

```
Use KursDB;
```

```
--KursAdi, SehirAdi,IlceAdi,AcilanKursEgitmenUcreti,AcilanKursKursUcreti
```

```
Select KursAdi,  
SehirAdi,  
IlceAdi,  
AcilanKursEgitmenUcreti,  
AcilanKursKursUcreti from AcilanKurs, Kurs, Ilce, Sehir  
where  
AcilanKurs.AcilanKursKursID = Kurs.KursID and  
AcilanKurs.AcilanKursIlceID = Ilce.IlceID and  
Ilce.IlceSehirID = Sehir.SehirID
```

```
-- Acilan Kurs adını, egitmen adını, şehir adını ve ilçe adını getiren sorguyu yazınız
```

```
Select KursAdi,EgitmenAdi,EgitmenSoyadi, SehirAdi, IlceAdi from AcilanKurs, Kurs,  
KursEgitmen, Egitmen,Ilce, Sehir  
where  
AcilanKurs.AcilanKursKursID = Kurs.KursID and  
AcilanKurs.AcilanKursIlceID = Ilce.IlceID and  
AcilanKurs.AcilanKursID = KursEgitmen.KursEgitmenAcilanKursID and  
Egitmen.EgitmenID = KursEgitmen.KursEgitmenEgitmenID and  
Ilce.IlceSehirID = Sehir.SehirID
```

```
-- açılan kurs adı ve kursu alan öğrenci sayılarını
```

```
Select KursAdi,  
(Select COUNT(*) from KursOgrenci  
where  
KursOgrenci.KursOgrenciAcilanKursID = AcilanKurs.AcilanKursKursID)  
from AcilanKurs, Kurs  
where Kurs.KursID = AcilanKurs.AcilanKursKursID
```

```
use KursDB;
```

```
--Açılan kursların adlarını ve şehirlerin adlarını ekrana getiren sql sorgusu
```

```
Select KursAdi, SehirAdi from AcilanKurs, Kurs, Ilce, Sehir  
where AcilanKurs.AcilanKursKursID = Kurs.KursID and  
AcilanKurs.AcilanKursIlceID = Ilce.IlceID and  
Ilce.IlceSehirID = Sehir.SehirID
```

```
-- Öğrenci Adi, Ogrenci soyadı,  
-- Öğrencinin gittiği kurs adı, bu kursun şehir adı  
-- Bu kursu veren eğitmenin adı, soyadı
```

```

Select Ogresci.OgresciAdi, Ogresci.OgresciSoyadi,
Kurs.KursAdi,Sehir.SehirAdi,
Egitmen.EgitmenAdi,Egitmen.EgitmenSoyadi
from Ogresci,KursOgresci,AcilanKurs,Kurs, Ilce,Sehir,KursEgitmen,Egitmen
Where Ogresci.OgresciID = KursOgresci.KursOgresciOgresciID and
KursOgresci.KursOgresciAcilanKursID = AcilanKurs.AcilanKursID and
Kurs.KursID = AcilanKurs.AcilanKursKursID and
AcilanKurs.AcilanKursIlceID = Ilce.IlceID and
Sehir.SehirID = Ilce.IlceSehirID and
KursEgitmen.KursEgitmenAcilanKursID = AcilanKurs.AcilanKursID and
KursEgitmen.KursEgitmenEgitmenID = Egitmen.EgitmenID

```

--INNER JOIN kullanarak Açılan kursların adlarını ve şehirlerin adlarını ekrana getiren sql sorgusu

```

SELECT k.KursAdi, s.SehirAdi FROM AcilanKurs AS ack
INNER JOIN Kurs AS k ON ack.AcilanKursKursID = k.KursID
INNER JOIN Ilce AS i ON ack.AcilanKursIlceID = i.IlceID
INNER JOIN Sehir AS s ON s.SehirID = i.IlceSehirID

