



İSTANBUL ÜNİVERSİTESİ
C | E | R | R | A | H | P | A | Ş | A

BIMU3064-Veri Tabanı Yönetim Sistemleri

İBRAHİM BAŞAR YARGICI
1306191467

Ödev4
Veritabanı = PostgreSQL

(21.12.2021)

Ödev 4: E-R ile Hastane VT Tasarımı

Teslim tarihi: 21.12.2021 Salı 24:00, Ödev AKSIS'ten teslim edilecektir.

Bir hastane bilgi sistemi için aşağıdaki gereksinimleri karşılayan bir veritabanını E-R ile tasarlayınız. Tasarım sonu elde ettiğiniz E-R diagramını ilişkisel VT şemasına çeviriniz. **E-R diagramı ve ilişkisel VT şemasını teslim ediniz.** E-R diagramında Varlık kümeleri için dikdörtgen, ilişki kümeleri için baklava kullanınız. Varlık ve ilişki kümelerinin özelliklerini ve anahtarlarını göstermeyi unutmayınız. İlişki kümelerinin mapping cardinality türlerini (1-m, m-m, 1-1) ve varlık kümelerinin ilişki kümelerini katılım türlerini (all/some, hepsi/bazısı) tek/çift veya ince/kalın çizgiyle, sanal ilişki kümelerini kesikli çizgi ile göstermeyi unutmayınız.

- Doktorların tckimlik no, ad, soyad, uzmanlık alan ('Dahiliye', 'KBB', 'Kalp', 'Psikiyatri', ...), tel, eposta bilgileri saklanacaktır.
- Hemşirelerin ve hastaların tckimlik no, ad, soyad, tel, eposta bilgileri saklanacaktır.
- Hemşire ve doktorların ayrıca maşş bilgisi desaklanacaktır.
- İlaçların barkodu, adı, fiyatı saklanacaktır.
- Hastane odalarının numaraları ve dahili tel no'ları ve alanları m2 cinsinden saklanacaktır.
- Hastalar hastalık belirtileri ('baş ağrısı', 'kusma', 'halsizlik', ...) gibi belirtiler gösterince hastanedeki bir doktordan belirli bir tarih ve saat için randevu alırlar. Randevuda doktor hastayı belirlenen odada muayene eder, bir hastalık ('grip', 'solunum enfeksiyonu', 'burkulma', ...) teşhisi koyar ve reçete yazar. Hasta muayene ücreti öder ve karşılığında bir fatura alır. Hastalık belirtileri ve hastalık bilgisi de saklanacaktır.
- Bir reçetede reçete numarası, tarihi, hasta bilgileri ve reçetenin içerdiği ilaçların listesi bulunur. Bir recetede minimum 1 ilaç vardır.
- Bazı hastalara ameliyat yani operasyon yapılır. Operasyon bir hastaya minimum bir doktor ve minimum bir hemşire tarafından belirli bir tarih ve saatte yapılan, belirli tıbbi adı ve ücreti olan bir işlemdir. Operasyonda da ücret karşılığı fatura kesilir.
- Bir faturada fatura numarası, tarihi, saati, müşteri yani hastanın adı ve adresi bilgilerinin yanı sıra ayrı kalemler listesi şeklinde her kalemin adı ('muayene', 'mide ameliyatı', ...) birim fiyatı, adedi, birim fiyat x adet = ücreti bilgileri ve faturanın en altında fatura toplamı, ödeme şekli ('nakit', 'kredi kart', ...) bilgileri saklanacaktır.

