

# Adli Süreç Takip ve Yönetim Sistemi

## Proje Ekibindeki Kişiler

- 225260030 – Sıla Rabia Bayram
- 235260134 – İrem Allış
- 235260182 – Merve Beril Yılmaz

## 1. Proje Kullanıcıları Gereksinimleri ve Yetkileri

Adli Süreç Takip ve Yönetim Sistemi'nde beş ana kullanıcı türü bulunur: **Dava Tarafları**, **Avukat**, **Savcı**, **Hakim** ve **Admin**. Her kullanıcı türü, veri tabanındaki verilere farklı düzeylerde erişim ve müdahale yetkisine sahiptir.

### 1.1 Dava Tarafları (Davalı, Davacı, Tanık)

- Sadece kendi dava bilgilerini görüntüleyebilir.
- Duruşma yeri, saati gibi bilgileri görüntüleyebilir.
- Yalnızca görüntüleme yetkisine sahiptir, veri girişi yapamaz.

### 1.2 Avukat

- Baktığı davalarla ilgili dosyalara ve bilgilere erişebilir.
- Randevularıyla ilgili bilgileri görüntüleyebilir.
- Müvekkil adına dosya ekleyebilir.

### 1.3 Hakim

- Baktığı davalarla ilgili dosyalara ve bilgilere erişebilir.
- Dava duruşma tarihlerini ve kararlarını güncelleyebilir.
- Tanık bilgilerini görüntüleyebilir, kararları ve ceza detaylarını kaydedebilir.

### 1.4 Savcı

- Yürüttüğü soruşturmalara erişebilir.
- Delil ve tanık bilgilerini güncelleyebilir.
- İddianameler, suçlamalar gibi dokümanları sisteme ekleyebilir.
- Davalar ve dava tarafları hakkında işlem yapabilir.

### 1.5 Admin

- Sistem genelinde tam yetkiye sahiptir.
- Kullanıcı ve rolleri yönetir.
- Tüm dava ve kullanıcı kayıtlarına erişebilir.

## 2. Tablolar ve Varlıkların Özellikleri

Bu veri tabanı modeli, bir dava yönetim sistemindeki davalar, kişiler, duruşmalar, belgeler ve ilişkili diğer yapılar arasındaki ilişkileri organize etmektedir. Aşağıda her bir tablonun içerdiği alanlar ve bu alanların açıklamaları yer almaktadır.

### 2.1 DavaDosyalari

Dava dosyalarına ait temel bilgileri içerir.

Alan Adı	Açıklama	Veri Tipi	Özellikler
dosyaID	Dava dosyasının benzersiz kimliği	INTEGER	Primary Key
davaNo	Dava numarası	STRING	
mahkemeID	Davanın görüldüğü mahkemenin ID'si	INTEGER	Foreign Key, Mahkemeler tablosuna referans
davaAcilmaTarihi	Davanın açılış tarihi	DATE	
davaTürü	Davanın türü	STRING	
davaDurumu	Davanın durumu (açık, kapalı, beklemede)	STRING	

**İlişkiler:** Bir mahkemede (adliye) birden fazla dava dosyası bulunabilir (1:N).

### 2.2 Kisiler

Sistemdeki kişilerin bilgilerini içerir.

Alan Adı	Açıklama	Veri Tipi	Özellikler
kisiID	Kişinin benzersiz kimliği	INTEGER	Primary Key
kisiTC	Kişinin TC kimlik numarası	STRING	
kisiAdSoyad	Kişinin adı ve soyadı	STRING	
kisiDogumTarihi	Kişinin doğum tarihi	DATE	
kisiTelefon	Kişinin telefon numarası	STRING	
kisiMail	Kişinin e-posta adresi	STRING	
kisiAdres	Kişinin adres bilgisi	STRING	
kisiSifresi	Kişinin sisteme giriş yapabilmesi için şifresi	STRING	
kisiRolu	Kişinin rolü (avukat, hakim, savcı, davatafları, admin)	STRING	

### 3. DavaTaraflari

Dava dosyalarındaki taraflara ilişkin bilgiler.

Alan Adı	Açıklama	Veri Tipi	Özellikler
tarafID	Tarafın benzersiz kimliği	INTEGER	Primary Key
dosyaID	Tarafın dahil olduğu dava dosyasının ID'si	INTEGER	Foreign Key, DavaDosyaları tablosuna referans
kisiID	Taraf olarak atanmış kişinin ID'si	INTEGER	Foreign Key, Kisiler tablosuna referans
tarafRol	Tarafın davadaki rolü (davalı, davacı, tanık)	STRING	
ekBilgi	Taraf hakkında ek bilgiler	STRING	

**İlişkiler:** Bir dava dosyasının birden fazla tarafı olabilir, aynı zamanda bir dava tarafının da birden fazla dava dosyası olabilir (N:N). Kişi ile dava tarafı arasında bire bir ilişki vardır (1:1).

