daire, logo, simge, sembol, ticari marka içeren bir resim

Açıklama otomatik olarak oluşturuldu

**T.C.**

**EGE ÜNİVERSİTESİ**

**TİRE KUTSAN MESLEK YÜKSEKOKULU**

1-) BURCU DİLAY BOZDALI

2-) BERFİN SEFEROĞLU

3-) MURATHAN SARGIN

4-) SARA BERİL KARAMAN

VERİ TABANI II PROJESİ

**Danışman**

**Öğr. Gör. Yeşim AKTAŞ**

İzmir, 2024

**BEYANNAME**

Proje içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi, görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu, kullanılan verilerde herhangi bir tahrifat yapmadığımı ve bu projenin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir proje çalışması olarak sunmadığımı beyan ederim.

**…/…/………**

**1-) Burcu Dilay BOZDALI**

**2-)Berfin SEFEROĞLU**

**3-)Murathan SARGIN**

**4-)Sara Beril KARAMAN**

ÖNSÖZ VE TEŞEKKÜR

Bu projeyi yaratmamızda büyük bir payı olan ve bizi bu nedenle daha da yakınlaştırarak grup çalışmalarıyla ilgili ilham olan sevgili Yeşim Aktaş Hocama tüm samimiyetimle teşekkür ediyorum. Bunun yanında benim için çok kıymetli olan bu süreci karşılıklı anlayış ve saygı ile yürüttüğümüz grup arkadaşlarıma da çok teşekkür ediyorum.

* Sara Beril KARAMAN

Bu projeyi yapma fırsatı veren Yeşim Aktaş'a sonsuz teşekkürlerimi sunuyorum, büyük bir adım attığımız eğitim ve öğrenim hayatımızda büyük destekçimiz olan öğretmenlerimize de ayriyeten teşekkür ederim.

* Berfin SEFEROĞLU

Bu öğretim yılında, başta hocalarım olmak üzere sonrasında bu projede emeği geçen arkadaşlarıma teşekkür etmek istiyorum. Göstermiş olduğunuz özveri ve destek için minnettarım. Bilgi ve deneyimlerinizi bizlerle paylaştığınız için teşekkürler.

* Burcu Dilay BOZDALI

Bilgisayar programcılığı bölümünde veri tabanı dersi için ben ve sevgili takım arkadaşlarım adına Yeşim AKTAŞ öğretmenime bu proje ödevini yapma şansı tanıdığı için teşekkürlerimi sunuyorum.

* Murathan SARGIN

İÇİNDEKİLER

ÖNSÖZ VE/VEYA TEŞEKKÜR i

1. GEREKSİNİM ANALİZİ 1

2. TABLOLAR 2

3. TABLO İLİŞKİLERİ(VIEWS) 7

4. TEK TABLO SELECT SORGUSU 8

5. İKİ TABLO SELECT SORGUSU 9

6. ÜÇ TABLO SELECT SORGUSU 10

7. INSERT INTO KODLARI 11

8. UPDATE KODLARI 14

9. DELETE KODLARI 15

10. ÖZGEÇMİŞ 1 16

11. ÖZGEÇMİŞ 2 17

12. ÖZGEÇMİŞ 3 18

13. ÖZGEÇMİŞ 4 19

**1.GEREKSİNİM ANALİZLERİ**

1-) Bir hastanın birden fazla şikayeti olabilir.

2-) Bir hastanın birden fazla muayene kaydı olabilir.

3-) Bir servis birden fazla odaya sahip olabilir.

4-) Bir hastanın birden fazla randevusu olabilir.

5-) Bir doktor birden fazla randevuya sahip olabilir.

6-) Bir doktorun birden fazla hastası olabilir.

7-) Bir ilaç birden fazla reçetede yer alabilir.

8-) Bir Personelin en az bir adresi olabilir.

9-) Bir hastanın en fazla bir tane T.C kimlik numarası olabilir.

10-) Bir oda en fazla bir servise ait olabilir.

11-) Bir hastanın en fazla bir hasta ID'si olabilir.

13-) Her hastanın en fazla bir adet doğum tarihi olmalıdır.

14-) Her ilacın en az bir adı olmalıdır.

15-) Her ilacın en fazla bir tane ID'si olmalıdır.

