



T.C

**FIRAT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**AKILLI İLAÇ TAKİBİ VE RANDEVU
YÖNLENDİRME SİSTEMİ
RAPOR**

HACER ÇADIRCI 220260004

EMİNE POLAT 220260014

İLKİN TANIK 220260072

Akıllı İlaç Takip ve Randevu Yönlendirme Sistemi

Bu raporda, Akıllı İlaç Takip Sistemi projemiz için tasarlanan veritabanı yapısı detaylı bir şekilde anlatılmaktadır. Veritabanı, iki ana şema ve bu şemalara bağlı tablolar üzerinden yapılandırılmıştır: **KullaniciYonetimi** ve **SaglikSistemi**.

Şema Yapısı

1. KullaniciYonetimi Şeması

2. Kullanıcıların yönetimi ile ilgili tabloları barındırır:

- Kullanıcılar Tablosu:** Sistemdeki tüm kullanıcıların temel bilgilerini tutar. Her kullanıcı bir **Kullanıcı_ID** ile tanımlanır. Kullanıcı rolleri, isim-soyisim, e-posta (benzersiz) ve şifre bilgileri bu tabloda yer alır.
- Hasta Yakını Tablosu:** Hasta yakınlarının bilgilerini saklar. Hasta ve kullanıcı bilgileri ile ilişkilendirilmiştir. Ayrıca, yakınlık durumunu belirtmek için bir alan içerir.
- Sağlık Profesyoneli Tablosu:** Sağlık profesyonellerinin bilgilerini içerir. Her sağlık profesyoneli bir kullanıcı hesabına sahiptir ve bu tablo, **Kullanıcılar** tablosuyla ilişkilendirilmiştir.

3. SaglikSistemi Şeması

Sağlıkla ilgili verilerin tutulduğu şemadır:

- Hasta Tablosu:** Hastaların temel bilgilerini içerir. Bu tabloda cinsiyet, doğum tarihi, kronik hastalıklar, teşhis tarihi ve alerji bilgileri saklanır. Her hasta bir kullanıcı hesabı ile ilişkilendirilmiştir.
- İlaçlar Tablosu:** Sistemdeki ilaçların bilgilerini içerir. İlaç adı, dozaj, kullanım talimatı ve yan etkiler bu tabloda tutulur.
- İlaç Kullanımı Tablosu:** Hastaların ilaç kullanım bilgilerini içerir. Hangi hastanın hangi ilacı, hangi tarihler arasında ve hangi sıklıkta kullandığı bu tabloda yer alır. Ayrıca, son kullanım tarihi de kayıt altına alınır.
- Sağlık Profesyoneli-Hasta Tablosu:** Sağlık profesyonelleri ile takip ettikleri hastalar arasındaki ilişkiyi yönetir. Bu tablo, profesyonel ve hasta bilgilerinin yanı sıra ilişkilendirme tarihini içerir.
- Bildirim Durum Tablosu:** İlaç kullanımı ile ilgili bildirimlerin durumlarını tanımlar. Örneğin, "Tamamlandı", "Bekleniyor" gibi durumlar burada tutulur.
- Takvim ve Bildirimler Tablosu:** İlaç kullanım bildirimlerini ve hatırlatmalarını yönetir. Her bildirim, ilgili ilaç kullanımı ve durum bilgisi ile ilişkilendirilmiştir. Kullanım tarihi ve saati gibi detaylar da bu tabloda yer alır.

