



HASTANE VERİTABANI

Profesör İBRAHİM ARPACI



SAKHR MOHAMMED AL MASNAEA 2111505218
MUHAMMED YILDIZ 201505044
RIDVAN BÜYÜKKAYA 2111505056
Muhammad Kalumian 2111505201

Hastane veri tabanı

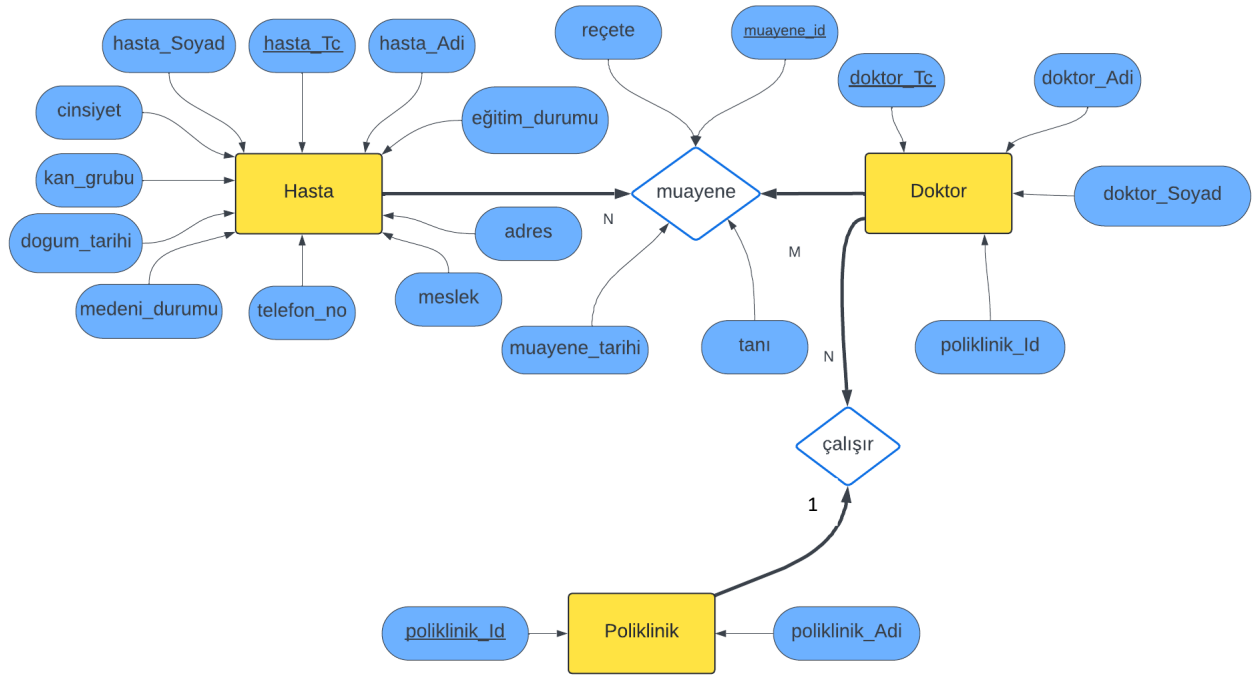
Hastanede muayene ve hasta kayıt işlemleri için bir otomasyon sistemi gerekmektedir. Hastane otomasyon sisteminde çeşitli poliklinikler bulunmakta ve her polikliniğin benzersiz bir numarası ve adı vardır. Her poliklinikte o polikliniğe özgü doktorlar bulunmaktadır. Doktorlar için, kimlik numarası, adı, soyadı ve poliklinik numarası verileri tutulmalıdır. Hastalar, poliklinik ve tercih ettikleri doktoru seçer. Doktorlar hastaları muayene eder, her muayenenin eşsiz bir id vardır. Muayene sonucunda ise hastalara tanı koyar ve reçete yazar. Doktorlar, hastalarının muayene ettiklerinde muayene tarihi saklanmalıdır. Hastanın genel kimlik bilgileri (adı, soyadı, cinsiyet, doğum tarihi, T.C. kimlik numarası, medeni durum, eğitim durumu, meslek, kan grubu, telefon numarası ve adresi) saklanmalıdır.