## 4. Duruşmalar

### Açıklama

Duruşmaların detaylarını içeren tablo, davaların hangi tarihlerde ve hangi mahkemelerde görüldüğünü takip etmek amacıyla oluşturulmuştur.

### Alanlar

- durusmaID:** Duruşmanın benzersiz kimliği (INTEGER, Primary Key).
- dosyaID:** Duruşmanın bağlı olduğu dava dosyasının kimliği (INTEGER, Foreign Key).
- durusmaTarihi:** Duruşma tarihi (DATE).
- durusmaSaati:** Duruşma saati (TIME).
- durusmaYeri:** Duruşmanın yapıldığı yer (STRING).
- mahkemeID:** Duruşmanın görüldüğü mahkemenin kimliği (INTEGER, Foreign Key).
- hakimID:** Duruşmayı yöneten hakimin kimliği (INTEGER, Foreign Key).
- avukatID:** Duruşmada yer alan avukatın kimliği (INTEGER, Foreign Key).
- durusmaSonuc:** Duruşmanın sonucu (STRING).
- tarafID:** Duruşmada yer alan tarafın kimliği (INTEGER, Foreign Key).

### İlişkiler

- Bir dava dosyası birden fazla duruşmada yer alabilir.
- Bir mahkemede birden fazla duruşma gerçekleştirilebilir.
- Bir hakim birden fazla duruşmada görev alabilir.
- Bir avukat birden fazla duruşmada yer alabilir.
- Bir duruşmada birden fazla taraf yer alabilir.

## 5. Belgeler

## Açıklama

Dava dosyalarına eklenen belgeleri kaydeder.

### Alanlar

- **belgeID**: Belgenin benzersiz kimliği (INTEGER, Primary Key).
- **dosyaID**: Belgenin bağlı olduğu dava dosyası kimliği (INTEGER, Foreign Key).
- **belgeTürü**: Belge türü (delil, ifade, karar, rapor) (STRING).
- **belgeAciklama**: Belge hakkında açıklama (STRING).
- **yuklenmeTarihi**: Belgenin sisteme yüklendiği tarih (DATE).
- **kisiID**: Belgeyi yükleyen kişinin kimliği (INTEGER, Foreign Key).

### İlişkiler

- Bir dava dosyasına birden fazla belge eklenebilir.
- Bir belge birden fazla dosyada yer alabilir.
- Bir kişi birden fazla belge yükleyebilir.

---

## 6. Kararlar

### Açıklama

Davalarla ilgili alınan kararların detaylarını içerir.

### Alanlar

- **kararID**: Kararın benzersiz kimliği (INTEGER, Primary Key).
- **dosyaID**: Kararın bağlı olduğu dava dosyasının kimliği (INTEGER, Foreign Key).
- **kararTarihi**: Kararın tarihi (DATE).
- **kararDurumu**: Kararın durumu (kesin, temyiz) (STRING).
- **kararDetayi**: Kararın detayları (TEXT).
- **hakimID**: Kararı veren hakimin kimliği (INTEGER, Foreign Key).

### İlişkiler

- Bir dava dosyasına birden fazla karar verilebilir.
- Bir hakim birden fazla kararda görev alabilir.

---

## 7. Mahkemeler

### Açıklama

Mahkemelerle ilgili temel bilgileri içerir.

### Alanlar

- **mahkemeID**: Mahkemenin benzersiz kimliği (INTEGER, Primary Key).
- **mahkemeAd**: Mahkemenin adı (STRING).

- **mahkemeAdres:** Mahkemenin adresi (STRING).
  - **mahkemeTelefon:** Mahkemenin telefon numarası (STRING).
  - **mahkemeMail:** Mahkemenin e-posta adresi (STRING).
- 

## 8. Loglar

### Açıklama

Sistemde yapılan işlemleri kaydeder.

### Alanlar

- **logID:** Log kaydının benzersiz kimliği (INTEGER, Primary Key).
- **kisiID:** İşlemi yapan kişinin kimliği (INTEGER, Foreign Key).
- **dosyaID:** İlgili dava dosyası kimliği (INTEGER, Foreign Key).
- **logTarihi:** Log kaydının tarihi (DATE).
- **islemTipi:** İşlem tipi (oluşturma, güncelleme, silme) (STRING).
- **islemDetayi:** İşlemin detayı (STRING).