Sistem İşleyişi

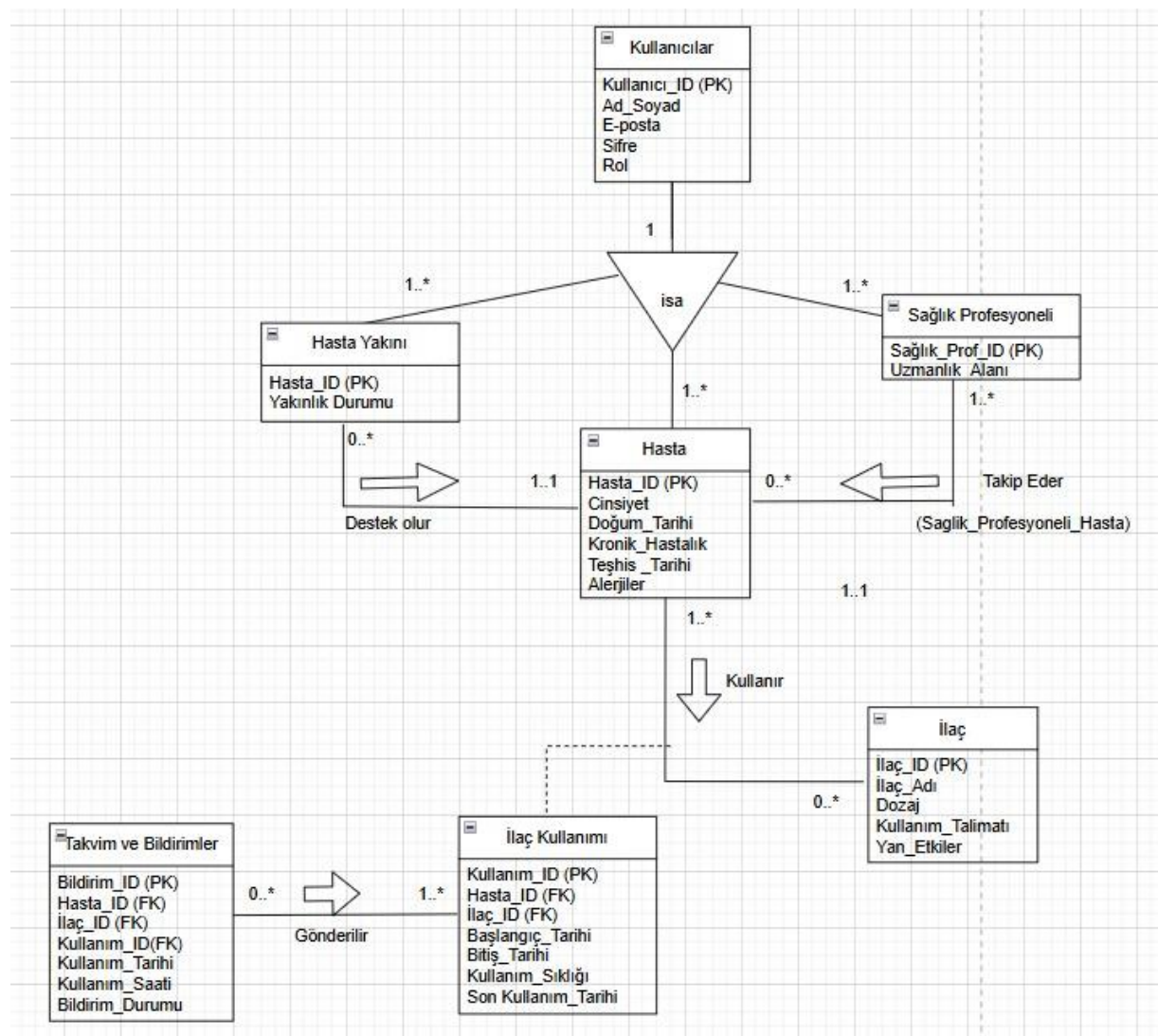
- Kullanıcı Girişi:** Sisteme giriş yapan kullanıcı, rolüne göre farklı verilere erişim sağlar. Hasta ve hasta yakınları, ilaç takibi ve geçmiş bilgilerini görüntüleyebilirken, sağlık profesyonelleri hastaların durumunu takip edebilir.
- İlaç Takibi:** Hasta ilaç kullanım bilgileri **İlaç Kullanımı Tablosu** üzerinden takip edilir. Sistem, ilaçların kullanım tarihlerini ve sıklığını kontrol ederek hastaya hatırlatmalar yapar.

- **Bildirim Yönetimi: Takvim ve Bildirimler Tablosu**, ilaç hatırlatmalarını ve durumlarını yönetir. Bu tablo, hasta ve sağlık profesyoneli arasındaki iletişimi destekler.
- **Hasta ve Sağlık Profesyoneli İlişkisi**: Sağlık profesyonelleri, takip ettikleri hastaların bilgilerine erişebilir. Bu ilişki, **Sağlık Profesyoneli-Hasta Tablosu** ile yönetilir.

Sonuç

Bu yapı, kullanıcı dostu bir sağlık yönetim sistemi oluşturmak için tasarlanmıştır. Veritabanı, ilişkisel yapısı sayesinde hastaların ilaç kullanımını düzenli bir şekilde takip etmeyi ve sağlık profesyonellerinin hasta bilgilerine kolayca erişmesini sağlar. Sistemin bildirim mekanizması, ilaçların zamanında kullanılmasını teşvik eder ve hasta yakınlarını da süreçlere dahil eder.

E-R DİYAGRAMI



İLİŞKİSEL ŞEMASI

Kullanıcılar (Kullanıcı_ID, Rol, İsim, Soyisim, Eposta, Şifre)

Hasta (*Hasta_ID*, Kullanıcı_ID, Cinsiyet, Doğum_Tarihi, Kronik_Hastalık, Teşhis_Tarihi, Alerjiler)

- **Yabancı Anahtar (FK):** Kullanıcı_ID → Kullanıcılar.Kullanıcı_ID

Hasta_Yakını (*Yakın_ID*, Hasta_ID, Kullanıcı_ID, Yakınlık_Durumu)

- **Yabancı Anahtar (FK):** Hasta_ID → Hasta.Hasta_ID
- **Yabancı Anahtar (FK):** Kullanıcı_ID → Kullanıcılar.Kullanıcı_ID

Sağlık_Profesyoneli (*Sağlık_Prof_ID*, Kullanıcı_ID, Uzmanlık_Alanı)

- **Yabancı Anahtar (FK):** Kullanıcı_ID → Kullanıcılar.Kullanıcı_ID

İlaçlar (*İlaç_ID*, İlaç_Adı, Dozaj, Kullanım_Talimatı, Yan_Etkiler)

İlaç_Kullanımı (*Kullanım_ID*, Hasta_ID, İlaç_ID, Başlangıç_Tarihi, Bitiş_Tarihi, Kullanım_Sıklığı, Son_Kullanım_Tarihi)

