



PROJENİN ADI: DEPREMZEDE KONUT TAKİP SİSTEMİ

PROJE EKİBİ: KEREM BERKE BAŞAK-235260154

MELİH KERİMGİL-205260012

MUHAMMED TALHA TOPÇU-225260120

PROJE TANIMI VE AMACI :

Depremzede Konut Takip Sistemi projesi, depremzedelere sađlanan konutların takibini ve yönetimini kolaylaştırmak için tasarlanmıştır. Bu sistem, depremzedelerin başvuru süreçlerini, konut tahsisatlarını, konutların durumunu ve ilgili raporları merkezi bir veritabanında takip eder. Projenin amacı, konut başvurularını, tahsisatlarını ve personel görevlerini etkin bir şekilde yönetmek ve gerektiğinde hızlı erişim sađlamak suretiyle depremzedelere yönelik hizmetlerin daha verimli ve şeffaf hale gelmesini sađlamaktır. Bu sayede, kullanıcılar arasındaki koordinasyonu güçlendirerek, afet sonrası hızlı müdahale ve yardım süreçlerinin iyileştirilmesi hedeflenmektedir.

PROJE İÇİN KULANILAN TABLOLAR:

Bu sistemde aşağıdaki tablolar ve her birine ait alanlar tanımlanmıştır. Her tablo, veritabanındaki farklı bir işlevi veya süreci temsil eder.

1. Adres Tablosu

AdresID (Primary Key)

DepremzedeID (Foreign Key -> Depremzede)

EskiAdres

YeniAdres

2. Depremzede Tablosu

DepremzedeID (Primary Key)

Ad

Soyad

TC_KimlikNo

DogumTarihi

3. İletişim Tablosu

İletişimID (Primary Key)

DepremzedeID (Foreign Key -> Depremzede)

Telefon

Email

4. Konut Tablosu

KonutID (Primary Key)

Adres

Metrekare

OdaSayisi

KonutTuruID (Foreign Key -> Konut_Turu)

DurumID (Foreign Key -> Konut_Durumu)

5. Konut_Durumu Tablosu

DurumID (Primary Key)

Durum

6. Konut_Sahipligi Tablosu

KonutSahipligiID (Primary Key)

KonutID (Foreign Key -> Konut)

SahipID (Foreign Key -> Depremzede)

7. Konut_Turu Tablosu

KonutTuruID (Primary Key)

Turu

8. Tahsis Tablosu

TahsisID (Primary Key)

DepremzedeID (Foreign Key -> Depremzede)

KonutID (Foreign Key -> Konut)

TahsisTarihi

9. Tahsis_Tarihi Tablosu

TahsisTarihiID (Primary Key)

KonutID (Foreign Key ->Konut)

TahsisTarihi

İLİŞKİSEL KÜME FORMATI:

Depremzede (DepremzedeID, Ad, Soyad, TC_KimlikNo, DogumTarihi, Cinsiyet, Telefon, Email)

Adres (AdresID, DepremzedeID, EskiAdres, YeniAdres, Mahalle, Ilce, Il)

Adres. DepremzedeID ->Depremzede. DepremzedeID

Konut (KonutID, KonutTuru, KonutMetreKare, OdaSayisi, Durum, AdresID)

Konut. AdresID -> Adres. AdresID

Konut_Sahipligi (KonutSahipligiID, KonutID, DepremzedelD)

Konut_Sahipligi. KonutID ->Konut. KonutID

Konut_Sahipligi. DepremzedelD ->Depremzede. DepremzedelD

Tahsis (TahsisID, KonutID, DepremzedelD, TahsisTarihi, Durum)

Tahsis. KonutID ->Konut. KonutID

Tahsis. DepremzedelD ->Depremzede. DepremzedelD

Konut_Durumu (DurumID, Durum)

Konut_Turu (TurID, TurAdı)

İlişkisel Kümelerin Özeti:

Depremzede kümesinde her depremzede bir veya daha fazla Adres kaydına sahip olabilir.

Adres kümesindeki her kayıt, bir Depremzede'yi tanımlar.

Konut kümesindeki her konut bir Adres ile ilişkilidir.

Konut_Sahipligi kümesindeki her kayıt bir Konut ve bir Depremzede arasındaki ilişkiyi ifade eder.

Tahsis kümesindeki her kayıt bir Konut ve bir Depremzede arasındaki tahsis durumunu gösterir.