### İlişkiler

- Bir kişi birden fazla işlem yapabilir.
  - Bir dava dosyasında birden fazla işlem kaydedilebilir.
- 

## 9. Avukat

### Açıklama

Avukatlarla ilgili bilgileri içerir.

### Alanlar

- **avukatID:** Avukatın benzersiz kimliği (INTEGER, Primary Key).
- **kisiID:** Avukat olarak kayıtlı kişinin kimliği (INTEGER, Foreign Key).
- **mahkemeID:** Görev aldığı mahkeme kimliği (INTEGER, Foreign Key).
- **avBuro:** Avukatın bağlı olduğu büro (STRING).
- **davaSayisi:** Temsil ettiği dava sayısı (INTEGER).

### İlişkiler

- Bir avukat birden fazla mahkemede görev alabilir.
- Bir mahkemede birden fazla avukat bulunabilir.
- 

## 10. Savcı

### Açıklama

Savcılarla ilgili bilgileri içeren tablo, savcıların hangi dosyalarda ve mahkemelerde görev aldığını takip etmek için kullanılır.

### Alanlar

- **savciID**: Savcının benzersiz kimliği (INTEGER, Primary Key).
- **kisiID**: Savcı olarak kayıtlı kişinin kimliği (INTEGER, Foreign Key).
- **savciTuru**: Savcının türü (başsavcı, cumhuriyet) (STRING).
- **dosyaID**: Savcıya atanan dava dosyasının kimliği (INTEGER, Foreign Key).
- **mahkemeID**: Savcının görev yaptığı mahkemenin kimliği (INTEGER, Foreign Key).

### İlişkiler

- Bir savcı birden fazla dava dosyasına atanabilir.
- Bir dosyada birden fazla savcı görev alabilir.
- Bir mahkemede birden fazla savcı bulunabilir.

---

## 11. Hakim

### Açıklama

Hakimlerle ilgili bilgileri içerir.

### Alanlar

- **hakimID**: Hakimin benzersiz kimliği (INTEGER, Primary Key).
- **kisiID**: Hakim olarak kayıtlı kişinin kimliği (INTEGER, Foreign Key).
- **hakimTuru**: Hakimin türü (örneğin, başsavcı, cumhuriyet) (STRING).
- **dosyaID**: Hakimin baktığı dava dosyasının kimliği (INTEGER, Foreign Key).
- **mahkemeID**: Hakimin görev yaptığı mahkemenin kimliği (INTEGER, Foreign Key).

### İlişkiler

- Bir hakim birden fazla dava dosyasına bakabilir.
- Bir mahkemede birden fazla hakim bulunabilir.

---

## 12. Kanunlar

### Açıklama

Kanun ve maddelere ait bilgileri içerir.

### Alanlar

- **kanunID**: Kanunun benzersiz kimliği (INTEGER, Primary Key).
- **kanunKodu**: Kanun kodu (STRING).
- **kanunMaddeNo**: Kanunun madde numarası (INTEGER).

- **maddeMetni**: Kanun maddesinin tam metni (TEXT).
  - **cezaTuru**: Maddenin öngördüğü ceza türü (STRING).
  - **cezaSuresi**: Maddenin öngördüğü ceza süresi (STRING).
- 

## 13. Suçlar

### Açıklama

Davalarla ilişkili suçları takip etmek için kullanılır.

### Alanlar

- **sucID**: Suçun benzersiz kimliği (INTEGER, Primary Key).
- **dosyaID**: Suçun dahil olduğu dava dosyasının kimliği (INTEGER, Foreign Key).
- **kanunID**: Suçun ilgili olduğu kanunun kimliği (INTEGER, Foreign Key).
- **sucTipi**: Suç tipi (STRING).
- **sucYeri**: Suçun işlendiği yer (STRING).
- **tarafID**: Suç ile ilişkili tarafın kimliği (INTEGER, Foreign Key).
- **sucDurumu**: Suç durumu (inceleme, tamamlandı) (STRING).

### İlişkiler

- Bir suç birden fazla dava dosyasında yer alabilir.
  - Bir suç bir kanun ile bağlantılıdır.
  - Bir suçun birden fazla tarafı olabilir.
- 

## 14. Cezalar

### Açıklama

Suçlara verilen cezaları takip etmek için oluşturulmuştur.

### Alanlar

- **cezaID**: Cezanın benzersiz kimliği (INTEGER, Primary Key).
- **sucID**: Cezanın bağlı olduğu suçun kimliği (INTEGER, Foreign Key).
- **cezaTuru**: Cezanın türü (para, hapis) (STRING).
- **cezaSuresi**: Cezanın süresi (STRING).
- **cezaTutar**: Cezanın miktarı (DECIMAL).
- **basTarih**: Cezanın başlangıç tarihi (DATE).
- **bitTarih**: Cezanın bitiş tarihi (DATE).
- **cezaAciklama**: Cezaya ilişkin açıklama (STRING).

