

VERİ TABANI-II DERS NOTLARI

Okul Veri Tabanı Soruları

- 1) Öğrencilerin Tüm bilgilerini listele
`select * from ogrenci`
- 2) Öğretmenlerin iletişim bilgilerini listele
`select mail, tel from ogretmenler`
- 3) Vize notu 50'den küçük olan öğrencilerin numaralarını listele
`select ogr_no from notlar where vize < 50`
- 4) Final notu 20 ile 80 arasında olan öğrencilerin numaralarını listele
`Select ogr_no from notlar where final between 20 and 80`
- 5) Kayıtlı olan kaç öğrenci olduğu listeleyiniz.
`select count(*) from ogrenci`
- 6) Öğrencilerin adlarını A'dan Z'ye listele
`select ad from ogrenci order by ad ASC`
- 7) Bilgisayar programcılığı kayıtlı kaç öğrenci olduğunu listeleyiniz
`select count(*) from ogrenci where bol_no = (select bol_no from bolumler where bol_ad = 'bilgisayar Pr.')`
- 8) Öğrencilerin ikamet ettiği farklı şehirleri listeleyiniz
`select DISTINCT(adres) from ogrenci`
- 9) Samsun ordu ve izmirde ikamet eden öğrencilerin ad ve soyadlarını listeleyiniz
`select ad, soyad from ogrenci where adres IN ('samsun' , 'ordu' , 'izmir')`
- 10) İstanbul ilinde yaşamayan öğrencilerin tüm bilgilerini listeleyiniz
`select * from ogrenci where adres NOT IN ('istanbul')
veya
select * from ogrenci where adres <> ('istanbul')`
- 11) Muhasebe bölümü öğretmenlerinin adlarını listeleyiniz
`select ad from ogretmenler where bolum_no = (select bolum_no from bolumler where bolum_ad = 'muhasebe')`
- 12) Her bir dersten sınava giren öğrenci sayısını listeleyiniz.
`select count(*), ders_id from notlar group by ders_id`
- 13) Her bir bölümde kayıtlı öğrenci sayısını listeleyiniz
`select count(*), bolum_no from ogrenci group by bolum_no`
- 14) Öğretmenlerin adlarını uzunluklarını listeleyiniz
`select LEN(ad) from ogretmenler`
- 15) En yüksek final notunu listeleyiniz
`select max(final) from notlar`

16) Vize notları toplamını listeleyiniz

```
select SUM(vize) from notlar
```

17) Veri tabanı dersine kayıtlı öğrencilerin adlarını listeleyiniz

```
select ad from ogrenci where ogr_no IN (SELECT ogr_no from notlar where ders_id  
= (select ders_id from ders where ders_ad = 'Veri Tabani'))
```

18) Her bir derse ait vize notları ortalaması 60 üzerinde olan ortalamaları listeleyiniz

```
select AVG(vize), ders_id from notlar group by ders_id having AVG(vize) > 60
```

19) Öğrencilerin adlarının ilk 3 harfini listeleyiniz

```
select LEFT(ad,3) from ogrenci
```

20) Öğrencilerin adlarının son 2 harfini listeleyiniz

```
select RIGHT(ad,2) from ogrenci
```

21) Ders kodlarını yazdıktan sonra araya tire (-) işareti koyduktan sonra ders adlarını listeleyiniz

```
select ders_kodu + '-' + ders_ad from ders
```

22) Öğretmenlerinin adlarının ilk 2 harfini alıp büyük harfe dönüştüren sql cümlelerini yazınız.

```
select UPPER(LEFT(ad,2)) from ogretmenler
```

23) Öğrencilerin ad, soyadlarını ve kayıtlı oldukları bölümlerin adlarını listeleyiniz

```
select ogrenci.ad, ogrenci.soyad, bolumler.bolum_ad from ogrenci, bolumler where  
ogrenci.bolum_no = bolumler.bolum_no
```

