personel kayıtlı kart listesine basarsa sistemde kayıtlı kartları görebilecek ve bu kartları sil-güncelle yetkisiyle değiştirebilecektir. Buna ek panelden kart ekleme yetkisine de sahiptir. Bunlara ek, çıkış yap butonuna basarsa ana giriş paneline yönlendirilecektir.

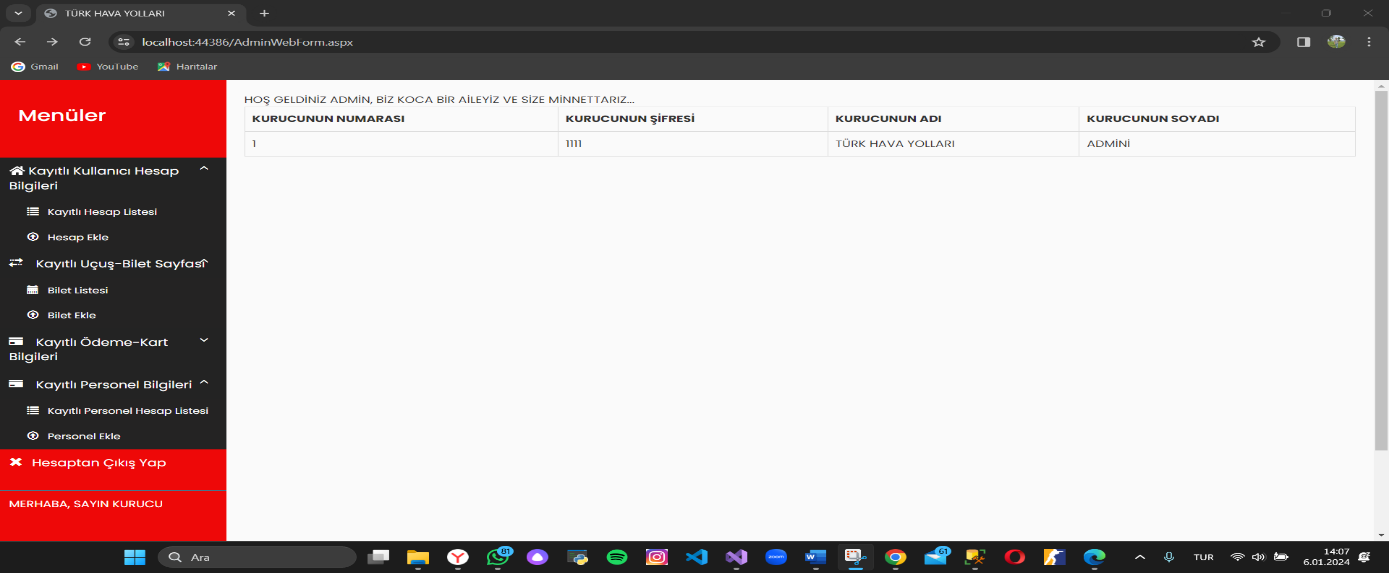
**6.ADMİN LOGİN PANEL**

**ekran görüntüsü, metin, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Eğer kurucu siz iseniz ana giriş panelinden ADMİN GİRİŞİ BUTONUNA basınız. Admin giriş panelinde Admin numaranızı ve Admin şifrenizi girip hesabınıza giriş yapınız.

**7.ADMİN WEB ARAYÜZ**

****

Hesaba giriş yapan adminin ana sayfası bu şekildedir ve yapabileceği işlemler soldaki panelde mevcuttur. Bu işlemlerden;

* 1-KAYITLI HESAP BİLGİLERİ
* 2-KAYITLI BİLET SAYFASI
* 3-KAYITLI KART LİSTESİ
* 4-KAYITLI PERSONEL HESAP LİSTESİ
* 5-HESAPTAN ÇIKIŞ YAP

Seçeneklerinden her biri hareketi, hesabı görme ve üzerinde değiştirme, ekleme, silme ve daha fazlasını yapabilme hakkına sahiptir.

