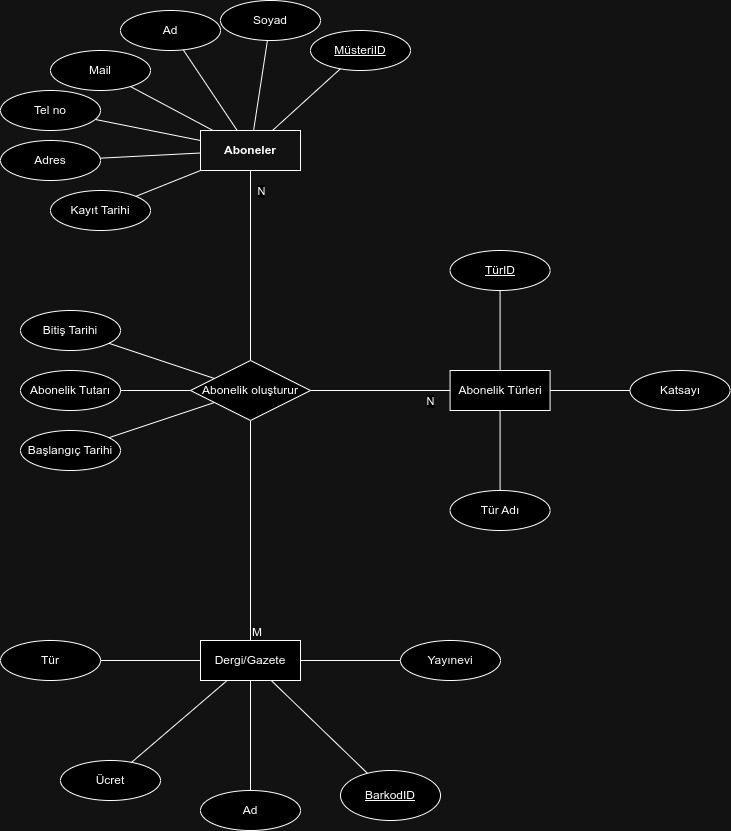
**VERİ TABANI YÖNETİM SİSTEMLERİ PROJE ÖDEVİ**

## **Hazırlayanlar:**

## 201001045 Umut Karakoç

## 201001015 Senanur Değirmenbaşı

# **ER Diyagramı**



## **Aboneler Tablosu**

İçerisinde abonelere ait kayıt tarihi, adres, tel no, mail, ad, soyad ve müsteriID verilerini tutar. Kayıt tarihi **date** veri tipinde tanımlanmıştır. Adres, tel no, mail, ad ve soyad **nvarchar** veri tipinde tanımlanmıştır. MusteriID **int** veri tipinde tanımlanmış olup **primary key** olarak kullanılır, yani benzersizdir.

## **Abonelik Tablosu**

Abonelik oluşturma eyleminin temel özelliklerini içeren tablodur. Başlangıç tarihi, abonelik tutarı ve bitiş tarihi verilerini tutar. Başlangıç tarihi ve bitiş tarihi **date** veri tipinde, abonelik tutarı **decimal** veri tipinde tanımlanmıştır.

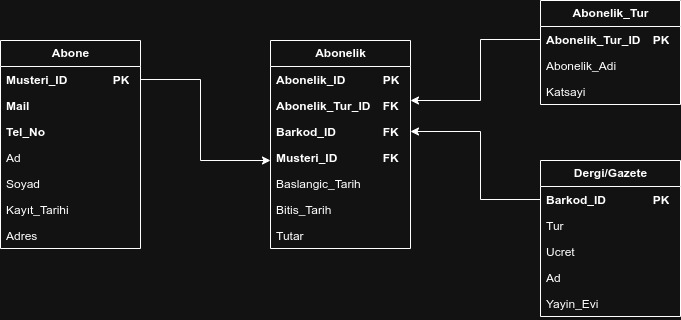
## **Abonelik Türleri Tablosu**

Tür adı, türID ve katsayı verilerini tutar. Çeşitli abonelik türleri vardır. Bu nedenle her türe özgü olması için türID, **primary key** olarak tanımlanmıştır. Abonelik türüne göre katsayı özelliği değişmektedir. Tür adı ise abonelik türünün adını ifade eder. Tür adı **nvarchar,** türID **int** ve katsayı **decimal** veri tipinde tanımlanmıştır.

## **Dergi/Gazete Tablosu**

Gazete ve dergilere ait ad, tür, yayınevi, ücret ve barkodID verilerini tutar. Dergi veya gazetelerin adları aynı olsa bile bunlar farklı bir dergi veya gazete olabilir. Bu nedenle her bir dergi ve gazete için barkodID **primary key** olarak kullanılır. Ad, tür ve yayınevi **nvarchar** veri tipinde, ücret **decimal** veri tipinde, barkodID **int** veri tipinde tanımlanmıştır. Ücretin **decimal** veri tipinde tanımlanmasının nedeni ücretin hassas bir ölçüm olmasıdır.

# **ER Diyagramının Veri Tabanı Şemasına Aktarımı ve Tablolar Arası İlişkiler**



**Abonelik tablosu** abonelik oluşturma eyleminin tablosudur. Abonelik tablosunda her bir abonelik benzersizdir. Bunun için **primary key** olarak bir Abonelik\_ID tanımlanmıştır. Sistemimizdeki bütün tablolarla doğrudan ilişki içerisindedir ve bu tablolara ait verileri **foreign key** olarak tutar. Bu ilişkileri inceleyelim:

## **Abone – Abonelik Tabloları Arasındaki İlişki**

Abone tablosunda bulunan **primary key** şeklinde tanımlı Musteri\_ID alanı, Abonelik tablosunda bulunan Musteri\_ID alanına **foreign key** olarak tanımlanmıştır. Bu alan, her bir aboneliğin hangi müşteriye ait olduğunu belirlemek için kullanılır.

## **Abonelik\_Tur – Abonelik Tabloları Arasındaki İlişki**

Abonelik\_Tur tablosunda bulunan  **primary key** şeklinde tanımlı Abonelik\_Tur\_ID alanı, Abonelik tablosunda **foreign key** olarak tanımlanmıştır. Bu alan, her bir aboneliğin hangi abonelik türünde olduğunu belirlemek için kullanılır.

## **Dergi/Gazete- Abonelik Tabloları Arasındaki İlişki**

Deri/Gazete tablosunda bulunan **primary key** şeklinde tanımlı Barko\_ID alanı, Abonelik tablosunda **foreign key** olarak tanımlanmıştır. Bu alan, her bir aboneliğin hangi dergi veya gazeteye ait olduğunu belirlemek için kullanılır.

# **SQL SERVER VERİ TABANIYLA GERÇEKLEŞTİRİLECEK İŞLEMLER İÇİN KULLANILACAK OLAN PROGRAMLAMA DİLİ**

Oluşturulacak olan SQL Server veri tabanı C# programlama diliyle kullanılacaktır. C#, SQL Server veri tabanına erişim ve bu veri tabanında çeşitli işlemler gerçekleştirme yeteneğine sahip bir programlama dilidir. Bunun yanı sıra C#’ın nesne yönelimli bir programlama dili olması, veri tablolarındaki verileri C# nesnelerine bağlama ve ilişkilendirme olanağı verir. Bu, veritabanındaki kayıtları daha kolay ve anlamlı bir şekilde kullanabilmek için bir avantaj sağlar.

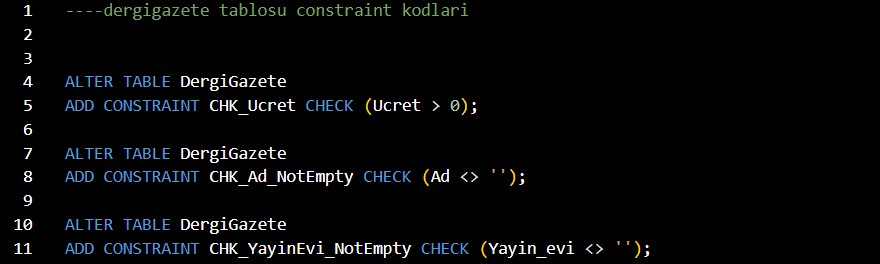
# **İŞLEMLER**

## **A.1) Constraint Kodları**

## **Dergi Gazete Tablosu Constraint Kodları**

1. CHK\_Ucret **CHECK**: Ücretsiz bir dergi veya gazete bulunmamaktadır. Bu nedenle DergiGazete tablosunda"Ucret" sütunundaki değerlerin 0'dan büyük olmasını zorunlu kılan bir kısıtlama kullanılmıştır.
2. CHK\_Ad\_NoEmpty **CHECK**: DergiGazete tablosundaki "Ad" sütunundaki değerlerin boş olmamasını zorunlu kılan bir kısıtlamadır. Çünkü her dergi ve gazetenin bir adı vardır.
3. CHK\_YayinEvi\_NotEmpty **CHECK**:

DergiGazete tablosundaki "Yayin\_evi" sütunundaki değerlerin boş olmamasını zorunlu kılan bir kısıtlamadır. Çünkü her dergi ve gazeteyi basan bir yayınevi vardır.

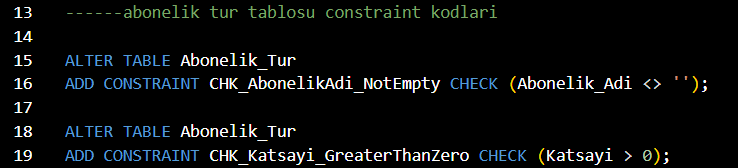


## **Abonelik\_Tur Tablosu Constraint Kodları**

1. CHK\_AbonelikAdi\_NotEmpty **CHECK**:

Abonelik\_Tur tablosundaki "Abonelik\_Adi" sütununda değerlerin boş olmamasını zorunlu kılan bir kısıtlamadır.

1. CHK\_Katsayi\_GreaterThanZero **CHECK**: Abonelik\_Tur tablosundaki "Katsayi" sütununda değerlerin 0'dan büyük olmasını zorunlu kılan bir kısıtlamadır.



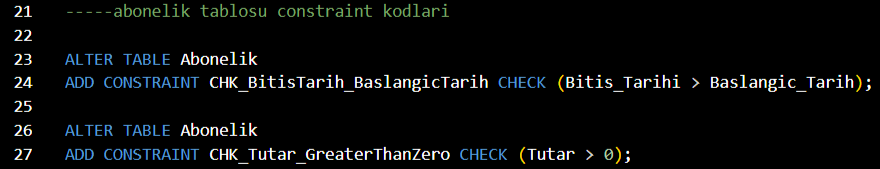
## **Abonelik Tablosu Constraint Kodları**

1. CHK\_BitisTarih\_BaslangicTarih **CHECK**:

Abonelik tablosundaki "Bitis\_Tarihi" sütunundaki değerlerin "Baslangic\_Tarih" sütunundaki değerlerden büyük olmasını zorunlu kılan bir kısıtlamadır.