Veri Tabanı Şeması

Database Schema:

Entity Sets:

- Uzmanlik (id, ad)
- Maas (id, maas)
- Hemsire (id, tc_no, eposta, ad, soyad)
- Doktor (id, tc_no, eposta, ad, soyad)
- Hasta (id, tc_no, eposta, ad, soyad, adres)
- Teshis (id, teshis)
- Oda (id, kapasite)
- Telefon (id, tel_no)
- Operasyon (id, tarih)
- Islem (id, ucret, tibbi_ad)
- HastalikBelirtileri (id, belirti)
- OdemeSekli (id, tur)
- Ilac (barkod_no, ad, fiyat)
- Recete (recete_no, tarih)
- Fatura (fatura_no, tarih, ucret)

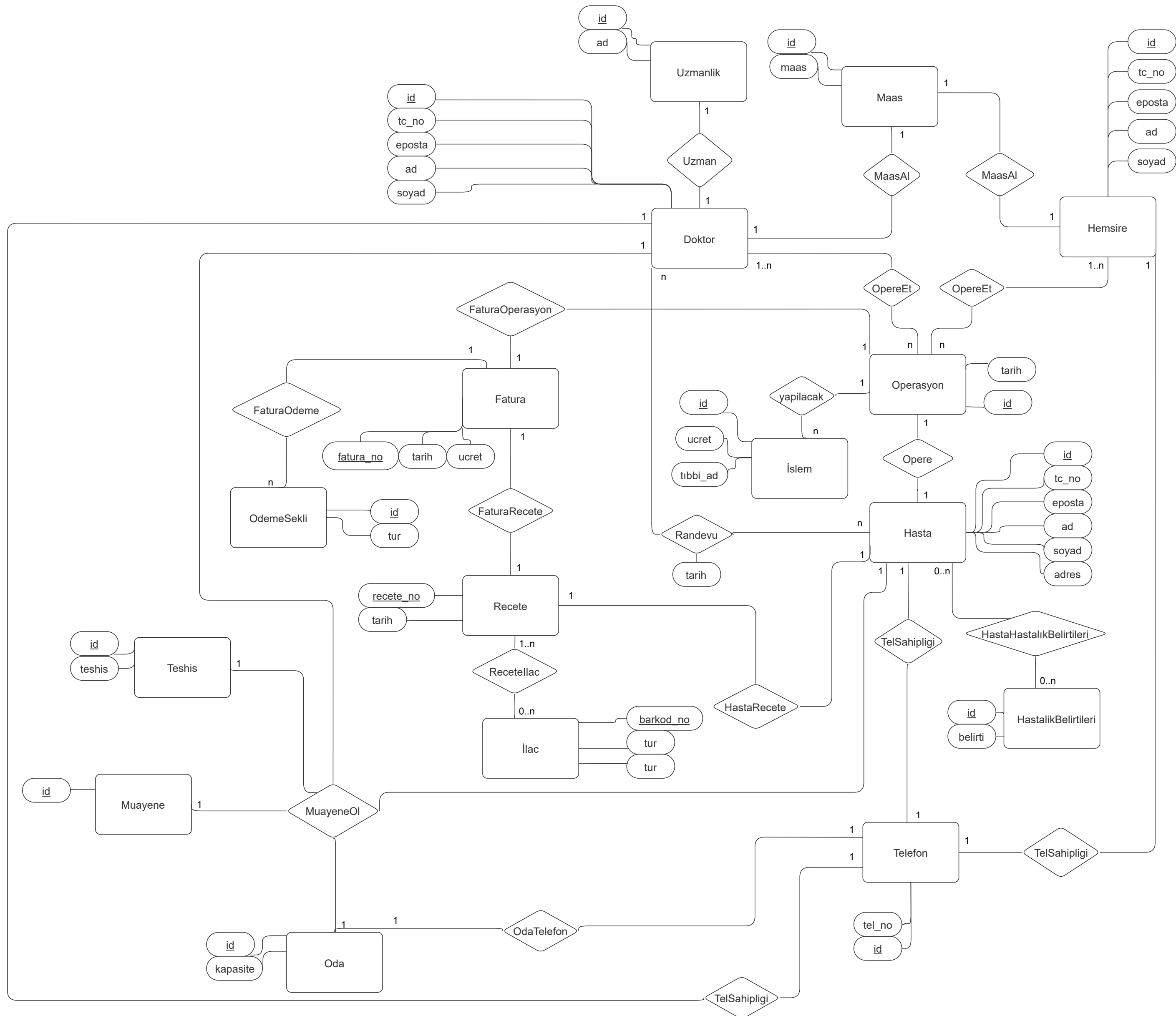
Relationship Sets:

- Uzman(doktor(1)-(1)uzmanlık)
- MaasAl(doktor(1)-(1)maas)
- MaasAl(hemsire(1)-(1)maas)
- OpereEt(doktor(M)-(M)operasyon)
- OpereEt(hemsire(M)-(M)operasyon)
- Opere(hasta(1)-(1)operasyon)
- Yapilacak(operasyon(1)-(M)islem)
- TelSahipligi(doktor(1)-telefon(1))
- TelSahipligi(hemsire(1)-telefon(1))
- TelSahipligi(hasta(1)-telefon(1))
- Randevu (hasta(M)-(M)doktor)[tarih]
- HastaRecete(recete(1)-(1)hasta)
- FaturaRecete(fatura(1)-(1)recete)
- FaturaOperasyon(fatura(1)-(1)operasyon)
- FaturaOdeme(fatura(1)-(M)odemeSekli)
- Recetellac(recete(M)-(M)ilac)
- OdaTelefon(oda(1)-(1)telefon)
- Muayene [(doktor(1)-(1)hasta) (hasta(1)-(1)oda) (oda(1)-(1)teshis)]
- HastaHastalikBelirtileri(hasta(M)-(M)HastalikBelirtileri)

ER Diagramı

ER (Entity Relationship) Diagram Of a Hospital System

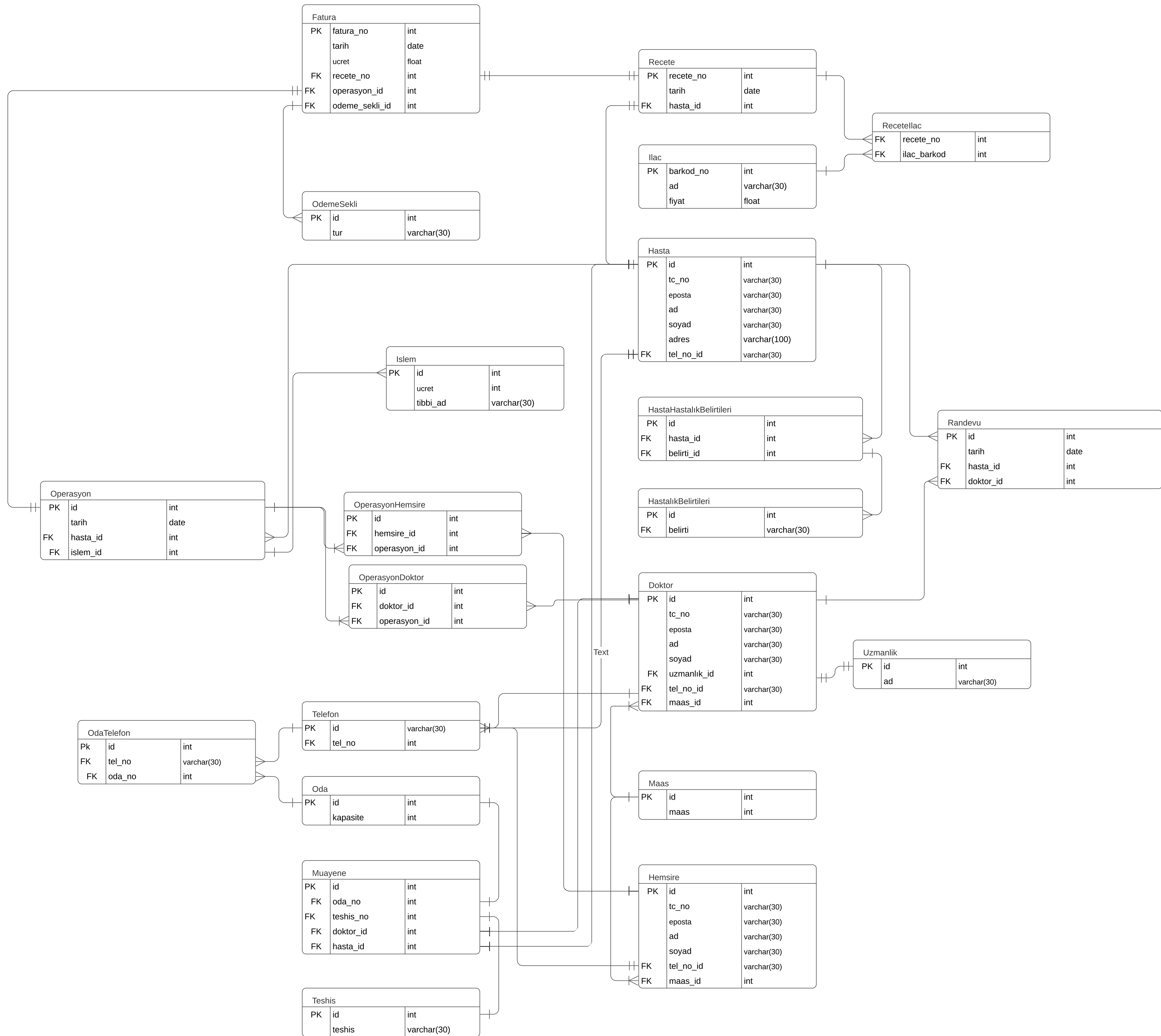
by İbrahim Başar YARGICI



UML Diagramı

UML (Unified Modeling Language) Diagram Of a Hospital System

by İbrahim Başar YARGICI



Veri Tabanı İçin SQL kodları

```
CREATE TABLE Hasta (  
    id int,  
    tc_no varchar(30),  
    eposta varchar(30),  
    ad varchar(30),  
    soyad varchar(30),  
    adres varchar(100),  
    tel_no_id varchar(30),  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE Operasyon (  
    id int,  
    tarih date,  
    hasta_id int,  
    islem_id int,  
    PRIMARY KEY (id),  
    CONSTRAINT FK_Operasyon.hasta_id  
        FOREIGN KEY (hasta_id)  
        REFERENCES Hasta(id)  
);
```

```
CREATE TABLE Maas (  
    id int,  
    maas int,  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE Hemsire (  
    id int,  
    tc_no varchar(30),  
    eposta varchar(30),  
    ad varchar(30),  
    soyad varchar(30),  
    tel_no_id varchar(30),  
    maas_id int,  
    PRIMARY KEY (id),  
    CONSTRAINT FK_Hemsire.maas_id  
        FOREIGN KEY (maas_id)  
            REFERENCES Maas(id)  
);
```

```
CREATE TABLE OperasyonHemsire (  
    id int,  
    hemsire_id int,  
    operasyon_id int,  
    PRIMARY KEY (id),  