- **Yabancı Anahtar (FK):** Hasta_ID → Hasta.Hasta_ID
- **Yabancı Anahtar (FK):** İlaç_ID → İlaçlar.İlaç_ID

Sağlık_Profesyoneli_Hasta (*Sağlık_Prof_Hasta_ID*, Sağlık_Prof_ID, Hasta_ID, Tarih)

- **Yabancı Anahtar (FK):** Sağlık_Prof_ID → Sağlık_Profesyoneli.Sağlık_Prof_ID
- **Yabancı Anahtar (FK):** Hasta_ID → Hasta.Hasta_ID

Bildirim_Durum (*Durum_ID*, Bildirim_Durumu)

Takvim_Bildirimler (*Bildirim_ID*, Kullanım_ID, Kullanım_Tarihi, Kullanım_Saati, Durum_ID)

- **Yabancı Anahtar (FK):** Kullanım_ID → İlaç_Kullanımı.Kullanım_ID
- **Yabancı Anahtar (FK):** Durum_ID → Bildirim_Durum.Durum_ID

-- Şema Oluşturma

CREATE SCHEMA KullaniciYonetimi AUTHORIZATION dbo;

CREATE SCHEMA SaglikSistemi AUTHORIZATION dbo;

-- Kullanıcılar Tablosu

CREATE TABLE KullaniciYonetimi.Kullanıcılar (

Kullanıcı_ID INT IDENTITY(1,1) PRIMARY KEY,

Rol NVARCHAR(50) NOT NULL,

İsim NVARCHAR(100) NOT NULL,

Soyisim NVARCHAR(100) NOT NULL,

Eposta NVARCHAR(150)

UNIQUE NOT NULL,

```
Şifre NVARCHAR(200) NOT NULL );
```

```
-- Hasta Yakını Tablosu
```

```
CREATE TABLE KullaniciYonetimi.Hasta_Yakını (  
  
Yakin_ID INT IDENTITY(1,1) PRIMARY KEY,  
  
Hasta_ID INT NOT NULL,  
  
Kullanici_ID INT UNIQUE NOT NULL,  
  
Yakınlık_Durumu NVARCHAR(50),  
  
FOREIGN KEY (Hasta_ID) REFERENCES SaglikSistemi.Hasta(Hasta_ID),  
  
FOREIGN KEY (Kullanici_ID) REFERENCES KullaniciYonetimi.Kullanıcılar(Kullanici_ID)  
  
);
```

```
-- Sağlık Profesyoneli Tablosu
```

```
CREATE TABLE KullaniciYonetimi.Sağlık_Profesyoneli (  
  
Sağlık_Prof_ID INT IDENTITY(1,1) PRIMARY KEY,  
  
Kullanici_ID INT UNIQUE NOT NULL,  
  
Uzmanlık_Alanı NVARCHAR(100),  
  
FOREIGN KEY (Kullanici_ID) REFERENCES KullaniciYonetimi.Kullanıcılar(Kullanici_ID)  
  
);
```

```
-- Hasta Tablosu
```

```
CREATE TABLE SaglikSistemi.Hasta (  
  
Hasta_ID INT IDENTITY(1,1) PRIMARY KEY,  
  
Kullanici_ID INT UNIQUE NOT NULL,  
  
Cinsiyet NVARCHAR(10),  
  
Doğum_Tarihi DATE,  
  
Kronik_Hastalık NVARCHAR(255),  
  
Teşhis_Tarihi DATE,  
  
Alerjiler NVARCHAR(255),
```

```
FOREIGN KEY (Kullanıcı_ID) REFERENCES KullaniciYonetimi.Kullanıcılar(Kullanıcı_ID)

);
```

-- İlaçlar Tablosu

```
CREATE TABLE SaglikSistemi.İlaçlar (

İlaç_ID INT IDENTITY(1,1) PRIMARY KEY,

İlaç_Adı NVARCHAR(100) NOT NULL,

Dozaj NVARCHAR(50),

Kullanım_Talimatı NVARCHAR(255),

Yan_Etkiler NVARCHAR(255)

);
```

-- İlaç Kullanımı Tablosu

```
CREATE TABLE SaglikSistemi.İlaç_Kullanımı (

Kullanım_ID INT IDENTITY(1,1) PRIMARY KEY,

Hasta_ID INT NOT NULL,

İlaç_ID INT NOT NULL,

Başlangıç_Tarihi DATE NOT NULL,

Bitiş_Tarihi DATE NOT NULL,

Kullanım_Sıklığı NVARCHAR(50),

Son_Kullanım_Tarihi DATE,

FOREIGN KEY (Hasta_ID) REFERENCES SaglikSistemi.Hasta(Hasta_ID),

FOREIGN KEY (İlaç_ID) REFERENCES SaglikSistemi.İlaçlar(İlaç_ID)

);
```

```
CREATE TABLE SaglikSistemi.Saglik_Profesyoneli_Hasta (

Sağlık_Prof_Hasta_ID INT IDENTITY(1,1) PRIMARY KEY,

Sağlık_Prof_ID INT NOT NULL,

Hasta_ID INT NOT NULL,
```