SQL KODLARI:

TABLULAR

```
CREATE TABLE [dbo].[Adres] (  
    [AdresID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
    [DepremzedeID] INT NULL FOREIGN KEY REFERENCES [dbo].[Depremzede] ([DepremzedeID]),  
    [EskiAdres] VARCHAR(255) NULL,  
    [YeniAdres] VARCHAR(255) NULL  
);
```

```
CREATE TABLE [dbo].[Depremzede] (  
    [DepremzedeID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
    [Ad] VARCHAR(100) NULL,  
    [Soyad] VARCHAR(100) NULL,  
    [TC_KimlikNo] VARCHAR(11) NULL UNIQUE NONCLUSTERED,  
    [DogumTarihi] DATE NULL  
);
```

```
CREATE TABLE [dbo].[Iletisim] (  
    [IletisimID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
    [DepremzedeID] INT NULL FOREIGN KEY REFERENCES [dbo].[Depremzede] ([DepremzedeID]),  
    [Telefon] VARCHAR(20) NULL,  
    [Email] VARCHAR(100) NULL  
);
```

```
CREATE TABLE [dbo].[Konut] (  
    [KonutID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
    [Adres] VARCHAR(255) NULL,  
    [Metrekare] INT NULL,  
    [OdaSayisi] INT NULL,  
    [KonutTuruID] INT NULL FOREIGN KEY REFERENCES [dbo].[Konut_Turu] ([KonutTuruID]),  
    [DurumID] INT NULL FOREIGN KEY REFERENCES [dbo].[Konut_Durumu] ([DurumID])  
);
```

```
CREATE TABLE [dbo].[Konut_Durumu] (  
    [DurumID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
    [Durum] VARCHAR(50) NULL  
);
```

```
CREATE TABLE [dbo].[Konut_Sahipligi] (  
    [KonutSahipligiID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
    [KonutID] INT NULL FOREIGN KEY REFERENCES [dbo].[Konut] ([KonutID]),  
    [SahipID] INT NULL  
);
```

```
CREATE TABLE [dbo].[Konut_Turu] (  
    [KonutTuruID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
    [Turu] VARCHAR(50) NULL  
);
```

```
CREATE TABLE [dbo].[Tahsis] (  
    [TahsisID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
    [DepremzedeID] INT NULL FOREIGN KEY REFERENCES [dbo].[Depremzede] ([DepremzedeID]),  
    [KonutID] INT NULL FOREIGN KEY REFERENCES [dbo].[Konut] ([KonutID]),  
    [TahsisTarihi] DATE NULL  
);
```

```
CREATE TABLE [dbo].[Tahsis_Tarihi] (  
    [TahsisTarihiID] INT IDENTITY(1,1) NOT NULL PRIMARY KEY CLUSTERED,  
    [KonutID] INT NULL FOREIGN KEY REFERENCES [dbo].[Konut] ([KonutID]),  
    [TahsisTarihi] DATE NULL  
);
```

TRIGGER

Aşağıdaki trigger Tahsis tablosuna yeni bir kayıt eklendiğinde, tahsis edilen konutun durumunu otomatik olarak 'Tahsis Edilmiştir' şeklinde günceller.

```
CREATE TRIGGER TRG_Tahsis_Insert
ON [dbo].[Tahsis]
AFTER INSERT
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @KonutID INT;
    SELECT @KonutID = KonutID FROM inserted;
    UPDATE Konut
    SET DurumID = (SELECT DurumID FROM Konut_Durumu WHERE Durum = 'Tahsis Edilmiştir')
    WHERE KonutID = @KonutID;
    PRINT 'Konut durumu güncellendi.';
END;
GO
```

Stored Procedure

Depremzedelere boş durumdaki bir konutun tahsis edilmesini (özel olarak ayırmasını) sağlar.

```
CREATE PROCEDURE [dbo].[TahsisYap]
    @DepremzedelID INT
AS
BEGIN
    DECLARE @KonutID INT;
    SELECT TOP 1 @KonutID = KonutID
    FROM Konut
    WHERE DurumID = (SELECT DurumID FROM Konut_Durumu WHERE Durum = 'Boş')
    ORDER BY NEWID();

    IF @KonutID IS NOT NULL
    BEGIN
```

```

INSERT INTO Tahsis (DepremzedeID, KonutID, TahsisTarihi)
VALUES (@DepremzedeID, @KonutID, GETDATE());

UPDATE Konut
SET DurumID = (SELECT DurumID FROM Konut_Durumu WHERE Durum = 'Tahsis Edildi')
WHERE KonutID = @KonutID;

PRINT 'Konut başarıyla tahsis edildi.';

END

ELSE

BEGIN

    PRINT 'Uygun konut bulunamadı.';

END

END;