16-) Her hastanenin en az bir adı vardır.

17-) Her hastanenin en az bir telefon numarası vardır.

18-) Bir ilaç birden fazla reçetede yer alabilir.

19-) Bir hasta birden fazla doktora gidebilir.

20-) Bir hastanede birden fazla doktor çalışabilir.

21-) Bir ilacı birden fazla kişi alabilir.

**2.TABLOLAR**

CREATE TABLE hasta (

hasta\_id INT PRIMARY KEY,

hasta\_tc CHAR(11),

hasta\_ad VARCHAR(25),

hasta\_soyad VARCHAR(30),

hasta\_dogum\_tarihi DATE,

hasta\_cinsiyet CHAR(1),

hasta\_telno CHAR(11),

hasta\_mail VARCHAR(50),

hasta\_medeni\_hal VARCHAR(10),

oda\_id INT,

FOREIGN KEY (oda\_id) REFERENCES oda(oda\_id)

);

CREATE TABLE doktor (

doktor\_id INT PRIMARY KEY,

doktor\_ad VARCHAR(25),

doktor\_soyad VARCHAR(30),

doktor\_brans VARCHAR(50),

doktor\_telno CHAR(11),

doktor\_mail VARCHAR(50),

klinik\_id INT,

FOREIGN KEY (klinik\_id) REFERENCES klinik(klinik\_id)

);

CREATE TABLE hemsire (

hemsire\_id INT PRIMARY KEY,

hemsire\_ad VARCHAR(25),

hemsire\_soyad VARCHAR(30),

hemsire\_telno CHAR(11),

hemsire\_mail VARCHAR(50),

hemsire\_brans VARCHAR(50),

klinik\_id INT,

FOREIGN KEY (klinik\_id) REFERENCES klinik(klinik\_id)

);

CREATE TABLE klinik (

klinik\_id INT PRIMARY KEY,

klinik\_ad VARCHAR(50),

klinik\_brans VARCHAR(50),

klinik\_kapasite INT,

hastane\_id INT,

FOREIGN KEY (hastane\_id) REFERENCES hastane(hastane\_id)

);

CREATE TABLE oda (

oda\_id INT PRIMARY KEY,

oda\_no VARCHAR(10),

oda\_tipi VARCHAR(30),

oda\_kapasite INT,

oda\_durum VARCHAR(20),

hastane\_id INT,

FOREIGN KEY (hastane\_id) REFERENCES hastane(hastane\_id)

);

CREATE TABLE hastane (

hastane\_id INT PRIMARY KEY,

hastane\_ad VARCHAR(50),

hastane\_adres TEXT,

hastane\_telno CHAR(11),

hastane\_sehir VARCHAR(50),

hastane\_tur VARCHAR(50)

);

CREATE TABLE ameliyathane (

ameliyathane\_id INT PRIMARY KEY,

ameliyathane\_no INT,

ameliyathane\_kapasite INT,

ameliyathane\_ekipman TEXT,

ameliyathane\_durum VARCHAR(100),

hastane\_id INT,

FOREIGN KEY (hastane\_id) REFERENCES hastane(hastane\_id)

);

CREATE TABLE randevu (

randevu\_id INT PRIMARY KEY,

hasta\_id INT,

doktor\_id INT,

klinik\_id INT,

randevu\_tarihi DATETIME,

FOREIGN KEY (hasta\_id) REFERENCES hasta(hasta\_id),

FOREIGN KEY (doktor\_id) REFERENCES doktor(doktor\_id),

FOREIGN KEY (klinik\_id) REFERENCES klinik(klinik\_id)

);

CREATE TABLE recete (

recete\_id INT PRIMARY KEY,

hasta\_id INT,

doktor\_id INT,

ilac\_id INT,

kullanim\_talimat TEXT,

baslangic\_tarihi DATE,

bitis\_tarihi DATE,

FOREIGN KEY (hasta\_id) REFERENCES hasta(hasta\_id),

FOREIGN KEY (doktor\_id) REFERENCES doktor(doktor\_id),

FOREIGN KEY (ilac\_id) REFERENCES ilac(ilac\_id)

);