```

-- INNER JOIN kullanarak
-- Öğrenci Adi, Ogresci soyadı,
-- Ogrescinin gittiği kurs adı, bu kursun sehir adı
-- Bu kursu veren eğitmenin adı, soyadı

```

SELECT o.OgresciAdi,o.OgresciSoyadi,k.KursAdi,s.SehirAdi,e.EgitmenAdi,e.EgitmenSoyadi FROM
Ogresci AS o
INNER JOIN KursOgresci AS ko ON ko.KursOgresciOgresciID = o.OgresciID
INNER JOIN AcilanKurs AS ack ON ko.KursOgresciAcilanKursID = ack.AcilanKursID
INNER JOIN Kurs AS k ON k.KursID = ack.AcilanKursKursID
INNER JOIN Ilce AS i ON i.IlceID = ack.AcilanKursIlceID
INNER JOIN Sehir AS s ON s.SehirID = i.IlceSehirID
INNER JOIN KursEgitmen AS ke ON ke.KursEgitmenAcilanKursID = ack.AcilanKursID
INNER JOIN Egitmen AS e ON ke.KursEgitmenEgitmenID = e.EgitmenID

```

-- Acilan kursun ücretini ve ilçesini getirin.

```

Select IlceAdi, AcilanKursKursUcreti from AcilanKurs AS ack
INNER JOIN Ilce AS i ON i.IlceID = ack.AcilanKursIlceID

```

-- Acilan kursun ücretini ve ilçesini getirin. İlçesi girilmemiş olanlar da gelsin

```

Select IlceAdi, AcilanKursKursUcreti from AcilanKurs AS ack
LEFT JOIN Ilce AS i ON i.IlceID = ack.AcilanKursIlceID

```

-- Bütün ilçeleri getir. İlçelerdeki kursların ücreti, kurs yoksa ücretre NULL getir.

```

Select IlceAdi, AcilanKursKursUcreti from AcilanKurs AS ack
RIGHT JOIN Ilce AS i ON i.IlceID = ack.AcilanKursIlceID

```

--Açılan Kurs adı, ücreti sehir adı, ilce adı. ilçesi null verilerde gelsin

```

Select k.KursAdi, ack.AcilanKursKursUcreti,s.SehirAdi, i.IlceAdi from AcilanKurs AS ack
INNER JOIN Kurs AS k ON k.KursID = ack.AcilanKursKursID
LEFT JOIN Ilce AS i ON i.IlceID = ack.AcilanKursIlceID
LEFT JOIN Sehir AS s ON s.SehirID = i.IlceSehirID

```

```
use KursDB;
```

```
--ilcede açılan İlçe isimleri, kursların sayısını
```

```
Select İlceAdi, COUNT(KursAdi) from AcilanKurs as ack  
left join Kurs as k on k.KursID = ack.AcilanKursKursID  
left join İlce as i on i.İlceID = ack.AcilanKursİlceID  
group by İlceAdi  
having İlceAdi like '%a'
```

```
-- Eğer ilçede açılan kurs yoksa 0 değerini getirsin.
```

```
--(Once ilçe sonra açılan kurs ilişkilendirilecek)
```

```
Select İlceID, COUNT(*) as Sayi from İlce,AcilanKurs where İlceID=AcilanKursİlceID  
group by İlceID
```

```
--Oğrencinin aldığı kurs sayısını öğrencinin adı soyadıyla birlikte
```

```
Select o.OğrenciAdi, OğrenciSoyadi, COUNT(KursOğrenciAcilanKursID) from KursOğrenci as ko  
inner join Öğrenci as o on o.OğrenciID = ko.KursOğrenciOğrenciID  
group by OğrenciAdi, OğrenciSoyadi  
--having OğrenciSoyadi like '%E%'
```

```
--Oğrencinin adını, soyadını, doğum tarihini ve aldığı kurs sayısını
```

```
-- ÖğrenciID ile gruplandırma yapılsın.
```

```
Select ÖğrenciID , COUNT(KursOğrenciAcilanKursID) as Sayi from Öğrenci as o  
inner join KursOğrenci ko on o.OğrenciID=ko.KursOğrenciOğrenciID  
group by ÖğrenciID
```

```
Select *,  
(Select ÖğrenciAdi from Öğrenci where Öğrenci.OğrenciID = SorguSonucu.OğrenciID) as  
OğrAdi,  
(Select ÖğrenciSoyadi from Öğrenci where Öğrenci.OğrenciID = SorguSonucu.OğrenciID) as  
OğrSoyadi,  
(Select ÖğrenciDogumTar from Öğrenci where Öğrenci.OğrenciID = SorguSonucu.OğrenciID) as  
OğrDogumTarihi  
from (Select ÖğrenciID , COUNT(KursOğrenciAcilanKursID) as Sayi from Öğrenci as o  
inner join KursOğrenci ko on o.OğrenciID=ko.KursOğrenciOğrenciID  
group by ÖğrenciID) as SorguSonucu
```

```
--CREATE ile oluşturulur. ALTER ile düzenleme yapılır.
```

```
--Create ile oluşturulduktan sonra aynı isimle CREATE yapılamaz.
```

```
-----
```

```
--CREATE VIEW VwOğrenciAdiKursAdi AS
```

```
--Select ÖğrenciAdi,OğrenciSoyadi, KursAdi from KursOğrenci as ko
```

```
--inner join Öğrenci as o on o.OğrenciID = ko.KursOğrenciOğrenciID
```

```
--inner join AcilanKurs as ack on ack.AcilanKursKursID = ko.KursOğrenciAcilanKursID
```

```
--inner join Kurs as k on k.KursID = ack.AcilanKursKursID
```

```
Select * from VwOğrenciAdiKursAdi
```

```
ALTER PROCEDURE Sp_OğrAdiKursAdi
```

```

@soyadi nvarchar(5),
@adi nvarchar(5)
AS
BEGIN
Select OgrenciAdi,OgrenciSoyadi, KursAdi from KursOgrenci as ko
inner join Ogrenci as o on o.OgrenciID = ko.KursOgrenciOgrenciID
inner join AcilanKurs as ack on ack.AcilanKursKursID = ko.KursOgrenciAcilanKursID
inner join Kurs as k on k.KursID = ack.AcilanKursKursID
where OgrenciSoyadi like '%' + @soyadi + '%' and
OgrenciAdi like '%' + @adi + '%'
END

