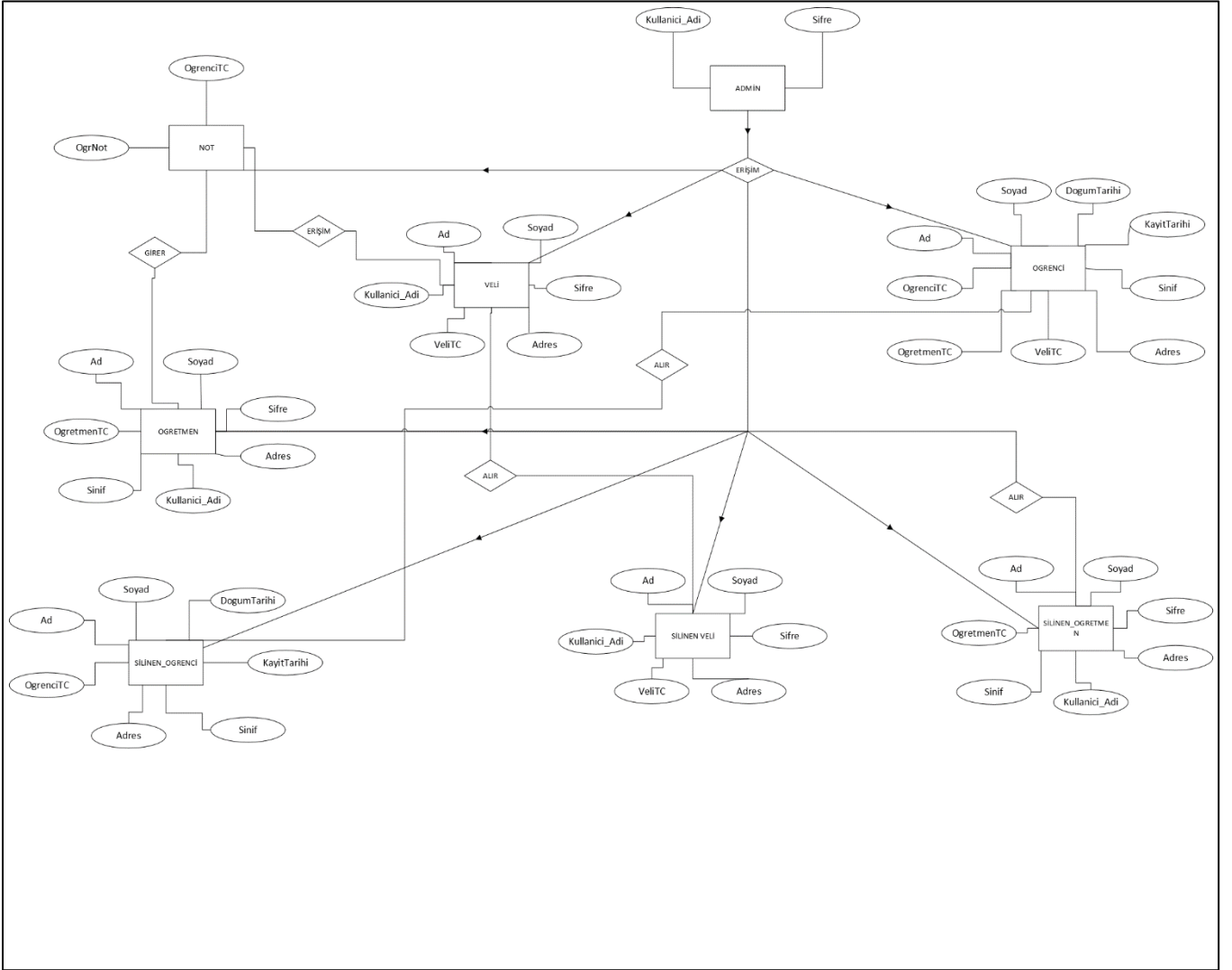


**2022-2023 AKADEMİK YILI
GÜZ DÖNEMİ****BM-309 Veritabanı Yönetim Sistemleri****Ders Sorumlusu:****Dr. Öğr. Üyesi Fatih KAYAALP**

**Anaokulu/kreşin kayıtlarını tutacak olan bir veritabanı
projesi.**

Hazırlayanlar:**Enes Emre KOÇAK- 191001025****Furkan TAŞER- 201001087****Muhammed ÖZGİRAY- 211002331****Yiğit GÖKALP- 191001055**

Projemiz bir anaokulunda veli, öğrenci ve öğretmenlerin her türlü verilerini tutacak bir projedir. Projemiz C# form uygulaması ile yazılmış olup veritabanı SQL ile yapılmıştır. Projemiz ER diyagramı şekil-1’de verilmiştir.



Veri tabanımızda 4 adet birbirleri ile çeşitli bağlantılı ve birbiri ile ilişkisiz 4 tablo, toplam 8 tablo bulunmaktadır. Bunlar şu şekildedir.

