

```
use PersonelDB;
```

```
create table Kisi (  
KisiID int IDENTITY(1,1) not null,  
KisiAdi nvarchar(150) not null,  
KisiSoyadi nvarchar(150) not null,  
KisiMaas smallint not null,  
KisiYoneticisiMi bit not null,  
KisiBirim nvarchar(150) null  
)
```

```
use PersonelDB;
```

```
insert into Kisi (KisiAdi, KisiSoyadi, KisiBirim, KisiMaas, KisiYoneticisiMi)  
values ('Süleyman Gökhan', 'TAŞKIN', 'Yazılım', 9000, 1)
```

```
use PersonelDB;
```

```
alter table Kisi drop column KisiBirim
```

```
use PersonelDB;
```

```
create table Birim (  
BirimID int IDENTITY(1,1) not null,  
BirimAdi nvarchar(250) not null,
```

```
Constraint PK_BirimID Primary Key(BirimID)  
)
```

```
use PersonelDB;
```

```
insert into Birim (BirimAdi) values ('Yazılım')  
insert into Birim (BirimAdi) values ('Muhasebe')  
insert into Birim (BirimAdi) values ('İnsan Kaynakları')  
insert into Birim (BirimAdi) values ('Teknik Servis')  
insert into Birim (BirimAdi) values ('Yemekhane')
```

```
use PersonelDB;
```

```
alter table Kisi add KisiBirimID int
```

```
use PersonelDB;
```

```
Select * from Kisi  
Select KisiAdi, KisiSoyadi from Kisi  
Select Kisi.KisiAdi, Kisi.KisiSoyadi from Kisi  
Select K.KisiAdi, K.KisiSoyadi from Kisi as K  
Select K.KisiAdi as Adi, K.KisiSoyadi as Soyadi from Kisi as K  
  
Select K.KisiAdi Adi, K.KisiSoyadi Soyadi from Kisi K
```

```
use PersonelDB;
```

```
select distinct KisiAdi from Kisi
```

```
--Aggregate fonksiyonları
```

```
Select Min(KisiMaas) as EnDusukMaas from Kisi
```

```
Select Max(KisiMaas) as EnYuksekMaas from Kisi
```

```
Select Avg(KisiMaas) as OrtalamaMaas from Kisi
```

```
Select Sum(KisiMaas) as MaasToplami from Kisi
```

```
Select Count(KisiMaas) as NullOlmayanlarınSayisi from Kisi
```

```
Select Count(*) as SatirSayisi from Kisi
```

```
Select Min(KisiMaas) as EnDusukMaas,
```

```
Max(KisiMaas) as EnYuksekMaas,
```

```
Avg(KisiMaas) as OrtalamaMaas
```

```
from Kisi
```

```
use PersonelDB;
```

```
Select * from Kisi where KisiMaas >= 5000
```

```
Select * from Kisi where KisiYoneticisiMi != 0 --eşit değil
```

```
Select * from Kisi where KisiYoneticisiMi <> 0 --eşit değil
```

```
Select * from Kisi where KisiAdres is not null
```

```
Select * from Kisi where KisiAdres is null
```

```
Select * from Kisi where KisiMaas >=5000 and KisiYoneticisiMi=1
```

```
Select * from Kisi where KisiMaas >=5000 or KisiAdres is null
```

```
Select * from Kisi where KisiMaas>=4200 and KisiMaas<=6000
```

```
Select * from Kisi where KisiMaas between 4200 and 6000
```

```
use PersonelDB;
```

```
Select * from Kisi where KisiAdi like '___han' -- 6 karakterli ve han ile bitenler
```

```
Select * from Kisi where KisiAdi like '%han' -- tüm han ile bitenler
```

```
Select * from Kisi where KisiSoyadi like '%TÜRK' -- Sonu TÜRK ile bitenler
```

```
Select * from Kisi where KisiSoyadi like '%TÜRK%' -- içinde TÜRK geçenler
```

```
Select * from Kisi where KisiAdi like 'G%' -- baş harfi G olanlar
```

```
Select * from Kisi where KisiAdi like '_____' -- 5 karakterli olanları getir
```

```
Select * from Kisi where KisiAdi not like '_____' -- 5 karakterli olmayanları getir
```

```
use PersonelDB;
```

```
Select * from Kisi where KisiBirimID =1 or KisiBirimID =2 or KisiBirimID =3
```

```
select * from Kisi where KisiBirimID in (1, 2, 3)
```

```
Select * from Kisi where KisiBirimID != 1 and KisiBirimID != 2 and KisiBirimID !=3
```

```
select * from Kisi where KisiBirimID not in (1, 2, 3)
```

```
--Kişi Birim id 1, 3, 5 olan ve adresi null olmayanları getir.
```

```
Select * from Kisi where KisiBirimID in (1,3,5) and KisiAdres is null
```

```
use PersonelDB;
```

```
Select K.KisiAdi, K.KisiSoyadi, K.KisiBirimID, B.BirimID, B.BirimAdi  
from Kisi as K, Birim as B  
where K.KisiBirimID = B.BirimID
```

```
--Select * from Kisi, Birim
```

```
-- Maaşı 5000 den büyük olan Kisilerin çalıştığı birimleri getiren sorgu (Birden fazla olmayacak)
```

```
Select distinct BirimAdi from Kisi,Birim where KisiBirimID = BirimID and KisiMaas>5000
```

```
use PersonelDB;
```

```
Select * from Birim order by BirimAdi      -- küçükten büyüğe (A-Z)  
Select * from Birim order by BirimAdi asc -- küçükten büyüğe (A-Z)  
Select * from Birim order by BirimAdi desc -- büyükten küçüğe sıralama (Z-A)
```

```
Select * from Kisi order by KisiAdi, KisiSoyadi
```

```
Select * from Kisi where KisiMaas >= 5000 order by KisiAdi, KisiSoyadi
```

```
Select KisiAdi,KisiSoyadi, ASCII('s') from Kisi -- karaktern ascii kodunu verir.