Tarih DATE,

FOREIGN KEY (Sağlık_Prof_ID) REFERENCES KullaniciYonetimi.Sağlık_Profesyoneli(Sağlık_Prof_ID),
FOREIGN KEY (Hasta_ID) REFERENCES SaglikSistemi.Hasta(Hasta_ID)

);

CREATE TABLE SaglikSistemi.Bildirim_Durum (

Durum_ID INT IDENTITY(1,1) PRIMARY KEY,

Bildirim_Durumu NVARCHAR(50) UNIQUE);

CREATE TABLE SaglikSistemi.Takvim_Bildirimler (

Bildirim_ID INT IDENTITY(1,1) PRIMARY KEY,

Kullanım_ID INT NOT NULL,

Kullanım_Tarihi DATE NOT NULL,

Kullanım_Saati TIME NOT NULL,

Durum_ID INT,

FOREIGN KEY (Kullanım_ID) REFERENCES SaglikSistemi.İlaç_Kullanımı(Kullanım_ID),

FOREIGN KEY (Durum_ID) REFERENCES SaglikSistemi.Bildirim_Durum(Durum_ID)

);

NORMALİZASYON İŞLEMİNİN UYGULANMASI

1. Kullanıcılar Tablosu

Kullanıcılar (Kullanıcı_ID, Rol, İsim, Soyisim, Eposta, Şifre)

- **Açıklama:**
 - Her kullanıcı için benzersiz bir ***Kullanıcı_ID*** (Primary Key) tanımlandı.
 - **Eposta** UNIQUE olarak belirlendi, böylece her kullanıcıya ait e-posta adresi tekil ve ayırt edici oldu.
 - Tüm sütunlar, yalnızca ***Kullanıcı_ID*** anahtarına bağımlıdır (tam fonksiyonel bağımlılık).
 - **3NF:** Hiçbir sütun diğer sütunlara transitif bağımlı değil.
 - **2. Hasta Tablosu**

Hasta (Hasta_ID, Kullanıcı_ID, Cinsiyet, Doğum_Tarihi, Kronik_Hastalık, Teşhis_Tarihi, Alerjiler)

- **Açıklama:**
 - ***Hasta_ID*** Primary Key olarak tanımlandı.
 - Kullanıcı bilgileriyle ilişki kurmak için ***Kullanıcı_ID*** Foreign Key olarak eklendi.

- Tüm sütunlar, yalnızca **Hasta_ID** anahtarına bağlıdır.
- **3NF**: Kronik hastalık, teşhis tarihi ve alerjiler gibi bilgiler transitif bağımlılıktan arındırıldı.

3. Hasta Yakını Tablosu

Hasta_Yakını (Yakın_ID, Hasta_ID, Kullanıcı_ID, Yakınlık_Durumu)

- **Açıklama:**
 - **Yakın_ID** Primary Key olarak tanımlandı.
 - Hasta bilgileriyle ilişki kurmak için **Hasta_ID** ve kullanıcı bilgileriyle ilişki kurmak için **Kullanıcı_ID** Foreign Key olarak eklendi.
 - **Yakınlık_Durumu**, yalnızca **Yakın_ID** anahtarına bağlıdır.
 - **3NF**: Hiçbir transitif bağımlılık yoktur.

4. Sağlık Profesyoneli Tablosu

Sağlık_Profesyoneli (Sağlık_Prof_ID, Kullanıcı_ID, Uzmanlık_Alanı)

- **Açıklama:**
 - **Sağlık_Prof_ID** Primary Key olarak tanımlandı.
 - Kullanıcı bilgileriyle ilişki kurmak için **Kullanıcı_ID** Foreign Key olarak eklendi.
 - Uzmanlık alanı, yalnızca **Sağlık_Prof_ID** anahtarına bağlıdır.
 - **3NF**: Transitif bağımlılık bulunmamaktadır.

5. İlaçlar Tablosu

İlaçlar (İlaç_ID, İlaç_Adı, Dozaj, Kullanım_Talimatı, Yan_Etkiler)

- **Açıklama:**
 - **İlaç_ID** Primary Key olarak tanımlandı.
 - İlaç adı, dozaj ve yan etkiler gibi bilgiler yalnızca **İlaç_ID** anahtarına bağlıdır.
 - **3NF**: İlaç adı gibi bilgiler arasında transitif bağımlılık yoktur.

6. İlaç Kullanımı Tablosu

İlaç_Kullanımı (Kullanım_ID, Hasta_ID, İlaç_ID, Başlangıç_Tarihi, Bitiş_Tarihi, Kullanım_Sıklığı, Son_Kullanım_Tarihi)

- **Açıklama:**
 - **Kullanım_ID** Primary Key olarak tanımlandı.
 - Hasta ve ilaç bilgileriyle ilişki kurmak için **Hasta_ID** ve **İlaç_ID** Foreign Key olarak eklendi.
 - Kullanım detayları (başlangıç ve bitiş tarihi, sıklık) yalnızca **Kullanım_ID** anahtarına bağlıdır.
 - **3NF**: Tüm sütunlar tam fonksiyonel bağımlıdır ve transitif bağımlılık yoktur.

