



```

create database HospitalManagamentSystem
go
use HospitalManagamentSystem
go

--hasta bilgileri
create table Patient_Information
(
    Patient_ID int primary key identity(1,1) not null,
    Patient_Name nvarchar(50) not null,
    Patient_Surname nvarchar(50) not null,
    Gender nvarchar(10) not null,
    Birth_Date date not null,
    Birth_Place nvarchar(20) not null, --doğumyeri
    File_No int not null
)

--sigorta
create table Insurances
(
    Insurances_ID int primary key identity(1,1) not null,
    Insurances_Name nvarchar(50) not null,
    Active bit not null,
    Patient_Information int not null --birebir bağlı
)

--hasta detay bilgileri
create table Patient_Information_Details
(
    --birebir tablo
    Patient_Information_ID int primary key not null,
    Identification_Number nchar(20) not null, --tc kimlik no
    Address1 nvarchar(250) not null,
    File_No int not null,
    Phone nvarchar(15) not null
)

--randevu
create table Appointments
(
    Appointment_ID int primary key identity(1,1) not null,
    Appointment_Date date not null,
    Appointment_Time time not null,
    Patient_ID int not null,
    Polyclinics_ID int not null,
    Doctor_ID int not null,
    Appointment_Confirmation bit not null --randevu onay
)

```

```
--poliklinikler
create table Polyclinics
(
    Polyclinic_ID int primary key identity(1,1) not null,
    Polyclinic_Name nvarchar(50) not null,
    Active bit not null
)
```

```
--doktorlar
create table Doctors
(
    Doctor_ID int primary key identity(1,1) not null,
    Doctor_Identification_Number nchar(20) not null,
    Doctor_Name nvarchar(50) not null,
    Doctor_Surname nvarchar(50) not null,
    Title nvarchar(50) not null,
    Branch_ID int not null,
    Polyclinic_ID int not null,
    Doctor_Phone nchar(15) not null,
    Doctor_Mail nvarchar(100) not null,
    Doctor_Address nvarchar(50) not null,
    Active bit not null
)
```

```
--branşlar (branch)
create table Branchs
(
    Branch_ID int primary key identity(1,1) not null,
    Branch_Name int not null,
)
```

```
--hemşire
create table Nurses
(
    Nurse_ID int primary key identity(1,1) not null,
    Nurse_Identification_Number nchar(20) not null,
    Nurse_Name nvarchar(50) not null,
    Nurse_Surname nvarchar(50) not null,
    Title nvarchar(50) not null,
    Polyclinic_ID int not null,
    Nurse_Phone nchar(15) not null,
    Nurse_Mail nvarchar(100) not null,
    Nurse_Address nvarchar(50) not null,
    Active bit not null
)
```

```
--vizite
create table Fees
(
    Fee_ID int primary key identity(1,1) not null,
    Polyclinic_ID int not null,
    Doctor_ID int not null,
    Patient_ID int not null, --birebir
    Patient_File_No int not null,
    Fee_Date date not null,
    Fee_Start_Date date not null,
    Fee_End_Time time not null,
    Active bit not null,
    Registration_Date date not null, --kayıt tarihi
    Discharge_Date date not null --taburcu tarihi
)
```

```
--vizite tani
--!!!!ara tablo
create table Fee_Diagnoses
(
    Diagnosis_ID int not null,
    Fee_ID int not null
    primary key(Diagnosis_ID, Fee_ID)
)
```

```
--tani
create table Diagnoses
(
    Diagnosis_ID int primary key not null,
    Diagnosis_Code nvarchar(20) not null,
    Diagnosis_Name nvarchar(100) not null,
    Active bit not null,
    Registration_Date date not null
)
```

```
--hasta çıkış
create table Patient_Discharge
(
    Patient_Discharge_ID int primary key not null,
    Discharge_Date date not null,
    Explanation nvarchar(100) not null --açıklama
)
```

```
--reçete
create table Prescriptions
(
    Prescription_ID int primary key not null,
    Prescription_No int not null,
    --muayene ve reçete arasında birebir ilişki
    Examination_ID int not null,
    Medicine_Piece int not null,
)
```

```

--ilaçlar
create table Medicines
(
    Medicine_ID int primary key identity(1,1) not null,
    Medicine_Name nvarchar(50) not null,
    SGK bit not null,
    Price decimal(18,2) not null
)

--reçete ilaç olacak
--!!!!ara tablo
create table Prescription_Medicine
(
    Prescription_ID int,
    Medicine_ID int
    primary key(Prescription_ID, Medicine_ID)
)

--hasta yatış
create table Patient_Hospitalization
(
    Hospitalization_ID int not null,
    Service_ID int not null,
    Pat_Hos_Date date not null, --hasta yatış tarihi
    Hospitalization bit not null --yatış yapıldı
)

--hasta refakat
create table Patient_Accompaniment
(
    Pat_Acc_ID int primary key identity(1,1) not null,
    Fee_No int not null,
    Identification_Number nchar(11) not null,
    Acc_Name nvarchar(50) not null,
    Acc_Surname nvarchar(50) not null,
    Start_Date date not null,
    End_Date date not null,
    Acc_Address nvarchar(250) not null,
    Acc_Phone nchar(15) not null ,
    Degree_Proximity nvarchar(50) not null --yakınlık derecesi
)

--ameliyat
create table Surgeries
(
    Surgery_ID int primary key identity(1,1) not null,
    Doctor_ID int not null,
    Patient_ID int not null,
    Nurse_ID int not null,
    Surgery_Date date not null,
    Fee_No int not null
)