24) Öğrencilerin ad ve soyadlarını ve vize_final notlarını listeleyiniz

```
select ogrenci.ad, ogrenci.soyad, notlar.vize, notlar.final from ogrenci, notlar  
where ogrenci.ogr_no = notlar.ogr_no
```

25) Ders odu "123" olan derse kayıtlı öğrencilerin ad, soyad ve numaralarını listeleyiniz.

```
select ogrenci.ad, ogrenci.soyad, ogrenci.ogr_no from ogrenci, notlar, ders  
where notlar.ders_kodu = "123" and ders.ders_kodu = notlar.ders_kodu and  
notlar.ogr_no = ogrenci.ogr_no
```

VERİ TABANI-II DERS NOTLARI

Database yedeği alma:

Database ismine sağ tıkla > Tasks > Back Up (.bak uzantısı ile kaydedilir)

Database getirme:

Database ismine sağ tıkla > Tasks > Restore

Group by

alt alta yazılacak ürünleri tek satırda yazmamıza yardımcı olur

```
select count(xx), gruplanacak_sütun from tablo_adi Group By Gruplanacan_sütun
```

*Her bir temizlik ürününden kaç tane var?

*Her bir giyim ürününden kaç tane var?

Örnek:

Her bir kategoride kaç tane ürün olduğunu saydırır:

```
select count(*), kategori_no from urunler Group By kategori_no
```

Ürün miktarı	Kategori no
313	1
13	2
218	3

Having

Gruplandırılmış yapılarda şart varsa having kullanılır

Normal yapılarda where kullanılır.

```
select count(*), kategori_no from urunler Group By kategori_no Having Count(*) > 100
```

Ürün miktarı	Kategori no
313	1
13	2
218	3

Örnek:

```
select SUM(fiyat), kategori_no from urunler where fiyat>100 Group By kategori_no  
Having SUM(fiyat)>5000
```

Kategori_no	Fiyat	Ürün no
1	80	11
1	125	12
1	300	13
1	20	14
1	250	15

VERİ TABANI-II DERS NOTLARI

1. Her bir kategoriye ait ürünleri ortalamalarını bulunuz.

```
select AVG(ucret), k_no from urunler group by k_no
```

2. Her bir derse ait final notu ortalamalarını listeleyiniz.

```
select AVG(final), ders_kodu from notlar group by ders_kodu
```

3. Her bir bölümde kayıtlı kaç öğrenci olduğunu listeleyiniz

```
select COUNT(*), bolum_no from ogrenci group by bolum_no
```

4. Final notu 50 üzerinde Olan her bir derse ait final notlarını ort. Listeleyiniz.

```
select avg(final), ders_kodu from notlar where final >= 50 group by  
ders_kodu
```

5. Toplam öğrenci sayısı 20 üzerinde olan her bir bölüme ait öğrenci sayılarını listeleyiniz

```
select COUNT(*), bolum_no from ogrenciler group by bolum_no Having  
COUNT(*) >= 20
```