7. Takvim Bildirimler Tablosu

Takvim_Bildirimler (Bildirim_ID, Kullanım_ID, Kullanım_Tarihi, Kullanım_Saati, Durum_ID)

- **Açıklama:**
 - **Bildirim_ID** Primary Key olarak tanımlandı.
 - İlaç kullanımıyla ilişki kurmak için **Kullanım_ID** Foreign Key olarak eklendi.
 - Bildirim durumu, yalnızca **Bildirim_ID** anahtarına bağlıdır.
 - **3NF**: Transitif bağımlılık bulunmamaktadır.

8. Bildirim Durum Tablosu

Bildirim_Durum (Durum_ID, Bildirim_Durumu)

- **Açıklama:**
 - **Durum_ID** Primary Key olarak tanımlandı.
 - Bildirim durumu, yalnızca **Durum_ID** anahtarına bağlıdır.
 - **3NF**: Tüm sütunlar tam fonksiyonel bağımlıdır ve transitif bağımlılık yoktur.

TABLolar ARASINDAKİ İLİŞKİLER ŞU ŞEKİLDEDİR

1. Kullanıcı → Hasta (1..1)

- **Açıklama:**

Her kullanıcı, sisteme kayıtlı bir hasta olabilir. Bu durumda, her kullanıcının yalnızca bir hasta kaydıyla ilişkisi olur.

Örneğin, bir kullanıcı kendi bilgileriyle sisteme giriş yapar ve sistemde sadece kendisi için bir hasta profili oluşturulur.

2. Kullanıcı → Sağlık Profesyoneli (1..1)

- **Açıklama:**

Her kullanıcı, aynı zamanda sağlık profesyoneli olabilir. Ancak bir kullanıcı sadece bir sağlık profesyoneli kaydıyla ilişkilendirilebilir.

Örneğin, bir doktor sisteme giriş yaptığında, sağlık profesyoneli olarak tek bir kayıt oluşturulur.

3. Kullanıcı → Hasta Yakını (1..N)

- **Açıklama:**

Bir kullanıcı birden fazla hasta yakınıyla ilişkili olabilir. Her hasta yakını ise yalnızca bir kullanıcıya bağlıdır.

Örneğin, bir ebeveyn sisteme giriş yaparak birden fazla çocuğunun (hastalarının) bilgilerini yönetebilir.

4. Hasta Yakını → Hasta (1..N)

- **Açıklama:**

Bir hasta yakını birden fazla hastayla ilişkili olabilir. Her hasta ise yalnızca bir hasta yakını

tarafından temsil edilebilir.

Örneğin, bir anne veya baba, birden fazla çocuğu için hasta bilgilerini yönetebilir.

5. Hasta ↔ Sağlık Profesyoneli (N..M)

- **Açıklama:**

Bir hasta birden fazla sağlık profesyoneli tarafından takip edilebilir ve bir sağlık profesyoneli birden fazla hastayı takip edebilir.

Örneğin, bir hasta hem dahiliye hem de kardiyoloji doktorları tarafından kontrol ediliyorken, aynı doktorlar başka hastaları da takip edebilir.

6. Hasta → İlaç Kullanımı (1..N)

- **Açıklama:**

Bir hasta birden fazla ilaç kullanımı kaydına sahip olabilir. Ancak her ilaç kullanımı yalnızca bir hastaya bağlıdır.

Örneğin, bir hasta farklı zamanlarda farklı ilaçlar kullanabilir ve bu ilaçların her biri, hastanın ilaç kullanım geçmişinde kaydedilir.

7. İlaç → İlaç Kullanımı (1..N)

- **Açıklama:**

Bir ilaç, birden fazla ilaç kullanımı kaydına dahil olabilir. Ancak her ilaç kullanımı yalnızca bir ilaç ile ilişkilidir.

Örneğin, "Aspirin" birden fazla hastaya reçete edilebilir, ancak bir ilaç kullanım kaydı sadece o ilaca aittir.

8. İlaç Kullanımı → Bildirimler (1..N)

- **Açıklama:**

Bir ilaç kullanımı kaydı, birden fazla bildirimle ilişkili olabilir. Ancak her bildirim yalnızca bir ilaç kullanımı kaydına bağlıdır.

Örneğin, bir hastaya sabah ve akşam ilacı içmesi için hatırlatma yapılırsa, her bildirim o hastanın belirli bir ilaç kullanımına ait olacaktır.

9. Hasta → Takip (1..N)

- **Açıklama:**

Bir hasta birden fazla takip kaydına sahip olabilir. Ancak her takip kaydı yalnızca bir hastaya bağlıdır.

Örneğin, bir hasta, düzenli olarak yapılan doktor kontrollerinde her kontrol için bir takip kaydına sahip olur.

10. Sağlık Profesyoneli → Takip (1..N)

- **Açıklama:**

Bir sağlık profesyoneli birden fazla takip kaydına sahip olabilir. Ancak her takip kaydı yalnızca bir sağlık profesyoneline bağlıdır.

Örneğin, bir doktor, birden fazla hastayı takip ediyor olabilir ve her hastanın kontrolü için ayrı bir takip kaydı oluşturulur.