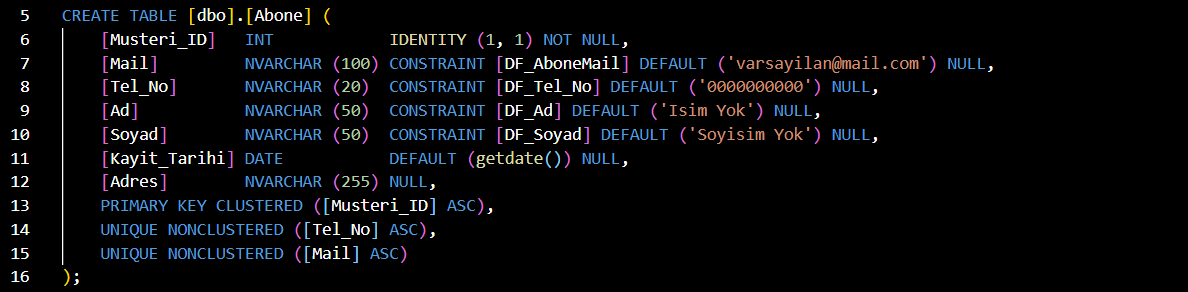
1. CHK\_Tutar\_GreaterThanZero **CHECK**:

Abonelik tablosundaki "Tutar" sütunundaki değerlerin 0'dan büyük olmasını zorunlu kılan bir kısıtlamadır.



## **A.2) Her Tabloya Ait Oluşturulma, Default Değerlerinin ve Trigger Kodlarının Toplu İncelenmesi**

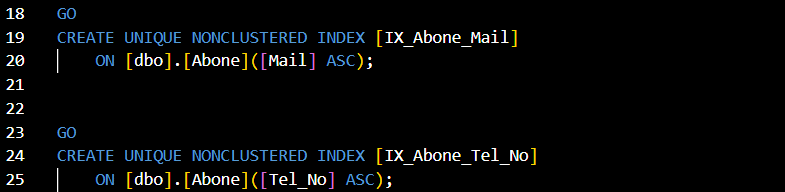
## **Abone Tablosunun Oluşum Kodları ve Default Değerleri**



Tablo, müşteriye ait bilgileri saklamak için oluşturulmuştur.

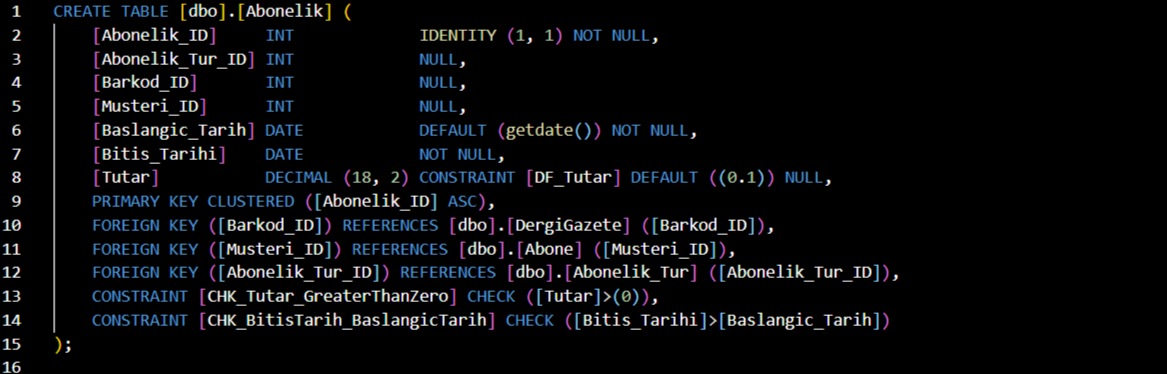
- Musteri\_ID: Otomatik artan (IDENTITY) olarak tanımlanmış, tablonun birincil anahtarıdır.

- Mail, Tel\_No, Ad, Soyad, Kayit\_Tarihi, Adres: Müşteriye ait e-posta, telefon numarası, ad, soyad, kayıt tarihi ve adres bilgilerini saklamak için NVARCHAR veya DATE formatında tanımlanmış alanlardır. Bazıları varsayılan değerlerle (DEFAULT) veya NULL olarak tanımlanmıştır.



Yukarıda belirtilen kod kümesi Mail ve Tel\_No alanları üzerinde benzersiz **indeksler** oluşturarak, aynı e-posta veya telefon numarasıyla birden fazla kaydın oluşmasını engeller.

## **Abonelik Tablosu**

Bu tablo, abonelik bilgilerini saklamak için tasarlanmıştır.

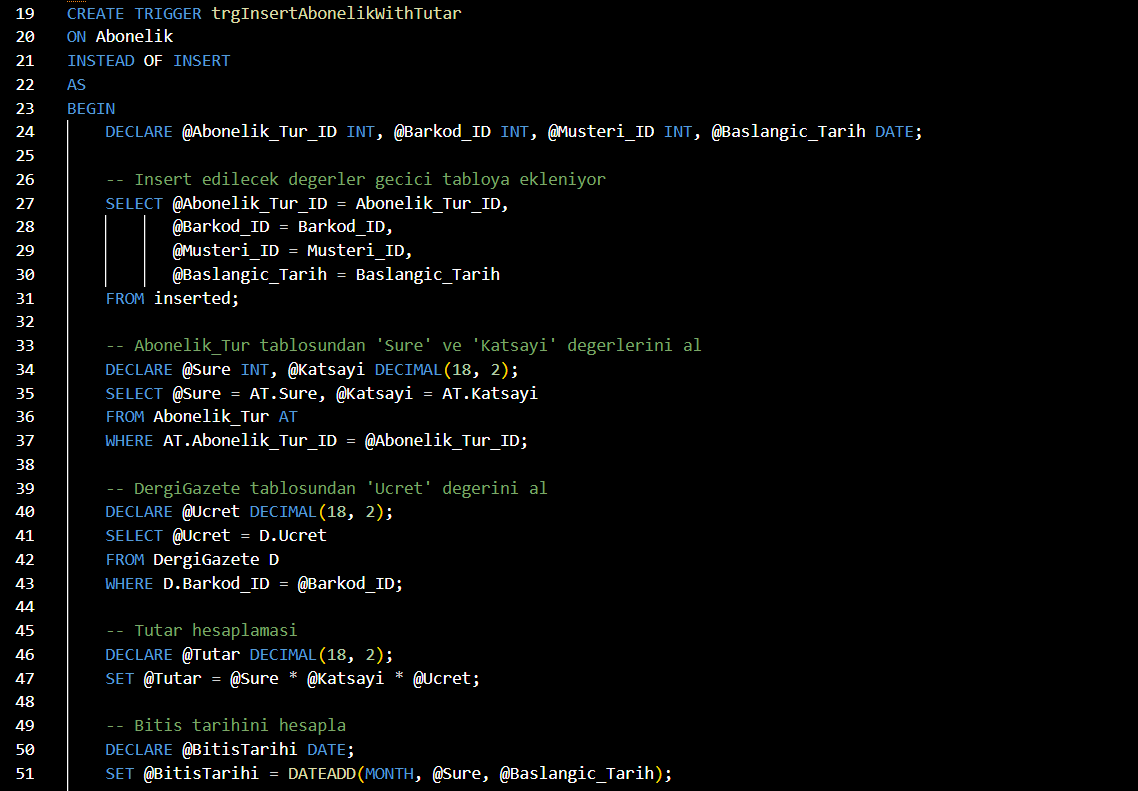
- Abonelik\_ID: Otomatik artan (IDENTITY) olarak tanımlanmış, tabloya ait bir **primary key** değeridir.

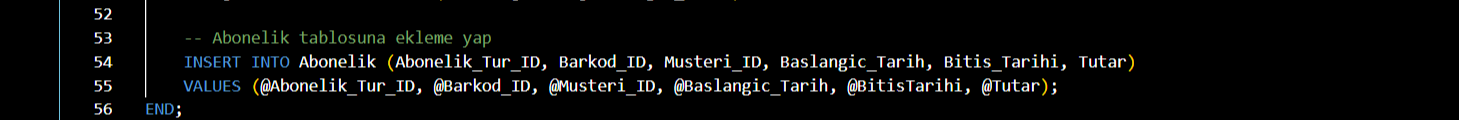
- Abonelik\_Tur\_ID, Barkod\_ID, Musteri\_ID: Diğer tablolarla ilişkili anahtar alanlardır ve **foreign key** kısıtlamaları ile bağlanmışlardır. Bu alanlar, Abonelik\_Tur, DergiGazete ve Abone tablolarındaki ilgili alanlara referans oluşturur.

- Baslangic\_Tarih, Bitis\_Tarihi, Tutar: Aboneliğin başlangıç ve bitiş tarihleri ile birlikte abonelik tutarını saklamak için tanımlanmıştır. Bazıları varsayılan değerlerle (DEFAULT) veya NULL olarak tanımlanmıştır. Ayrıca, Tutar alanı için CHECK kısıtlaması ile sıfırdan büyük olma kontrolü yapılmıştır.

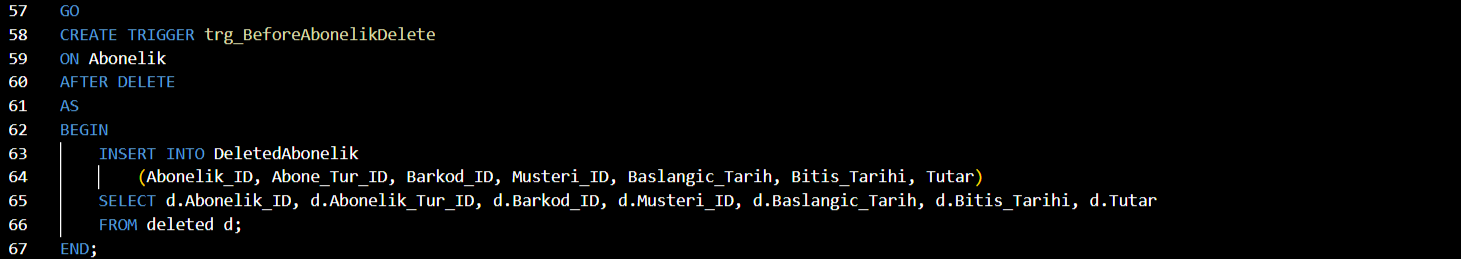
Triggerlar:

1. trgInsertAbonelikWithTutar: Bu tetikleyici, Abonelik tablosuna yeni bir kayıt eklenirken çalışır. Eklenen kaydın abonelik tutarını hesaplar ve bu bilgileri Abonelik tablosuna ekler. Abonelik süresi, katsayı ve ücret gibi bilgiler diğer tablolardan alınarak Tutar hesaplanır ve Bitis\_Tarihi belirlenir.