```

```

--servisler
create table Services
(
    Service_ID int primary key identity(1,1) not null,
    Gender nvarchar(10) not null,
    Service_Name nvarchar(50),
)

--yataklar
create table Beds
(
    Bed_ID int primary key identity(1,1) not null,
    Room_ID int not null,
    Active bit not null,
    --???FullorEmpty bit not null --dolu mu
)

--odalar
create table Rooms
(
    Room_ID int primary key identity(1,1) not null,
    Room_Number int not null,
    Service_ID int not null,
)

--lab
create table Labs
(
    Lab_ID int primary key identity(1,1) not null,
    Lab_Name nvarchar(50),
    Lab_No int
)

--lab test
create table Lab_Test
(
    Lab_Test_ID int primary key identity(1,1) not null,
    Test_Name nvarchar(50)
)

--muayene ve test arasındaki ara tablo
create table Examination_LabTest
(
    Lab_Test_ID int not null,
    Examination_ID int not null,
    primary key(Lab_Test_ID, Examination_ID)
)

```

```

create table Fee_LabTest
(
    Lab_Test_ID int not null,
    Fee_ID int not null,
    primary key(Lab_Test_ID, Fee_ID)
)

--lab sonuçları
create table LabReports
(
    LabReports_ID int primary key not null,
    ----LabTest ve LabReports arasında birebir ilişki kur
    --Doctor_ID int not null,
    Patient_ID int not null,
    Descirption nvarchar(100)
)

create table Radiology_Test
(
    Radiology_Test_ID int primary key not null,
    Radiology_Test_Type nvarchar(50) not null,
)

create table Radiology_Reports
(
    Radiology_Report_ID int primary key not null,
    Patient_ID int not null
)

create table Examination_Radiology
(
    Radiology_Test_ID int not null,
    Examination_ID int not null
    primary key(Radiology_Test_ID, Examination_ID)
)

create table Fee_Radiology
(
    Radiology_Test_ID int not null,
    Fee_ID int not null
    primary key (Radiology_Test_ID, Fee_ID)
)

--muayene
create table Examination
(
    Examination_ID int primary key identity(1,1) not null,
    Doctor_ID int not null,
    Examination_Date date not null
)