**Projem İçin Oluşturduğum Tablolar ve İlişkileri**

**metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

Projem için 4 tablo oluşturdum THYWEBOTOMASYON veri tabanıma. Bunlar;

1-TBL\_UyeEkleme

2-TBL\_BiletOlusturma

3-TBL\_OdemeBilgisi

4-TBL\_Personel

Bu ana tablolarım haricinde trigger’larım için de 3 tane tablo ekledim. Bunlar;

1-TBL\_DurumSilme

2-TBL\_SilmeDurum2

3-TBL\_DurumEklen

**1.**

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

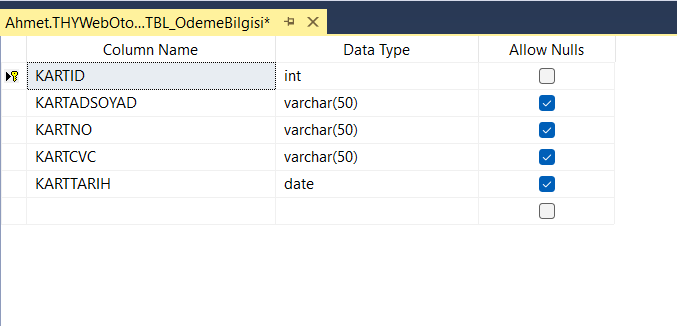
Açıklama otomatik olarak oluşturuldu**

**2.**

**metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**3.**

****

**4.**

**metin, ekran görüntüsü, sayı, numara, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**5.**

**metin, ekran görüntüsü, ekran, görüntüleme, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**6.**

**7.metin, ekran görüntüsü, yazı tipi, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**8.**

**metin, ekran görüntüsü, ekran, görüntüleme, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**STORED PROCEDURE(SP) ÖRNEKLERİ**

metin, ekran görüntüsü, yazılım, bilgisayar içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Bu güncelleme SP’sinde “Uptade” ifadesi kullanılarak TBL\_UyeEkleme tablosundaki belirli bir kullanıcının verisi güncellenir. İlk olarak paremetrelerin tanımlanması yapılır, sonra set ile ilgili paremetreler ilgili sütunlara atanır ve where koşulu ile girdiğimiz numara parametresi tablodaki numara parametresi ile eşleşir. Son olarak güncelleme işlemi başarılı başarılı olursa Results kısmında” Üye bilgileri başarıyla güncellendi” mesajı çıkmaktadır.

Buna ek olarak, aşağıda yorum satırı içinde bu Sp’nin nasıl çalışacağına örnek bir kod yazdım. Bu kodları yazıp execute edince results kısmında başarı mesajı gelirken girdiğim bilgiler TBL\_UyeEkleme tablosuna eklenmiş olmaktadır.

**metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

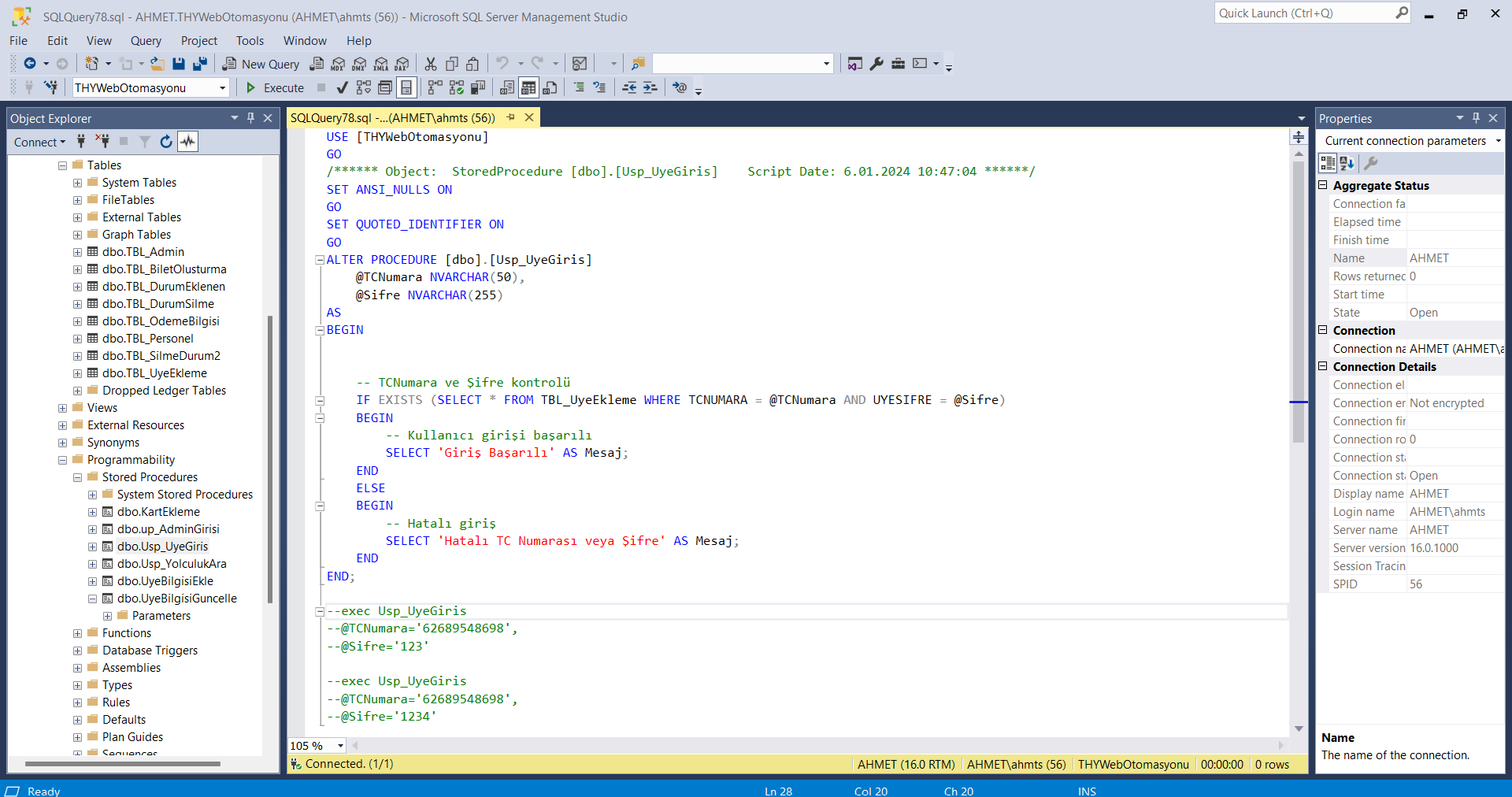
1. Bu ekleme SP’sinde “İnsert” ifadesi kullanılarak bilgileri girilen kullanıcı TBL\_UyeEkleme tablosuna eklenir. İlk olarak paremetrelerin tanımlanması yapılır, sonra insert into komutumuz ile ilgili paremetreler ilgili sütunlara eklenir. Son olarak güncelleme işlemi başarılı başarılı olursa Results kısmında” Ekleme işlemi başarılı bir şekilde gerçekleşti” mesajı çıkmaktadır.

Buna ek olarak, aşağıda yorum satırı içinde bu Sp’nin nasıl çalışacağına örnek bir kod yazdım. Bu kodları yazıp execute edince results kısmında başarı mesajı gelirken girdiğim bilgiler TBL\_UyeEkleme tablosuna eklenmiş olmaktadır. Bu sayede her seferinde yeni üye için kod yazıcağımıza hızlıca bu SP’i tekrardan çalıştırabiliriz.

