

SQL Sorguları

Ekleme, Silme, Güncelleme

•(Maaşı 3000 ve altında olan personellerin maaşlarını 4000 yapar.)

```
UPDATE personelbilgi  
SET MAAS=4000 WHERE MAAS<=3000
```

•(İlk sorgu ilaçların stoklarını artan şekilde sıralar. İkinci sorgu ise elde edilen bilgiye göre stoku az olan ilaçtan sipariş verir.)

```
SELECT ILACADI, STOK FROM ilacbilgi  
GROUP BY STOK ASC
```

```
INSERT INTO siparisbilgi(`siparisID`, `SIPARISTARIHI`, `ADET`, `ilacID`, `nakFirmaBilgi`,  
`personelID`) VALUES (NULL, '2021-05-30', '100', '4', '3', '1');
```

•(Stoku 0 olan ilaçları siler.)

```
DELETE FROM ilacbilgi WHERE STOK=0
```

•(Yusuf adlı personeli depo müdürü yapar.)

```
INSERT INTO depomudur(AD, SOYAD, CINSIYET, TELEFON, DOGUMTARIHI,
BASLAMATARIHI, MAAS)
SELECT AD, SOYAD, CINSIYET, TELEFON, DOGUMTARIHI, BASLAMATARIHI, MAAS
FROM personelbilgi WHERE personelbilgi.AD="Yusuf"
```

•(1 numaralı ilacın fiyatını 10, stokunu 100 yapar.)

```
UPDATE ilacbilgi SET STOK=100, FIYAT=10
WHERE ilacID=1
```

Basit Sorgular

•(Selami Caddesinde bulunan depoların kapasitesini verir.)

```
SELECT ADRES, KAPASITE
FROM depobilgi
WHERE ADRES LIKE "%Selami Cad%"
```

•(Kapasitesi 300'ün üstünde olan depoları verir.)

```
SELECT * FROM depobilgi
WHERE KAPASITE>300
```

•(İlaçların fiyatlarını artan şekilde verir.)

```
SELECT ILACADI, FIYAT
FROM ilacbilgi
ORDER BY FIYAT ASC
```

•(Cinsiyeti erkek olan çalışanların ad ve soyad bilgilerini verir.)

```
SELECT AD, SOYAD, CINSIYET
FROM personelbilgi WHERE CINSIYET=0
```

•(Kategorisi "Takviye Vitamin" olan ilaçları döndürür.)

```
SELECT ilacbilgi.ILACADI, ilackategori.KATEGORI
FROM ilacbilgi, ilackategori
WHERE ilacbilgi.kategoriID=ilackategori.kategoriID AND
ilackategori.KATEGORI="Takviye Vitamin"
```

Karmaşık Sorgular

- (Selami Caddesindeki deponun kapasite bilgisini ve kalan yer bilgilerini döner.)

```
SELECT depobilgi.depoID, depobilgi.KAPASITE, depobilgi.KAPASITE-SUM(ilacbilgi.STOK)
AS "KALAN YER" FROM ilacbilgi, depobilgi
WHERE ilacbilgi.depoID=depobilgi.depoID AND depobilgi.ADRES LIKE "%Selami%"
```

- (Hastaların ismini, şikayetlerini, kullandıkları ilacı ve hastayla ilgilenen personeli döner.)

```
SELECT CONCAT(hastabilgi.AD," ",hastabilgi.SOYAD) AS "HASTA ADI",
sikayetbilgi.SIKAYET, ilacbilgi.ILACADI,
CONCAT(personelbilgi.AD," ",personelbilgi.SOYAD) AS "ILGILENEN PERSONEL"
FROM hastabilgi, sikayetbilgi, ilacbilgi, personelbilgi
WHERE hastabilgi.ilacID=ilacbilgi.ilacID AND
hastabilgi.sikayetID=sikayetbilgi.sikayetID AND
hastabilgi.personelID=personelbilgi.personelID GROUP BY hastabilgi.AD
```

- (Konuma göre çalışan personel sayısını döndürür.)

```
SELECT 'Eczane Çalışanları' AS "YER", COUNT(*) AS "TOPLAM" FROM personelbilgi UNION
SELECT 'Depo Personelleri', COUNT(*) FROM depopersonel UNION
SELECT 'Depo Müdürleri', COUNT(*) FROM depomudur
```