CREATE TABLE ilac (

ilac\_id INT PRIMARY KEY,

ilac\_ad VARCHAR(50),

ilac\_tur VARCHAR(30),

ilac\_firma VARCHAR(50),

ilac\_son\_kullanma\_tarihi DATE,

ilac\_kullanim\_talimat TEXT,

tedarikci\_id INT,

FOREIGN KEY (tedarikci\_id) REFERENCES tedarikci(tedarikci\_id)

);

CREATE TABLE tedarikci (

tedarikci\_id INT PRIMARY KEY,

tedarikci\_ad VARCHAR(50),

tedarikci\_telno CHAR(11),

tedarikci\_mail VARCHAR(50),

tedarikci\_adres TEXT,

urun\_tur VARCHAR(50)

);

CREATE TABLE sigorta (

sigorta\_id INT PRIMARY KEY,

sigorta\_ad VARCHAR(50),

toplam\_ucret VARCHAR(20),

sigorta\_kapsami TEXT,

hasta\_id INT,

police\_no VARCHAR(20),

bitis\_tarih DATE,

FOREIGN KEY (hasta\_id) REFERENCES hasta(hasta\_id)

);

CREATE TABLE fatura (

fatura\_id INT PRIMARY KEY,

toplam\_ucret VARCHAR(20),

odeme\_tarihi DATE,

hasta\_id INT,

FOREIGN KEY (hasta\_id) REFERENCES hasta(hasta\_id)

);

CREATE TABLE ziyaretci (

ziyaretci\_id INT PRIMARY KEY,

ziyaretci\_ad VARCHAR(25),

ziyaretci\_soyad VARCHAR(30),

ziyaretci\_telno CHAR(11),

ziyaretci\_iliski VARCHAR(50),

hasta\_id INT,

ziyaret\_tarihi DATETIME,

FOREIGN KEY (hasta\_id) REFERENCES hasta(hasta\_id)

);

CREATE TABLE ambulans (

ambulans\_id INT PRIMARY KEY,

plaka\_no VARCHAR(15),

ekipmanlar TEXT,

kapasite INT,

durum VARCHAR(20),

sofor\_id INT,

hastane\_id INT,

FOREIGN KEY (hastane\_id) REFERENCES hastane(hastane\_id)

);

CREATE TABLE ekipman (

ekipman\_id INT PRIMARY KEY,

ekipman\_ad VARCHAR(50),

ekipman\_tur VARCHAR(30),

ekipman\_durum VARCHAR(20),

satin\_alma\_tarihi DATE,

garanti\_bitis\_tarihi DATE,

hastane\_id INT,

FOREIGN KEY (hastane\_id) REFERENCES hastane(hastane\_id)

);