1) İlişkili Tablolar

a) Notlar

- Notlar ve öğrenciler tablosu arasında Foreign Key bağlantısı oluşturulmuştur ve öğrenciler tablosundan notlar tablosuna doğru 1- ∞ yani bir öğrencinin birden fazla notu olması amaçlanmıştır.
- Notlar tablosu bileşenleri: OgrenciTC ve ogrNot.

b) Öğrenciler

- Öğrenciler tablosu ana tablo olarak seçilmiş ve bütün hareketler öğrenci tablosuna bağlı olarak planlanmıştır.
- Öğrenciler tablosu bileşenleri: TC, Ad, Soyad, DogumTarihi, KayitTarihi, Sinif, Adres, VeliTC, OgretmenTC.

c) Öğretmenler

- Öğretmenler ve Öğrenci tablosu arasında Foreign Key bağlantısı oluşturulmuştur ve öğretmenler tablosundan öğretmen tablosuna doğru 1- ∞ yani bir öğretmenin birden fazla öğrencisi olması amaçlanmıştır.
- Öğretmenler tablosu bileşenleri: OgretmenTC, Ad, Soyad, Sinif, Kullanici_Adi, Sifre, Adres.

d) Veli

- Veli ve Öğrenci tablosu arasında Foreign Key bağlantısı oluşturulmuştur ve veliler tablosundan öğretmen tablosuna doğru 1- ∞ yani bir velinin birden fazla öğrencisi olması amaçlanmıştır.
- Veli tablosu bileşenleri: VeliTC, Ad, Soyad, Kullanici_Adi, Sifre, Adres.

2) İlişkisiz Tablolar

a) Admin

b) Silinen öğrenciler

c) Silinen Öğretmenler

d) Silinen Veliler

İlişkisiz tablolarda gösterilen b, c ve d seçeneği ödev föyünde verilen senaryonun 7. maddesi gereği ilişkilendirme gereksinimi duyulmamıştır. Admin tablosu ile program üstünde tam yetkilendirme ile yetkilendirilmiştir.

Projede Kullanılan bazı SQL örnek kodları

Aşağıda verilen kod bloğunda veri tabanı özellikleri gösterilmiştir.

```
CREATE DATABASE [AnaOkulu]
  CONTAINMENT = NONE
  ON PRIMARY
( NAME = N'AnaOkulu', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\AnaOkulu.mdf' , SIZE = 8192KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )
  LOG ON
( NAME = N'AnaOkulu_log', FILENAME = N'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\AnaOkulu_log.ldf' , SIZE = 8192KB ,
MAXSIZE = 2048GB , FILEGROWTH = 65536KB )
  WITH CATALOG_COLLATION = DATABASE_DEFAULT
```

Aşağıda verilen kod bloğu ile Öğrenciler tablosu oluşturulmuştur.

```
CREATE TABLE [dbo].[Ogrenci] (
  [TC] [varchar](11) NOT NULL,
  [Ad] [varchar](30) NOT NULL,
  [Soyad] [varchar](30) NOT NULL,
  [DogumTarihi] [date] NOT NULL,
  [KayitTarihi] [date] NOT NULL,
  [Sinif] [varchar](3) NULL,
  [Adres] [varchar](150) NOT NULL,
  [VeliTC] [varchar](11) NULL,
  [OgretmenTC] [varchar](11) NULL,
  PRIMARY KEY CLUSTERED
(
  [TC] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY =
OFF) ON [PRIMARY]
) ON [PRIMARY]
```

Aşağıda verilen kod bloğu ile Notlar ve Öğrenci tabloları arasında Foreign Key bağlantısı yapılmıştır.

```
ALTER TABLE [dbo].[Notlar] WITH CHECK ADD CONSTRAINT [FK_Notlar_Ogrenci]
FOREIGN KEY([OgrenciTC])
REFERENCES [dbo].[Ogrenci] ([TC])
```

Aşağıda verilen kod bloğunda stored procedure kullanılmıştır. Bu kod sayesinde veli paneline giriş yapıldığında veli ad soyadı ile öğrenci soyadı eşitlenerek velinin öğrencileri kendi sayfasında yer almasını sağlanıyor.

```
create proc [dbo].[VeliProc]
as
begin

Select o.TC,o.Ad,o.Soyad,n.OgrNot from Notlar n
Inner Join Ogrenci o on n.OgrenciTC =o.TC
Inner Join Veli v on o.Soyad=v.Soyad
end
```

Aşağıda verilen kod bloğunda stored procedure kullanılmıştır. Bu kod bloğu sayesinde seçilen sınıf şubesine göre öğrenciler listelenmektedir.

```
create proc [dbo].[SinifB]
as
begin

select TC,Ad,Soyad,n.OgrNot from Ogrenci o Left Outer Join Notlar n on
o.TC=n.OgrenciTC where Sinif like 'B'

end
```

Aşağıda verilen kod bloğunda herhangi bir silme (delete) işlemi sonrasında trigger tetiklenerek ilgili veriyi silinenler tablosuna taşımaktadır.

```
CREATE TRIGGER tr_SilinenVeli ON Veli
AFTER DELETE
AS
BEGIN
DECLARE @VeliTC varchar(11)
DECLARE @Ad varchar(20)
DECLARE @Soyad varchar(20)
DECLARE @Kullanici_Adi varchar(30)
DECLARE @Sifre varchar(20)
DECLARE @Adres varchar(150)

SELECT @VeliTC=VeliTC FROM deleted
SELECT @Ad=Ad FROM deleted
SELECT @Soyad=Soyad FROM deleted
SELECT @Kullanici_Adi=Kullanici_Adi FROM deleted
SELECT @Sifre=Sifre FROM deleted
SELECT @Adres=Adres FROM deleted

INSERT INTO SilinenVeli VALUES
(@VeliTC,@Ad,@Soyad,@Kullanici_Adi,@Sifre,@Adres)

END
```

Kullanıcı Arayüzü Tasarımı