TRIGGERLAR

--Aşağıdaki trigger, yeni bir hasta eklerken, kullanıcının rolünün doğru olup olmadığını kontrol eder.

--Eğer eklenen kullanıcının rolü "Hasta" değilse, işlemi iptal eder ve bir hata mesajı döner.

--Sağlık profesyoneli veya hasta yakını için de ayrı trigger'lar yazılabilir.

```
CREATE TRIGGER TRG_ValidateHastaRole
```

```
ON SaglikSistemi.Hasta
```

```
AFTER INSERT
```

```
AS
```

```
BEGIN
```

```
    SET NOCOUNT ON;
```

```
    IF EXISTS (
```

```
        SELECT 1
```

```
        FROM Inserted I
```

```
        JOIN KullaniciYonetimi.Kullanıcılar K
```

```
        ON I.Kullanıcı_ID = K.Kullanıcı_ID
```

```
        WHERE K.Rol <> 'Hasta'
```

```
    )
```

```
    BEGIN
```

```
        ROLLBACK TRANSACTION;
```

```
    THROW 50000, 'Eklenen kullanıcı hasta rolüne sahip olmalıdır.',1;
```

```
    END
```

```
END;
```

-- sağlık profesyoneli olan bir kullanıcı oluşturduk

```
INSERT INTO KullaniciYonetimi.Kullanıcılar (Rol, İsim, Soyisim, Eposta, Şifre)
```

```
VALUES ('Sağlık Profesyoneli', 'Mehmet', 'Kaya', 'mehmet@example.com', 'sifre456');
```

-- bu sağlık profesyonelini Hasta tablosuna eklemeye çalıştığımızda hata alırsız

```
INSERT INTO SaglikSistemi.Hasta (Kullanıcı_ID, Cinsiyet, Doğum_Tarihi)
```

```
VALUES (10, 'Erkek', '1995-05-15');
```

-- Bu işlem hata verecek ve rollback yapılacaktır.

SQL SORGULARI

----- Kullanıcılar Tablosu

INSERT INTO KullaniciYonetimi.Kullanıcılar (Rol, İsim, Soyisim, Eposta, Şifre)

VALUES

('Hasta', 'Emine', 'Polat', 'ep.60@example.com', '123456'),

('Hasta', 'İlkin', 'Tanık', 'it.56@example.com', 'abcdef'),

('Sağlık Profesyoneli', 'rabia', 'cad', 'rb.23@example.com', '2323'),

('Hasta Yakını', 'Hacer', 'Cadirci', 'hcd@example.com', '2323'),

'Sağlık Profesyoneli', 'Medine', 'Şa', 'medi@example.com', '23223');

INSERT INTO KullaniciYonetimi.Kullanıcılar (Rol, İsim, Soyisim, Eposta, Şifre)

VALUES

('Hasta Yakını', 'magnus', 'carlsen', 'chess@example.com', '2323');

	Kullanıcı_ID	Rol	İsim	Soyisim	Eposta	Şifre
1	1	Hasta	Emine	Polat	ep.60@example.com	123456
2	2	Hasta	İlkin	Tanık	it.56@example.com	abcdef
3	3	Sağlık Profesyoneli	rabia	cad	rb.23@example.com	2323
4	4	Hasta Yakını	Hacer	Cadirci	hcd@example.com	2323
5	5	Sağlık Profesyoneli	Medine	Şa	medi@example.com	23223

----- Hasta Tablosu

INSERT INTO SaglikSistemi.Hasta (Kullanıcı_ID, Cinsiyet, Doğum_Tarihi, Kronik_Hastalık, Teşhis_Tarihi, Alerjiler)

VALUES

(1, 'Kadın', '1990-03-15', 'Diyabet', '2015-06-10', 'Penisilin'),

(2, 'Erkek', '1985-11-02', 'Astım', '2012-09-18', 'Polen');

	Hasta_ID	Kullanıcı_ID	Hasta_Adi	Hasta_Soyadi	Hasta_Eposta	Cinsiyet	Doğum_Tarihi	Kronik_Hastalık	Teşhis_Tarihi	Alerjiler
1	1	1	Emine	Polat	ep.60@example.com	Kadın	1990-03-15	Diyabet	2015-06-10	Penisilin
2	2	2	İlkin	Tanık	it.56@example.com	Erkek	1985-11-02	Astım	2012-09-18	Polen

----- Hasta Yakını Tablosu

```
INSERT INTO KullaniciYonetimi.Hasta_Yakini (Hasta_ID, Kullanici_ID, Yakinlik_Durumu)
```

```
VALUES
```

```
(1, 4, 'Anne');
```

-----bir hasta yakınının en fazla bir hastası olabilir. o yüzden bu kod duplicate key value hatası verecektir -----çünkü daha önce kullanıcı id 4 olan kişiyi bir hasta ile ilişkilendirmiştik.

```
INSERT INTO KullaniciYonetimi.Hasta_Yakini (Hasta_ID, Kullanici_ID, Yakinlik_Durumu)
```

```
VALUES
```

```
(2, 4, 'Anne');
```