### İlişkiler

- Bir suç birden fazla cezaya neden olabilir.
- Bir ceza birden fazla suçla ilişkili olabilir.

---

## 15. Randevular

### Açıklama

Avukatlar ve dava tarafları arasındaki görüşmeleri kaydeder.

### Alanlar

- **randevuID**: Randevunun benzersiz kimliği (INTEGER, Primary Key).
- **avukatID**: Randevunun düzenlendiği avukatın kimliği (INTEGER, Foreign Key).
- **tarafID**: Randevunun düzenlendiği tarafın kimliği (INTEGER, Foreign Key).
- **randevuTarih**: Randevu tarihi (DATE).
- **randevuSaat**: Randevu saati (TIME).
- **randevuDurumu**: Randevu durumu (STRING).
- **randevuAciklama**: Randevuya ilişkin açıklama (STRING).

### İlişkiler

- Bir avukat birden fazla randevu düzenleyebilir.
- Bir taraf birden fazla avukatla randevu yapabilir.

---

## 16. Ödemeler

### Açıklama

Randevulara ilişkin yapılan ödemeleri kaydeder.

### Alanlar

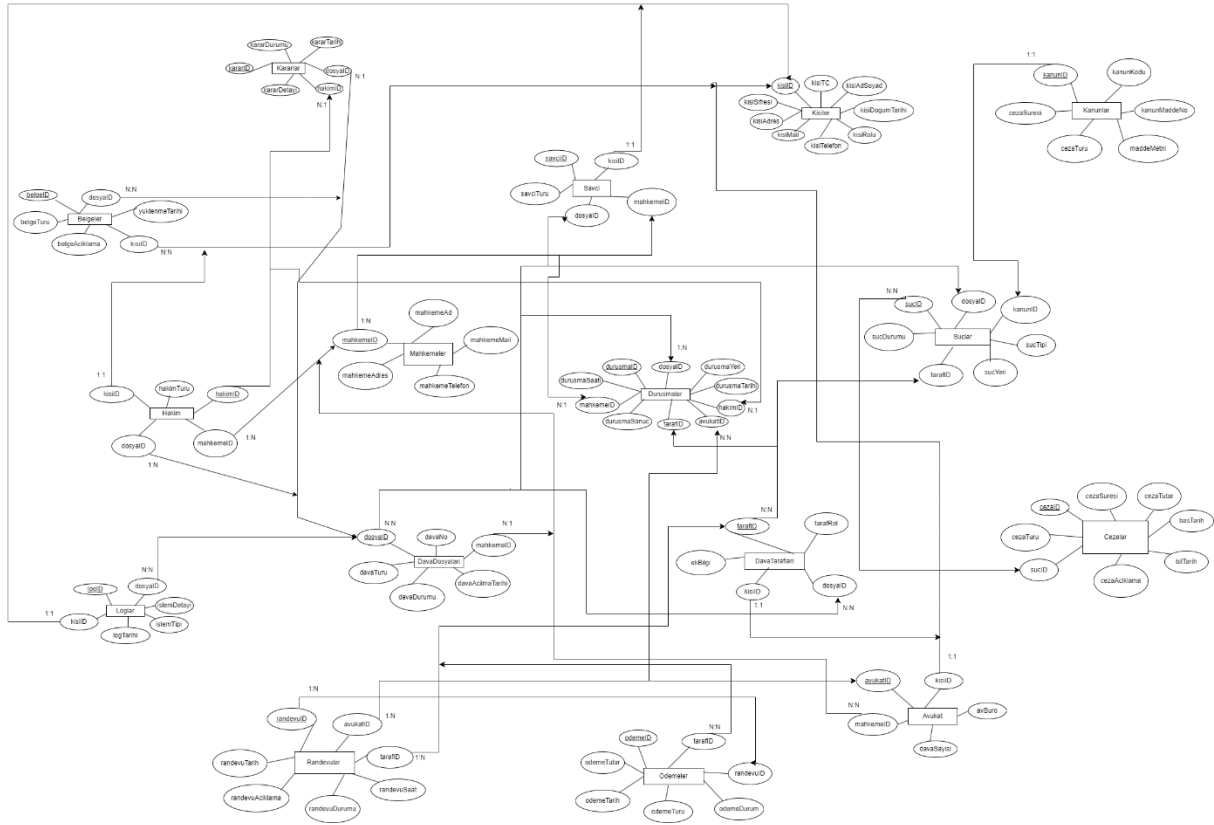
- **odemeID**: Ödemenin benzersiz kimliği (INTEGER, Primary Key).
- **tarafID**: Ödemeyi yapan tarafın kimliği (INTEGER, Foreign Key).
- **randevuID**: Ödemeye ilişkin randevunun kimliği (INTEGER, Foreign Key).
- **odemeTutar**: Ödeme tutarı (DECIMAL).
- **odemeTarih**: Ödeme tarihi (DATE).
- **odemeTuru**: Ödeme türü (nakit, kart) (STRING).
- **odemeDurum**: Ödeme durumu (ödendi, ödenmedi) (STRING).