CREATE TABLE laboratuvar (

laboratuvar\_id INT PRIMARY KEY,

test\_adi VARCHAR(50),

test\_tipi VARCHAR(30),

hasta\_id INT,

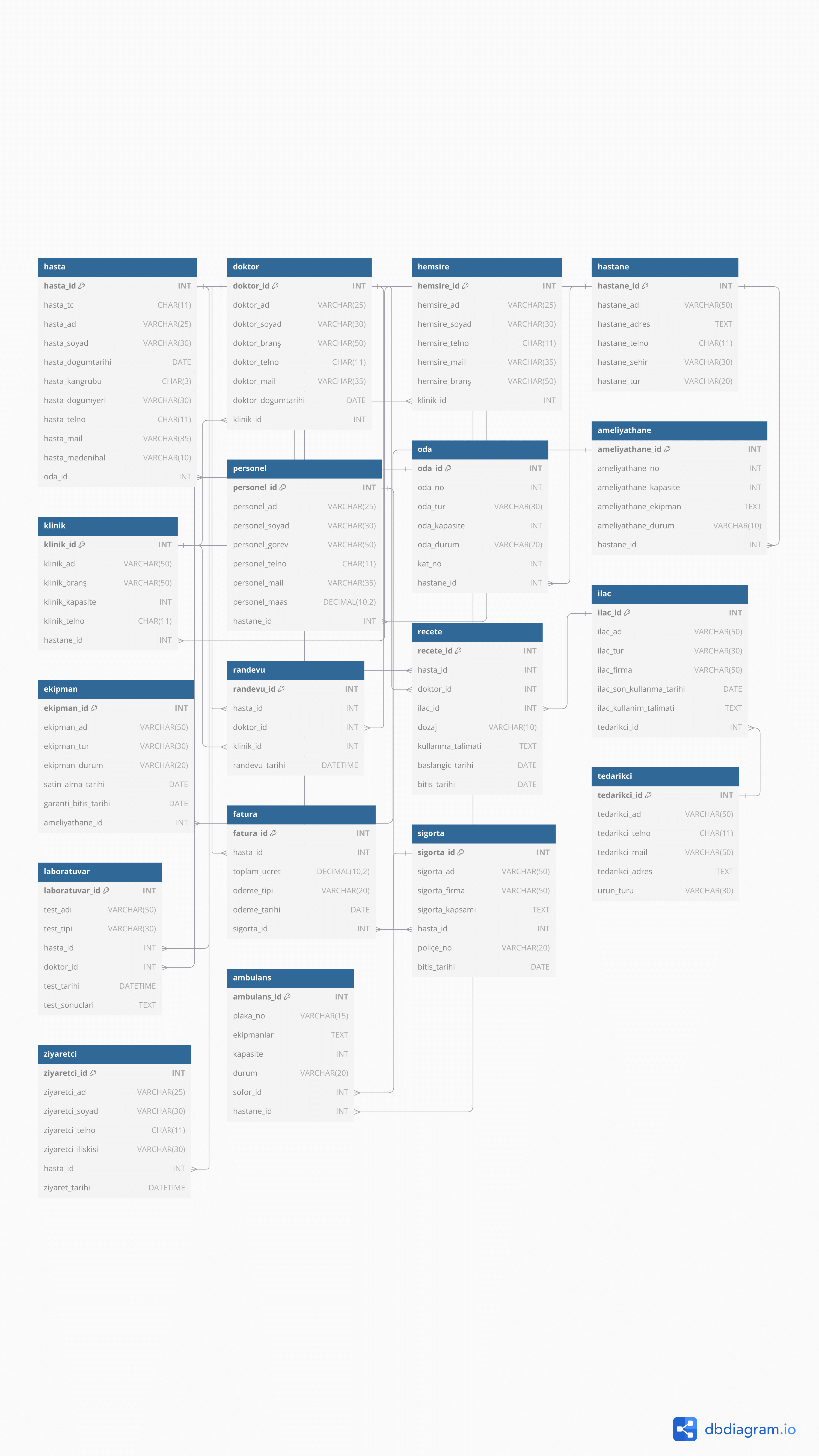
doktor\_id INT,

test\_tarihi DATETIME,

test\_sonuclari TEXT,

FOREIGN KEY (hasta\_id) REFERENCES hasta(hasta\_id),

FOREIGN KEY (doktor\_id) REFERENCES doktor(doktor\_id) );



**3. TABLO İLİŞKİLERİ (WIEWS)**

**TEK TABLO SELECT SORGUSU:**

**Sorgu 1:**

Select \* from Hastalar

**Açıklama 1:**

Bu sorgu Hastalar tablosundaki verileri çekme işlemi yapmaktadır.

Sorgu 2:

Select \* HastaID, Ad ,Soyad, Telefon from Hastalar where HastaID = 3;

**Açıklama 2:**

Bu sorgu HastaID’si 3 olan hastanın Hastalar tablosundan verilerini çekme işlemi yapmaktadır.

**Sorgu 3:**

Select \* DoktorID, Ad ,Soyad, PolinklinikID from Doktorlar order by PoliklinikID;

**Açıklama 3:**

Bu sorgu doktorlar tablosundan doktorların çalıştıkları polikliniklere göre listeleme işlemi yapmaktadır.

**İKİ TABLO SELECT SORGUSU**

**Sorgu 1:**

SELECT Hastalar.Ad AS HastaAd, Hastalar.Soyad AS HastaSoyad, Receteler.ReceteID, Receteler.YazimTarihi FROM Hastalar JOIN Receteler ON Hastalar.HastaID=Receteler.HastaID;

**Açıklama 1:**

Bu sorgu, Hastalar tablosu ve Reçeteler tablosu arasındaki HastaID sütunu üzerinden birleştirme işlemi yaparak, her bir hastanın aldığı reçeteleri ve bu reçetelerin yazım tarihlerini listeler.