```

```

EXEC Sp_OgrAdiKursAdi @soyadi='Ş', @adi = 'a'

```

```

-- AcilanKursUcreti 18000 ve daha fazla olan kursların adını
-- ve egitmenlerin adını getiren
-- View ve stored procedure oluşturunuz.
--stored procedure kısmında 18000 dışarıdan alsın
--hafta11Soru.sql dosyasında cevabı
-----

```

```

--WITH CHECK OPTION
-- View üzerinden veri eklerken WITH CHECK OPTION sözcüğü kullanılırsa
-- Where kısıtındaki değerlere uygun olarak tabloya ekleme yaptırılabilir.

```

```

ALTER VIEW VwAcilanKursEkle AS
SELECT * FROM AcilanKurs where
AcilanKursKursUcreti>=10000 and
AcilanKursEgitmenUcreti>=15000
WITH CHECK OPTION

```

```

INSERT INTO VwAcilanKursEkle
(AcilanKursKursID,AcilanKursIlceID,AcilanKursEgitmenUcreti,AcilanKursKursUcreti)
VALUES
(2,4,15000,11000)

```

```

-- Acilan kurs isimlerini, Şehir isimlerini ve
-- eğitmen ad soyadını getiren stored procedure yazınız.
-- Kullanıcı dışarıdan şehir adını ve
-- maksimum kurs ücretini girebilsin.

```

```

ALTER PROCEDURE SpKursAdiEgitmenAdiSehirAdi
@sehirAdi nvarchar(50),
@maxKursUcreti int

```

```

AS
BEGIN
Select KursAdi, SehirAdi, EgitmenAdi,EgitmenSoyadi, AcilanKursKursUcreti from AcilanKurs
as ack
inner join Kurs as k on k.KursID = ack.AcilanKursKursID
inner join Ilce as i on i.IlceID = ack.AcilanKursIlceID
inner join Sehir as s on i.IlceSehirID = s.SehirID
inner join KursEgitmen as ke on ke.KursEgitmenAcilanKursID = ack.AcilanKursKursID
inner join Egitmen as e on e.EgitmenID = ke.KursEgitmenEgitmenID
where SehirAdi = @sehirAdi and AcilanKursKursUcreti <= @maxKursUcreti
END

```

```

EXEC SpKursAdiEgitmenAdiSehirAdi @sehirAdi='Bursa', @maxKursUcreti = 14000

```

```

-- Öğrenci tablosuna veri eklenirken Doğum tarihi
-- 01.01.1950 ve 01.01.2003 arasında olan öğrencilerin eklenmesi
-- isteniyor bunun için yazılacak olan Viewı yazınız.
-- WITH CHECK OPTION

CREATE VIEW VwOgrenciEkle AS
    Select * from Ogrenci
        where OgrenciDogumTar>'01.01.1950' and OgrenciDogumTar<'01.01.2003'
WITH CHECK OPTION

INSERT INTO VwOgrenciEkle(OgrenciAdi,OgrenciSoyadi,OgrenciDogumTar)
VALUES
('sadsa','dsad','11.11.2001')

