

Bu projede, bir hastane veritabanı uygulaması geliştirilmesi hedeflenmiştir. Veritabanı, hastane yönetimi ve işlemleri için gerekli olan çeşitli bilgileri depolamak amacıyla tasarlanmıştır. Bu amaçla, beş temel tablo oluşturulmuştur: Doktor\_Bilgileri, Hasta\_Bilgileri, Randevular, TibbiKayıtlar ve Faturalandırma.

Doktor\_Bilgileri tablosu, hastanede çalışan doktorların bilgilerini içerir. Bu tabloda her doktor için benzersiz bir kimlik numarası (DoktorID) atanmış, doktorların adları, soyadları, uzmanlık alanları, işe giriş tarihleri, aktiflik durumları ve iletişim bilgileri (telefon, e-posta ve adres) gibi bilgiler yer almaktadır.

Hasta\_Bilgileri tablosu ise hastaların bilgilerini saklamak için kullanılır. Bu tabloda, her hasta için benzersiz bir kimlik numarası (HastaID) bulunmaktadır. Hastaların adları, soyadları, doğum tarihleri, cinsiyetleri ve iletişim bilgileri (telefon, e-posta ve adres) bu tabloda depolanır.

Randevular tablosu, hastaların doktorlardan aldıkları randevuları yönetmek için oluşturulmuştur. Bu tabloda, her randevu için benzersiz bir kimlik numarası (RandevuID) bulunur. Ayrıca, randevunun alındığı hasta ve doktor kimlik numaraları, randevu tarihi, detayları ve durumu gibi bilgiler de yer alır.

TibbiKayıtlar tablosu, hastaların tıbbi kayıtlarını saklamak için kullanılır. Bu tabloda her kayıt için benzersiz bir kimlik numarası (KayitID) atanır. Doktor ve hasta kimlik numaraları, kayıt tarihi, tanı, tedavi ve notlar gibi bilgiler bu tabloda depolanır.

Faturalandırma tablosu, hastane hizmetlerinin faturalandırılması ile ilgili bilgileri içerir. Bu tabloda her fatura için benzersiz bir kimlik numarası (FaturaID) bulunur. Ayrıca, fatura ile ilgili hasta kimlik numarası, tutar, fatura tarihi ve ödeme durumu gibi bilgiler de yer alır.

## STORRED PROCEDURE

**Randevu oluşturduğumuz zaman 1000 liralık randevu oluşturma ücreti ödenmemiş olarak faturalandırma tablosuna eklenir.**

```
EXEC sp_AddRandevuVeFatura
    @HastaID = 5,
    @DoktorID = 6,
    @RandevuTarihi = '2024-05-30 13:00:00',
    @Detay = 'İlk muayene',
    @Durum = 'Planlandı';
```

```
SELECT * FROM Faturalandırma
WHERE HastaID = 5
ORDER BY FaturaTarihi DESC
```

```
SELECT * FROM Randevular
WHERE HastaID = 5
ORDER BY RandevuTarihi DESC
```

## TRIGGER

**Bir randevu oluşturulduğunda ilgili doktorun Aktif hasta sayısı 1 tane artar.**

```
SELECT * FROM Doktor_Bilgileri
```

```
INSERT INTO Randevular (HastaID, DoktorID, RandevuTarihi, Detay, Durum)
VALUES (8, 13, '2024-05-28 16:00:00', 'Endoskopi', 'Planlandı');
```

```
SELECT * FROM Randevular
WHERE HastaID = 8
```

## VIEW

Hastanede hangi bölümün nasıl iş yaptığını görmek için 2 adet view oluşturulmuştur. Birincisinde Bölüme göre kaç randevu olduğu, ikincisi de bu bölümlerin hastaneye ne kadar kazandırdığını gösterir.

```
SELECT * FROM vw_UzmanlikAlaninaGoreToplamRandevu;
```

```
SELECT * FROM vw_UzmanlikAlaninaGoreToplamFatura;
```

## SQL SORGULARI

### 1. Belirli Bir Doktorun Randevularını Listele

```
SELECT
    r.RandevuID,
    r.HastaID,
    d.DoktorAdı,
    d.DoktorSoyadı,
    d.UzmanlikAlanı AS DoktorBolumu,
    h.Ad AS HastaAdı,
    h.Soyad AS HastaSoyadı,
    r.RandevuTarihi,
    r.Detay,
    r.Durum
FROM
    Randevular r
JOIN
    Hasta_Bilgileri h ON r.HastaID = h.HastaID
JOIN
    Doktor_Bilgileri d ON r.DoktorID = d.DoktorID
WHERE
    r.DoktorID = 13;
```

## 2. Belirli Bir Hastanın Tüm Randevularını Listeleme

```
SELECT
    r.RandevuID,
    r.HastaID,
    h.Ad AS HastaAdı,
    h.Soyad AS HastaSoyadı,
    r.DoktorID,
    d.DoktorAdı,
    d.DoktorSoyadı,
    r.RandevuTarihi,
    r.Detay,
    r.Durum
FROM
    Randevular r
JOIN
    Hasta_Bilgileri h ON r.HastaID = h.HastaID
JOIN
    Doktor_Bilgileri d ON r.DoktorID = d.DoktorID
WHERE
    r.HastaID = 13;
```

## 3. Tüm Doktorların Toplam Randevu Sayısını Listeleme

```
SELECT
    d.DoktorAdı,
    d.DoktorSoyadı,
    COUNT(r.RandevuID) AS ToplamRandevuSayisi
FROM
    Doktor_Bilgileri d
LEFT JOIN
    Randevular r ON d.DoktorID = r.DoktorID
GROUP BY
    d.DoktorAdı, d.DoktorSoyadı
ORDER BY
    ToplamRandevuSayisi DESC;
```

## 4. Ödenmemiş Faturaları Listeleme

```
SELECT
    f.FaturalID,
    f.HastaID,
    h.Ad AS HastaAdı,
    h.Soyad AS HastaSoyadı,
    f.Tutar,
    f.FaturaTarihi
FROM
```

```
Faturalandırma f
JOIN
  Hasta_Bilgileri h ON f.HastaID = h.HastaID
WHERE
  f.OdemeDurumu = 0;
```

### **5.Toplam Borç**

```
SELECT
  h.HastaID,
  h.Ad AS HastaAdı,
  h.Soyad AS HastaSoyadı,
  SUM(f.Tutar) AS ToplamBorc
FROM
  Hasta_Bilgileri h
JOIN
  Faturalandırma f ON h.HastaID = f.HastaID
WHERE
  f.OdemeDurumu = 0
GROUP BY
  h.HastaID, h.Ad, h.Soyad
ORDER BY
  ToplamBorc DESC;
```