    CONSTRAINT FK_OperasyonHemsire.operasyon_id  
        FOREIGN KEY (operasyon_id)  
            REFERENCES Operasyon(id),  
    CONSTRAINT FK_OperasyonHemsire.hemsire_id  
        FOREIGN KEY (hemsire_id)  
            REFERENCES Hemsire(id)  
);
```

```
CREATE TABLE Doktor (  
    id int,  
    tc_no varchar(30),  
    eposta varchar(30),  
    ad varchar(30),  
    soyad varchar(30),  
    uzmanlik_id int,  
    tel_no_id varchar(30),  
    maas_id int,  
    PRIMARY KEY (id),  
    CONSTRAINT FK_Doktor.maas_id  
        FOREIGN KEY (maas_id)  
            REFERENCES Maas(id)  
);
```

```
CREATE TABLE Muayene (  
    id int,  
    oda_no int,  
    teshis_no int,  
    doktor_id int,  
    hasta_id int,  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE Islem (  
    id int,  
    ucret int,  
    tibbi_ad varchar(30),  
    PRIMARY KEY (id),  
    CONSTRAINT FK_Islem.id  
        FOREIGN KEY (id)  
            REFERENCES Operasyon(islem_id)  
);
```

```
CREATE TABLE Oda (  
    id int,  
    kapasite int,  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE Telefon (  
    id varchar(30),  
    tel_no int,  
    PRIMARY KEY (id),  
    CONSTRAINT FK_Telefon.id  
        FOREIGN KEY (id)  
            REFERENCES Doktor(tel_no_id)  
);
```

```
CREATE TABLE OdaTelefon (  
    id int,  
    tel_no varchar(30),  
    oda_no int,  
    CONSTRAINT FK_OdaTelefon.tel_no  
        FOREIGN KEY (tel_no)  
            REFERENCES Telefon(id),  
    CONSTRAINT FK_OdaTelefon.oda_no  
        FOREIGN KEY (oda_no)  
            REFERENCES Oda(id)  
);
```

```
CREATE TABLE Recete (  
    recete_no int,  
    tarih date,  
    hasta_id int,  
    PRIMARY KEY (recete_no)  
);
```

```
CREATE TABLE Ilac (  
    barkod_no int,  
    ad varchar(30),  
    fiyat float,  
    PRIMARY KEY (barkod_no)  
);
```

```
CREATE TABLE Teshis (  
    id int,  
    teshis varchar(30),  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE OperasyonDoktor (  