```

ROLLBACK TRANSACTION

İşlem sırasında hata oluşursa tüm işlemleri geri almak için bir **ROLLBACK TRANSACTION kullandık**

```

CREATE PROCEDURE [dbo].[TahsisYapTransaction]
    @DepremzedeID INT
AS
BEGIN
    BEGIN TRANSACTION;

    DECLARE @KonutID INT;

    BEGIN TRY
        SELECT TOP 1 @KonutID = KonutID
        FROM Konut
        WHERE DurumID = (SELECT DurumID FROM Konut_Durumu WHERE Durum = 'Boş')
        ORDER BY NEWID();

        IF @KonutID IS NOT NULL
        BEGIN
            INSERT INTO Tahsis (DepremzedeID, KonutID, TahsisTarihi)
            VALUES (@DepremzedeID, @KonutID, GETDATE());

```



```
UPDATE Konut

SET DurumID = (SELECT DurumID FROM Konut_Durumu WHERE Durum = 'Tahsis Edildi')

WHERE KonutID = @KonutID;


COMMIT TRANSACTION;

PRINT 'Konut başarıyla tahsis edildi.';

END

ELSE

BEGIN

    PRINT 'Uygun konut bulunamadı.';

    ROLLBACK TRANSACTION;

END

END TRY

BEGIN CATCH

    PRINT 'Bir hata oluştu.';

    ROLLBACK TRANSACTION;

END CATCH

END;
```

İlişki Kontrol

Adres tablosundaki DepremzedeID alanını, Depremzede tablosundaki DepremzedeID alanına bağlar.

```
ALTER TABLE [dbo].[Adres] WITH CHECK ADD FOREIGN KEY([DepremzedeID])
REFERENCES [dbo].[Depremzede] ([DepremzedeID])
```

İletişim tablosundaki DepremzedeID alanını, Depremzede tablosundaki DepremzedeID alanına bağlar.

```
ALTER TABLE [dbo].[Iletisim] WITH CHECK ADD FOREIGN KEY([DepremzedeID])
REFERENCES [dbo].[Depremzede] ([DepremzedeID])
```

Konut tablosundaki DurumID alanını, Konut_Durumu tablosundaki DurumID alanına bağlar.

```
ALTER TABLE [dbo].[Konut] WITH CHECK ADD FOREIGN KEY([DurumID])
REFERENCES [dbo].[Konut_Durumu] ([DurumID])
```

Konut tablosundaki KonutTuruID alanını, Konut_Turu tablosundaki KonutTuruID alanına bağlar.

```
ALTER TABLE [dbo].[Konut] WITH CHECK ADD FOREIGN KEY([KonutTuruID])  
REFERENCES [dbo].[Konut_Turu] ([KonutTuruID])
```

Konut_Sahipligi tablosundaki KonutID alanını, Konut tablosundaki KonutID alanına bağlar.

```
ALTER TABLE [dbo].[Konut_Sahipligi] WITH CHECK ADD FOREIGN KEY([KonutID])  
REFERENCES [dbo].[Konut] ([KonutID])
```

Tahsis tablosundaki DepremzedeID alanını, Depremzede tablosundaki DepremzedeID alanına bağlar.

```
ALTER TABLE [dbo].[Tahsis] WITH CHECK ADD FOREIGN KEY([DepremzedeID])  
REFERENCES [dbo].[Depremzede] ([DepremzedeID])
```

Tahsis tablosundaki KonutID alanını, Konut tablosundaki KonutID alanına bağlar.

```
ALTER TABLE [dbo].[Tahsis] WITH CHECK ADD FOREIGN KEY([KonutID])  
REFERENCES [dbo].[Konut] ([KonutID])
```

Tahsis_Tarihi tablosundaki KonutID alanını, Konut tablosundaki KonutID alanına bağlar.

```
ALTER TABLE [dbo].[Tahsis_Tarihi] WITH CHECK ADD FOREIGN KEY([KonutID])  
REFERENCES [dbo].[Konut] ([KonutID])
```