1. Varlıklar ve Nitelikler

Varlıklar	Nitelikler ve türleri
Hasta	hasta_Tc: VARCHAR(11) PRIMARY KEY hasta_Adi VARCHAR(255) hasta_Soyadı VARCHAR(255) cinsiyet ENUM ('Erkek', 'Kadın', 'Diğer') doğum_tarihi DATE medeni_durum ENUM ('Bekar', 'Evli', 'Boşanmış', 'Dul') eğitim_durumu VARCHAR(255) meslek VARCHAR(255) kan_grubu ENUM('A+', 'A-', 'B+', 'B-', 'AB+', 'AB-', 'O+', 'O-') telefon_no VARCHAR(15) adres VARCHAR(500)
Doktor	doktor_Tc VARCHAR(11) PRIMARY KEY doktor_Adi VARCHAR(255) doktor_Soyadı VARCHAR(255) poliklinik_id INT (10)
Poliklinik	poliklinik_id INT(10) PRIMARY KEY poliklinik_adı VARCHAR(255)
Muayene	Mayene_Id INT(10) Ddoktor_Tc VARCHAR(11) Hhasta_Tc VARCHAR(11) Muayene_tarihi () tanı VARCHAR(255) reçete VARCHAR(255)

İlişkiler:

- Bir poliklinikte birden fazla doktor olabilir
- Bir doktor bir poliklinikte çalışabilir
- Bir hastaya birden fazla muayene yapılmış olabilir
- Bir doktor birden fazla hastayı muayene etmiş olabilir



Adım 1: güçlü varlıklar

Hasta

<u>hasta_Tc</u>	hasta_Adi	hasta_Soyadı	cinsiyet	Eğitim_durumu	adres	meslek	Telefon_no	Medeni_durumu	Doğum_tarihi	Kan_grubu
-----------------	-----------	--------------	----------	---------------	-------	--------	------------	---------------	--------------	-----------

Doktor

<u>doktor_Tc</u>	doktor_Adi	doktor_Soyadı	poliklinik_Id
------------------	------------	---------------	---------------

Poliklinik

<u>poliklinik_Id</u>	poliklinik_Adi
----------------------	----------------

- Adım2:zayıf varlıklar yoktur
- Adım 3: 1:1 ilişki yoktur
- Adım 4: 1:N ilişkiler

Poliklinik ve doktor arasında

- Adım 5: N:M İLİŞKİLER

Muayene

<u>Muayene_Id</u>	Hhasta_Tc	Ddoktor_tc	reçete	tanı	Muayene_tarihi
-------------------	-----------	------------	--------	------	----------------

- Step 6: Mapping of Multi-valued attributes There is no multi-valued attribute.
- Step 7: Mapping of N-ary Relationship Types There is no N-ary relationship type.

Tablolar arasındaki ilişkiler şu şekilde gösterilebilir

Hasta

<u>hasta_Tc</u>	hasta_Adi	hasta_Soyadı	cinsiyet	Eğitim_durumu	adres	meslek	Telefon_no	Medeni_durumu	Doğum_tarihi	Kan_grubu
-----------------	-----------	--------------	----------	---------------	-------	--------	------------	---------------	--------------	-----------

Doktor

<u>doktor_Tc</u>	doktor_Adi	doktor_Soyadi	poliklinik_Id
------------------	------------	---------------	---------------

Poliklinik

<u>poliklinik_Id</u>	poliklinik_Adi
----------------------	----------------

Muayene

<u>Muayene_Id</u>	Hhasta_Tc	Ddoktor_tc	Muayene_tarihi	reçete	tanı
-------------------	-----------	------------	----------------	--------	------

RELATION	Functional Dependencies
HASTA	hasta_Tc \longrightarrow {hasta_Adi, hasta_Soyadi, cinsiyet, kan_grubu, meslek, medeni_durumu, doğum_tarihi, telefon_no, adres} hasta_Tc:Primary Key
DOKTOR	doktor_Tc \longrightarrow {doktor_Adi, doktor_Soyadi, cinsiyet, poliklinik_Id } doktor_Tc:Primary Key poliklinik_Id : Foreign Key references poliklinik
POLİKLİNİK	poliklinik_Id \longrightarrow { poliklinik_Adi } poliklinik_Id:Primary Key
MUAYENE	muayene_Id \longrightarrow { muayene_tarihi, tanı, reçete, hasta_Tc, doktor_Tc } muayene_Id:Primary Key hasta_Tc: Foreign Key references hasta doktor_Tc: Foreign Key references doktor