### İlişkiler

- Bir taraf birden fazla ödeme yapabilir.
- Bir randevu birden fazla ödeme ile ilişkili olabilir.



## E-R Diyagramı



## İlişkisel Veri Modeli

DavaDosyaları(dosyaID, davaNo, mahkemeID, davaAcilmaTarihi, davaTuru, davaDurumu)

Kisiler(kisiID, kisiTC, kisiAdSoyad, kisiDogumTarihi, kisiTelefon, kisiMail, kisiAdres, kisiSifresi, kisiRolu)

DavaTarafları(tarafID, dosyaID, kisiID, tarafRol, ekBilgi)

Durusmalar(durusmaID, tarafID, avukatID, dosyaID, durusmaTarihi, durusmaSaati, durusmaYeri, mahkemeID, hakimID, durusmaSonuc)

Belgeler(belgelID, dosyaID, kisiID, belgeTuru, belgeAciklama, yuklenmeTarihi)

Kararlar(kararID, dosyaID, kararTarihi, kararDurumu, kararDetayi, hakimID)

Mahkemeler(mahkemeID, mahkemeAd, mahkemeAdres, mahkemeTelefon, mahkemeMail)

Loglar(logID, dosyaID, kisiID, logTarihi, islemTipi, islemDetayi)

Avukat(avukatID, mahkemeID, kisiID, avBuro, davaSayisi)

Savci(savciID, dosyaID, kisiID, savciTuru, mahkemeID)

Hakim(hakimID, kisiID, hakimTuru, dosyaID, mahkemeID)

Kanunlar(kanunID, kanunKodu, kanunMaddeNo, maddeMetni, cezaTuru, cezaSuresi)

Suclar(sucID, dosyaID, tarafID, kanunID, sucTipi, sucYeri, sucDurumu)

Cezalar(cezaID, sucID, cezaTuru, cezaSuresi, cezaTutar, basTarih, bitTarih, cezaAciklama)

Randevular(randevuID, avukatID, tarafID, randevuTarih, randevuSaat, randevuDurumu, randevuAciklama)

Odemeler(odemeID, tarafID, randevuID, odemeTutar, odemeTarih, odemeTuru, odemeDurum)

## Adli Süreç Takip ve Yönetim Sistemi - BCNF Normalizasyon Sonucu

---

### 1. DavaDosyalari Tablosu (BCNF)

Bu tablo **BCNF**'ye uygundur. Değişiklik gerekmez.

```
CREATE TABLE DavaDosyalari (  
    dosyaID INTEGER PRIMARY KEY IDENTITY(1,1),  
    davaNo NVARCHAR(20) NOT NULL UNIQUE,  
    mahkemeID INTEGER NOT NULL,  
    davaAcilmaTarihi DATE,  
    davaTuru NVARCHAR(50),  
    davaDurumu NVARCHAR(50),  
    FOREIGN KEY (mahkemeID) REFERENCES Mahkemeler(mahkemeID)  
);
```

---

### 2. Kisiler Tablosu (BCNF)

Bu tablo **BCNF**'ye uygundur. Değişiklik gerekmez.

```
CREATE TABLE Kisiler (  
    kisiID INTEGER PRIMARY KEY IDENTITY(1,1),  
    kisiTC NVARCHAR(11) NOT NULL UNIQUE,  
    kisiAdSoyad NVARCHAR(100) NOT NULL,  
    kisiDogumTarihi DATE,  
    kisiTelefon NVARCHAR(15),  
    kisiMail NVARCHAR(100),  
    kisiAdres NVARCHAR(255),  
    kisiSifresi NVARCHAR(255),  
    kisiRolu NVARCHAR(50)  
);
```

---

### 3. DavaTaraflari Tablosu (BCNF)

Bu tablo **BCNF**'ye uygundur. Değişiklik gerekmez.

```
CREATE TABLE DavaTaraflari (  
  
    tarafID INTEGER PRIMARY KEY IDENTITY(1,1),  
    dosyaID INTEGER NOT NULL,  
    kisiID INTEGER NOT NULL,  
    tarafRol NVARCHAR(50),  
    ekBilgi NVARCHAR(255),  
    FOREIGN KEY (dosyaID) REFERENCES DavaDosyolari(dosyaID),  
    FOREIGN KEY (kisiID) REFERENCES Kisiler(kisiID)  
);
```

---

#### 4. Durusmalar Tablosu (BCNF)

Durusmalar tablosunda potansiyel bağımlılıkların giderilmesi için tablo ikiye bölünmüştür.