Veri ekleme

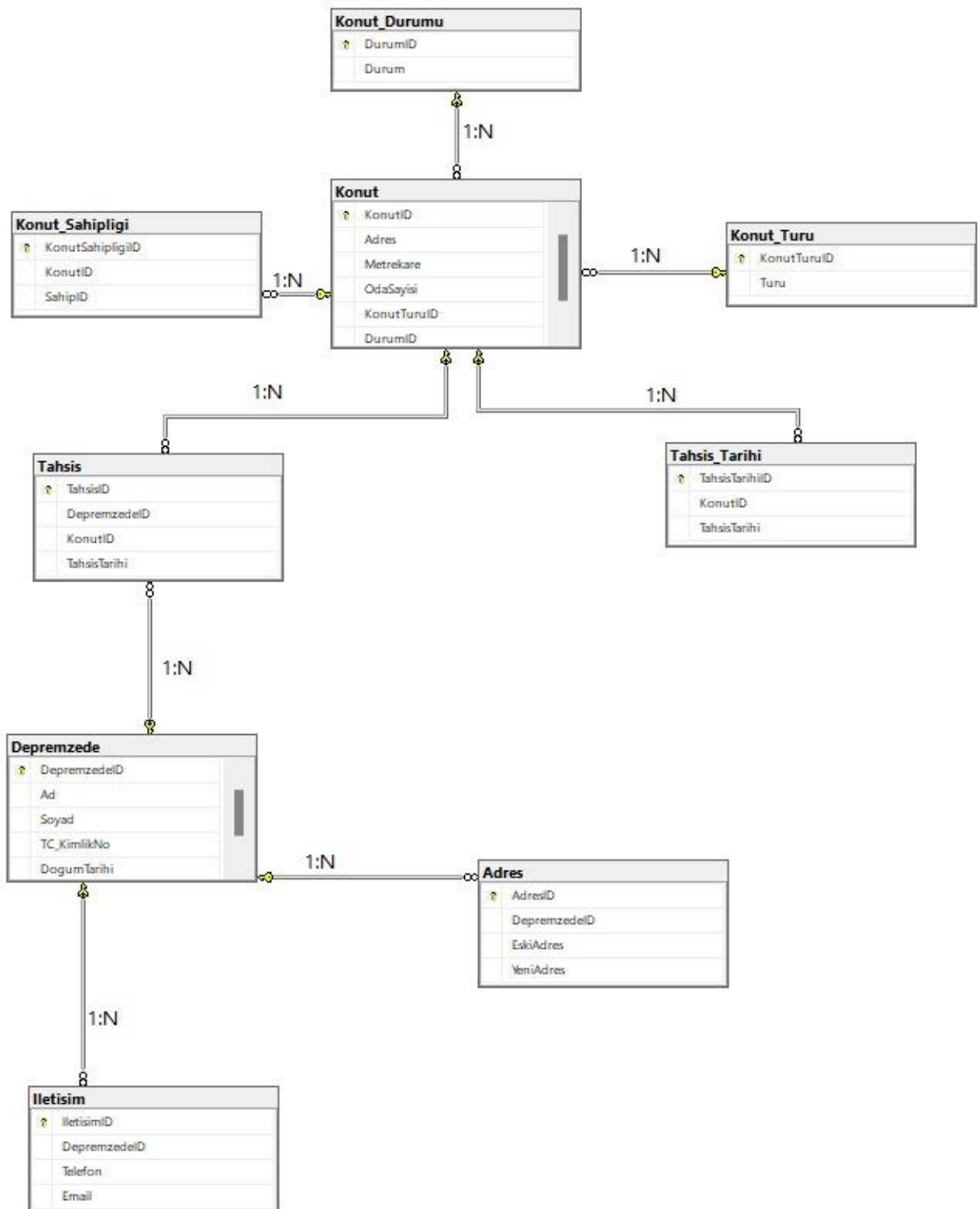
```
INSERT INTO Depremzede (Ad, Soyad, TC_KimlikNo, DogumTarihi)
```

```
VALUES
```

```
('Ali', 'Yılmaz', '12345678901', '1985-05-20'),
```

```
('Ayşe', 'Kara', '98765432109', '1990-11-15');
```

UML



E-R Diyagramı