Sql kodları:

CREATE TABLE Poliklinik (

poliklinik_Id INT İDENTITY(1,1) PRIMARY KEY,

poliklinik_adı VARCHAR(255) NOT NULL

);

CREATE TABLE Doktor (

doktor_Tc VARCHAR(11) PRIMARY KEY,

doktor_Adi VARCHAR(255) NOT NULL,

doktor_Soyadı VARCHAR(255) NOT NULL,

poliklinik_Id INT NOT NULL,

FOREIGN KEY (poliklinik_Id) REFERENCES Poliklinik(poliklinik_Id)

);

```
CREATE TABLE HASTA (  
    hasta_Tc VARCHAR(11) PRIMARY KEY,  
    hasta_Adı VARCHAR(255) NOT NULL,  
    hasta_Soyadı VARCHAR(255) NOT NULL,  
    cinsiyet NOT NULL,  
    doğum_tarihi DATE NOT NULL,  
    medeni_durum NOT NULL,  
    eğitim_durumu VARCHAR(255) NOT NULL,  
    meslek NOT NULL,  
    kan_grubu NOT NULL,  
    telefon_no VARCHAR(15) NOT NULL,  
    adres VARCHAR(500) NOT NULL  
);
```

```
CREATE TABLE Muayene (  
    doktor_Tc VARCHAR(11) NOT NULL,  
    hasta_Tc VARCHAR(11) NOT NULL,  
    tanı VARCHAR(255) NOT NULL,  
    reçete VARCHAR(255) NOT NULL,  
    FOREIGN KEY (doktor_Tc) REFERENCES DoKTOR(doktor_Tc),  
    FOREIGN KEY (hasta_tc) REFERENCES Hasta(hasta_Tc)  
);
```

1NF(1. Normal Form)

Bir veri tabanının 1NF olabilmesi için aşağıdaki özellikleri karşılayabilmesi gerekir:

- Aynı tablo içinde tekrarlayan kolonlar bulunamaz.
- Her kolonda yalnızca bir değer bulunabilir.
- Her satır bir eşsiz anahtarla tanımlanmalıdır.

Hastane veri tabanında tekrarlayan hiçbir kolon bulunmuyor ve her kolonda yalnızca bir değer bulunduğu ve birincil anahtarın açıklayamadığı hiçbir kolon olmadığı için 1NF uygundur.

2NF(2.Normal Form)

Bir veri tabanının 2NF olabilmesi için aşağıdaki özellikleri karşılayabilmesi gerekir:

- Tablo 1NF olmalıdır.
- Anahtar olmayan değerler ile kompozit (bileşik) anahtarlar arasında kısmi (partial) bağımlılık durumu oluşmamalıdır. Kısmi bağımlılık durumu, anahtar olmayan herhangi bir değer kompozit bir anahtarın yalnızca bir kısmına bağlı ise oluşur.
- Herhangi bir veri alt kümesi birden çok satırda tekrarlanmamalıdır. Bu tür veri alt kümeleri için yeni tablolar oluşturulmalıdır.
- Ana tablo ile yeni tablolar arasında, dış anahtarlar (foreign key) kullanılarak ilişkiler tanımlanmalıdır.

Hastane veritabanı 2.NF uygundur.