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**3.**Bu SP’de “İlk olarak paremetrelerin tanımlanması yapılır (Sonucunu görmek istediğimiz), sonra ilgili paremetreler ilgili sütunlara atanır ve where koşulu ile girdiğimiz konum bilgisi parametresi tablodaki halihazırda olan konum parametresi ile eşleşir. Son olarak, yorum satırı içine aldığım kod ile SP çağırılır ve çalıştırılır. Örnekte konum bilgisi olarak “Nereden=Van” ve “Nereye=Muş” bilgilerini girdiğimden ötürü execute edince ilgili konumların bilgisi tabloda var ise ilgili tablonun tüm elemanlarıyla ilgili verinin olduğu satırı Result kısmında bize vermektedir.



**4.**Bu SP kullanıcı girişini doğrulamak için yazıldı. İlk olarak parametrelerin tanımlanması yapılır işlem yapılıcak sonra IF EXISTS ifadesi ile TBL\_UyeEkleme tablosundan tüm TCNUMARA ve UYESIFRE sütunları, verilen @TCNumara ve @Sifre değerleri ile eşleşen bir kayıt olup olmadığını, şartın sağlanıp sağlanmadığını kontrol edildi. Eğer eşleşen bir kayıt yoksa, giriş başarısız olarak kabul edilir ve @Sonuc değişkenine 'Hatalı giriş mesajı atanır. Eğer eşleşen bir kayıt varsa, giriş başarılı olarak kabul edilir ve @Sonuc değişkenine 'Giriş Başarılı mesajı atanır. Yorum satırlarında olan SP çağırma kodlarım ile iki duruma da örnek vermekteyim. İlk olarak if kısmına örnek, ikinci olarak else durumuna örnek vermekteyim.

metin, yazılım, bilgisayar simgesi, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

**5.** Bu SP de admin girişini doğrulamak için yazıldı. İlk olarak parametrelerin tanımlanması yapılır işlem yapılıcak sonra IF EXISTS ifadesi ile ADMINNUMARA ve ADMINSIFRE sütunları, verilen @AdmininNumarası ve @AdmininSifresi değerleri ile eşleşen bir kayıt olup olmadığını, şartın sağlanıp sağlanmadığını kontrol edildi. Eğer eşleşen bir kayıt yoksa, giriş başarısız olarak kabul edilir ve @Sonuc değişkenine 'Hatalı giriş mesajı atanır. Eğer eşleşen bir kayıt varsa, giriş başarılı olarak kabul edilir ve @Sonuc değişkenine 'Başarılı giriş. Sayın admin, hoş geldiniz.' mesajı atanır. Yorum satırlarında olan SP çağırma kodlarım ile iki duruma da örnek vermekteyim. İlk olarak if kısmına örnek, ikinci olarak else durumuna örnek vermekteyim.