```

```

CREATE VIEW VwSoru AS
SELECT Kurs.KursAdi, Egitmen.EgitmenAdi, Egitmen.EgitmenSoyadi,
AcilanKurs.AcilanKursKursUcreti
FROM AcilanKurs
INNER JOIN Kurs ON AcilanKurs.AcilanKursKursID = Kurs.KursID
INNER JOIN KursEgitmen ON AcilanKurs.AcilanKursID = KursEgitmen.KursEgitmenAcilanKursID
INNER JOIN Egitmen ON KursEgitmen.KursEgitmenEgitmenID = Egitmen.EgitmenID

Where AcilanKursKursUcreti >=18000

Select * from VwSoru

CREATE PROCEDURE SpSoru
@minUcret int
AS
BEGIN
SELECT Kurs.KursAdi, Egitmen.EgitmenAdi, Egitmen.EgitmenSoyadi,
AcilanKurs.AcilanKursKursUcreti
FROM AcilanKurs
INNER JOIN Kurs ON AcilanKurs.AcilanKursKursID = Kurs.KursID
INNER JOIN KursEgitmen ON AcilanKurs.AcilanKursID = KursEgitmen.KursEgitmenAcilanKursID
INNER JOIN Egitmen ON KursEgitmen.KursEgitmenEgitmenID = Egitmen.EgitmenID

Where AcilanKursKursUcreti >=@minUcret
END

EXEC SpSoru @minUcret = 21000

```

```
DECLARE @ad nvarchar(25), @soyad nvarchar(25), @tarih datetime
DECLARE @numara int
```

```
SELECT @numara=10
Set @numara=10
```

```
DECLARE @EgitmenAdi varchar(30)
SELECT @EgitmenAdi='%Gökhan%'
SELECT * FROM Egitmen WHERE EgitmenAdi like @EgitmenAdi
Print @@error
Print @@rowcount
```

```
SELECT *,
CASE
WHEN KursEgitmenUcreti<17000 THEN 'maaş Az'
WHEN KursEgitmenUcreti>17000 THEN 'maaş çok'
ELSE 'Maaş Normal'
END
AS MaasDurumu,
CASE KursSehirID
WHEN 1 THEN 'Bursa'
WHEN 2 THEN 'Balıkesir'
WHEN 3 THEN 'İzmir'
END
AS Sehir,
CASE KursSehirID
WHEN 1 THEN KursEgitmenUcreti*1.3
WHEN 2 THEN KursEgitmenUcreti*1.1
else KursEgitmenUcreti
end as ZamliMaas
FROM Kurs
```

```
Select *, IIF(KursEgitmenUcreti>17000, 'maaş çok', 'maaş az') from Kurs
```

--Eğer sql deyimi bir den fazla satır dan oluşursa BEGIN ... END arasında yazılır.

```
DECLARE @borc int
Select @borc=4000
IF @borc>5000
    Select * from Egitmen
ELSE
    Select * from KursEgitmen
```

--While yapısı bir sql deyiminin tekrarlanmasını sağlar. Koşul yanlış oluncaya kadar döngü çalışır.

--BREAK : döngüden koşul sağlanmadan yani erken çıkılmasını sağlar.

--CONTINUE : devamındaki satırlar işlenmeden döngünün ilk satırına dönülmesini sağlar.

```
DECLARE @sayi int
SELECT @sayi=1
```

```

WHILE @sayi<=12
BEGIN
PRINT @sayi
SELECT @sayi = @sayi + 1
END

```

```

DECLARE @sonEklenenID int

```

```

BEGIN TRANSACTION
BEGIN TRY
    INSERT INTO Egitmen (EgitmenAdi, EgitmenSoyadi)
    VALUES ('Betül', 'ESEN')

    SET @sonEklenenID = SCOPE_IDENTITY()

    INSERT INTO KursEgitmen (KursEgitmenAcilanKursID, KursEgitmenEgitmenID)
    VALUES (14, @sonEklenenID)

    COMMIT

END TRY
BEGIN CATCH
    ROLLBACK
END CATCH

```