6. Ücreti 250TL ve üzeri olan, her bir kategoriye ait ücretler toplamını yazınız.
`select SUM(fiyat), k_no from urunler where fiyat >= 250 group by k_no`
-
7. Her bir araç modeline ait toplam kiralama ücretini listeleyiniz.
`select SUM(ucret), model_id from araclar group by model_id`
8. Her bir öğretmenin girdiği ders sayılarını listeleyiniz
`select COUNT(ders_id), sicil_no from dersler group by sicil_no`
veya
`select COUNT(*), sicil_no from dersler group by sicil_no`
9. Ücreti 100TL'nin üzerinde(where) olan ve toplam ücretleri 1500TL(having) üzerinde olan her bir kategoriye ait toplam ücretleri listeleyiniz.
`select SUM(fiyat), kategori_no from urunler where fiyat > 100 group by kategori_no having SUM(fiyat) > 1500`
10. Her bir bölümde kaç doktor olduğunu listeleyiniz
`select COUNT(*), bolum_no from doktorlar group by bolum_no`
11. Kira ücreti 120TL'nin altında olan,(where) her bir araç modeline ait ortalama kira ücretlerini listeleyiniz
`select AVG(fiyat), model_id from araclar where fiyat < 120 group by model_id`
12. Toplam kiralama sayısı 20'nin(having) üzerinde olan her bir kitaba Ait toplam kiralama sayılarını listeleyiniz.
`select COUNT(*), ISBN from odunc group by ISBN Having COUNT(*) > 20`
veya
`select COUNT(*), kitap_id from odunc group by kitap_id Having COUNT(*) > 20`
-
13. Her bir isimden kaç öğrenci olduğunu listeleyiniz.
`select ad, Count(*) from ogrenciler Group by ad`
veya
`select Count(*), ad from ogrenciler Group by ad`
14. Her bir şehirde kaç öğrencinin ikamet ettiğini listeleyiniz.
`select sehir, count(*) from ogrenciler Group by sehir`
15. Her sınıftaki/bölümdeki kız ve erkek öğrenci sayılarını listeleyiniz.
`select count(*), bol_no, cinsiyet from ogrenciler Group by bol_no, cinsiyet`
16. Her üyenin okuduğu kitap sayısını listeleyiniz.
`select uye_no, Count(*) from odunc/kira Group by uye_no`

17. Her odanın toplam kaç kere rezerve olduğunu listeleyiniz.

```
select oda_no, Count(*) from rezervasyon Group by oda_no
```

18. Her aracın toplamda kaç kere kiralandığını listeleyiniz.

```
select plaka, count(*) from kiralama Group by plaka
```

19. Cinsiyete göre yaşça en büyük öğrencilerin ad ve doğum tarihlerini listeleyiniz.

```
select ad, cinsiyet MIN(dogum_tarihi) from ogrenciler Group by ad, Cinsiyet
```

20. Beş harfli ve her bir isimden kaç öğretmen olduğunu listeleyiniz

```
select count(*), ad from ogretmenler where ad LIKE '-----' Group by ad
```

21. Adının içinde “E” harfi geçen(**where**), her bir kategoriye ait ürünlerin toplam fiyatlarını listeleyiniz.

```
select k_no SUM(fiyat) from urunler where urun_adi LIKE '%E%' Group by k_no
```

22. Toplam rezervasyon sayısı(**having**) 2 ve üzeri olan her odanın rezervasyon sayılarını listeleyiniz.

```
select oda_no, Count(*) from rezervasyon group by oda_no having Count(*) >= 2
```

23. Fiyatı 100 ile 200(**where**) arasında olan ve stoğu 10(**where**) üzerinde olan ürünlere(?) ait tüm kategorilere ait ortalama fiyatı 120TL’den büyük(**having**) olanları listeleyiniz.

```
select k_no, AVG(fiyat) from urunler where fiyat between 100 and 200 and stok > 10 Group by k_no Having AVG(fiyat) >= 120
```

24. Her bir departmandaki en yüksek maaşlı kadın ve erkek personellerin maaşlarını listeleyiniz.

```
select departman_no, MAX(maas), cinsiyet from personel Group by departman_no, cinsiyet
```

VERİ TABANI-II DERS NOTLARI

1-Bilgisayar bölümü öğrencilerinin ad ve soyadlarını listeleyiniz

```
select ad,soyad from ogrenci where bolum_no=(select bolum_no from bolumler where bolum_ad = 'bilgisayar')
```

2- Veri tabanı dersini alan öğrencilerin numaralarını listeleyiniz.

```
select ogr_no from notlar where ders_kodu = (select ders_kodu from ders where ders_ad = 'veri tabanı')
```

3- Serkan varan’ın vermiş olduğu derslerin adlarını listeleyiniz.

```
select ders_adi from dersler where sicil_no = (select sicil_no from ogretmenler where ogretmen_ad = 'serkan' and ogretmen_soyad = 'varan')
```