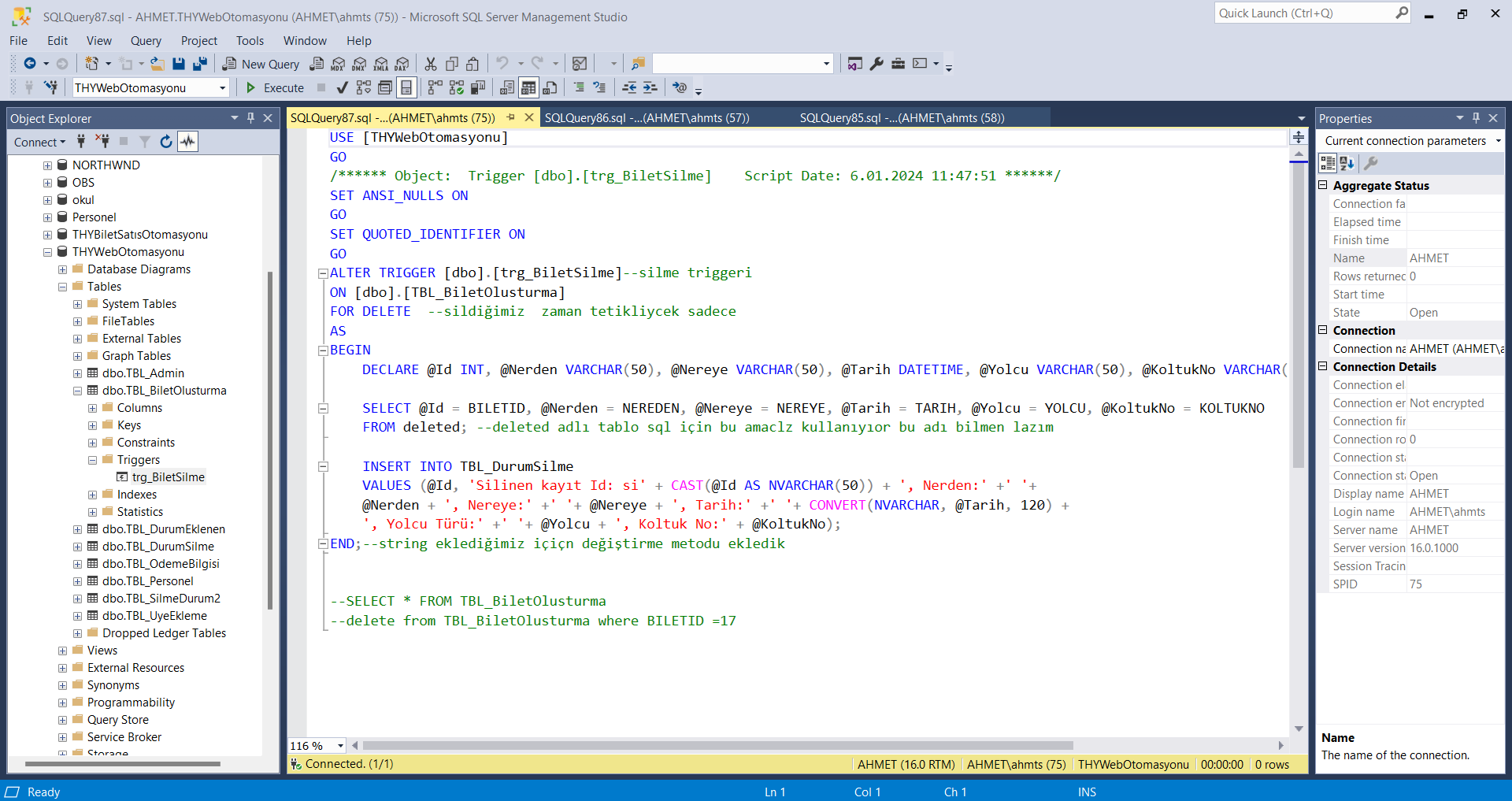
**metin, yazılım, bilgisayar simgesi, bilgisayar içeren bir resim

Açıklama otomatik olarak oluşturuldu**

**6.** Bu ekleme SP’sinde “İnsert” ifadesi kullanılarak bilgileri girilen kullanıcı TBL\_OdemeBilgisi tablosuna eklenir. İlk olarak paremetrelerin tanımlanması yapılır, sonra insert into komutumuz ile ilgili paremetreler ilgili sütunlara eklenir. Son olarak güncelleme işlemi başarılı başarılı olursa Results kısmında” Kartınız başarılı bir şekilde eklenmiştir” mesajı çıkmaktadır.

Buna ek olarak, aşağıda yorum satırı içinde bu Sp’nin nasıl çalışacağına örnek bir kod yazdım. Bu kodları yazıp execute edince results kısmında başarı mesajı gelirken girdiğim bilgiler TBL\_OdemeBilgisi tablosuna eklenmiş olmaktadır. Bu sayede her seferinde yeni kart için kod yazıcağımıza hızlıca bu SP’i tekrardan çalıştırabiliriz.