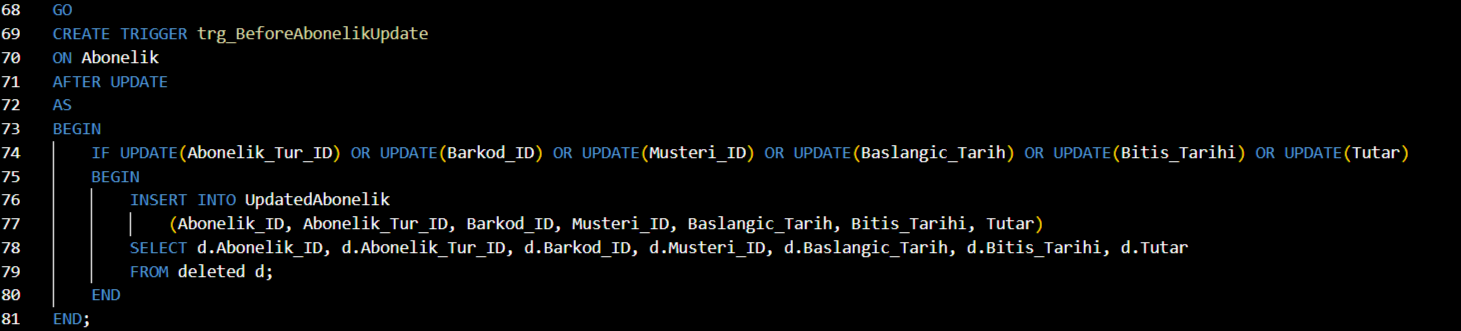




2. trg\_BeforeAbonelikDelete: Bu tetikleyici, Abonelik tablosundan bir kayıt silindiğinde çalışır. Silinen kaydın bilgilerini 'DeletedAbonelik' adlı tabloya yedekler. Bu şekilde, silinen kayıtların geçmişi tutulur.



3. trg\_BeforeAbonelikUpdate:Bu tetikleyici, Abonelik tablosunda bir kayıt güncellendiğinde çalışır. Güncellenen kaydın bilgilerini 'UpdatedAbonelik' adlı tabloya yedekler. UPDATE işlemi gerçekleştirilen alanlardaki değişiklikleri izler.



Bu tetikleyiciler, Abonelik tablosunda yapılan değişikliklerin izini sürmek ve silinen/güncellenen verileri başka tablolarda saklayarak geçmişi takip etmek için kullanılır.

Önem ve İşlev:

- Abonelik Bilgileri Saklama: Abonelik tablosu, aboneliklerle ilgili detaylı bilgileri saklamak için kullanılır.

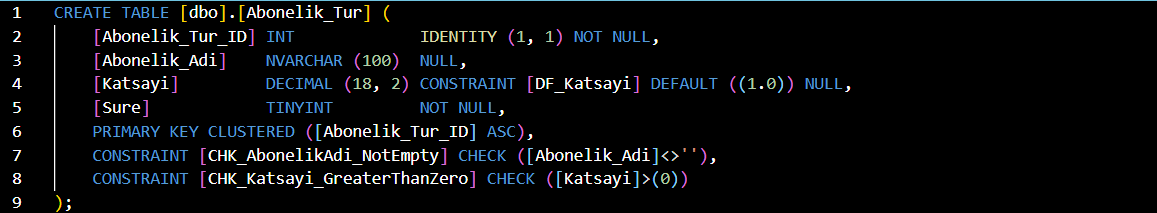
- Geçmiş Takibi: Tetikleyiciler, silinen veya güncellenen kayıtların geçmişini 'DeletedAbonelik' ve 'UpdatedAbonelik' tablolarında saklayarak, veri değişikliklerinin takibini yapar.

- Benzersiz Değer Kontrolü: Bir abonenin aynı tarih aralığında aynı abonelik için birden fazla kaydının oluşmasının önüne geçmek için kısıtlamalar eklenmiştir.

- Tutar ve Bitiş Tarihi Hesaplama: Tetikleyici, abonelik tutarını ve bitiş tarihini diğer tablolardan aldığı bilgilere göre hesaplar ve Abonelik tablosuna ekler.

## **Abonelik\_Tur Tablosu**

Abonelik\_Tur Tablosu Oluşumu:



Bu tablo, abonelik türlerini ve bu türlerle ilgili bilgileri saklamak için tasarlanmıştır. İlgili alanlar ve yapıları şu şekildedir:

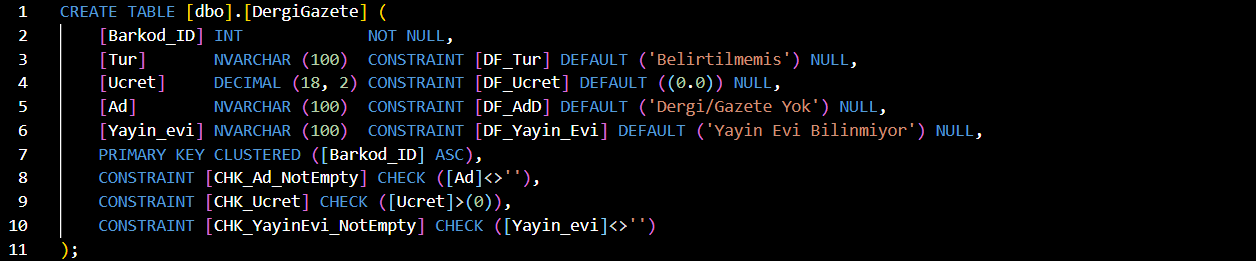
- Abonelik\_Tur\_ID:\*Otomatik artan (IDENTITY) olarak tanımlanmış, tablonun **birincil anahtarıdır**.

- Abonelik\_Adi: Abonelik türünün adını saklamak için tanımlanmıştır. Bu alan NULL olabilir, ancak bir kısıtlama ile boş bir değer alamaz.

- Katsayi: Abonelik türünün katsayı değerini saklamak için tanımlanmıştır. Bu alanın sıfırdan büyük olma kısıtlaması bulunmaktadır.

-Sure: Abonelik türünün süresini saklamak için tanımlanmıştır.

## **DergiGazete Tablosu Oluşumu**

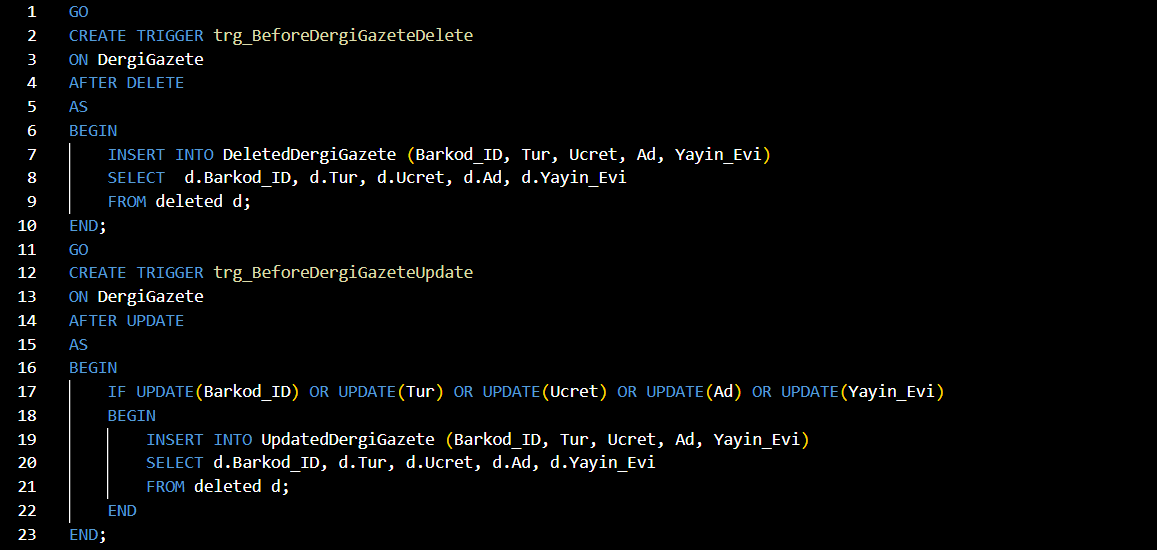


Bu tablo, dergi ve gazete bilgilerini saklamak için tasarlanmıştır. İlgili alanlar ve yapıları şu şekildedir:

- Barkod\_ID: Otomatik artan bir değer olarak tanımlanmış, tablonun birincil anahtarıdır.

- Tur, Ucret, Ad, Yayin\_evi: Dergi veya gazete türü, ücret, ad ve yayınevi bilgilerini saklamak için tanımlanmıştır. Bazı alanlar için kısıtlamalar eklenmiştir; örneğin, ücret alanı sıfırdan büyük olmalıdır.

Tetikleyiciler:



- trg\_BeforeDergiGazeteDelete ve trg\_BeforeDergiGazeteUpdate: Bu tetikleyiciler, DergiGazete tablosunda yapılan DELETE ve UPDATE işlemleri için çalışır. Silinen veya güncellenen kayıtların bilgilerini başka tablolara yedekler. Bu şekilde, geçmiş verilerin takibi sağlanır.

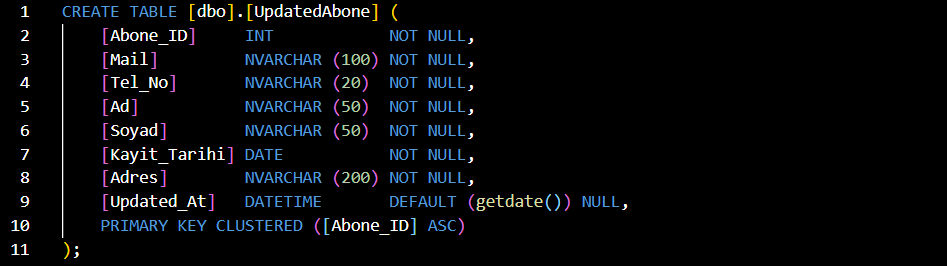
Bu tablolar ve tetikleyiciler, veri tabanı işlemlerinin gerçekleştirilmesi ve yapılan değişikliklerin izlenmesi amacıyla kullanılır. Kısıtlamalar, veri bütünlüğünü korumak ve veritabanı işlemlerini düzenli hale getirmek için eklenmiştir. Bu yapılar, veri tabanı yönetimi ve güvenliği açısından önemli bir rol oynar.

## **Belirli Türdeki Verilerin Takibi İçin Kullanılacak 'UpdatedAbone', 'UpdatedAbonelik', 'UpdatedDergiGazete', 'DeletedAbone' ve 'DeletedAbonelik' Tabloları**

Bu tablolar, veritabanındaki belirli tabloların güncellenmiş veya silinmiş verilerini takip etmek için kullanılır. Güncellenmiş veya silinmiş verilerin izini sürmek, veri bütünlüğünü korumak ve gerektiğinde geri dönüş yapabilmek için önemli bir rol oynar. Bu tür tablolar, genellikle veri takibi, denetimi ve geri dönüş işlemleri için kullanılır.