**Sorgu 2:**

SELECT Receteler.ReceteID, Receteler.YazimTarihi, Doktorlar.Ad AS DoktorAd, Doktorlar.Soyad AS DoktorSoyad FROM Receteler JOIN Doktorlar ON Receteler.DoktorID = Doktorlar.DoktorID;

**Açıklama 2:**

Bu sorgu, Reçeteler tablosundaki ReçeteID ve Yazım tarihi bilgilerini, her reçeteyi yazan Doktorlar tablosundaki doktor bilgileriyle birleştirir.

**Sorgu 3:**

SELECT Personel.Ad, Personel.Soyad, Klinikler.KlinikAd FROM Personel JOIN Klinik ON Personeller.KlinikID = Klinik.KlinikID;

**Açıklama 3:**

Bu sorgu, Personeller ve Klinik tablolarını KlinikID üzerinden birleştirir ve her bir personelin çalıştığı kliniklerin adlarını listeler.

**ÜÇ TABLO SELECT SORGUSU**

**Sorgu 1:**

SELECT HastaAd, HastaSoyad, Receteler.ReceteID, Receteler.YazimTarihi, DoktorAd, DoktorSoyad FROM Hastalar JOIN Receteler ON Hasta.HastaID = Recete.HastaID JOIN Doktor ON Recete.DoktorID = DoktorID;

**Açıklama 1:**

Bu sorgu, reçeteler tablosundaki DoktorID ile Doktor tablosundaki doktor bilgileri ile birleştirir yani her hastanın aldığı reçeteyi yazan doktorların ad ve soyad bilgileri ile birlikte bir liste oluşturulur.

**Sorgu 2:**

SELECT HastaAd, HastaSoyad, Randevu.RandevuID, Randevu.RandevuTarihi, Klinik.KlinikAd FROM Randevu JOIN Hastalar ON Randevu.HastaID = Hasta.HastaID JOIN Klinik ON Randevu.KlinikID = Klinik.KlinikID;

**Açıklama 2:**

Bu sorgu, her hastanın aldığı randevu bilgilerini, randevuların tarihlerini ve bu randevuların hangi klinikte olduğunu gösterir. Randevu, Hasta ve Klinikler tabloları arasındaki Join işlemi ile veri birleştirilir

**7. INSERT-INTO KODLARI**

|  |  |
| --- | --- |
| INSERT INTO  Oda (oda\_id, oda\_no, oda\_tur, oda\_kapasite, oda\_durum, kat\_no, hastane\_id)  VALUES  (1, 24, 'yoğun bakım', 4, 'boş', 6, 1); | Oda tablosuna oda ID, oda numarası, oda türü, oda kapasitesi, oda durumu, odanın hangi katta olduğu ve hastane ID bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Hasta (hasta\_id, hasta\_ad, hasta\_soyad, hasta\_dogumtarihi, hasta\_kangrubu,  hasta\_dogumyeri, hasta\_telno, hasta\_mail, hasta\_medenihal, oda\_id)  VALUES  (3, 'Hasan', 'Uzuner', '2004-04-28', 'Arh+', 'İstanbul', '5550991865',  'hasanuz@gmail.com', 'Bekar', 1); | Hasta tablosuna hasta ID, hasta adı, hasta soyadı, hasta doğum tarihi, hasta kan grubu, hasta doğum yeri, hasta telefon numarası, hasta e-posta adresi, hasta medeni hali ve oda ID bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Klinik (klinik\_id, klinik\_ad, klinik\_branş, klinik\_kapasite, klinik\_telno, hastane\_id)  VALUES  (89, 'kırmızı alan', 'yoğun bakım', 100, '2353452134', 6); | Klinik tablosuna klinik ID, klinik adı, klinik branşı, klinik kapasitesi, klinik telefon numarası ve hastane ID bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Doktor (doktor\_id, doktor\_ad, doktor\_soyad, doktor\_branş, doktor\_telno, doktor\_mail, doktor\_dogumtarihi, klinik\_id)  VALUES  (5, 'Mahmut', 'Tuncer', 'ebe', '5592143456', 'hasantuncer@hotmail.com', '2004-05-24', 8); | Doktor tablosuna doktor ID, doktor adı, doktor soyadı, doktor branşı, doktor telefon numarası, doktor e-posta adresi, doktor doğum tarihi ve klinik ID bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Personel (personel\_id, personel\_ad, personel\_soyad, personel\_görev, personel\_telno, personel\_mail, personel\_maas, hastane\_id)  VALUES  (10, 'Kerimcan', 'Durmaz', 'hademe', '5646326644', 'Kerimcandurmaz@gmail.com', 25000, 6); | Personel tablosuna personel ID, personel adı, personel soyadı, personel görevi, personel telefon numarası, personel e-posta adresi, personel maaşı ve hastane ID bilgilerinin girilmesini sağlayan kod |
| INSERT INTO  Randevu (randevu\_id, hasta\_id, doktor\_id, klinik\_id, randevu\_tarihi)  VALUES  (20, 3, 5, 89, '2025-07-18'); | **Randevu tablosuna randevu ID, hasta ID, doktor ID, klinik ID ve randevu tarihi bilgilerinin girilmesini sağlayan kod.** |
| INSERT INTO  Laboratuvar (laboratuvar\_id, test\_ad, test\_tip, hasta\_id, doktor\_id, test\_tarih, test\_sonuclari)  VALUES  (56, 'coronavirus', 'PCRtesti', 3, 5, '2024-02-23', 'pozitif'); | Laboratuvar tablosuna laboratuvar ID, test adı, test tipi, hasta ID, doktor ID, test tarihi ve test sonuçları bilgilerinin girilmesini sağlayan kod. |