**Trigger Örnekleri**

**metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

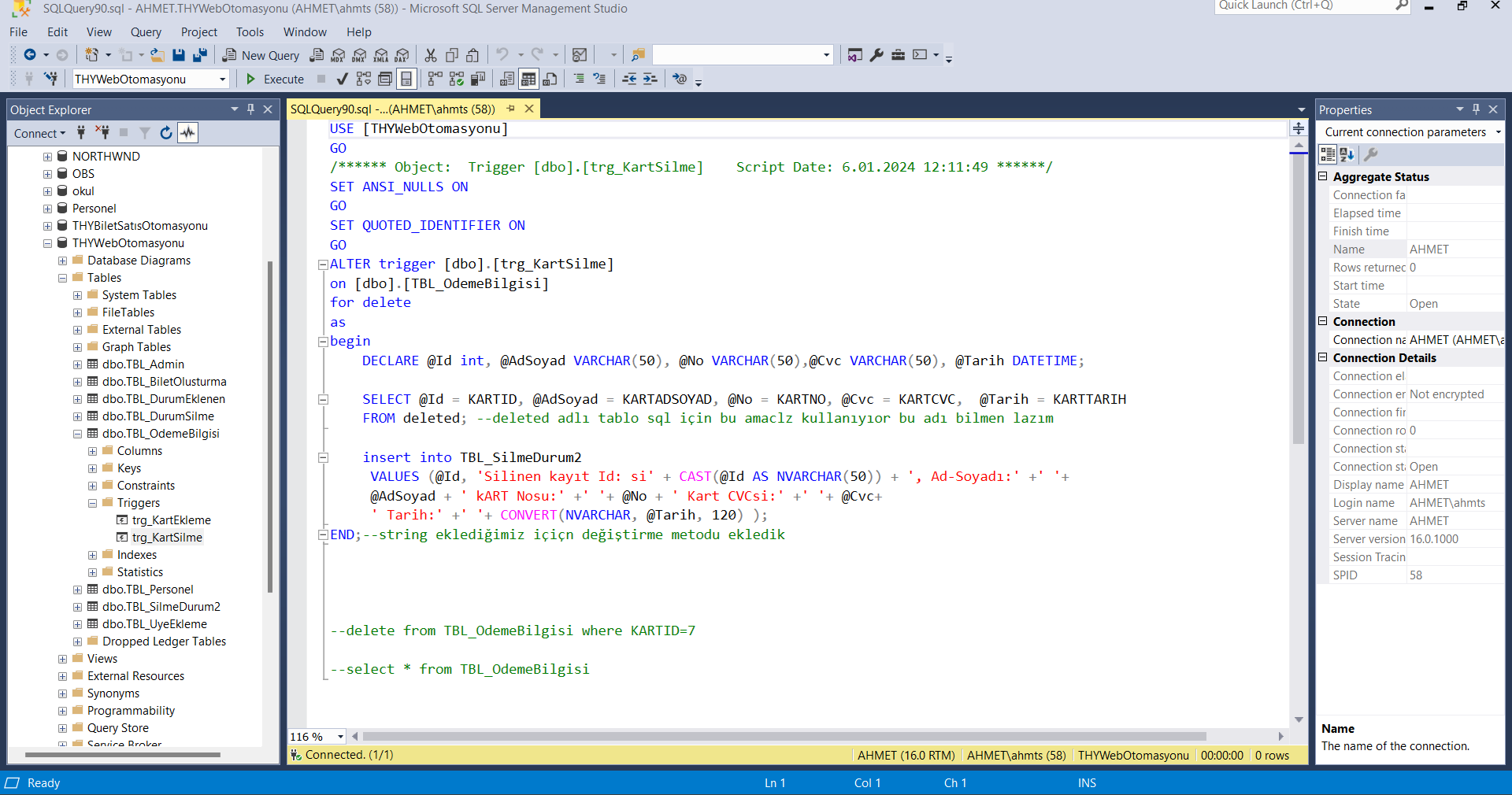
Açıklama otomatik olarak oluşturuldu**

**1.**İlk trigger örneğim, silme(delete) triggerı. Sadece TBL\_BiletOlusturma üzerinde silme işlemi yapacağım zaman tetikleyici çalışır. DECLARE ifadesi ile değişkenler tanımlanır ve bu değişkenlere SELECT ifadesi ile deleted(bu tablo SQL de bu amaçla kullanılıyor!) tablosundan gelen bilgiler atanır. INSERT INTO TBL\_DurumSilme ifadesi ile TBL\_DurumSilme adlı başka bir tabloya yeni bir kayıt eklenir. Bu yeni kayıt, silinen kaydın bilgilerini ve bir mesajı içerir. Mesaj, silinen kaydın Id'sini, nerden ve nereye olduğunu, tarihini, yolcu türünü ve koltuk numarasını içerir. Triggerın altında yorum satırı şeklindeki kodu execute edince örneğin where BILETID = 17 ise bu ID’e sahip bileti siler ve TBL\_DurumSilme tabloma tüm verileriyle ekler. Buna ek önceden sildiğim biletlerin de durum tablosuna eklenişi görseldeki gibidir.

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**2.**İkinci trigger örneğimde, insert triggerını kullandım. Bu tetikleyici ile TBL\_OdemeBilgisi tablosu üzerinde çalışırken tabloya yeni bir ekleme yaptığımız zaman insert işlemi tetikleyiciye “Yeni kart kaydı başarılı bir şekilde yapıldı” şeklinde mesaj döndürtür. Bu triggera örnek olarak yorum satırı içindeki insert komutunu yazınca Results kısmında mesaj çıkmaktadır, TBL\_OdemeBilgisi tablosuna gittiğimizde ise yazdığımız ekleme sorgusundaki bilgiler tabloya eklenmiş olacaktır.



metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**3.**Üçüncü trigger örneğim, silme(delete) triggerı. Sadece TBL\_OdemeBilgisi üzerinde silme işlemi yapacağım zaman tetikleyici çalışır. DECLARE ifadesi ile değişkenler tanımlanır ve bu değişkenlere SELECT ifadesi ile deleted(bu tablo SQL de bu amaçla kullanılıyor!) tablosundan gelen bilgiler atanır. INSERT INTO TBL\_DurumSilme2 ifadesi ile TBL\_DurumSilme2 adlı başka bir tabloya silinen veri yeni bir kayıt şeklinde eklenir. Bu yeni kayıt, silinen kaydın bilgilerini ve bir mesajı içerir. Mesaj, silinen kaydın Id'sini, Ad-Soyadını, Kart no’sunu, Kart cvc’sini ve Tarihi içerir. Triggerın altında yorum satırı şeklindeki kodu execute edince örneğin where KARTID = 7 ise bu ID’e sahip Kartu siler ve TBL\_DurumSilme2 tabloma tüm verileriyle ekler. Buna ek önceden sildiğim kartların da durum2 tablosuna eklenişi görseldeki gibidir.

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**4.**Dördüncü trigger örneğimde, insert triggerını kullandım. Bu tetikleyici ile TBL\_Personel tablosu üzerinde çalışırken tabloya yeni bir ekleme yaptığımız zaman insert işlemi tetikleyiciye “Personel başarılı bir şekilde sisteme kayıt edilmiştir “şeklinde mesaj döndürtür. Bu triggera örnek olarak yorum satırı içindeki insert komutunu yazınca Results kısmında mesaj çıkmaktadır,TBL\_Personel tablosuna gittiğimizde ise yazdığımız ekleme sorgusundaki bilgiler tabloya eklenmiş olucaktır.