	Yakin_ID	Hasta_ID	Hasta_Adi	Hasta_Soyadi	Yakin_Adi	Yakin_Soyadi	Yakinlik_Durumu
1	1	1	Emine	Polat	Hacer	Cadirci	Anne
2	3	1	Emine	Polat	magnus	carlsen	Baba

--ama bir hastanın birden fazla yakını olabilir. bu durumda 1 idli hastaya daha önce bir hasta yakını -- atamıştık fakat yeni bir hasta yakını daha ekleyebildik.

```
INSERT INTO KullaniciYonetimi.Hasta_Yakini (Hasta_ID, Kullanici_ID, Yakinlik_Durumu)
```

```
VALUES
```

```
(1, 6, 'Abla');
```

----- Sağlık Profesyoneli Tablosu

```
INSERT INTO KullaniciYonetimi.Saglik_Profesyoneli (Kullanici_ID, Uzmanlik_Alani)
```

```
VALUES
```

```
(3, 'Kardiyoloji'), (4, 'Onkoloji');
```

	Saglik_Prof_ID	Kullanici_ID	Uzmanlik_Alani
1	1	3	Kardiyoloji
2	2	4	Onkoloji

----- İlaçlar Tablosu

```
INSERT INTO SaglikSistemi.Ilaclar (Ilac_Adi, Dozaj, Kullanim_Talimatı, Yan_Etkiler)
```

```
VALUES
```

```
('Parol', '500 mg', 'Günde 3 kez', 'Baş dönmesi'),
```

```
('Aspirin', '100 mg', 'Yemekten sonra', 'Mide bulantısı'),
```

('Metformin', '850 mg', 'Sabah ve akşam', 'Baş ağrısı'),

('Ventolin', '100 mcg', 'Günde 2 kez', 'Boğaz kuruluğu'),

('Lipitor', '20 mg', 'Akşam yatmadan önce', 'Kas ağrısı');

	İlaç_ID	İlaç_Adı	Dozaj	Kullanım_Talimatı	Yan_Etkiler
1	1	Parol	500 mg	Günde 3 kez	Baş dönmesi
2	2	Aspirin	100 mg	Yemekten sonra	Mide bulantısı
3	3	Metformin	850 mg	Sabah ve akşam	Baş ağrısı
4	4	Ventolin	100 mcg	Günde 2 kez	Boğaz kuruluğu
5	5	Lipitor	20 mg	Akşam yatmadan önce	Kas ağrısı

----- İlaç Kullanımı Tablosu

INSERT INTO SaglikSistemi.İlaç_Kullanımı (Hasta_ID, İlaç_ID, Başlangıç_Tarihi, Bitiş_Tarihi, Kullanım_Sıklığı, Son_Kullanım_Tarihi)

VALUES

(1, 1, '2023-01-01', '2023-03-01', 'Günde 3 kez', NULL),

(2, 2, '2022-05-15', '2022-12-15', 'Günde 1 kez', NULL);

	Kullanım_ID	Hasta_ID	Hasta_Adı	Hasta_Soyadı	İlaç_ID	İlaç_Adı	Başlangıç_Tarihi	Bitiş_Tarihi	Kullanım_Sıklığı	Son_Kullanım_Tarihi
1	1	1	Emine	Polat	1	Parol	2023-01-01	2023-03-01	Günde 3 kez	NULL
2	2	2	İlkin	Tanık	2	Aspirin	2022-05-15	2022-12-15	Günde 1 kez	NULL

----Bir hasta birden fazla ilaç kullanabilir

INSERT INTO SaglikSistemi.İlaç_Kullanımı (Hasta_ID, İlaç_ID, Başlangıç_Tarihi, Bitiş_Tarihi, Kullanım_Sıklığı, Son_Kullanım_Tarihi)

VALUES

(1, 2, '2024-01-01', '2023-03-01', 'Günde 3 kez', NULL);

----bir ilacı da birden fazla hasta kullanabilir

INSERT INTO SaglikSistemi.İlaç_Kullanımı (Hasta_ID, İlaç_ID, Başlangıç_Tarihi, Bitiş_Tarihi, Kullanım_Sıklığı, Son_Kullanım_Tarihi)

VALUES

(2, 1, '2024-01-01', '2023-03-01', 'Günde 3 kez', NULL);

----- Sağlık Profesyoneİi-Hasta Tablosu

INSERT INTO SaglikSistemi.Sağlık_Profesyoneİi_Hasta (Sağlık_Prof_ID, Hasta_ID, Tarih)

VALUES

(1, 1, '2023-02-01'),

(2, 2, '2023-03-15');

	ID	Doktor_ID	Doktor_Adı	Doktor_Soyadı	Hasta_ID	Hasta_Adı	Hasta_Soyadı	Görüşme_Tarihi
1	1	1	rabia	cad	1	Emine	Polat	2023-02-01
2	2	2	Hacer	Cadirci	2	İlkin	Tanık	2023-03-15

---bir sağlık profesyonelinin takip ettiği birden fazla hasta olabilir ---aynı hastayı takip eden birden fazla sağlık profesyoneli olabilir

INSERT INTO SaglikSistemi.Saglik_Profesyoneli_Hasta (Sağlık_Prof_ID, Hasta_ID, Tarih)

VALUES

(1, 2, '2023-02-01');