4- Adı Ahmet olan öğrencilerin almış oldukları vize ve final notlarını listeleyiniz.

```
select vize, final from notlar where ogr_no IN (select ogr_no from ogrenci where ad = 'ahmet')  
(Birden fazla Ahmet olabileceği için = yerine IN kullandık.)
```

5- Muhasebe bölümüne ait dersleri listeleyiniz.

```
select ders_adi from dersler where bolum_no = (select bolum_no from bolumler where bolum_adi = 'Muhasebe')
```

1-Ofis programları dersini alan öğrencilerin ad ve soyadlarını listeleyiniz.

```
select ad,soyad from ogrenci where ogr_no IN (select ogr_no from notlar where ders_kodu = (select ders_kodu from dersler where ders_adi = 'Ofis programlari'))
```

2-Bilgisayar bölümü öğrencilerinin aldığı notları listeleyiniz.

```
select vize, final, but from notlar where ogr_no IN (select ogr_no from ogrenci where bolum_no = (select bolum_no from bolumler where bolum_adi = 'Bilgisayar'))
```

3-Tuğba Cansu'nun vermiş olduğu dersleri alan öğrencilerin numaralarını listeleyiniz.

```
select ogr_no from notlar where ders_kodu = (select ders_kodu from dersler where sicil_no = (select sicil_no from ogretmenler where ad = 'tuğba' and soyad = 'Cansu'))
```

4-Giyim kategorisine ait ürünlerin adlarını listeleyiniz.

```
select urun_adi from urunler where kategori_no = (select kategori_no from kategoriler where kategori_adi = 'Giyim')
```

5- “Ayşe” adlı müşterilerin yapmış olduğu alışveriş bilgilerini listeleyiniz.

```
select * from alisveris where muster_i_no IN (select muster_i_no from musteriler where muster_i_adi= 'ayse')
```

6- “Ayşe” adlı müşterilerin yapmış olduğu alışverişe ait ürünlerin adlarını listeleyiniz.

```
select urun_adi from urunler where urun_id IN (SELECT urun_id from alisveris where muster_i_no= (select muster_i_no from musteriler where muster_i_adi='ayse'))
```