## **Updated Tabloları**

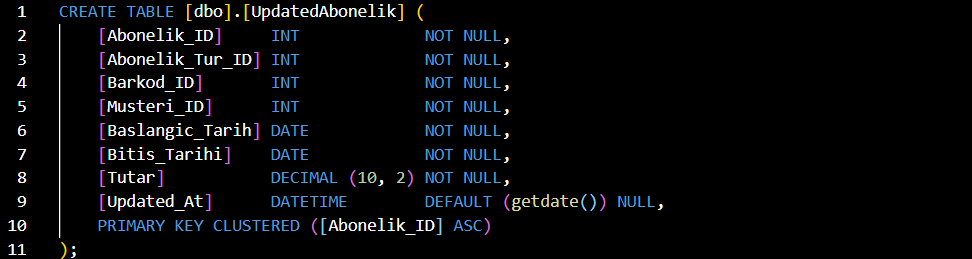
UpdatedAbone Tablosu:



- Bu tablo, 'Abone' tablosundaki güncellenmiş verilerin izini sürmek için oluşturulmuştur.

- Alanlar, Abone tablosundaki alanların bir benzeri olup, verilerin güncellendiği tarihi saklamak için 'Updated\_At' adında bir alan içerir.

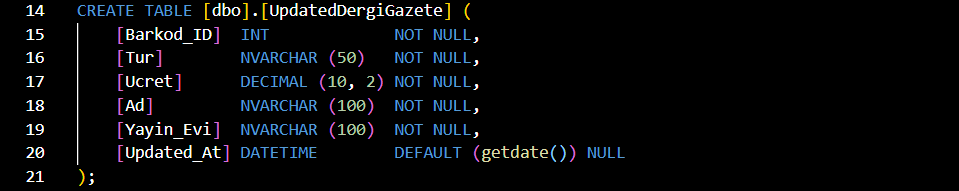
UpdatedAbonelik Tablosu:



- Bu tablo, 'Abonelik' tablosundaki güncellenmiş verilerin izini sürmek için oluşturulmuştur.

- Alanlar, Abonelik tablosundaki alanların bir benzeri olup, verilerin güncellendiği tarihi saklamak için 'Updated\_At' adında bir alan içerir.

UpdatedDergiGazete Tablosu:

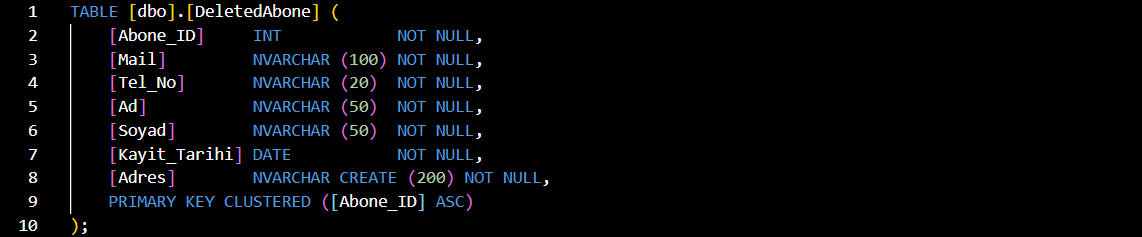


- Bu tablo, 'DergiGazete' tablosundaki verilerin güncelleştirilmeden önceki verileri triggerlar yardımıyla tutar.

- Alanlar, DergiGazete tablosundaki alanların bir benzeri olup, verilerin güncellendiği tarihi saklamak için 'Updated\_At' adında bir alan içerir.

## **Deleted Tabloları**

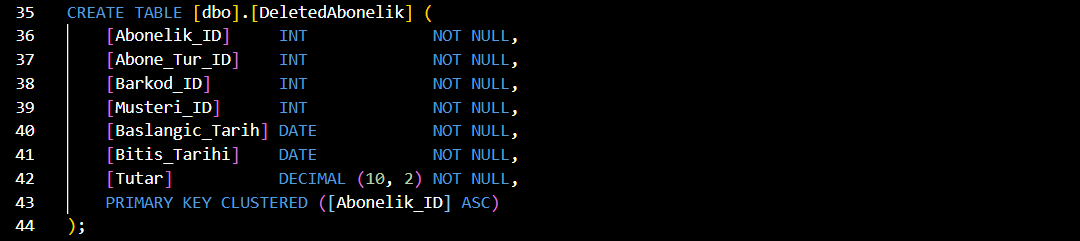
DeletedAbone Tablosu:



- Bu tablo, 'Abone' tablosundan silinen verilerin izini sürmek için oluşturulmuştur.

- Alanlar, Abone tablosundaki alanların bir benzeri olup, silinen verilerin kayıt tarihini içerir.

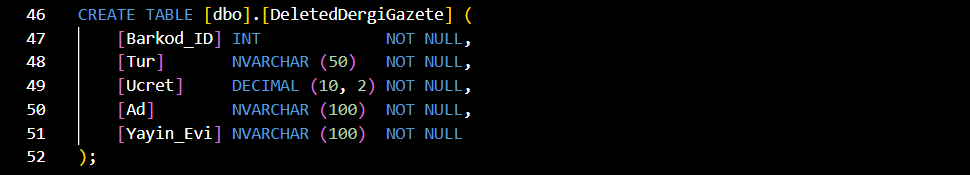
DeletedAbonelik Tablosu:



- Bu tablo, 'Abonelik' tablosundan silinen verilerin izini sürmek için oluşturulmuştur.

- Alanlar, Abonelik tablosundaki alanların bir benzeri olup, silinen verilerin kayıt tarihini içerir.

DeletedDergiGazete Tablosu:



- Bu tablo, 'DergiGazete' tablosundan silinen verilerin izini sürmek için oluşturulmuştur.

- Alanlar, DergiGazete tablosundaki alanların bir benzeri olup, silinen verilerin kayıt tarihini içerir.

.

## **B) Triggerlar**

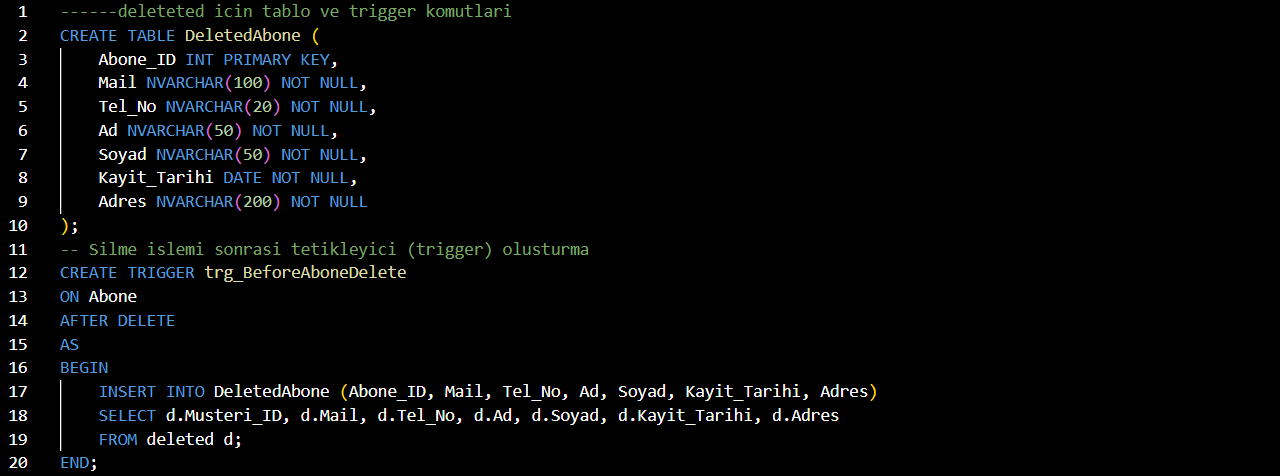
Veri tabanındaki verilerin yönetilmesini sağlayan işlemler için otomatik olarak tetiklenir. Belirli bir tabloya yapılan değişiklikleri izlemek, kontrol etmek veya başka bir işlemi tetiklemek için kullanılır. Genellikle sıralanan olaylara tepki olarak çalışırlar:

1. Insert: Yeni bir kayıt eklenirse
2. Update: Bir kayıt güncellenirse
3. Delete: Bir kayıt silinirse

Bir trigger kodunun içerisinde yer alan AFTER ifadesi belirtilen olayların gerçekleşmesinden sonra BEFORE ifadesi ise olay gerçekleşmeden önce tetikleneceğini belirtir. Bu sekmede Abone ve Abonelik tablolarındaki kodlar örnek alınarak trigger kodlarının daha detaylı bir açıklaması yapılacaktır.

## **Deleted Durumunda Abone Tablosu İçin Trigger**

Öncelikle “Abone” tablosuna ait bir kayıt silinmesi durumunda, bu silinmiş verilerin depolanabileceği bir “DeletedAbone” adlı bir tablo oluşturulur. “DeletedAbone” tablosu için tanımlanan tüm değerlerin NOT NULL olmasının nedeni silinen kayıttaki tüm sütunların kaydedilecek olmasıdır. Delete işlemi öncesi “trg\_BeforeAboneDelete” trigger’ı tetiklenir ve oluşturmuş olduğumuz “DeletedAbone” tablosuna Delete komutunun uygulandığı kaydı ekler. Böylelikle veri kaybından kaçınmış oluruz. Gerektiğinde bu verileri geri çağırabiliriz.



