

# Sorgular

Tags

```
Sorgu Örnekleri
USE HastaneOtomasyon
GO

--İstanbulda ya da Bursada olan hastaneleri gösteren sorgu (Joinsiz)
SELECT * FROM Hastaneler h, Iller i
WHERE (i.ILAdi!= 'istanbul' OR i.ILAdi='Bursa') AND i.ILID=h.ILID;

--(Joinli)
SELECT * FROM Hastaneler h
JOIN Iller i ON h.ILID=i.ILID
WHERE i.ILAdi!= 'istanbul' OR i.ILAdi='Bursa';

--Süte alerisi olan hastaların adı ve soyadı bilgisini getiriniz
SELECT h.HastaID,h.HastaAdi,h.HastaSoyadi FROM Hastalar h
JOIN HastaGecmisi hs ON h.HastaID=hs.HastaID
WHERE hs.SahipOlduguAlerjiler='süt'

--Kan grubu 0 rh - olan ilaç kullanmayan hastaların telefon no ad soyad bilgisini getiren sorgu
SELECT h.HastaAdi+h.HastaSoyadi AS Hasta,h.HastaTelefon,hg.SurekliKullanilanIlaclar FROM Hastalar h
JOIN HastaGecmisi hg ON hg.HastaID=h.HastaID
JOIN KanGrubu kg ON kg.KanGrubuID=h.KanGrubuID
WHERE kg.KanGrubuAdi='0 rh-' AND hg.SurekliKullanilanIlaclar IS NULL

select * from Hastalar h
JOIN KanGrubu kg ON kg.KanGrubuID=h.KanGrubuID
where KanGrubuAdi='0 rh-'

--Fiyatı 20 ile 50 arasında olan ilaçların bilgisini fiyata göre azalan sıralı getiren sorgu (20 dahil 50 dahil degil)
SELECT * FROM Ilacilar i WHERE i.Fiyat>=20 AND i.Fiyat<50
SELECT * FROM Ilacilar i WHERE i.Fiyat BETWEEN 20 AND 50 ORDER BY i.Fiyat DESC

--Adana, Ankara şehirlerinde bulunan polikliniklerin adını getiren sorgu
SELECT p.PoliklinikAdi,h.HastaneAdi,i.ILAdi FROM Poliklinikler p
JOIN HastanePoliklinikDetay hp ON hp.PoliklinikID=p.PoliklinikID
JOIN Hastaneler h ON h.HastaneID=hp.HastaneID
JOIN Iller i ON i.ILID=h.ILID
WHERE i.ILAdi IN('Adana','Ankara');

--Reçete toplam fiyatı
SELECT SUM(i.Fiyat) AS [Toplam Reçete Tutarı],rid.ReceteID from Ilacilar i ,ReceteIlacDetay rid
WHERE rid.IlacID=i.IlacID
GROUP BY rid.ReceteID

--Hastaların ortalama yaşı
SELECT AVG(DATEDIFF(Year,H.HastaDogumTarihi,GETDATE())) FROM Hastalar h;

--Fiyatı en yüksek ilaç
SELECT MAX(i.Fiyat) AS [En yüksek fiyat] from Ilacilar i

--Toplam aile hekimi sayısı
SELECT COUNT(p.PersonelID) from Personeller p WHERE p.UnvanID=1;

--Kaç adet reçete var
SELECT COUNT(DISTINCT rid.ReceteID) from ReceteIlacDetay rid
SELECT COUNT(r.ReceteID) FROM Receteler r

--İsmi 10 karakter olan personel adlarını getir
SELECT p.PersonelAdi from Personeller p WHERE LEN(p.PersonelAdi)=10

--'C' harfi ile başlayan personellerin adlarını göster
SELECT p.PersonelAdi FROM Personeller p WHERE p.PersonelAdi LIKE 'C%';

--İlk harfi abcde lerden herhangi biri olan hastaların adını getiren sorgu
SELECT * FROM Hastalar h WHERE h.HastaAdi LIKE '[abcde]%' ORDER BY h.HastaAdi asc

--İlk harfi a ile başlayan İkinci harfi n ve d olmayan sonu a ve ya n ile biten illeri getiren sorgu
SELECT * FROM Iller i WHERE i.ILAdi LIKE 'a[^nd]%[an]';
```

```