metin, yazılım, bilgisayar simgesi, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

**5.**beşinci trigger örneğim, ekleme(insert) triggerı. Sadece TBL\_UyeEkleme üzerinde ekleme işlemi yapacağım zaman tetikleyici çalışır. DECLARE ifadesi ile değişkenler tanımlanır ve bu değişkenlere SELECT ifadesi ile inserted(bu tablo SQL de bu amaçla kullanılıyor!) tablosundan gelen bilgiler atanır. INSERT INTO TBL\_DurumEklenen Id ve Durum sütun ifadesi ile TBL\_DurumEklenen adlı başka bir tabloya eklenen veri yeni bir kayıt şeklinde eklenir. Bu yeni kayıt, eklenen kaydın bilgilerini ve bir mesajı içerir. Mesaj, eklenen üyenin Id'sini, T.C. Numarasını, şifresini, adını, soyadını, telefonunu, ülkesini, şehrini ve emailini içerir. Triggerın altında yorum satırı şeklindeki insert into kodunu execute edince row affected yazar TBL\_UyeEkleme tabloma ekler ve TBL\_DurumEkleme tabloma da tüm verileriyle ekler. Buna ek önceden eklediğim üyelerin de durumEkleme tablosuna eklenişi görseldeki gibidir.

**Fonksiyon Örnekleri**



**1.**İlk fonksiyon örneğim scaler-value function türünden, İsimleriBuyut adlı varchar türünde tek parametre alan ve varchar türünde değer döndüren fonksiyonumuzu yazdık. Bu Fonksiyon, aldığı metin değerini büyük harfe dönüştürmek için SQL'in UPPER fonksiyonunu kullanır. Bu fonksiyon, bir metin değerini alır ve tüm harfleri büyük harfe dönüştürerek geri döndürür. Altındaki Yorum satırı içindeki fonksiyonu çalıştırdığım örnekte TBL\_OdemeBilgisi tablomdan kartadsoyad sütununu getiriyor ama tüm isim bilgileri büyük harfle Results kısmında bize görünür.

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**2.**Fonksiyon örneğim table valued function türünden, Fonksiyon adı UlkeyeGoreUyeGetir ve bir parametre olarak @ulke değerini alır. RETURNS TABLE ile bir tablo türünde bir sonuç kümesi döndürür. WHERE UYEULKE = @ulke sorgumuz ile Fonksiyon, TBL\_UyeEkleme tablosunda UYEULKE sütununun girdiğimiz @ulke parametresi ile eşleşen değerlere sahip olan üyeleri seçer ve bu üyeleri içeren bir sonuç kümesini döndürür. Bu sayede aynı ülkeye sahip birçok kullanıcı üye görülebilir. Aşağıda yorum satırı içindeki fonksiyonu çağırdığım sorguyu çalıştırınca TBL\_UyeEkleme tablosunda ülke bilgisi Almanya girilen üyelerin o tablo içerisindeki tüm verisini Results kısmında görebiliyoruz.

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

**3.**Fonksiyon örneğim table valued function türünden, Fonksiyon adı UcusBilgisiGetir ve bir parametre olarak @Nereden değerini alır. RETURNS TABLE ile bir tablo türünde bir sonuç kümesi döndürür. WHERE NEREDEN = @Nereden sorgumuz ile Fonksiyon, TBL\_BiletOlusturma tablosunda NEREDEN sütununun girdiğimiz @Nereden parametresi ile eşleşen değerlere sahip olan Biletleri seçer ve bu biletleri içeren bir sonuç kümesini döndürür. Bu sayede aynı bilete sahip birçok kullanıcı biletleri görülebilir. Aşağıda yorum satırı içindeki fonksiyonu çağırdığım sorguyu çalıştırınca TBL\_BiletOlusturma tablosunda NEREDEN bilgisi Van girilen biletlerin o tablo içerisindeki tüm verisini Results kısmında görebiliyoruz.