



**1. Varlıklar**

Varlıklar, her tabloyu temsil eder:

* **Avukatlar**
* **Müvekkiller**
* **Davalar**
* **Dava\_Taraflari** (Ara varlık/ilişkiyi temsil eden tablo)
* **Mahkemeler**
* **Belgeler**
* **Ödemeler**
* **Şikayetler**

**2. Nitelikler**

**Her varlığa ait nitelikler:**

#### **Avukatlar**

* AvukatID (Birincil anahtar)
* Ad
* Soyad
* UzmanlikAlanı
* BaroNumarasi

#### **Müvekkiller**

* MuvekkilID (Birincil anahtar)
* Ad
* Soyad
* Telefon
* Eposta
* Adres

#### **Davalar**

* DavaID (Birincil anahtar)
* DavaTuru
* BaslangicTarihi
* Durum
* AvukatID (Yabancı anahtar)
* MahkemeID (Yabancı anahtar)

#### **Dava\_Taraflari** (Ara Tablo)

* DavaID (Birincil anahtar parçası ve yabancı anahtar)
* MuvekkilID (Birincil anahtar parçası ve yabancı anahtar)
* TarafRol (Davacı, Davalı gibi rolleri ifade eder)

#### **Mahkemeler**

* MahkemeID (Birincil anahtar)
* MahkemeAdi
* Sehir
* Adres

#### **Belgeler**

* BelgeID (Birincil anahtar)
* DavaID (Yabancı anahtar)
* BelgeTuru
* YuklenmeTarihi

#### **Ödemeler**

* OdemeID (Birincil anahtar)
* MuvekkilID (Yabancı anahtar)
* Tutar
* Tarih
* OdemeTuru

#### **Şikayetler**

* SikayetID (Birincil anahtar)
* DavaID (Yabancı anahtar)
* SikayetMetni
* SikayetTarihi

### ****3. İlişkiler****

İlişkiler, tablolar arasındaki bağlantıları ifade eder:

#### **Avukatlar ↔ Davalar**

* **İlişki Türü**: 1:N (Bir avukat birden fazla davaya bakabilir.)
* **İlişki Adı**: **"Temsil Eder"**

#### **Müvekkiller ↔ Dava\_Taraflari**

* **İlişki Türü**: 1:N (Bir müvekkil birden fazla davaya taraf olabilir.)
* **İlişki Adı**: **"Taraflık İlişkisi"**

#### **Davalar ↔ Dava\_Taraflari**

* **İlişki Türü**: 1:N (Bir dava birden fazla tarafa sahip olabilir.)
* **İlişki Adı**: **"Taraf Olur"**

#### **Davalar ↔ Mahkemeler**

* **İlişki Türü**: N:1 (Bir dava yalnızca bir mahkemede görülür.)
* **İlişki Adı**: **"Görülür"**

#### **Davalar ↔ Belgeler**

* **İlişki Türü**: 1:N (Bir dava birden fazla belgeye sahip olabilir.)
* **İlişki Adı**: **"İçerir"**

#### **Davalar ↔ Şikayetler**

* **İlişki Türü**: 1:N (Bir dava hakkında birden fazla şikayet olabilir.)
* **İlişki Adı**: **"Hakkında Yapılır"**

#### **Müvekkiller ↔ Ödemeler**

* **İlişki Türü**: 1:N (Bir müvekkil birden fazla ödeme yapabilir.)
* **İlişki Adı**: **"Yapar"**

#### **İlişkiler**:

1. Avukatlar → Davalar: **Temsil Eder**
2. Davalar → Dava\_Taraflari: **Taraf Olur**
3. Müvekkiller → Dava\_Taraflari: **Taraflık İlişkisi**
4. Davalar → Mahkemeler: **Görülür**
5. Davalar → Belgeler: **İçerir**
6. Davalar → Şikayetler: **Hakkında Yapılır**
7. Müvekkiller → Ödemeler: **Yapar**

**1. Avukatlar Tablosu**

CREATE TABLE Avukatlar (

AvukatID INT PRIMARY KEY IDENTITY(1,1),

Ad NVARCHAR(50),

Soyad NVARCHAR(50),

UzmanlikAlanı NVARCHAR(100),

BaroNumarasi NVARCHAR(20)

);