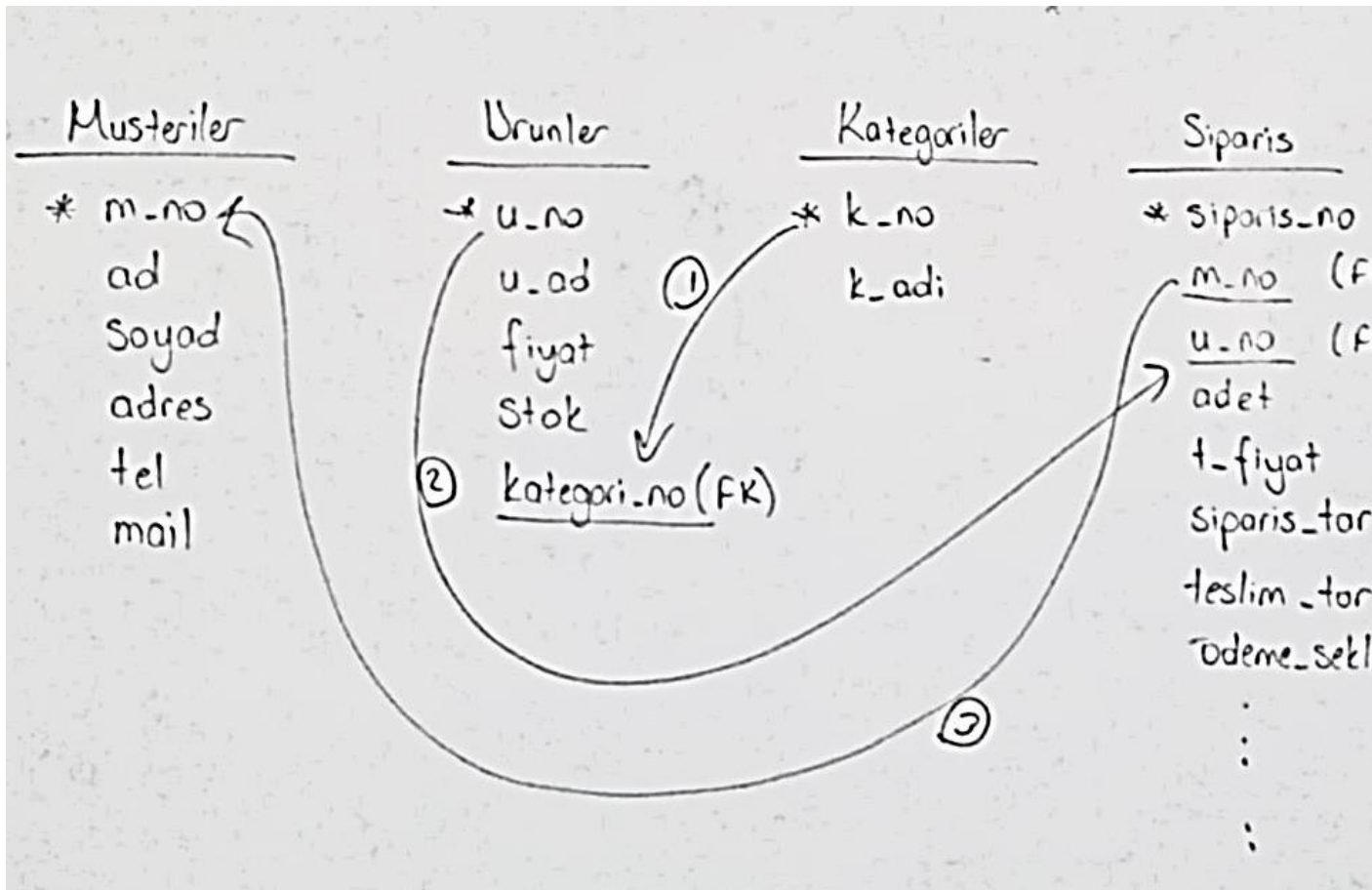
Müşteriler	Ürünler	Kategoriler	Sipariş
*m_no	*u_no	*k_no	*sipariş_no
Ad	U_ad	K_adi	M_no(FK)
Soyad	Fiyat		U_no (FK)
Adres	Stok		Adet
Tel	Kategori_no(FK)		T_fiyat
mail			Sipariş_tar
			Teslim_tar
			Odeme_sekli

1-Fiyatı 100TL altında olan ürünleri satın alan müşteri numaralarını listeleyiniz.

```
select m_no from siparis where urun_no = (select u_no from urunler where fiyat < 100)
```

2- Adı Ahmet olan müşterilerin satın almış oldukları ürünlerin kategorilerinin adlarını listeleyiniz.

```
select k_ad from kategoriler where k_no = (select k_no from urunler where u_no =(select u_no from siparis where m_no = (select m_no from musteriler where m_ad = 'Ahmet')))
```



VERİ TABANI-II DERS NOTLARI

1)Giyim kategorisine ait ürün adlarını listeleyiniz

```
Select urun_adi from urunler where k_no= (select k_no from kategoriler where k_adi='giyim')
```

2)Veri tabanı dersinden alınan notları listeleyiniz.

```
Select vize,final from notlar where ders_kodu= (select ders_kodu from dersler where d_adi='Veri Tabanı')
```

3) Veri tabanı dersini alan öğrencilerin adlarını ve soyadlarını listeleyiniz.

```
select ad,soyad from ogrenciler where ogr_no = (select ogr_no from notlar where ders_kodu = (select ders_kodu from dersler where ders_adi='Veri Tabanı'))
```