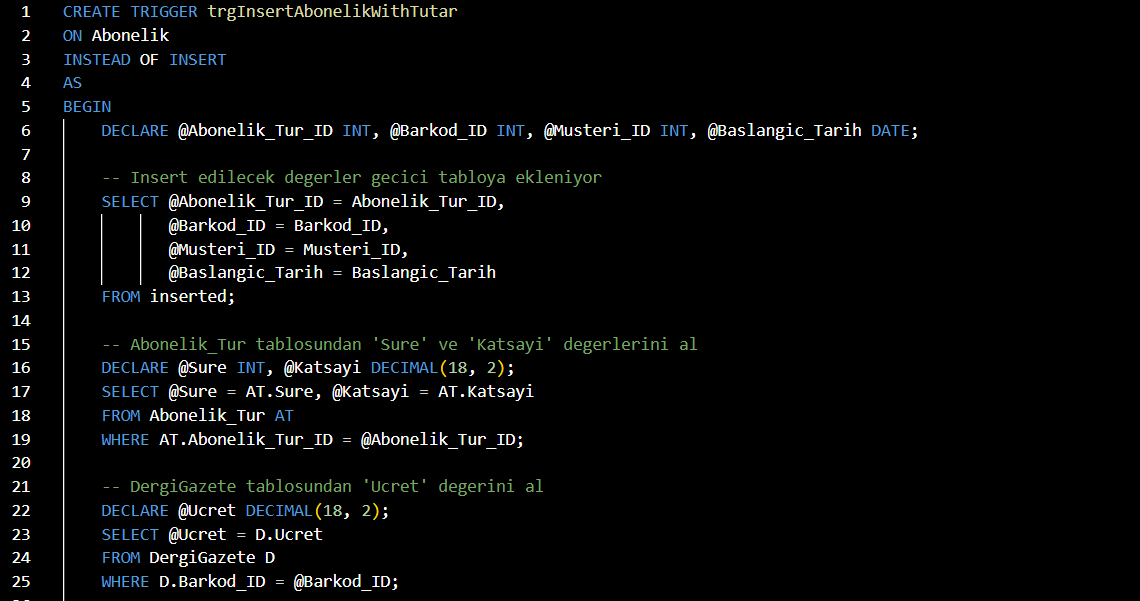
Açıklamalar:

1. CREATE TABLE DeletedAbone: “DeletedAbone” adlı tabloyu oluşturur. Bu tablo, silinen "Abone" kayıtlarının bilgilerini saklamak için kullanılacaktır. Bu tablo, silinen kaydın tüm sütunlarını içermektedir.
2. “Abone” tablosuna ait tüm veri tipleri DeletedAbone tablosunda da aynı şekilde tanımlanmıştır.
3. CREATE TRIGGER trg\_BeforeAboneDelete ON Abone AFTER DELETE: "Abone" tablosunda DELETE işlemi gerçekleştiğinde çalışacak bir tetikleyici oluşturur.
4. AS BEGIN ... END;: Tetikleyicinin içinde yer alan işlemleri tanımlar.
5. INSERT INTO Deleted Abone(): "DeletedAbone" tablosuna yeni bir kayıt ekler. Bu kayıt, "Abone" tablosundan silinen bir kaydın bilgilerini taşır.
6. SELECT d.Musteri\_ID, d.Mail, d.Tel\_No, d.Ad, d.Soyad, d.Kayit\_Tarihi, d.Adres FROM deleted d;: Silinen kaydın tüm sütunlarını içeren bilgileri seçer. DeletedAbone tablosuna eklenecek olan silinmiş “Abone” verilerini elde ettiğimiz komuttur.

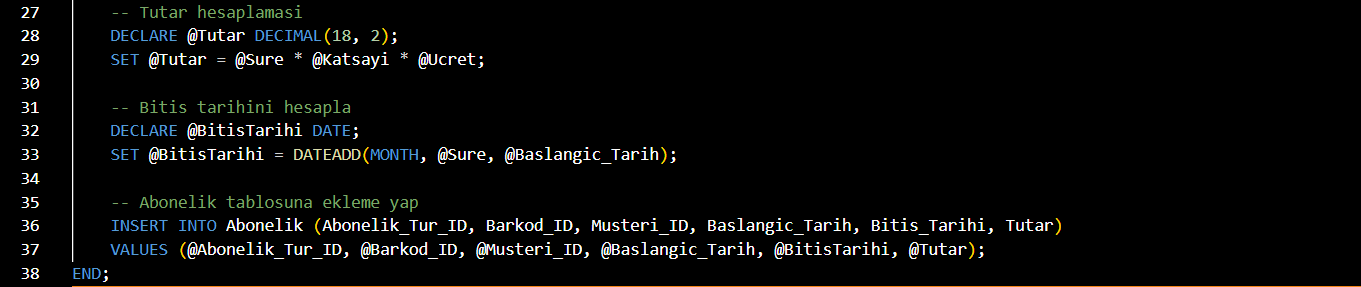
## **Abonelik Tablosuna İnstead Of Insert Tetikleyicisi**

Bu tetikleyicinin amacı “Abonelik” tablosuna yeni veriler eklerken önceden de eklenmiş olan verilere ait alanları zaman kaybı olmaması için dinamik olarak kendiliğinden girmektir. Bunun için bazı tablolardan veri çağırır.

Örneğin aşağıda ekran görüntüsü verilmiş olan tabloda Abonelik\_Tur ve DergiGazete tablolarından veriler alınacaktır. Bu veriler tutar ve bitiş tarihi hesaplama işlemlerinde kullanılacaktır.



Hesaplamaların ardından yeni veriler Abonelik tablosuna aşağıdaki gibi kaydedilir:



## **C) Sorgular**

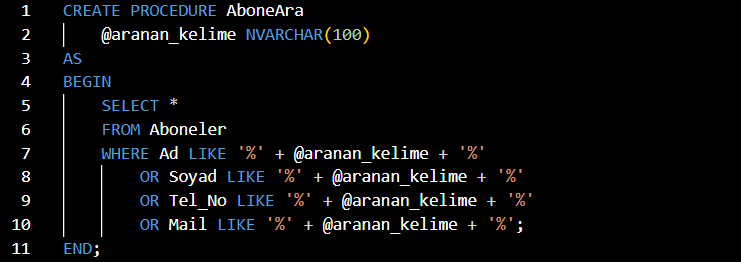
Aşağıda verilen sorgu ve procedure'ler, belirli arama kriterlerine göre veritabanı tablolarını sorgulamak ve sonuçları getirmek amacıyla tasarlanmıştır.

Kullanılacak Kodlar Hakkında Açıklamalar:

1. CREATE PROCEDURE ProcedureAdi : Bir procedure oluşturmamıza yarar ve ProcedureAdi yerine tanımlanacak olan procedure adı yazılır.
2. SELECT \*: SQL sorgularında tüm sütunları seç anlamına gelir.
3. FROM TabloAdi: SELECT \* komutunun seçeceği sütunların hangi tabloya ait olacağı TabloAdi yerine yazılır.

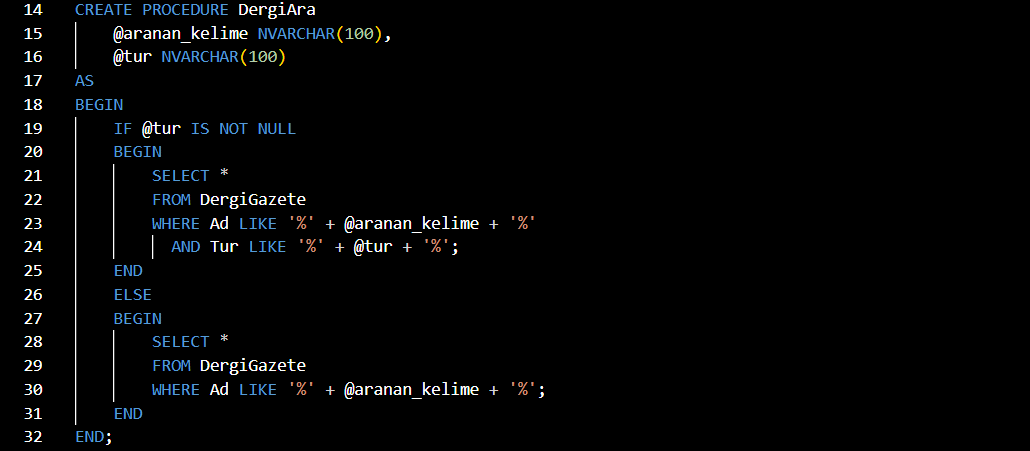
## **1) Abone Ara Procedure**

Bu procedure Aboneler tablosuna ait ad, soyad, telefon numarası ve mail adresine göre arama yapılabilmesini sağlar.



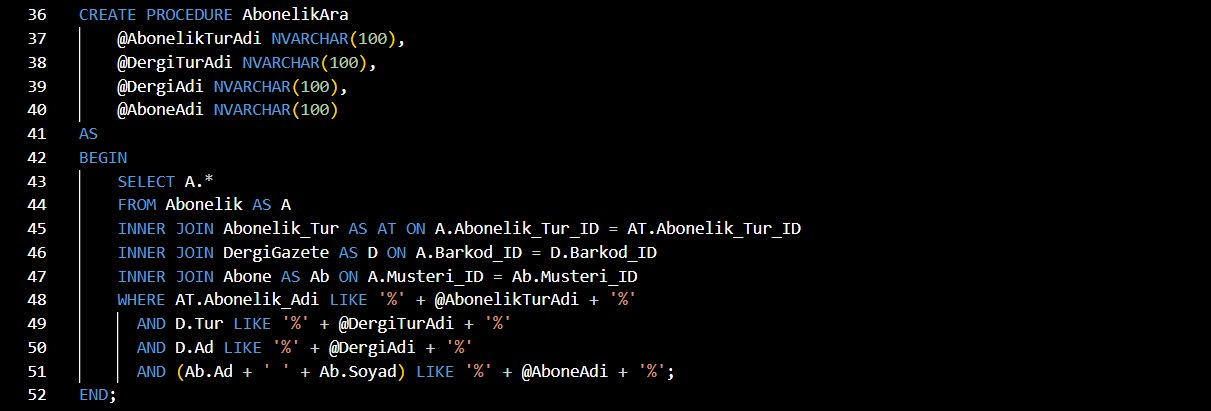
## **2) DergiAra Procedure**

Bu procedure, DergiGazete tablosunda belirli bir türe ve/veya belirli bir kelimeye göre arama yapılabilmesini sağlar.



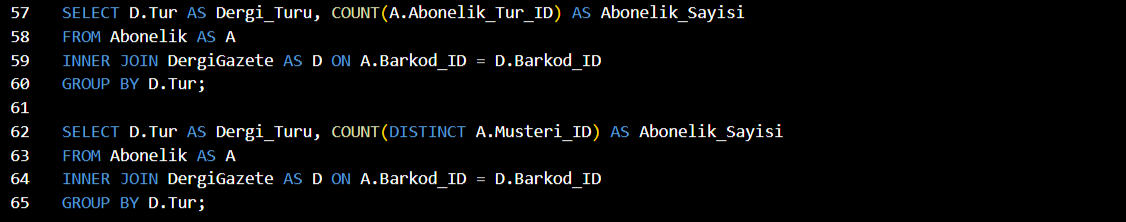
## **3) AbonelikAra Procedure**

Bu procedure Abonelik tablosundaki abonelik türü, dergi türü, dergi adı ve abone adına göre arama yapabilmeyi sağlar.