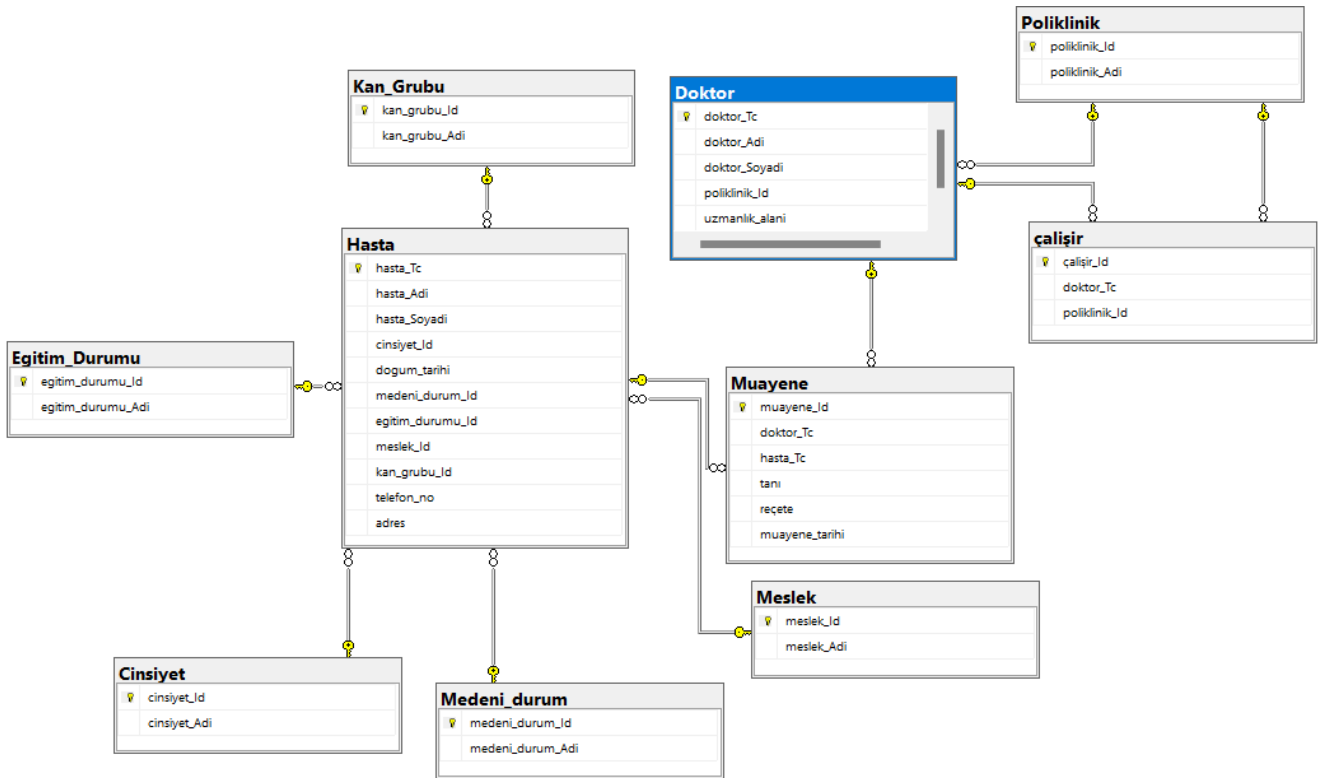
3NF(3.Normal Form)

Bir veri tabanının 3NF olabilmesi için aşağıdaki özellikleri karşılayabilmesi gerekir:

- Veri tabanı 2NF olmalıdır,

- Anahtar olmayan hiçbir kolon bir diğerine (anahtar olmayan başka bir kolona) bağlı olmamalı ya da geçişken fonksiyonel bir bağımlılığı (transitional functional dependency) olmamalıdır. Başka bir deyişle her kolon eşsiz anahtara tam bağımlı olmak zorundadır.

Hastane veri tabanı 3NF uygundur bu yüzden herhangi bir işleme gerek yoktur.



```

CREATE TABLE Poliklinik (
    poliklinik_Id INT identity(1,1) PRIMARY KEY not null,
    poliklinik_Adi VARCHAR(255) NOT NULL
);

CREATE TABLE Doktor (
    doktor_Tc VARCHAR(11) PRIMARY KEY,
    doktor_Adi VARCHAR(255) NOT NULL,
    doktor_Soyadi VARCHAR(255) NOT NULL,
    poliklinik_Id INT NOT NULL,
    FOREIGN KEY (poliklinik_Id) REFERENCES Poliklinik(poliklinik_Id)
);

CREATE TABLE Cinsiyet (
    cinsiyet_Id INT NOT NULL PRIMARY KEY,
    cinsiyet_Adi VARCHAR(255) NOT NULL,
);

CREATE TABLE Meslek(
    meslek_Id INT NOT NULL PRIMARY KEY,