    id int,  
    doktor_id int,  
    operasyon_id int,  
    PRIMARY KEY (id),  
    CONSTRAINT FK_OperasyonDoktor.doktor_id  
        FOREIGN KEY (doktor_id)  
            REFERENCES Doktor(id),  
    CONSTRAINT FK_OperasyonDoktor.operasyon_id  
        FOREIGN KEY (operasyon_id)  
            REFERENCES Operasyon(id)  
);
```

```
CREATE TABLE Uzmanlik (  
    id int,  
    ad varchar(30),  
    PRIMARY KEY (id)  
);
```



```
CREATE TABLE HastaHastalıkBelirtileri (  
    id int,  
    hasta_id int,  
    belirti_id int,  
    PRIMARY KEY (id),  
    CONSTRAINT FK_HastaHastalıkBelirtileri.hasta_id  
        FOREIGN KEY (hasta_id)  
        REFERENCES Hasta(id)  
);
```

```
CREATE TABLE HastalıkBelirtileri (  
    id int,  
    belirti varchar(30),  
    PRIMARY KEY (id),  
    CONSTRAINT FK_HastalıkBelirtileri.id  
        FOREIGN KEY (id)  
        REFERENCES HastaHastalıkBelirtileri(belirti_id)  
);
```

```
CREATE TABLE Fatura (  
    fatura_no int,  
    tarih date,  
    ucret float,  
    recete_no int,  
    operasyon_id int,  
    odeme_sekli_id int,  
    PRIMARY KEY (fatura_no)  
);
```

```
CREATE TABLE OdemeSekli (  
    id int,  
    tur varchar(30),  
    PRIMARY KEY (id),  
    CONSTRAINT FK_OdemeSekli.id  
        FOREIGN KEY (id)  
            REFERENCES Fatura(odeme_sekli_id)  
);
```

```
CREATE TABLE Randevu (  
    id int,  
    tarih date,  
    hasta_id int,  
    doktor_id int,  
    PRIMARY KEY (id),  
    CONSTRAINT FK_Randevu.id  
        FOREIGN KEY (id)  
            REFERENCES Hasta(id),  
    CONSTRAINT FK_Randevu.doktor_id  
        FOREIGN KEY (doktor_id)  
            REFERENCES Doktor(id)  
);
```

```
CREATE TABLE Recetellac (  
    recete_no int,  
    ilac_barkod int,  
    CONSTRAINT FK_Recetellac.ilac_barkod  
        FOREIGN KEY (ilac_barkod)  
            REFERENCES Ilac(barkod_no),  
    CONSTRAINT FK_Recetellac.recete_no  
        FOREIGN KEY (recete_no)  
            REFERENCES Recete(recete_no)  
);
```