----- Bildirim Durum Tablosu

INSERT INTO SaglikSistemi.Bildirim_Durum (Bildirim_Durumu)

VALUES

('Tamamlandı'), ('Bekliyor'), ('İptal Edildi'), ('Başarısız'), ('Ertelendi');

	Durum_ID	Bildirim_Durumu
1	4	Başarısız
2	2	Bekliyor
3	5	Ertelendi
4	3	İptal Edildi
5	1	Tamamlandı

----- Takvim Bildirimler Tablosu

INSERT INTO SaglikSistemi.Takvim_Bildirimler (Kullanım_ID, Kullanım_Tarihi, Kullanım_Saati, Durum_ID)

VALUES

(1, '2022-07-10', '08:00:00', 1), (2, '2019-11-22', '09:00:00', 2);

	Bildirim_ID	Kullanım_Tarihi	Kullanım_Saati	Durum_ID	Durum_Bilgisi	Hasta_ID	Hasta_Adı	Hasta_Soyadı	İlaç_ID	İlaç_Adı
1	1	2022-07-10	08:00:00.00000000	1	Tamamlandı	1	Emine	Polat	1	Parol
2	2	2019-11-22	09:00:00.00000000	2	Bekliyor	2	İlkin	Tanık	2	Aspirin

----bir ilaç kullanımı için birden fazla bildirim gönderilebilir

```
INSERT INTO SaglikSistemi.Takvim_Bildirimler (Kullanım_ID, Kullanım_Tarihi, Kullanım_Saati, Durum_ID)
```

```
VALUES
```

```
(1, '2016-07-18', '08:00:00', 2);
```

---bir bildirim farklı ilaç kullanımları için gönderilebilir

```
INSERT INTO SaglikSistemi.Takvim_Bildirimler (Kullanım_ID, Kullanım_Tarihi, Kullanım_Saati, Durum_ID)
```

```
VALUES
```

```
(2, '2020-05-21', '08:00:00', 1);
```

STORED PROCEDURE

--Amaç

--Hastanın ilaç kullanımını kaydetmek (başlangıç ve bitiş tarihleri ile birlikte).

--Oluşan kullanım için bildirimler oluşturmak (her gün için bildirim eklemek).

--Hatalı bir işlemde tüm değişiklikleri geri almak (rollback).

---Hem ilaç kullanımı hem de bildirim ekleme işlemleri tek bir işlemde yürütülür. Hata durumunda tüm değişiklikler geri alınır.

--- İşlemler modüler hale getirilir ve parametrelerle esnek bir yapı sağlanır.

---İşlemin başarılı veya başarısız olmasına göre değişikliklerin durumu net bir şekilde yönetilir.

```
CREATE PROCEDURE AddMedicationAndNotification
```

```
    @HastaID INT, @IlacID INT,
```

```
    @BaslangicTarihi DATE, @BitisTarihi DATE,
```

```
    @KullanimSikligi NVARCHAR(50),
```

```
    @BildirimTarihi DATE
```

```
AS
```

```
BEGIN
```

```
    BEGIN TRY
```



```

BEGIN TRANSACTION;

-- İlaç kullanımı ekle
INSERT INTO SaglikSistemi.İlaç_Kullanımı (Hasta_ID, İlaç_ID,
Başlangıç_Tarihi, Bitiş_Tarihi, Kullanım_Sıklığı, Son_Kullanım_Tarihi)
VALUES (@HastaID, @IlacID, @BaslangicTarihi, @BitisTarihi,
@KullanımSikligi, NULL);

-- Yeni eklenen Kullanım_ID'yi al
DECLARE @YeniKullanımID INT = SCOPE_IDENTITY();

-- Bildirim ekle
INSERT INTO SaglikSistemi.Takvim_Bildirimler (Kullanım_ID,
Kullanım_Tarihi, Kullanım_Saati, Durum_ID)
VALUES (@YeniKullanımID, @BildirimTarihi, '08:00:00', 2);

COMMIT TRANSACTION;
PRINT 'Başarılı: İlaç kullanımı ve bildirim eklendi.';
END TRY
BEGIN CATCH
    ROLLBACK TRANSACTION;
    PRINT 'Hata: İşlem sırasında bir hata oluştu. Tüm değişiklikler geri
alındı.';
    PRINT ERROR_MESSAGE();
END CATCH

END;

```

--- başlangıç tarihi bitiş tarihinden önce olduğu için sorun olmaz.

----başarılı senaryodur.

EXEC AddMedicationAndNotification

@HastaID = 1,

@IlacID = 1,

@BaslangicTarihi = '2025-01-01',

@BitisTarihi = '2025-03-01',

@KullanımSikligi = 'Günde 3 kez',

@BildirimTarihi = '2025-01-02';

----başlangıç tarihi bitiş tarihinden sonra olduğu için hata verir

----işlem geri alınır

EXEC AddMedicationAndNotification

@HastaID = 1,

@IlacID = 1,

@BaslangicTarihi = '2025-01-01',

@BitisTarihi = '2024-12-31',-- Başlangıç tarihi, bitiş tarihinden sonra

@KullanimSikligi = 'Günde 3 kez',

@BildirimTarihi = '2025-01-02';