##### Durusmalar:

```
CREATE TABLE Durusmalar (  
    durusmaID INTEGER PRIMARY KEY IDENTITY(1,1),  
    dosyaID INTEGER NOT NULL,  
    durusmaTarihi DATE,  
    durusmaSaati TIME,  
    durusmaYeri NVARCHAR(100),  
    FOREIGN KEY (dosyaID) REFERENCES DavaDosyolari(dosyaID)  
);
```

##### MahkemeDurusma:

```
CREATE TABLE MahkemeDurusma (  
    durusmaID INTEGER PRIMARY KEY,  
    mahkemeID INTEGER NOT NULL,  
    hakimID INTEGER,  
    avukatID INTEGER,  
    durusmaSonuc NVARCHAR(50),  
    FOREIGN KEY (durusmaID) REFERENCES Durusmalar(durusmaID),  
    FOREIGN KEY (mahkemeID) REFERENCES Mahkemeler(mahkemeID),  
    FOREIGN KEY (hakimID) REFERENCES Hakim(hakimID),  
    FOREIGN KEY (avukatID) REFERENCES Avukat(avukatID)  
);
```

---

#### 5. Belgeler Tablosu (BCNF)

Belgeler tablosu **BCNF**'ye uygundur. Değişiklik gerekmez.

```
CREATE TABLE Belgeler (  
    belgeID INTEGER PRIMARY KEY IDENTITY(1,1),  
    dosyaID INTEGER NOT NULL,  
    belgeTürü NVARCHAR(50),  
    belgeAciklama NVARCHAR(255),  
    yüklenmeTarihi DATE,  
    kisiID INTEGER,  
    FOREIGN KEY (dosyaID) REFERENCES DavaDosyolari(dosyaID),  
    FOREIGN KEY (kisiID) REFERENCES Kisiler(kisiID)  
);
```

---

## 6. Kararlar Tablosu (BCNF)

Kararlar tablosu **BCNF**'ye uygundur. Değişiklik gerekmez.

```
CREATE TABLE Kararlar (  
    kararID INTEGER PRIMARY KEY IDENTITY(1,1),  
    dosyaID INTEGER NOT NULL,  
    kararTarihi DATE,  
    kararDurumu NVARCHAR(50),  
    kararDetayi TEXT,  
    hakimID INTEGER,  
    FOREIGN KEY (dosyaID) REFERENCES DavaDosyolari(dosyaID),  
    FOREIGN KEY (hakimID) REFERENCES Hakim(hakimID)  
);
```

---

## 7. Mahkemeler Tablosu (BCNF)

Bu tablo **BCNF**'ye uygundur. Değişiklik gerekmez.

```
CREATE TABLE Mahkemeler (  
    mahkemeID INTEGER PRIMARY KEY IDENTITY(1,1),  
    mahkemeAd NVARCHAR(100) NOT NULL,  
    mahkemeAdres NVARCHAR(255),  
    mahkemeTelefon NVARCHAR(15),  
    mahkemeMail NVARCHAR(100)  
);
```

---

## 8. Loglar Tablosu (BCNF)

Loglar tablosu **BCNF**'ye uygundur. Değişiklik gerekmez.

```
CREATE TABLE Loglar (  
    logID INTEGER PRIMARY KEY IDENTITY(1,1),  
    kisiID INTEGER,  
    dosyaID INTEGER,  
    logTarihi DATE,  
    islemTipi NVARCHAR(50),  
    islemDetayi NVARCHAR(255),  
    FOREIGN KEY (kisiID) REFERENCES Kisiler(kisiID),  
    FOREIGN KEY (dosyaID) REFERENCES DavaDosyolari(dosyaID)  
);
```

---

## 9. Savci Tablosu (BCNF)

Savcı tablosu **BCNF**'ye uygundur. Değişiklik gerekmez.

```
CREATE TABLE Savci (  
    savciID INTEGER PRIMARY KEY IDENTITY(1,1),  
    kisiID INTEGER NOT NULL,  
    savciTuru NVARCHAR(50),  
    dosyaID INTEGER,  
    mahkemeID INTEGER,  
    FOREIGN KEY (kisiID) REFERENCES Kisiler(kisiID),
```

```
FOREIGN KEY (dosyaID) REFERENCES DavaDosyolari(dosyaID),  
FOREIGN KEY (mahkemeID) REFERENCES Mahkemeler(mahkemeID)  
);
```