|  |  |
| --- | --- |
| INSERT INTO  Ziyaretci (ziyaretci\_id, ziyaretci\_ad, ziyaretci\_soyad, ziyaretci\_telno, ziyaretci\_iliskisi, hasta\_id, ziyaret\_tarihi)  VALUES  (45, 'Melis', 'Coban', '5538657894', 'birinci\_dereceden\_yakini', 56, '2024-12-12'); | Ziyaretci tablosuna ziyaretçi ID, ziyaretçi adı, ziyaretçi soyadı, ziyaretçi telefon numarası, ziyaretçi ile ilişkisi, hasta ID ve ziyaret tarihi bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Fatura (fatura\_id, toplam\_ucret, odeme\_tipi, odeme\_tarihi, sigorta\_id)  VALUES  (64, 150000, 'kredi\_karti', '2025-01-22', 122); | Fatura tablosuna fatura ID, toplam ücret, ödeme tipi, ödeme tarihi ve sigorta ID bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Ambulans (ambulans\_id, plaka\_no, ekipmanlar, kapasite, durum, sofor\_id)  VALUES  (74, '112ABC12', 'solunum\_cihazi', 4, 'acil\_ambulans', 122); | Ambulans tablosuna ambulans ID, plaka numarası, ekipmanlar, kapasite, durum ve şoför ID bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Hemsire (hemşire\_id, hemşire\_ad, hemşire\_soyad, hemşire\_telno, hemşire\_mail, hemşire\_brans, klinik\_id)  VALUES  (76, 'Alev', 'Çalışkan', '5536214756', 'alevclskn@hotmail.com', 'Ameliyathane\_hemsiresi', 86); | Hemsire tablosuna hemşire ID, hemşire adı, hemşire soyadı, hemşire telefon numarası, hemşire mail adresi, hemşire branşı ve klinik ID bilgilerinin girilmesini sağlayan kod. |