**2. Müvekkiller Tablosu**

CREATE TABLE Muvekkiller (

MuvekkilID INT PRIMARY KEY IDENTITY(1,1),

Ad NVARCHAR(50),

Soyad NVARCHAR(50),

Telefon NVARCHAR(15),

Eposta NVARCHAR(100),

Adres NVARCHAR(200)

);

**3. Davalar Tablosu**

CREATE TABLE Davalar (

DavaID INT PRIMARY KEY IDENTITY(1,1),

DavaTuru NVARCHAR(100),

BaslangicTarihi DATE,

Durum NVARCHAR(50),

AvukatID INT,

MahkemeID INT,

FOREIGN KEY (AvukatID) REFERENCES Avukatlar(AvukatID),

FOREIGN KEY (MahkemeID) REFERENCES Mahkemeler(MahkemeID)

);

**4. Dava\_Tarafları Tablosu**

CREATE TABLE Dava\_Taraflari (

DavaID INT,

MuvekkilID INT,

TarafRol NVARCHAR(50),

PRIMARY KEY (DavaID, MuvekkilID),

FOREIGN KEY (DavaID) REFERENCES Davalar(DavaID),

FOREIGN KEY (MuvekkilID) REFERENCES Muvekkiller(MuvekkilID)

);

**5. Mahkemeler Tablosu**

CREATE TABLE Mahkemeler (

MahkemeID INT PRIMARY KEY IDENTITY(1,1),

MahkemeAdi NVARCHAR(100),

Sehir NVARCHAR(50),

Adres NVARCHAR(200)

);

**6. Belgeler Tablosu**

CREATE TABLE Belgeler (

BelgeID INT PRIMARY KEY IDENTITY(1,1),

DavaID INT,

BelgeTuru NVARCHAR(50),

YuklenmeTarihi DATE,

FOREIGN KEY (DavaID) REFERENCES Davalar(DavaID)

);

**7. Ödemeler Tablosu**

CREATE TABLE Odemeler (

OdemeID INT PRIMARY KEY IDENTITY(1,1),

MuvekkilID INT,

Tutar DECIMAL(10, 2),

Tarih DATE,

OdemeTuru NVARCHAR(50),

FOREIGN KEY (MuvekkilID) REFERENCES Muvekkiller(MuvekkilID)

);

**8. Şikayetler Tablosu**

CREATE TABLE Sikayetler (

SikayetID INT PRIMARY KEY IDENTITY(1,1),

DavaID INT,

SikayetMetni NVARCHAR(MAX),

SikayetTarihi DATE,

FOREIGN KEY (DavaID) REFERENCES Davalar(DavaID)

);

### TABLOLARA ÖRNEK OLARAK VERİ EKLEME:

### 1. ****Avukatlar Tablosu****

INSERT INTO Avukatlar (Ad, Soyad, UzmanlikAlanı, BaroNumarasi)

VALUES

('Ahmet', 'Yılmaz', 'Ceza Hukuku', 'BR12345'),

('Elif', 'Demir', 'Aile Hukuku', 'BR23456'),

('Kemal', 'Kaya', 'Ticaret Hukuku', 'BR34567');

2. **Müvekkiller Tablosu**

INSERT INTO Muvekkiller (Ad, Soyad, Telefon, Eposta, Adres)

VALUES

('Mehmet', 'Can', '5551234567', 'mehmetcan@example.com', 'İstanbul, Kadıköy'),

('Ayşe', 'Öztürk', '5559876543', 'ayseozturk@example.com', 'Ankara, Çankaya'),

('Ali', 'Kurt', '5556543210', 'alikurt@example.com', 'İzmir, Karşıyaka');

3. **Mahkemeler Tablosu**

INSERT INTO Mahkemeler (MahkemeAdi, Sehir, Adres)

VALUES

('İstanbul 5. Ağır Ceza Mahkemesi', 'İstanbul', 'Adliye Sarayı, Çağlayan'),

('Ankara 2. Aile Mahkemesi', 'Ankara', 'Adliye Sarayı, Sıhhiye'),

('İzmir 3. Ticaret Mahkemesi', 'İzmir', 'Adliye Sarayı, Bayraklı');

4. **Davalar Tablosu**

INSERT INTO Davalar (DavaTuru, BaslangicTarihi, Durum, AvukatID, MahkemeID)

VALUES

('Dolandırıcılık', '2024-01-10', 'Devam Ediyor', 1, 1),

('Boşanma', '2024-02-15', 'Beklemede', 2, 2),

('Haksız Rekabet', '2024-03-01', 'Tamamlandı', 3, 3);