---

## 10. Hakim Tablosu (BCNF)

Hakim tablosu **BCNF**'ye uygundur. Değişiklik gerekmez.

```
CREATE TABLE Hakim (  
    hakimID INTEGER PRIMARY KEY IDENTITY(1,1),  
    kisiID INTEGER NOT NULL,  
    hakimTuru NVARCHAR(50),  
    dosyaID INTEGER,  
    mahkemeID INTEGER,  
    FOREIGN KEY (kisiID) REFERENCES Kisiler(kisiID),  
    FOREIGN KEY (dosyaID) REFERENCES DavaDosyolari(dosyaID),  
    FOREIGN KEY (mahkemeID) REFERENCES Mahkemeler(mahkemeID)  
);
```

## 11. Avukat Tablosu (BCNF)

**Mevcut Yapı:**

| avukatID (PK) | kisiID (FK) | mahkemeID (FK) | avBuro | davaSayisi |

**Analiz:**

- kisiID ve mahkemeID arasında bir **N:N** ilişki var.
- Avukat birden fazla mahkemede çalışabilir, bir mahkemede birden fazla avukat bulunabilir.

**Normalleştirilmiş Yapı:**

Tabloyu ikiye böleriz:

### 1. Avukat:

```
CREATE TABLE Avukat (  
    avukatID INTEGER PRIMARY KEY IDENTITY(1,1),  
    kisiID INTEGER NOT NULL UNIQUE,  
    avBuro NVARCHAR(100),  
    davaSayisi INTEGER,  
    FOREIGN KEY (kisiID) REFERENCES Kisiler(kisiID)  
);
```

### 2. AvukatMahkeme (N:N ilişkisini yönetir):

```
CREATE TABLE AvukatMahkeme (  
    avukatID INTEGER,  
    mahkemeID INTEGER,  
    PRIMARY KEY (avukatID, mahkemeID),  
    FOREIGN KEY (avukatID) REFERENCES Avukat(avukatID),  
    FOREIGN KEY (mahkemeID)  
);
```

## 12. Suçlar Tablosu (BCNF)

### Mevcut Yapı:

| suçID (PK) | dosyaID (FK) | kanunID (FK) | suçTipi | suçYeri | tarafID (FK) | suçDurumu |

### Analiz:

- Bir suç birden fazla dosyada yer alabilir ve bir taraf birden fazla suçla ilişkilendirilebilir (N:N ilişkisi).

### Normalleştirilmiş Yapı:

Tabloyu ikiye böleriz:

#### 1. Suçlar:

```
CREATE TABLE Suclar (  
    suçID INTEGER PRIMARY KEY IDENTITY(1,1),  
    kanunID INTEGER NOT NULL,  
    suçTipi NVARCHAR(50),  
    suçYeri NVARCHAR(100),  
    suçDurumu NVARCHAR(50),  
    FOREIGN KEY (kanunID) REFERENCES Kanunlar(kanunID)  
);
```

#### 2. SuçDosyaTaraf (N:N ilişkisini yönetir):

```
CREATE TABLE SucDosyaTaraf (  
    suçID INTEGER,  
    dosyaID INTEGER,  
    tarafID INTEGER,  
    PRIMARY KEY (suçID, dosyaID, tarafID),  
    FOREIGN KEY (suçID) REFERENCES Suclar(suçID),  
    FOREIGN KEY (dosyaID) REFERENCES DavaDosyaları(dosyaID),  
    FOREIGN KEY (tarafID) REFERENCES DavaTarafları(tarafID)  
);
```

---

## 13. Cezalar Tablosu (BCNF)

### Mevcut Yapı:

| cezaID (PK) | suçID (FK) | cezaTuru | cezaSuresi | cezaTutar | başTarih | bitTarih | cezaAciklama |

### Analiz:

- Bir suç birden fazla cezaya sahip olabilir.

### Normalleştirilmiş Yapı:

Değişiklik gerekmez çünkü zaten **BCNF**'ye uygundur.

```
CREATE TABLE Cezalar (  
    cezaID INTEGER PRIMARY KEY IDENTITY(1,1),  
    sucID INTEGER NOT NULL,  
    cezaTuru NVARCHAR(50),  
    cezaSuresi NVARCHAR(50),  
    cezaTutar DECIMAL(10,2),  
    basTarih DATE,  
    bitTarih DATE,  
    cezaAciklama NVARCHAR(255),  
    FOREIGN KEY (sucID) REFERENCES Suclar(sucID)  
);
```

---

## 14. Randevular ve Ödemeler Tabloları (BCNF)

Bu tablolar zaten **BCNF**'ye uygundur. Değişiklik gerekmez.