|  |  |
| --- | --- |
| INSERT INTO  Ameliyathane (ameliyathane\_id, ameliyathane\_no, ameliyathane\_kapasite, ameliyathane\_ekipman, ameliyathane\_durum)  VALUES  (555, 12, 8, 'neşter', 'kullanimda'); | Ameliyathane tablosuna ameliyathane ID, ameliyathane numarası, ameliyathane kapasitesi, ameliyathane ekipmanı ve ameliyathane durumu bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Ilac (ilac\_id, ilac\_adi, ilac\_tur, ilac\_firma, ilac\_son\_kullanim\_tarihi, ilac\_kullanim\_talimati)  VALUES  (124, 'majezik', 'sanovel', '2026-02-12', 'agizdan\_kullanim'); | Ilac tablosuna ilaç ID, ilaç adı, ilaç türü, ilaç firması, ilaç son kullanım tarihi ve ilaç kullanım talimatı bilgilerinin girilmesini sağlayan kod |
| INSERT INTO  Tedarikçi (tedarikçi\_id, tedarikci\_ad, tedarikçi\_telno, tedarikçi\_mail, tedarikçi\_adres, urun\_turu)  VALUES  (78, 'Kermansan', '5885891235', 'kermansan@outlook.com', 'İstanbul-Avcılar', 'tıbbi\_araclar'); | Tedarikçi tablosuna tedarikçi ID, tedarikçi adı, tedarikçi telefon numarası, tedarikçi mail adresi, tedarikçi adresi ve ürün türü bilgilerinin girilmesini sağlayan kod |

|  |  |
| --- | --- |
| INSERT INTO  Sigorta (sigorta\_id, sigorta\_ad, sigorta\_kapsami, hasta\_id, poliçe\_no, bitiş\_tarihi)  VALUES  (90, 'Sağlıksen', 'özel\_sigorta', 3, 666, '2026-05-23'); | Sigorta tablosuna sigorta ID, sigorta adı, sigorta kapsamı, hasta ID, poliçe numarası ve bitiş tarihi bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Ekipman (ekipman\_id, ekipman\_ad, ekipman\_tur, ekipman\_durum, satin\_alma\_tarihi, garanti\_bitis\_tarihi, ameliyathane\_id)  VALUES  (123, 'solunum\_cihazi', 'solunum\_yolu\_cihazı', 'kullanımda', '2012-07-24', '2028-07-24', 555); | Ekipman tablosuna ekipman ID, ekipman adı, ekipman türü, ekipman durumu, satın alma tarihi, garanti bitiş tarihi ve ameliyathane ID bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Hastahane (hastane\_id, hastane\_ad, hastane\_adres, hastane\_telno, hastane\_sehir, hastane\_tur)  VALUES  (1, 'Hayat\_hastanesi', 'İstanbul Bağcılar', '2248672509', 'İstanbul', 'Özel'); | Hastahane tablosuna hastane ID, hastane adı, hastane adresi, hastane telefon numarası, hastane şehri ve hastane türü bilgilerinin girilmesini sağlayan kod. |
| INSERT INTO  Recete(reçete\_id, hasta\_,doktor\_id, ilaç\_id,dozaj,  Kullanma\_talimati, baslangic\_tarihi, bitiş\_tarihi)  VALUES  (65,3,5,124,10mg,agizdan\_kullanim,12-02-2026,  06.05.2028) | Reçete tablosunda reçete ID, hasta ID, doktor ID, ilaç ID, dozajını, son kullanma tarihi, başlangıç ve  bitiş tarihi bilgilerini girmesini sağlayan kod. |

**8. UPDATE KODLARI**

|  |  |
| --- | --- |
| SORGU | AÇIKLAMA |
| UPDATE HastaBilgi  SET Telefon= ‘0556 658 7845’  WHERE HastaID= 10234 | HastaBilgi tablosundaki HastaID’si 10234 olan hastanın telefon numarasını güncelleyen kod. |
| UPDATE DoktorBilgi SET Email = 'dr.ahmetyilmaz@hastane.com' WHERE DoktorID = 2005 | DoktorBilgi tablosundaki DoktorID’si 2005 olan doktorun e-posta adresini güncelleyen kod. |
| UPDATE RandevuKayit SET RandevuTarihi = '2025-01-15' WHERE RandevuID = 3008 | RandevuKayit tablosundaki RandevuID’si 3008 olan randevunun tarihini güncelleyen kod. |
| UPDATE KlinikBilgi SET KlinikAdi = 'Genel Cerrahi' WHERE KlinikID = 5002 | KlinikBilgi tablosundaki KlinikID’si 5002 olan kliniğin adını güncelleyen kod. |
| UPDATE PersonelBilgi SET Maas = 7500 WHERE PersonelID = 1009 AND Departman = 'Temizlik' | PersonelBilgi tablosundaki PersonelID’si 1009 olan ve Temizlik departmanında çalışan personelin maaşını güncelleyen kod. |