VERİ TABANI-II DERS NOTLARI

Truncate Table deneme

(Tablonun tamamını siler)

Delete from deneme

(Tablo içeriğinin tamamını siler)

CREATE >
OLUŞTURMAK
ALTER >
DEĞİŞTİRMEK
DROP > SİLMEK

} Nesne (Database,table,Column)

Insert
Update
Delete

} İçerik

```
Create DataBase okull
---
Create Table bolumler(b_no int, b_adi varchar(30))
---
Create Table bolumler(b_no int Not Null, b_adi varchar(30))
---
Create Table bolumler(b_no int Identity(1,1) Primary Key Not Null, b_adi varchar(30))
---
Create Table ogrenciler (
ogr_no char(12) Primary Key Not Null,
ad varchar(40),
soyad varchar(50),
tel char (12),
tc char(11),
adres varchar(200),
mail varchar(50),
b_no int FOREIGN KEY REFERENCES bolumler(b_no)
)
---
```

Okul veri Tabanı kodları (Her kısım ayrı Query'de yapıldı)

```
use okull
go
Create Table ogretmenler (
sicil_no char(12) Primary Key Not Null,
ad varchar(50),
soyad varchar(50),
tel char(12),
mail varchar(50),
b_no int FOREIGN KEY REFERENCES bolumler(b_no)
)

```

```
use okull
go
Create Table dersler (
ders_id char(12) Primary Key Not Null,
ders_ad varchar(50),
kredi varchar(20),
ders_notu int,
ders_kodu varchar(20),
sicil_no char(12) FOREIGN KEY REFERENCES ogretmenler(sicil_no)
)

```

```
use okull
go
Create Table notlar (
not_id char(12) Primary Key Not Null,
ders_id char(12) FOREIGN KEY REFERENCES dersler(ders_id),
tc char(11) FOREIGN KEY REFERENCES ogrenciler(tc),
vize int,
final int
)

```

```
use okull
go
Create Table bolumler (
b_id char(12) Primary Key Not Null,
b_ad varchar(50)
```



```
)  
use okull  
go  
Create Table ogrenciler (  
tc char(11) Primary Key Not Null,  
ogr_no char(12),  
ad varchar(40),  
soyad varchar(50),  
tel char (12),  
adres varchar(200),  
mail varchar(50),  
b_no int FOREIGN KEY REFERENCES bolumler(b_no)  
)
```

VERİ TABANI-II DERS NOTLARI

Char = tel,tc,sicil, ISBN

Varchar = mail, adres... metinsel ifadeler

sütun ekleme

```
ALTER TABLE urunler  
add asd char
```

sütun silme

```
ALTER TABLE urunler  
drop column asd
```

Tablo ekleme

```
CREATE TABLE newtable(sutun1 char(10), sutun2 int)
```

Tablo silme

```
DROP TABLE newtable
```

Veri tipi değiştirme (char(10) olan sutun1'i int olarak değiştirdik)

```
ALTER TABLE newtable  
ALTER Column sutun1 int
```

FK'yi eklemeyi unuttuysak sonradan eklemek için:

```
Alter Table doktorlar  
Add Foreign key (bolum_no) references bolumler(bolum_no)
```

HASTANE VERİ TABANI T-SQL KODLARI

QUERY-1 (HASTA)

```
CREATE TABLE hasta (  
tc char(11) primary key not null,  
ad varchar(40),  
soyad varchar(40),  
cinsiyet varchar(10),  
adres varchar(50),  
tel varchar(11),  
mail varchar(50)  
)
```

QUERY-2 (Bolumler)

```
CREATE TABLE bolumler(  
bolum_no int primary key not null,  
bolum_adi varchar (30)  
)
```

QUERY-3 (Doktorlar)

```
CREATE TABLE doktorlar(  
sicil_no varchar(20) primary key not null,  
ad varchar (50),  
soyad varchar (50),
```

```
tel varchar (11),
mail varchar (40),
bolum_no int foreign key references bolumler(bolum_no)
)
```