### Randevular:

```
CREATE TABLE Randevular (  
    randevuID INTEGER PRIMARY KEY IDENTITY(1,1),  
    avukatID INTEGER,  
    tarafID INTEGER,  
    randevuTarih DATE,  
    randevuSaat TIME,  
    randevuDurumu NVARCHAR(50),  
    randevuAciklama NVARCHAR(255),  
    FOREIGN KEY (avukatID) REFERENCES Avukat(avukatID),  
    FOREIGN KEY (tarafID) REFERENCES DavaTaraflari(tarafID)  
);
```

### Ödemeler:

```
CREATE TABLE Odemeler (  
    odemeID INTEGER PRIMARY KEY IDENTITY(1,1),  
    tarafID INTEGER,  
    randevuID INTEGER,  
    odemeTutar DECIMAL(10,2),  
    odemeTarih DATE,  
    odemeTuru NVARCHAR(50),  
    odemeDurum NVARCHAR(50),  
    FOREIGN KEY (tarafID) REFERENCES DavaTaraflari(tarafID),  
    FOREIGN KEY (randevuID) REFERENCES Randevular(randevuID)  
);
```

---

### Sonuç:

- **Durusmalar, Avukatlar ve Suçlar** tablosu, potansiyel bağımlılıklar nedeniyle ikiye bölünerek **BCNF**'ye uygun hale getirilmiştir.
  - Diğer tüm tablolar zaten **BCNF** kriterlerini sağladığı için değişiklik yapılmamıştır.
- 

## Transaction Senaryosu

Bir **yeni dava** eklenirken, aynı zamanda bu davaya ait **tarafklar** ve **duruşma** bilgileri eklenir. Eğer işlemler sırasında bir hata oluşursa, tüm değişiklikler geri alınmalıdır.

---

## Transaction Yönetimi Kodu

Aşağıdaki kod, bir dava ekleme sürecini atomik bir işlem olarak ele alır. İşlemler sırasında bir hata oluşursa, tüm yapılan değişiklikler geri alınır.

```
BEGIN TRY
    BEGIN TRANSACTION;
    INSERT INTO DavaDosyolari (davaNo, mahkemeID, davaAcilmaTarihi,
davaTürü, davaDurumu)
    VALUES ('2024/001', 1, '2024-01-01', 'Ceza', 'Açık');

    DECLARE @davaID INT = SCOPE_IDENTITY();

    INSERT INTO DavaTaraflari (dosyaID, kisiID, tarafRol, ekBilgi)
    VALUES (@davaID, 1, 'Davacı', 'Davanın ana tarafı');

    INSERT INTO Durusmalar (dosyaID, durusmaTarihi, durusmaSaati,
durusmaYeri, mahkemeID)
    VALUES (@davaID, '2024-02-15', '10:00:00', 'Ankara Adliye 1. Salon',
1);
    COMMIT TRANSACTION;
    PRINT 'İşlem başarıyla tamamlandı.';
END TRY
BEGIN CATCH

    ROLLBACK TRANSACTION;
    PRINT 'Hata oluştu, işlem geri alındı.';
    PRINT ERROR_MESSAGE();
END CATCH;
```

---

## Kod Açıklaması:

- BEGIN TRANSACTION VE COMMIT TRANSACTION:**
    - Tüm işlemler bir transaction bloğu içerisinde gerçekleştirilir.
    - Eğer tüm işlemler başarılı olursa, COMMIT ile işlemler kalıcı hale getirilir.
  - Hata Yönetimi (TRY...CATCH):**
    - İşlemler sırasında bir hata oluşursa, CATCH bloğu devreye girer ve yapılan tüm değişiklikler ROLLBACK ile geri alınır.
  - SCOPE\_IDENTITY():**
    - Yeni eklenen davanın ID değerini alır ve sonraki işlemlerde kullanır.
- 

## Test Senaryosu:

- Başarılı İşlem Testi:**
  - Kodun hatasız çalıştırılmasını sağlayın ve verilerin düzgün şekilde eklendiğini kontrol edin.



## 2. Hata Testi:

- Durusmalar tablosuna eklenen kayıt için bilinçli bir hata oluşturun (örneğin, geçersiz bir tarih girin).
- Hata oluştuğunda tüm işlemlerin geri alındığını doğrulayın.

---

Bu kod ile bir hata oluştuğunda işlemlerin tamamı geri alınacak, böylece veri tutarlılığı sağlanmış olacaktır.