SELECT * FROM iller i WHERE i.IlAdi LIKE '[a]%' AND i.IlAdi NOT LIKE '_[nd]%' AND i.IlAdi LIKE '%[an]'

--Mart ayında ve 1990 dan sonra doğan hastaları doğum tarihine göre sıralı getiren sorgu
SELECT h.HastaAdi+h.HastaSoyadi AS [Hastanın Tam Adı],h.HastaDogumTarihi FROM Hastalar h
WHERE DATENAME(month,HastaDogumTarihi)='March' AND DATEPART (Year,HastaDogumTarihi)>=1990
ORDER BY h.HastaDogumTarihi;

--Son alınan randevunun bilgileri
SELECT TOP 1 r.Tarih, MAX(r.RandevuZamani)
FROM Randevular r
GROUP BY r.Tarih
ORDER BY r.Tarih DESC

--Recetelerin toplam tutarını hesaplayan sorgu
SELECT rid.ReceteId, SUM(i.Fiyat*rid.Adet) AS [Reçete toplam ücreti],i.IlacAdi,rid.Adet,i.Fiyat
FROM Receteler r
JOIN ReceteIlacDetay rid ON r.ReceteId=rid.ReceteID
JOIN Ilacilar i ON i.IlacId=rid.IlacId
GROUP BY rid.ReceteId,i.IlacAdi,rid.Adet,i.Fiyat
ORDER BY rid.ReceteId

--Recete toplam fiyatı 10 ile 20 arasında olan sorgu
SELECT rid.ReceteId, SUM(i.Fiyat*rid.Adet) AS [Reçete toplam ücreti],i.IlacAdi,rid.Adet
FROM Receteler r
JOIN ReceteIlacDetay rid ON r.ReceteId=rid.ReceteID
JOIN Ilacilar i ON i.IlacId=rid.IlacId
GROUP BY rid.ReceteId,i.IlacAdi,rid.Adet
HAVING SUM(i.Fiyat)BETWEEN 10 AND 20
ORDER BY rid.ReceteId

--Hastanın ilaç bilgileri ve aldığı ilaçların fiyat toplamını getiren sorgu (2)

select h.HastaAdi,re.Barkod,i.IlacAdi,i.IcerikBilgisi,rd.Do,rd.KullanimSekli,rd.Adet, SUM(rd.Adet*i.Fiyat) as ToplamFiyat
from Hastalar h
JOIN Randevular r ON h.HastaID=r.HastaID
JOIN Muayeneler m ON r.RandevuID=m.RandevuID
JOIN Receteler re ON m.ReceteID=re.ReceteID
JOIN Ilacilar i ON re.ReceteId=i.IlacId
JOIN ReceteIlacDetay rd ON i.IlacId=rd.IlacId
GROUP BY h.HastaAdi,re.Barkod,i.IlacAdi,i.IcerikBilgisi,rd.Do,rd.KullanimSekli,rd.Adet,i.Fiyat

-----
--VIEWS
-----

/*Muayene sırasında, hekim hastanın önceki başvurularındaki yakınmaları,
tetkik ve tedavi bilgilerine ulaşabilir. (Join+VIEW)*/

go
Create view vw_HastaKayit
AS
select h.HastaID,h.HastaAdi+h.HastaSoyadi AS [Hasta adı ve soyadı],hs.SurekliKullanilanIlacilar,hs.GecirdigiAmeliyatlar,hs.SahipOlduguAlerji
from Hastalar h
JOIN HastaGecmisi hs ON h.HastaID=hs.HastaID
JOIN KanGrubu k ON h.KanGrubuID=k.KanGrubuID
go

SELECT * FROM vw_HastaKayit

-----
--FUNCTIONS
-----
--recetenin toplam fiyatını hesaplayan fonksiyon

-- reçetede her farklı ilacın adet ile fiyatını çarpıyor.uçarpıyor.ucuncu Bir değer ile bütün ilaçların hesaplanması gerek. Eksik sorgu.
go
create FUNCTION fn_ReceteFiyat
(
@ilacAdet int,
@ilacFiyati decimal(18,2)
)
RETURNS decimal(18,2)
AS
BEGIN
RETURN @ilacAdet*@ilacFiyati
END
go

select dbo.fn_ReceteFiyat(rd.Adet,i.Fiyat) as ToplamFiyat,h.HastaAdi,re.Barkod,i.IlacAdi,i.IcerikBilgisi,rd.Do,rd.KullanimSekli,rd.Adet
from Hastalar h
JOIN Randevular r ON h.HastaID=r.HastaID
JOIN Muayeneler m ON r.RandevuID=m.RandevuID