```



```

meslek_Adi VARCHAR(255) NOT NULL,
);
CREATE TABLE Egitim_Durumu(
    egitim_durumu_Id INT NOT NULL PRIMARY KEY,
    egitim_durumu_Adi VARCHAR(255) NOT NULL,
);

CREATE TABLE Medeni_durum(
    medeni_durum_Id INT NOT NULL PRIMARY KEY,
    medeni_durum_Adi VARCHAR(255) NOT NULL,
);
CREATE TABLE Kan_Grubu(
    kan_grubu_Id INT NOT NULL PRIMARY KEY,
    kan_grubu_Adi VARCHAR(255) NOT NULL,
);

CREATE TABLE çalışır (
    çalışır_Id INT PRIMARY KEY not null ,
    doktor_Tc VARCHAR(11) not null ,
    poliklinik_Id INT identity(1,1) not null,
    FOREIGN KEY (doktor_Tc) REFERENCES Doktor(doktor_Tc),
    FOREIGN KEY (poliklinik_Id) REFERENCES Poliklinik(poliklinik_Id)
);
CREATE TABLE Hasta (
    hasta_Tc VARCHAR(11) PRIMARY KEY,
    hasta_Adi VARCHAR(255) NOT NULL,
    hasta_Soyadi VARCHAR(255) NOT NULL,
    cinsiyet_Id INT NOT NULL,
    dogum_tarihi DATE NOT NULL,
    medeni_durum_Id INT NOT NULL,
    egitim_durumu_Id INT NOT NULL,
    meslek_Id INT NOT NULL,
    kan_grubu_Id INT NOT NULL,
    telefon_no VARCHAR(15) NOT NULL,
    adres VARCHAR(500) NOT NULL
    FOREIGN KEY (cinsiyet_Id) REFERENCES Cinsiyet(cinsiyet_Id),
    FOREIGN KEY (medeni_durum_Id) REFERENCES Medeni_durum( medeni_durum_Id),
    FOREIGN KEY (kan_grubu_Id) REFERENCES Kan_Grubu(kan_grubu_Id),
    FOREIGN KEY (meslek_Id) REFERENCES Meslek(meslek_Id),
    FOREIGN KEY (egitim_durumu_Id) REFERENCES Egitim_durumu( egitim_durumu_Id),
);

CREATE TABLE Muayene (
    muayene_Id INT identity(1,1) PRIMARY KEY not null,
    doktor_Tc VARCHAR(11) NOT NULL,
    hasta_Tc VARCHAR(11) NOT NULL,
    tanı VARCHAR(255) NOT NULL,
    reçete VARCHAR(255) NOT NULL,
    muayene_tarihi DATETIME NOT NULL,
    FOREIGN KEY (doktor_Tc) REFERENCES Doktor(doktor_Tc),
    FOREIGN KEY (hasta_tc) REFERENCES Hasta(hasta_Tc)
);

```

View'lar:

1-) hastanın hangi poliklinikte muayene olduğunu gösteren tablo

```

create view Muayeneler_view as select
Hasta.hasta_Tc as [Hasta TC],
Hasta.hasta_Adi + ' ' + Hasta.hasta_Soyadi as [Hasta Tum Adi],
Cinsiyet.cinsiyet_Adi as [Cinsiyet],
Poliklinik.poliklinik_Adi as [Poliklinik],
Doktor.doktor_Adi + ' ' + Doktor.doktor_Soyadi as [Doktor],
Muayene.muayene_tarihi as [Muayene tarihi]

```

```

from Muayene
inner join Doktor on Doktor.doktor_Tc = Muayene.doktor_Tc
inner join Hasta on Hasta.hasta_Tc = Muayene.hasta_Tc
inner join Poliklinik on Doktor.poliklinik_Id = Poliklinik.poliklinik_Id
inner join Cinsiyet on Cinsiyet.cinsiyet_Id = Hasta.cinsiyet_Id
2-)Doktorların çalıştığı poliklinikler

```

```

create view DoktorPoliklinik
as
select doktor_Tc,doktor_Adi,uzmanlık_alani,poliklinik_Adi from
Doktor inner join Poliklinik on
Doktor.doktor_Tc=Poliklinik.poliklinik_Id