QUERY-4 (Randevu)

```
CREATE TABLE randevular(
randevu_no varchar(50) primary key not null,
tc char(11) foreign key references hasta(tc),
sicil_no varchar(20) foreign key references doktorlar(sicil_no),
tarih date,
)
```

Tablolara veri ekleme

```
insert INTO hasta VALUES (59911326565, 'hakan', 'ocal', 'erkek', 'samsun', 5071891806,
'hakan55@gmail.com')
insert INTO hasta VALUES (59911326365, 'mustafa', 'keser', 'erkek', 'istanbul',
5371891806, 'mustafa34@hotmail.com')
insert INTO hasta VALUES (59911326765, 'hatice', 'gün', 'kadın', 'trabzon',
5341891806, 'hatice@gmail.com')
```

KÜTÜPHANE T-SQL KODLARI

```
Use master
Go
Create database kütüphane
```

```
Use kütüphane
Go
Create Table yazalar (yazar_id int primary key identity (1,1) not null,
Yazar_ad varchar(15),
Yazar_soyad varchar(20),
Tel char(11),
Mail varchar(40)
)
```

```
Use kütüphane
Go
Create table tur (
tur_id int identity (1,1) not null primary key,
tur_ad varchar (30),
)
```

```
Create table kitaplar (
ISBN char(13) not null primary key,
k_ad varchar (30),
sayfa int,
stok int,
tur_id int foreign key references tur (tur_id),
yazar_id int foreign key references yazarlar (yazar_id),
yayin_evi_id int foreign key references yayinevleri (y_id)
)
```

```
use kutuphane
go
Create Table yayinevi
(
y_id int identity(1,1) not null primary key,
y_ad varchar (30),
adres varchar(50),
tel char (11),
)
```

```
Create Table uyeler
(
uye_id int identity (1,1) not null primary key,
ad varchar (20),
soyad varchar (25),
```

```

tc char(11),
tel char (11),
mail varchar (30)
)

Create Table odunc
(
odunc_id int identity (1,1) not null,
uye_id int,
uye_id int ..... , primary key
mail varchar (30),
ISBN char(13),
odunc_tar date,
teslim_tar date
)

```

VERİ TABANI-II DERS NOTLARI

Sadece belirlediğimiz Tabloları Görmek için View kullanılır:

```

Create View vw_urunler
as
select fiyat, stok from urunler

```

```

Create View vw_musteriler
as
select musteriler.m_ad, musteriler.m_soyad, urunler.u_ad from musteriler, urunler,
satis where musteriler.m_no = satis.m_no and satis.u_no = urunler.u_no

```

Şifreleme:

```

Create View vw_deneme
with ENCRYPTION
As
select musteriler.m_ad, musteriler.m_soyad, urunler.u_ad from musteriler, urunler,
siparis where musteriler.m_id = siparis.m_id and siparis.barkod = urunler.barkod

```

Viewleri görüntüler(şifreli olanlar görüntülenemez):

```

sp_helptext vw_urunler

```

Trigger (Tetikleme)

```

Create Trigger trg_stokazalt
on siparis
after insert
as
declare @u_id int
declare @satilan_adet int
select @u_id = u_id, @satilan_adet = adet from inserted
update urunler set stok = stok - @satilan_adet where u_id = @u_id

```

veya

```

Create Trigger trg_stokazalt
on siparis
after insert
as
declare @barkod char(10)
declare @satilan_adet int
select @barkod = barkod, @satilan_adet = adet from inserted

```

```
update urunler set stok = stok - @satilan_adet where barkod = @barkod
```

Satış ve stok azaltma

```
INSERT INTO siparis (m_id, barkod, adet, t_fiyat, odeme_sekli, siparis_tar,  
teslim_tar, u_id)  
VALUES (4, '2', 1, 3200, 'nakit', '01-10-2019', '01-11-2019', 4 );
```