5. **Dava\_Tarafları Tablosu**

INSERT INTO Dava\_Taraflari (DavaID, MuvekkilID, TarafRol)

VALUES

(1, 1, 'Davacı'),

(2, 2, 'Davalı'),

(3, 3, 'Davacı');

6. **Belgeler Tablosu**

INSERT INTO Belgeler (DavaID, BelgeTuru, YuklenmeTarihi)

VALUES

(1, 'İddianame', '2024-01-15'),

(2, 'Dilekçe', '2024-02-20'),

(3, 'Karar', '2024-03-10');

7. **Ödemeler Tablosu**

INSERT INTO Odemeler (MuvekkilID, Tutar, Tarih, OdemeTuru)

VALUES

(1, 5000.00, '2024-01-12', 'Nakit'),

(2, 3000.00, '2024-02-18', 'Havale'),

(3, 7000.00, '2024-03-05', 'Kredi Kartı');

8. **Şikayetler Tablosu**

INSERT INTO Sikayetler (DavaID, SikayetMetni, SikayetTarihi)

VALUES

(1, 'Sanık suçlamaları reddediyor.', '2024-01-20'),

(2, 'Mahkeme tarihi sürekli erteleniyor.', '2024-02-25'),

(3, 'Avukatın iletişimi yetersiz.', '2024-03-15');

**1.Saklı Yordam: Yeni Ödeme Ekleme**

CREATE PROCEDURE YeniOdemeEkle

@MuvekkilID INT,

@Tutar DECIMAL(10, 2),

@Tarih DATE,

@OdemeTuru NVARCHAR(50)

AS

BEGIN

INSERT INTO Odemeler (MuvekkilID, Tutar, Tarih, OdemeTuru)

VALUES (@MuvekkilID, @Tutar, @Tarih, @OdemeTuru);

PRINT 'Yeni ödeme başarıyla eklendi.';

END;

EXEC YeniOdemeEkle @MuvekkilID = 1, @Tutar = 2000.00, @Tarih = '2024-12-29', @OdemeTuru = 'Kredi Kartı';

**2. Tetikleyici: Ödeme Eklendiğinde Ödeme Geçmişini Güncelle**

CREATE TABLE OdemeGecmisi (

MuvekkilID INT PRIMARY KEY,

ToplamTutar DECIMAL(15, 2),

SonOdemeTarihi DATE,

FOREIGN KEY (MuvekkilID) REFERENCES Muvekkiller(MuvekkilID)

);

CREATE TRIGGER OdemeGuncelleTrigger

ON Odemeler

AFTER INSERT

AS

BEGIN

UPDATE OdemeGecmisi

SET

ToplamTutar = ToplamTutar + INSERTED.Tutar,

SonOdemeTarihi = INSERTED.Tarih

FROM OdemeGecmisi

INNER JOIN INSERTED ON OdemeGecmisi.MuvekkilID = INSERTED.MuvekkilID;

INSERT INTO OdemeGecmisi (MuvekkilID, ToplamTutar, SonOdemeTarihi)

SELECT MuvekkilID, Tutar, Tarih

FROM INSERTED

WHERE NOT EXISTS (

SELECT 1 FROM OdemeGecmisi WHERE OdemeGecmisi.MuvekkilID = INSERTED.MuvekkilID ); END;

**Saklı yordamın Test İşlemleri:**

EXEC YeniOdemeEkle

@MuvekkilID = 1,

@Tutar = 1500.00,

@Tarih = '2024-12-30',

@OdemeTuru = 'Havale';

SELECT \* FROM OdemeGecmisi;

**Transaction Yönetimi:**

BEGIN TRY

BEGIN TRANSACTION;

INSERT INTO Odemeler (MuvekkilID, Tutar, Tarih, OdemeTuru)

VALUES (1, 4000.00, '2024-12-29', 'Kredi Kartı');

UPDATE Davalar

SET Durum = 'Ödeme Yapıldı'

WHERE DavaID = 1;

INSERT INTO Belgeler (DavaID, BelgeTuru, YuklenmeTarihi)

VALUES (99, 'Ödeme Makbuzu', '2024-12-29');

COMMIT;

PRINT 'Tüm işlemler başarıyla tamamlandı.';

END TRY

BEGIN CATCH

ROLLBACK;

PRINT 'Bir hata oluştu. İşlem geri alındı. Hata: ' + ERROR\_MESSAGE();

END CATCH;