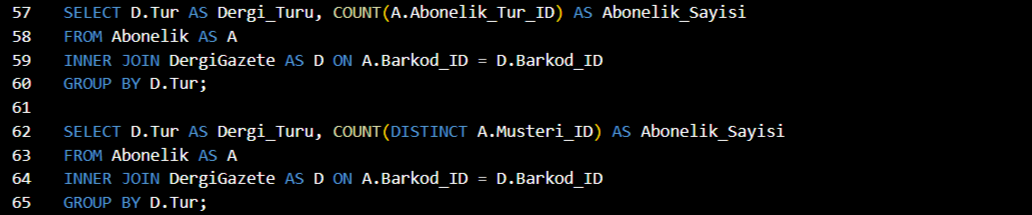
## **4) Dergi Türüne Göre Abonelik Sayıları İçin Sorgu**

Bu sorgu her bir dergi türüne ait toplam abonelik sayısını gösterir

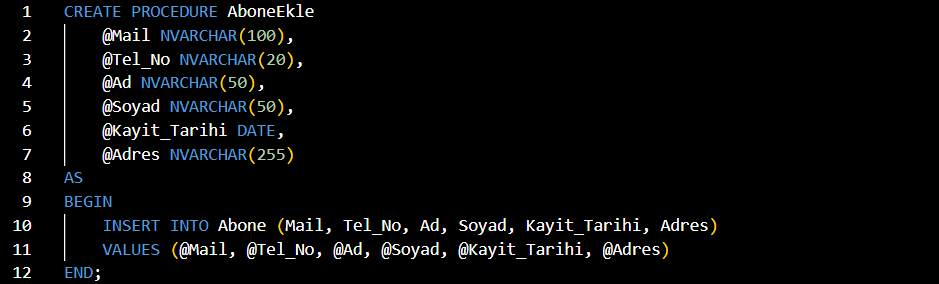


## **5) Dergi Türüne Göre Farklı Abonelerin Sayısı İçin Sorgu**

Bu sorgu, her bir dergi türüne ait farklı abonelerin sayısını gösterir. Bir müşteri birden çok dergiye veya gazeteye abone olabilir. Her bir abonenin sadece bir kere sayılması adına “**COUNT(DİSTİNCT A.Musteri\_ID)**  **AS Abonelik\_Sayisi FROM Abonelik**” kod satırı kullanılmıştır.



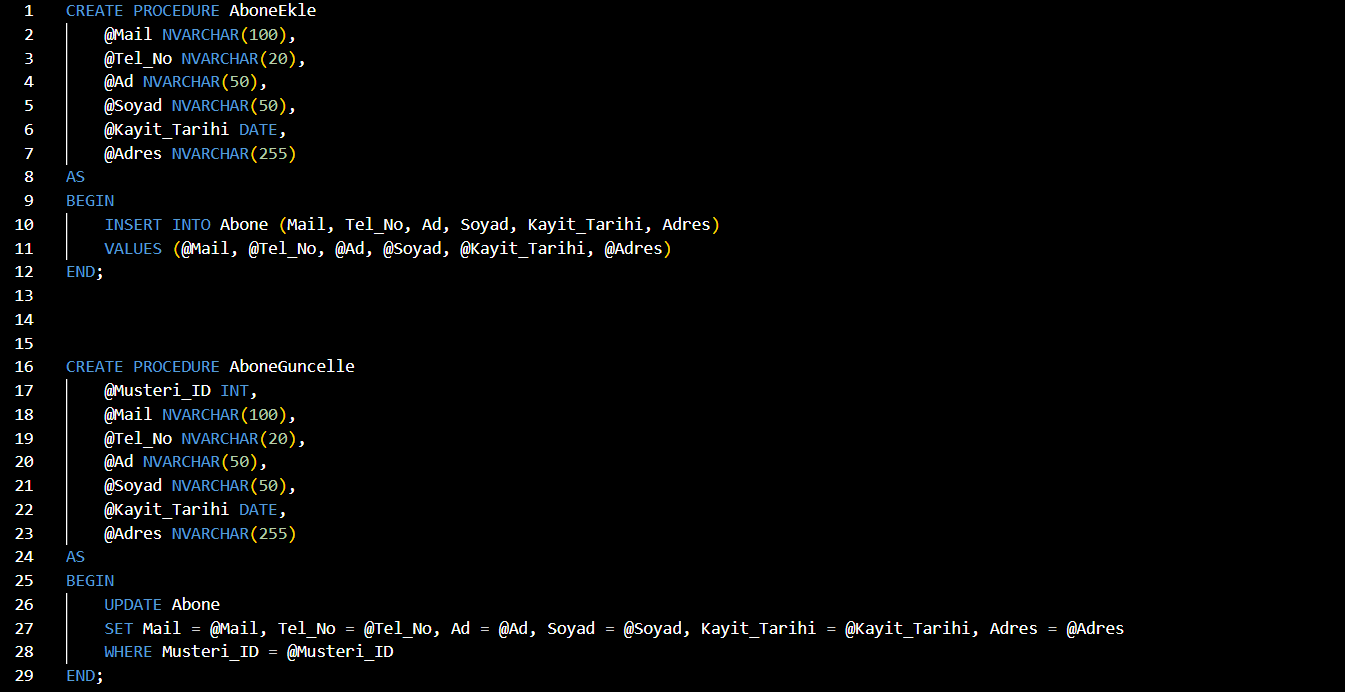
## **6) Abone Ekle Procedure**



Bu saklı prosedür, Abone tablosuna yeni kayıt eklemek için tasarlanmıştır.

- 'Mail', 'Tel\_No', 'Ad', 'Soyad', 'Kayit\_Tarihi' ve 'Adres' parametrelerini alarak yeni bir Abone kaydı ekler.

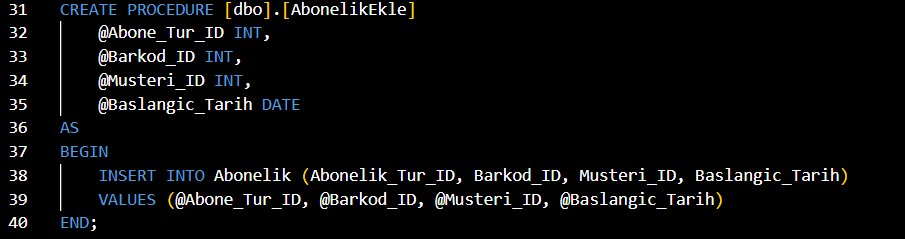
## **7) Abone Güncelle Procedure**



- Bu saklı prosedür, Abone tablosundaki bir kaydı güncellemek için tasarlanmıştır.

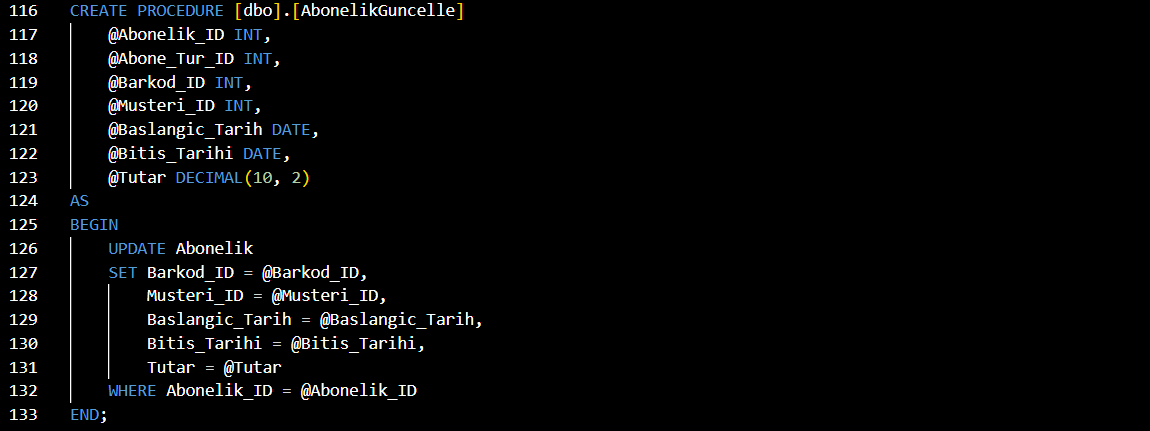
- 'Musteri\_ID' ile belirtilen Abone kaydının 'Mail', 'Tel\_No', 'Ad', 'Soyad', 'Kayit\_Tarihi' ve 'Adres' alanlarını günceller.

## **8) Abonelik Ekle Procedure**



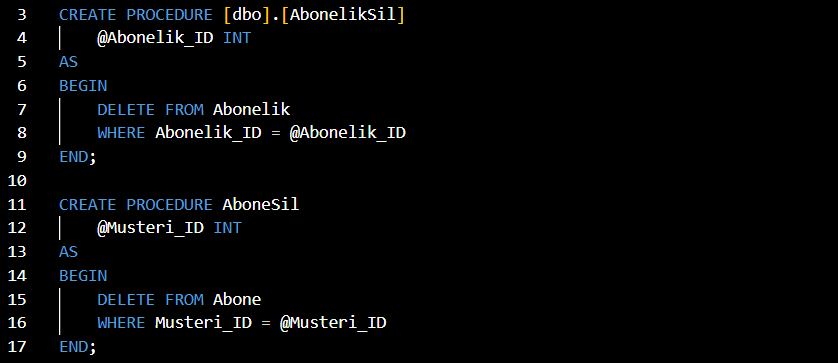
- Yeni bir abonelik eklemek için kullanılır. Parametreler aracılığıyla Abonelik tablosuna yeni bir kayıt ekler.

## **9) Abonelik Güncelle Procedure**



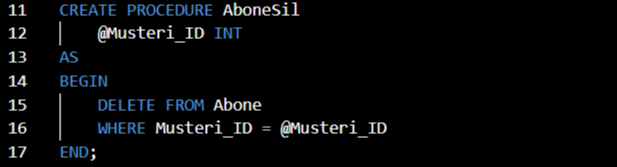
- Mevcut bir aboneliği güncellemek için kullanılır. Belirtilen Abonelik\_ID'ye sahip bir abonelik kaydının değerlerini günceller.

## **10) Abonelik Sil Procedure**



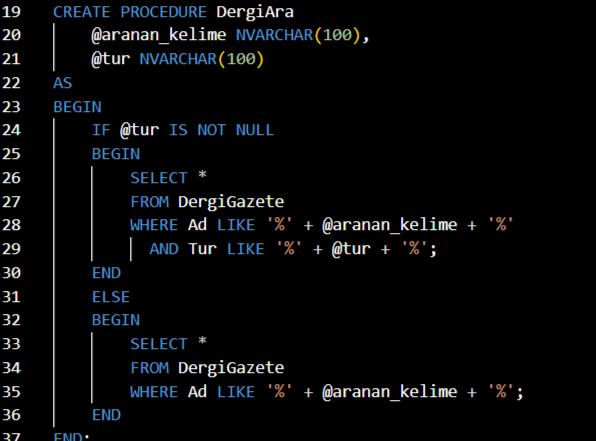
Belirtilen Abonelik\_ID'ye sahip bir abonelik kaydını silmek için kullanılır.

## **11) Abone Sil Procedure**



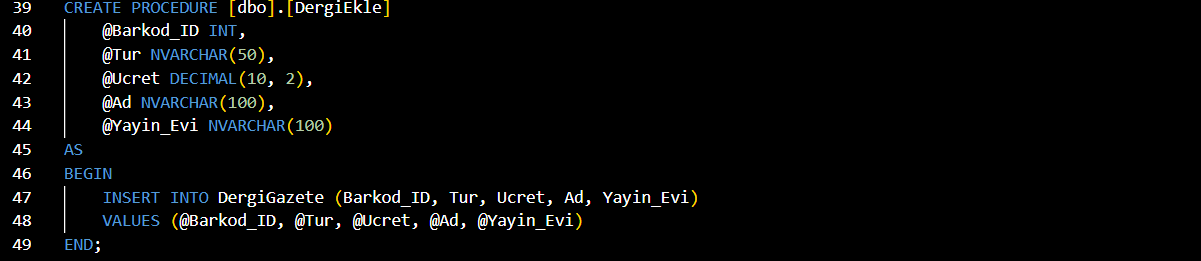
Belirtilen Musteri\_ID'ye sahip bir abone kaydını silmek için kullanılır.