**9. DELETE KODLARI**

|  |  |
| --- | --- |
| SORGU | AÇIKLAMA |
| DELETE FROM personel  WHERE personel\_id = 9; | Personel tablosunda personel id`si 9 olan personeli siler. |
| DELETE FROM randevu  WHERE randevu\_tarihi = `2024-11-14`; | Randevu tablosunda `2024-11-14` tarihinde olan randevunun randevu\_id, hasta\_id, doktor\_id, klinik\_id, randevu\_tarihi`ne ait kayıtları silen kod. |
| DELETE FROM fatura ; | Fatura tablosunda bulunan kayıtların tümünü silen kod. |
| DELETE FROM oda;  WHERE oda\_kapasite < 3 ; | Oda tablosunda oda kapasitesi 3 `ten küçük olan bütün kayıtları silen kod |
| DELETE FROM hemsire  WHERE klinik\_id = `13 ` | Hemşire tablosunda klinik id`si 13 olan tüm kayıtları siler |

|  |  |
| --- | --- |
| SORGU | AÇIKLAMA |
| DELETE FROM Ilac  WHERE IlacID = 18564 | Ilac tablosunda seçilen ID`nin bilgilerini silen kod. |
| DELETE FROM Tedarikci  WHERE TedarikciID = 6202 | Tedarikci tablosunda seçilen üyenin bilgilerini silen kod. |
| DELETE FROM Sigorta  WHERE Sigortafirma = `Demirkol Sigorta` | Sigorta tablosunda ismi girilen sigorta firmasını silen kod. |
| DELETE FROM Doktor  WHERE doktorbrans = `Nöroloji` | Doktor tablosunda Branş`ı seçilen doktorun kaydını silen kod. |
| DELETE FROM Hasta  WHERE HastaID = 567400 | Hasta tablosunda hasta ID`si girilen hastanın kayıtlarını silen kod. |

**ÖZGEÇMİŞ 1**

Adı Soyadı : Burcu Dilay BOZDALI

Doğum Yeri ve Tarihi : SİVAS, 2004

Yabancı Dil : İngilizce

Eğitim Durumu : Lise / Düzce 15 Temmuz Şehitleri Anadolu Lisesi

İş Deneyimi : -

İletişim (e-posta) : burcudilaybozdali@gmail.com

Bildiği Programlama Dilleri : Python, C#, Java, HTML, PHP

Veri Tabanı : Microsoft SQL Server

Hobiler : Spor yapmak, yemek yapmak

**ÖZGEÇMİŞ 2**

Adı Soyadı : Berfin SEFEROĞLU

Doğum Yeri ve Tarihi : İSTANBUL, 2004

Yabancı Dil : İngilizce

Eğitim Durumu : Lise / Hasan Aslanoba Anadolu Lisesi

İş Deneyimi : -

İletişim (e-posta) : berfin12@gmail.com

Bildiği Programlama Dilleri : Python, C#, Java, HTML, CSS

Veri Tabanı : Microsoft SQL Server

Hobiler : Spor yapmak, doğa yürüyüşü, eğitmenlik yapmak

**ÖZGEÇMİŞ 3**

Adı Soyadı : Murathan SARGIN

Doğum Yeri ve Tarihi : KÜTAHYA, 2004

Yabancı Dil : İngilizce

Eğitim Durumu : Lise / Şafak Öncü Fen Lisesi

İş Deneyimi : -

İletişim (e-posta) : Murathan.64.1907@gmail.com

Bildiği Programlama Dilleri : Python, C#, HTML

Veri Tabanı : Microsoft SQL Server

Hobiler : Basketbol oynamak, eksiz çizim yapmak

**ÖZGEÇMİŞ 4**

Adı Soyadı : Sara Beril KARAMAN

Doğum Yeri ve Tarihi : KÜTAHYA, 2005

Yabancı Dil : İngilizce

Eğitim Durumu : Lise / Şafak Öncü Fen Lisesi

İş Deneyimi : -

İletişim (e-posta) : berfin12@gmail.com

Bildiği Programlama Dilleri : Python, C#, HTML

Veri Tabanı : Microsoft SQL Server

Hobiler : Basketbol oynamak, eksiz çizim yapmak