```

3-)hastaların hangi doktor ve hangi poliklinikte ve hangi tarihte muayene olduğunu gösteren tablo

```

create view hastadoktorpoliklinik
as
select hasta_Adi,doktor_Adi,poliklinik_Adi,muayene_tarihi from
Hasta inner join Muayene on
Hasta.hasta_Tc=Muayene.hasta_Tc inner join
Doktor on Muayene.doktor_Tc=Doktor.doktor_Tc inner join
Poliklinik on Doktor.poliklinik_Id=Poliklinik.poliklinik_Id

```

Triggerlar

1-)hasta eklenince silinince ve güncelleme yapılınc tetiklenen yerler

```

create trigger hastaekleme
on Hasta
after insert
as begin
select 'hasta eklendi'
end

```

```

create trigger hastasilindi
on Hasta
after delete
as begin
select 'hasta silindi'
end

```

```

create trigger hastabilgilerigüncellendi
on Hasta
after update
as begin
select 'hasta bilgileri güncellendi'
end

```

2-)Muayene eklenince ve silinince tetiklenen yerler

```

create trigger muayeneolusturma
on Muayene
after insert
as begin
select 'yeni muayene olusturuldu'
end

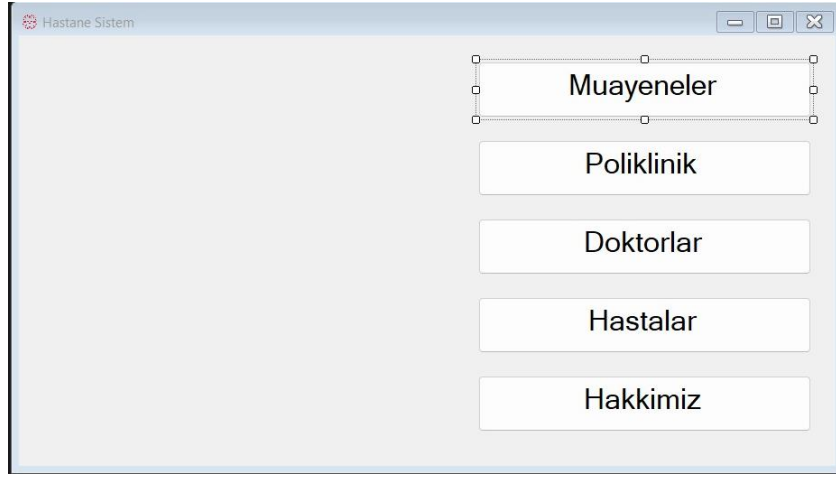
```

```

create trigger muayenesilme
on Muayene
after delete
as begin
select 'muayene silindi'
end

```

Anasayfa:



Hasta kayıt ve yeni muayene sayfası:

Muayene arama ve oluşturma sayfası:

Hastane Sistem - Muayeneler

Muayeneler

T.C Numarasi

Adi Soyadi

Doktor

Poliklinik

Tarih

Thursday , May 2, 2024

Ara

Yeni Muayene Oluşturma

Hastalar sayfası(hasta arama, yeni hasta oluşturma):

Hastane Sistem - Hastalar

Hastalar

T.C Numarasi

Adi Soyadi

Cinsiyet Cinsiyet

Medeni Durum Eğitim Durumu

Kan Grubu Telefon Numarasi

Doğum Tarihi

Thursday , May 2, 2024

Ara

Yeni Hasta Oluştur

Önceki Menüye Dön

Yeni hasta oluřturma:

Hasta Oluřturma

Yeni Hasta Oluřturma

Hasta'nin Bilgileri

T.C Numarasi

Adi

Soyadi

Cinsiyet

Eęitim Durumu

Adres

Meslek

Telefon Nomarasi

Medeni Durumu

Kan Grubu

Doęum Tarihi

Thursday , May 2, 2024

Oluřtur

Önceki Menüye Dön

Doktorlar sayfası:

Hastane Sistem - Doktorlar

Doktorlar

Poliklinik

Doktor

Ara

Önceki Menüye Dön

Poliklinikler sayfası:

Hastane Sistem - Poliklinikler

Poliklinikler

Poliklinik

Ara

Önceki Menüye Dön