## **12) DergiAra Procedure**



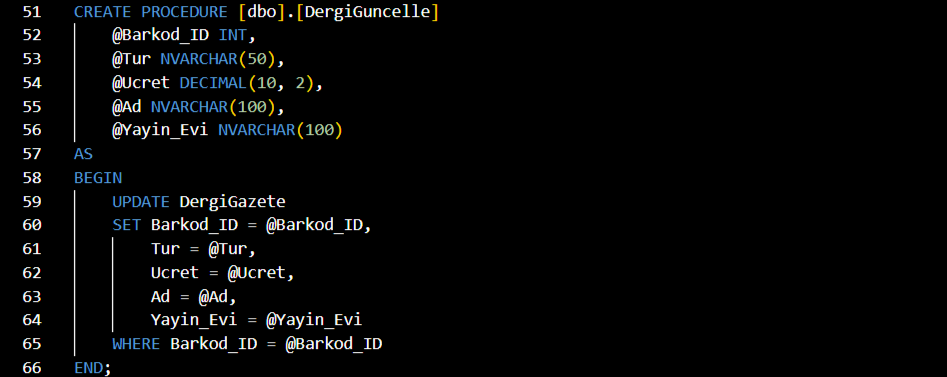
DergiGazete tablosunda veri aramak için kullanılır. Verilen kelimelere göre 'Ad' ve 'Tur' alanlarında arama yapar

## **13) DergiEkle Procedure**



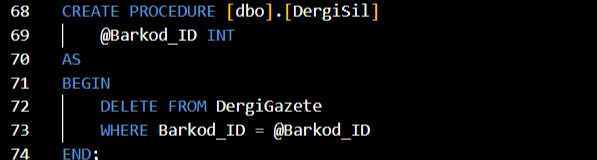
Yeni bir dergi veya gazete kaydı eklemek için kullanılır.

## **14) DergiGuncelle Procedure**



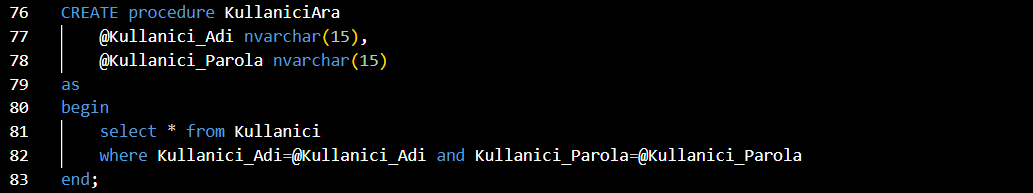
Varolan bir dergi veya gazete kaydını güncellemek için kullanılır.

## **15) DergiSil Procedure**



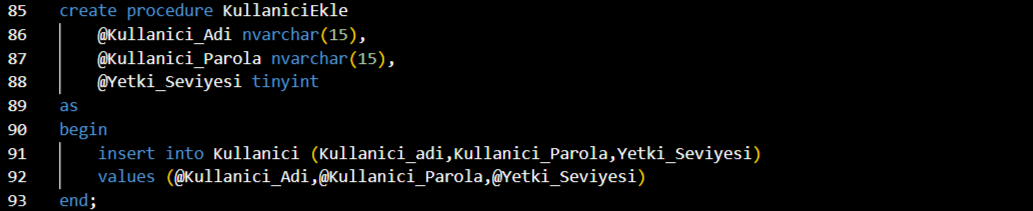
Belirtilen Barkod\_ID'ye sahip bir dergi veya gazete kaydını silmek için kullanılır.

## **16)KullaniciAra Procedure**



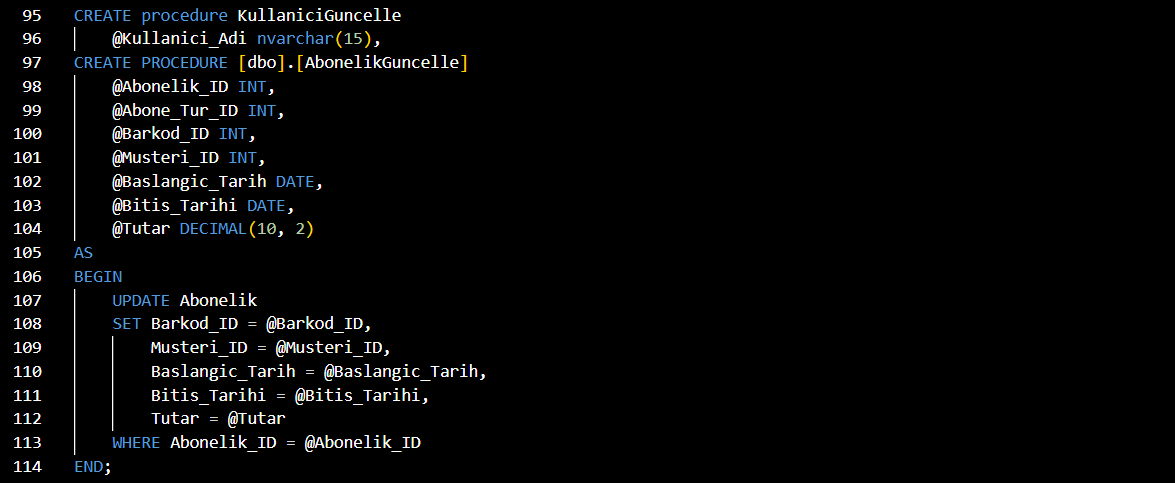
Kullanıcı adı ve şifreye göre Kullanici tablosunda arama yapmak için kullanılır.

## **17)KullaniciEkle Procedure**



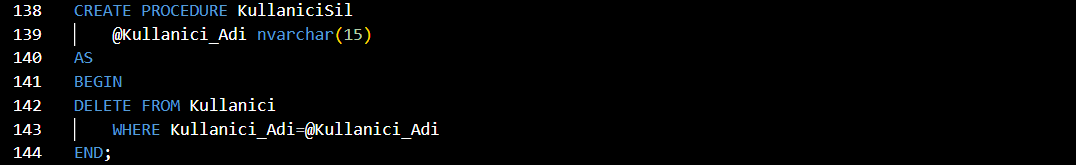
Yeni bir kullanıcı eklemek için kullanılır.

## **18) KullaniciGuncelle Procedure**



Varolan bir kullanıcı kaydını güncellemek için kullanılır.

## **19)KullaniciSil Procedure**

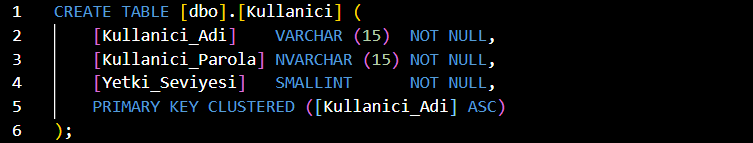


Belirtilen Kullanici\_Adi'ye sahip bir kullanıcı kaydını silmek için kullanılır.

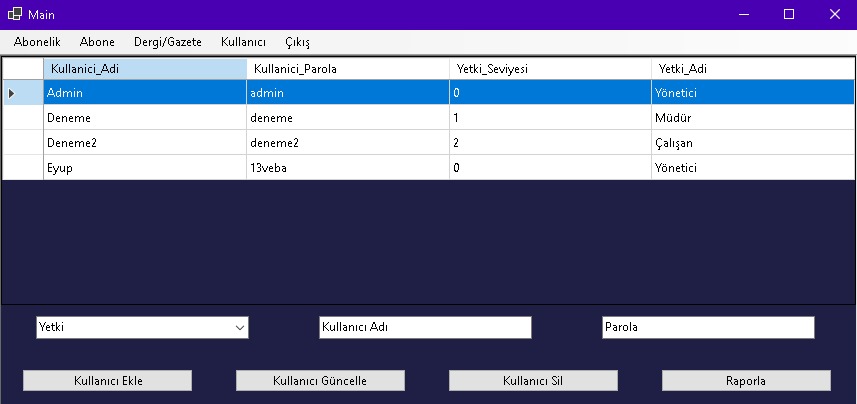
# **Kullanıcı Adı ve Şifre İle Güvenlik Kontrolü**

Kullanıcıların belirli veritabanı kaynaklarına erişimini kontrol etmek için kullanıcı adı ve şifre ile güvenlik kontrolü yapılır.

## **Kullanıcı Tablosu**



## **Kullanıcı Tablosundaki Yetkilendirme Sistemi**



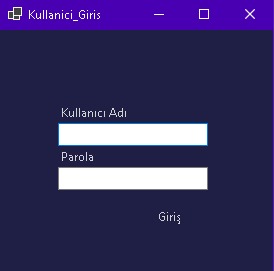
Yetki seviyeleri sırasıyla en kapsamlıdan en aza olacak şekilde “0,1 ve 2” olmak üzere tasarlanmıştır.

Yetki Seviyesi 0 (Sistem Yöneticisi Admin): En fazla yetkiye sahip olan kullanıcıdr. Veri tabanı nesneleri oluşturma, veri tabanı düzenleme, güncelleme, silme; yeni kullanıcı hesapları oluşturma veya var olan hesapları silme, kullanıcılara yetki seviyesi atama veya değiştirme gibi kapsamlı görevleri yerine getirebilir.

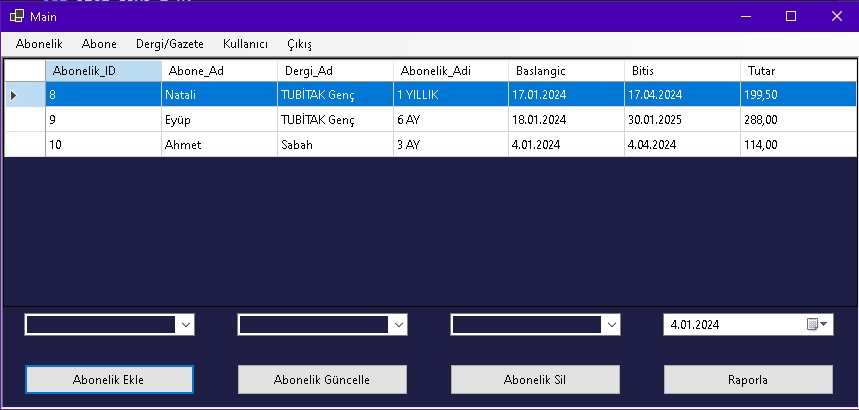
Yetki Seviyesi 1 (Müdür): Veri ekleyebilme (INSERT) ve veri silebilme (DELETE) yetkilerine sahiptir.

Yetki Seviyesi 2: Yalnızca veri okuma yetkisine sahiptir.