```

```

JOIN Receteler re ON m.ReceteID=re.ReceteID
JOIN Ilacilar i ON re.ReceteId=i.IlacId
JOIN ReceteIlacDetay rd ON i.IlacId=rd.IlacId
GROUP BY h.HastaAdi, re.Barkod, i.IlacAdi, i.IcerikBilgisi, rd.Do, rd.KullanimSekli, rd.Adet, i.Fiyat

--Yaş hesabı yapan fonksiyon

GO
CREATE FUNCTION FNC_YasHesapla
(
    @Tarih DATE
)
RETURNS INT
AS
BEGIN
    RETURN DATEDIFF(year, @Tarih, GETDATE());
END
GO

SELECT dbo.FNC_YasHesapla(h.HastaDogumTarihi) FROM Hastalar h WHERE h.HastaID=1;

-----
--STORE PROCEDURES
-----

--Randevuya gelmeyenler

go
Create PROC sp_RandevuyaGelmeyenler
(
    @Geldimi bit
)
AS
BEGIN
    select h.HastaAdi+ ' '+h.HastaSoyadi as Hastalar,r.Tarih as BugununTarihi, r.RandevuZamani as Saat,r.Geldimi
    from Randevular r
    JOIN Hastalar h ON r.HastaID=h.HastaID
    where Geldimi = @Geldimi
    ORDER BY R.Tarih ASC
END

select * from Randevular

exec sp_RandevuyaGelmeyenler 0

--Disaridan parametre alan x alerisi olan hastaların adı ve soyadı bilgisini getiriniz(StoreProcedure)(Parametreli)
go
CREATE PROCEDURE SP_AlerjidenGetir
@alerjiAdi NVARCHAR(50)
AS
BEGIN
select h.HastaID,h.HastaAdi,h.HastaSoyadi from Hastalar h
JOIN HastaGecmisi hs ON h.HastaID=hs.HastaID
WHERE hs.SahipOlduguAlerjiler=@alerjiAdi
END

EXEC SP_AlerjidenGetir 'Süt' --41 rows
EXEC SP_AlerjidenGetir 'Yazılım' -- xD sdhff
EXEC SP_AlerjidenGetir 'Yumurta' --9 rows

--Kan grubu A+ olan hastaların adı ve soyadı bilgisini getiriniz(StoreProcedure)(Parametresiz)
go
CREATE PROCEDURE SP_KanGrubuAPozitif
AS
BEGIN
SELECT h.HastaID, h.HastaAdi, h.HastaSoyadi FROM dbo.Hastalar h
JOIN KanGrubu kg ON h.KanGrubuID=kg.KanGrubuID
WHERE kg.KanGrubuAdi='A rh+'
END

EXEC SP_KanGrubuAPozitif --40 rows

-----
--TRIGGERS
-----
--Hasta kaydı silinmesin
go
create TRIGGER trg_IdEngelleme ON Hastalar

```

```

INSTEAD OF DELETE
AS
declare @catId int
select @catId = HastaID from deleted
PRINT 'Kaydı silemezsiniz'

delete from Hastalar where HastaID = 10

--HastaGecmisi silinmesin

go
create TRIGGER trg_IdEngelleme ON HastaGecmisi
INSTEAD OF DELETE
AS
declare @catId int
select @catId = GecmisID from deleted
PRINT 'Kaydı silemezsiniz'

delete from HastaGecmisi where GecmisID = 10
-----
--try-catch örneği
-----

BEGIN TRY
    IF EXISTS (select * from sys.databases where name = 'Hastane_Otomasyon')
        BEGIN
            DROP DATABASE Hastane_Otomasyon
            CREATE DATABASE Hastane_Otomasyon
        END
    ELSE
        BEGIN
            CREATE DATABASE Hastane_Otomasyon
        END
END TRY
BEGIN CATCH
    PRINT 'Beklenmedik bir hata oldu.'
END CATCH

```