- (Hiçbir hastayla ilgilenmeye personelleri döndürür.)

```
SELECT personelbilgi.AD, personelbilgi.SOYAD FROM personelbilgi
WHERE NOT EXISTS (SELECT * FROM hastabilgi WHERE
personelbilgi.personelID=hastabilgi.personelID)
```

- (Muhtar Caddesindeki deponun müdürünü, çalışanını ve açık adresini döndürür.)

```
SELECT CONCAT(depomudur.AD," ",depomudur.SOYAD) AS "DEPO MUDURU",
CONCAT(depopersonel.AD," ",depopersonel.SOYAD) AS "DEPO CALISANI",
depobilgi.depoID, depobilgi.ADRES FROM depobilgi, depopersonel, depomudur
WHERE depobilgi.depoMudurID=depomudur.depoMudurID AND
depopersonel.depoPersonelID=depobilgi.depoPersonelID AND
depobilgi.ADRES LIKE "%Muhtar%"
```

Trigger

- (Bir nakliye firması silindiğinde ilgili firmanın çalışanını veri tabanından silen trigger.)

```
CREATE TRIGGER kayitSilme BEFORE DELETE ON nakliyefirma
FOR EACH ROW DELETE nakliyecalisin, siparisbilgi
FROM nakliyecalisin INNER JOIN siparisbilgi
ON nakliyecalisin.nakFirmaBilgi = siparisbilgi.nakFirmaBilgi
WHERE nakliyecalisin.nakFirmaBilgi=OLD.nakFirmaBilgi
AND siparisbilgi.nakFirmaBilgi=OLD.nakFirmaBilgi
```

Prosedür

- (Depo numarasını, kapasitesini, toplam bulunan ilaç adedini ve kalan yeri döndüren prosedür.)

```
CREATE PROCEDURE depoKalanYer()
SELECT depobilgi.depoID, depobilgi.KAPASITE, SUM(ilacbilgi.STOK) AS "BULUNAN ILAC",
depobilgi.KAPASITE-SUM(ilacbilgi.STOK) AS "KALAN YER"
FROM ilacbilgi, depobilgi
WHERE ilacbilgi.depoID=depobilgi.depoID AND
depobilgi.depoID=1
UNION
SELECT depobilgi.depoID, depobilgi.KAPASITE, SUM(ilacbilgi.STOK) AS "BULUNAN ILAC",
depobilgi.KAPASITE-SUM(ilacbilgi.STOK) AS "KALAN YER"
FROM ilacbilgi, depobilgi
WHERE ilacbilgi.depoID=depobilgi.depoID AND
depobilgi.depoID=2
```

- (Konuma göre verilen toplam maaşı döndüren prosedür.)

```
CREATE PROCEDURE toplamMaas()
SELECT 'Eczane Çalışanları' AS "YER", SUM(personelbilgi.MAAS) AS "TOPLAM" FROM
personelbilgi
UNION
SELECT 'Depo Personelleri', SUM(depopersonel.MAAS) FROM depopersonel
UNION
SELECT 'Depo Müdürleri', SUM(depomudur.MAAS) FROM depomudur
```

Fonksiyon

- (Girilen ilaç ismine göre anlık stok bilgisini döndüren fonksiyon.)

```
CREATE FUNCTION `canliStok`(`ilacismi` VARCHAR(45))
RETURNS int(11)
RETURN (SELECT ilacbilgi.STOK FROM ilacbilgi
WHERE ilacbilgi.ILACADI=ilacismi)
```

- (Girilen personel ismine göre, personelin işe girdiği tarihten bu yana çalıştığı gün bilgisini verir.)

```
CREATE FUNCTION `kacGundurCalisiyor`(`gununTarihi` DATE, `calisanIsmi` VARCHAR(45))
RETURNS int(11)
RETURN (SELECT DATEDIFF(gununTarihi, personelbilgi.BASLAMATARIHI)
FROM personelbilgi WHERE personelbilgi.AD=calisanIsmi)
```