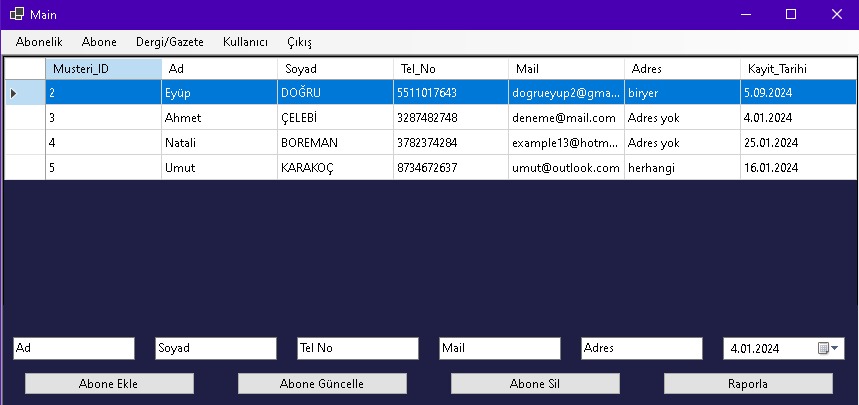
# **KULLANICI GİRİŞ EKRANI VE ARAYÜZÜ**



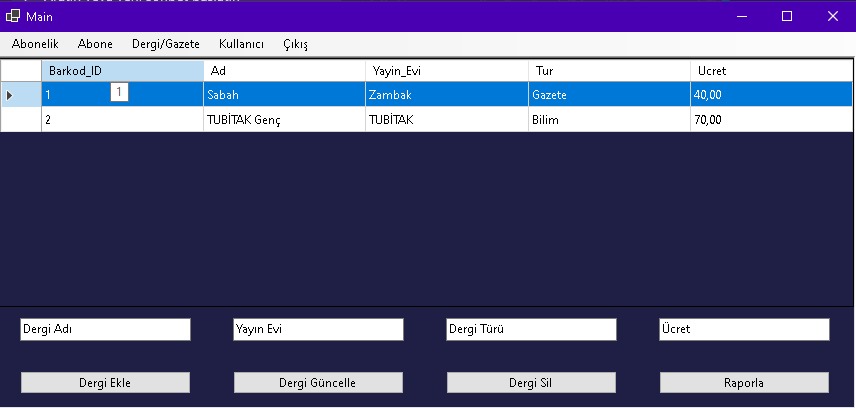
Kullanıcı, kullanıcı adını ve parolasını girerek arayüz ekranına erişim sağlar.



Kullanıcı yukarıdaki ekran görüntüsünde de örneği verildiği üzere tablolar üzerinde “Abone Ekle”, “Abonelik Güncelle”, “Abonelik Sil”, “Raporla” butonlarını kullanarak veri tabanı üzerinde yetki seviyesine göre, bu arayüzden kullanıcı tarafından işlem gerçekleştirilebilir.



Sayfanın en üst satırında bulunan “Abonelik”, “Abone”, “Dergi/Gazete”, “Kullanıcı” butonlarıyla işlem yapmak istediği tablolar arası geçiş yapabilir. Yukarıdaki ekran görüntüsünde Abone tablosuyla ilgili sayfada bulunduğumuzu görüyorsunuz.

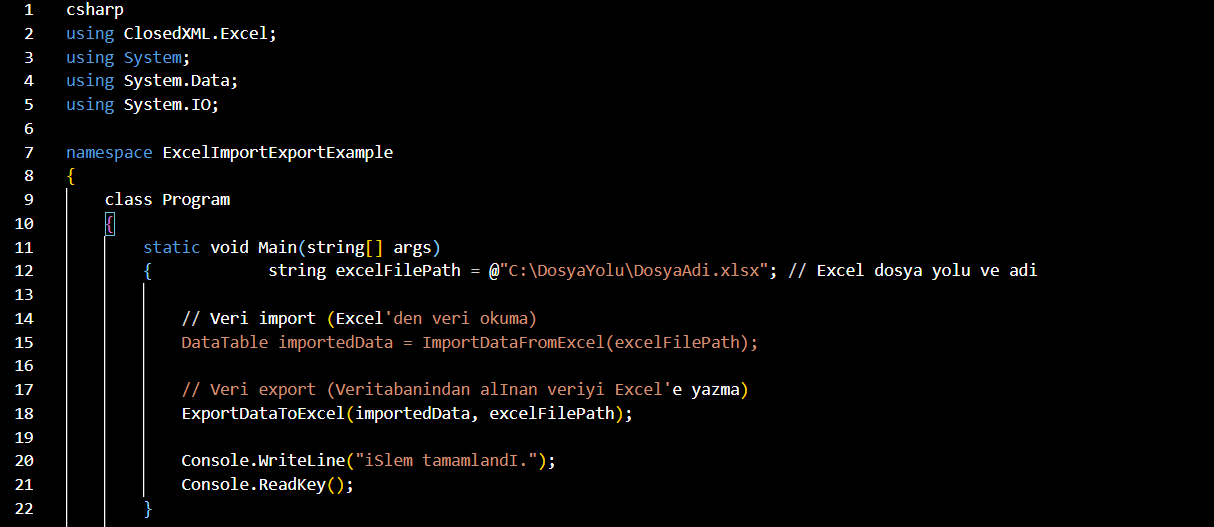


Dergi/Gazete tablosuna ait kullanıcı arayüzü yukarıdaki ekran görüntüsünde göründüğü şekildedir.

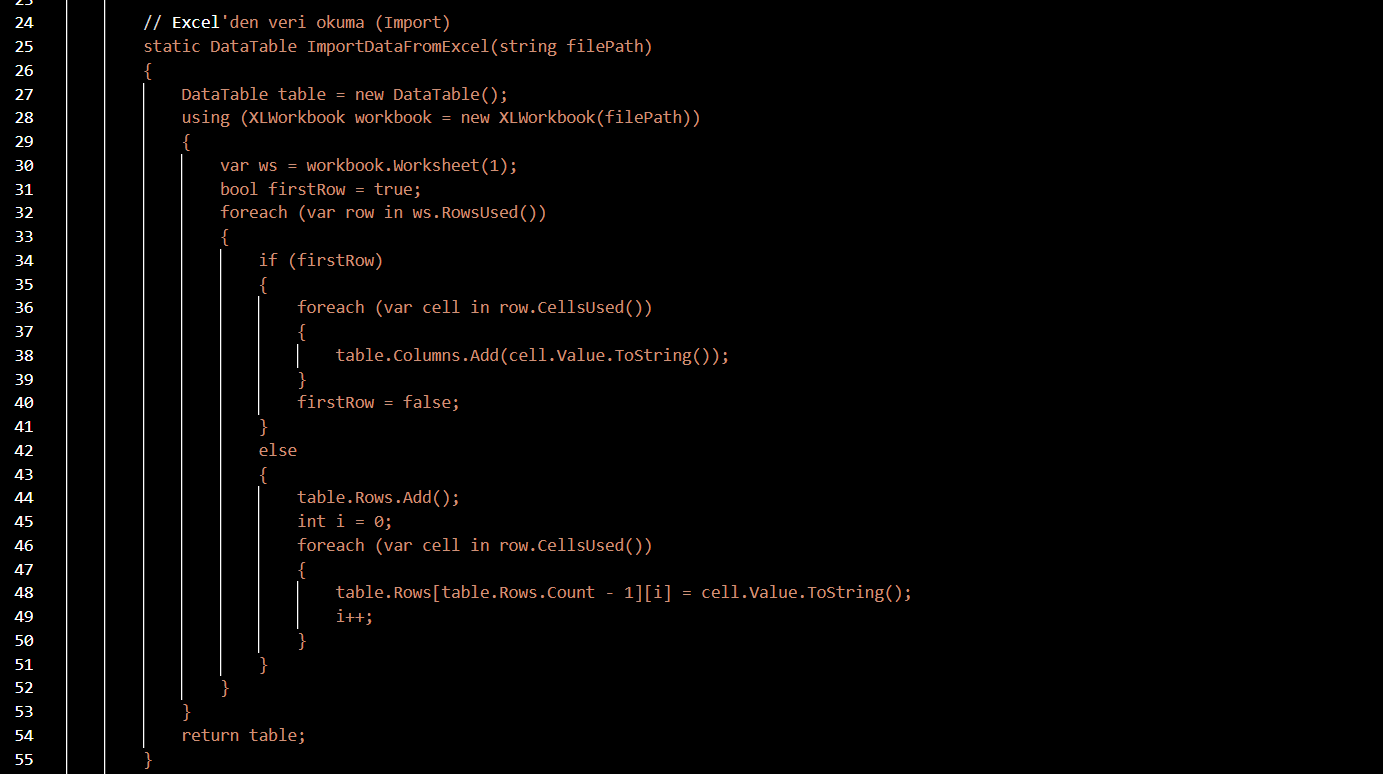
# 

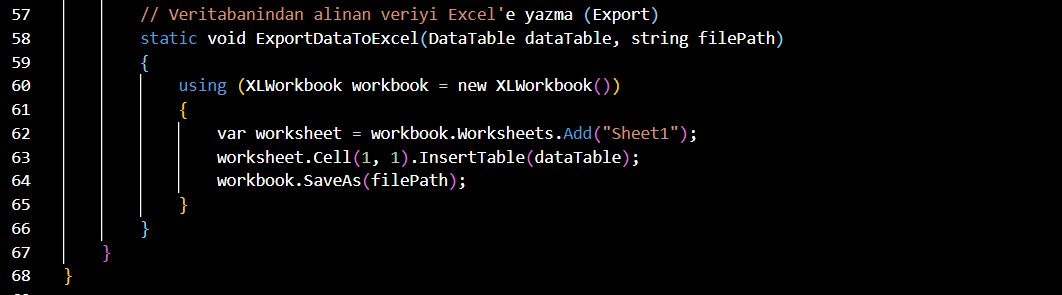
# **VERİ IMPORT/EXPORT UYGULAMASI**

Yazılan ClosedXML kütüphanesini kullanarak Excel'den veri okuma ve veriyi Excel'e yazma işlemlerini gerçekleştirir.



* ImportDataFromExcel metodunda, XLWorkbook kullanılarak Excel dosyası okunur.
* İlk satır başlık (sütun adları) olarak alınır ve DataTable'a sütunlar eklenir.
* Sonraki satırlar okunur ve DataTable'a satırlar eklenir.





* ExportDataToExcel metodunda, XLWorkbook oluşturulur ve bir çalışma sayfası eklenir.
* DataTable, çalışma sayfasına tablo olarak eklenir ve Excel dosyası olarak kaydedilir.
* Main metodunda, Excel dosyasının yolu belirlenir (excelFilePath).
* ImportDataFromExcel metodunu kullanarak Excel'den veri okunur ve bir DataTable'a atanır.
* ExportDataToExcel metodunu kullanarak alınan veri, aynı Excel dosyasına yazılır.