```

```

--birebir
alter table Patient_Information_Details add foreign key(Patient_Information_ID)
references Patient_Informations(Patient_ID)

--birebir
alter table Insurances add foreign key (Insurances_ID) references
Patient_Informations(Patient_ID)

--bire çok
alter table Fees add foreign key (Fee_ID) references
Patient_Informations(Patient_ID)
alter table Fees add foreign key (Polyclinic_ID) references
Polyclinics(Polyclinic_ID)
alter table Fees add foreign key (Doctor_ID) references Doctors(Doctor_ID)

--birebir
alter table Patient_Hospitalization add foreign key (Hospitalization_ID)
references Surgeries(Surgery_ID)

--bireçok
alter table Patient_Hospitalization add foreign key (Service_ID) references
Services(Service_ID)

--birebir
alter table Patient_Hospitalization add foreign key (Hospitalization_ID)
references Patient_Accompaniment(Pat_Acc_ID)

--bireçok
alter table Examination add foreign key (Doctor_ID) references Doctors(Doctor_ID)

--birebir
alter table Examination add foreign key (Examination_ID) references
Appointments(Appointment_ID)

--birebir
alter table Prescriptions add foreign key (Prescription_ID) references
Examination(Examination_ID)

--bireçok
alter table Appointments add foreign key (Patient_ID) references
Patient_Informations (Patient_ID)
alter table Appointments add foreign key (Polyclinics_ID) references
Polyclinics(Polyclinic_ID)
alter table Appointments add foreign key (Doctor_ID) references Doctors(Doctor_ID)

--bireçok
alter table Doctors add foreign key(Polyclinic_ID) references
Polyclinics(Polyclinic_ID)

--bireçok
alter table Nurses add foreign key(Polyclinic_ID) references
Polyclinics(Polyclinic_ID)

--bireçok
alter table Doctors add foreign key(Branch_ID) references Branches(Branch_ID)

--bireçok
alter table Surgeries add foreign key (Doctor_ID) references Doctors(Doctor_ID)

```



```

alter table Surgeries add foreign key (Patient_ID) references
Patient_Information(Patient_ID)
alter table Surgeries add foreign key (Nurse_ID) references Nurses(Nurse_ID)

--aratablo
alter table Prescription_Medicine add foreign key (Prescription_ID) references
Prescriptions(Prescription_ID)
alter table Prescription_Medicine add foreign key (Medicine_ID) references
Medicines(Medicine_ID)

--ara tablo
alter table Fee_Diagnoses add foreign key (Diagnosis_ID) references
Diagnoses(Diagnosis_ID)
alter table Fee_Diagnoses add foreign key (Fee_ID) references Fees(Fee_ID)

--birebir
alter table Patient_Discharge add foreign key (Patient_Discharge_ID) references
Fees(Fee_ID)

--bireçok
alter table Rooms add foreign key (Service_ID) references Services(Service_ID)

--bireçok
alter table Beds add foreign key (Room_ID) references Rooms(Room_ID)

--ara tablo
alter table Examination_LabTest add foreign key (Lab_Test_ID) references
Lab_Test(Lab_Test_ID)
alter table Examination_LabTest add foreign key (Examination_ID) references
Examination(Examination_ID)

--aratablo
alter table Fee_LabTest add foreign key (Lab_Test_ID) references
Lab_Test(Lab_Test_ID)
alter table Fee_LabTest add foreign key (Fee_ID) references Fees(Fee_ID)

--birebir
alter table LabReports add foreign key (LabReports_ID) references
Lab_Test(Lab_Test_ID)

--bireçok
alter table LabReports add foreign key (Patient_ID) references
Patient_Information(Patient_ID)

--bireçok
alter table LabReports add foreign key (Patient_ID) references
Patient_Information(Patient_ID)

--birebir
alter table Radiology_Test add foreign key (Radiology_Test_ID) references
Lab_Test(Lab_Test_ID)

--ara tablo
alter table Examination_Radiology add foreign key (Radiology_Test_ID) references
Radiology_Test(Radiology_Test_ID)
alter table Examination_Radiology add foreign key (Examination_ID) references
Examination(Examination_ID)

```

```
--aratablo
alter table Fee_Radiology add foreign key (Radiology_Test_ID) references
Radiology_Test(Radiology_Test_ID)
alter table Fee_Radiology add foreign key (Fee_ID) references Fees(Fee_ID)

--birebir
alter table Radiology_Reports add foreign key (Radiology_Report_ID) references
Radiology_Test(Radiology_Test_ID)

--bireçok
alter table Radiology_Reports add foreign key (Patient_ID) references
Patient_Information(Patient_ID)
```