```

```
Select KisiAdi, KisiSoyadi, LEN(KisiAdi) from Kisi -- sütündaki verinin karakter sayısını verir.
```

```
Select KisiAdi, KisiSoyadi, LEN(KisiAdi)+1 from Kisi
```

```
Select KisiAdi, KisiSoyadi, LEN(KisiAdi)+LEN(KisiSoyadi) from Kisi -- adı ve soyadı sütunlarındaki verilerin toplam karakter uzunluğu
```

```
Select KisiAdi, KisiSoyadi, UPPER(KisiSoyadi) as SoyadiBuyuk from Kisi -- sütundaki veriyi büyük harfe çevirir.
```

```
Select KisiAdi, KisiSoyadi, LOWER(KisiSoyadi) as SoyadiKucuk from Kisi -- sütundaki veriyi küçük harfe çevirir.
```

```
Select KisiAdi, KisiSoyadi, SUBSTRING(KisiSoyadi, 1, 2) from Kisi -- SUBSTRING(sütun, başlangıç, karaktersayısı)
```

```
Select KisiAdi, KisiSoyadi,  
ASCII(SUBSTRING(KisiSoyadi, 1, 1)) from Kisi
```

```
Select KisiAdi,KisiSoyadi, LEFT(KisiSoyadi, 2) from Kisi -- soldan başlayarak iki karakter getiriyor.
```

```
Select KisiAdi,KisiSoyadi, RIGHT(KisiSoyadi, 2) from Kisi -- sağdan başlayarak iki karakter getiriyor.
```

```
Select KisiAdi, KisiSoyadi, LTRIM(KisiAdi), LTRIM(KisiSoyadi) from Kisi -- baştaki boşlukları siler
```

```
Select KisiAdi, KisiSoyadi, RTRIM(KisiAdi), RTRIM(KisiSoyadi) from Kisi -- sonrakı boşlukları siler
```

```
Select KisiAdi, KisiSoyadi, TRIM(KisiAdi), TRIM(KisiSoyadi) from Kisi -- her iki taraftaki boşlukları siler.
```

```
Select KisiAdi,KisiSoyadi, REPLACE(KisiAdi, ' ', '#') from Kisi
```

```
Select KisiAdi,KisiSoyadi, REPLACE(RTRIM(KisiAdi), ' ', '#') from Kisi
```

```
Select UPPER('Yazılım Mühendisliği')
```

```
Select
```

```
REPLACE(REPLACE(REPLACE('MDBF {Yazılım Mühendisliği} [Veritabanı Yönetimi]',  
'{', '('), '}', ')'), '[', '('), ']', ')')
```

```
Select TRANSLATE('MDBF {Yazılım Mühendisliği} [Veritabanı Yönetimi]', '{}[]', '()()')  
--üstteki replace gibi parantezleri normal paranteze çevirir.
```

```
Select KisiAdi, KisiSoyadi, REPLICATE(KisiAdi, 5) from Kisi -- kişi adını yanyana 5  
kere yazdırır
```

```
Select KisiAdi, REVERSE(KisiAdi) from Kisi -- stringi ters olarak yazdırır.
```

```
Select KisiAdi, KisiSoyadi, CONCAT(KisiAdi, ' ', KisiSoyadi) from Kisi
```

```
Select ABS(-8) -- mutlak değerini alır
```

```
Select POWER(4,3)
```

```
Select SIN(90)
```

```
Select COS(90)
```

```
Select TAN(90)
```

```
Select COT(90)
```

```
Select LOG(5)
```

```
Select LOG10(5)
```

```
Select CEILING(3.4) -- ondalıklı sayıyı yukarı yuvarlar
```

```
Select FLOOR(3.8) -- ondalıklı sayıyı aşağı yuvarlar
```

```
Select ROUND(145.546321564, 2) -- virgülden sonrasını yakın olan sayıya yuvarlar.
```

```
Select ROUND(145.546321564, -2) -- tam sayı kısmını yakın olan tarafa yuvarlar
```

```
Select CAST('1990' as datetime)
```

```
Select CAST('1990' as int)
```

```
Select CONVERT(datetime, '1990')
```

```
Select KisiMaas, CONVERT(nvarchar, KisiMaas) from Kisi
```

```
select parse('02.05.1990' as datetime using 'en-US')
```

```
select parse('456,46' as float using 'tr-TR') -- 456.46 hata verir
```

```
select parse('456.46' as float using 'en-US') -- 456,46 hata verir
```

```
select try_parse('45' as int)
```

```
select try_parse('fsdfs' as int)
```

```
select try_parse('456,46' as float using 'en-US')
```

```
Select KisiAdi,
```

```
CONCAT(
```

```
LEFT(REVERSE(UPPER(KisiAdi)),1),
```

```
RIGHT(REVERSE(LOWER(KisiAdi)),LEN(KisiAdi)-1)) from Kisi --Ters çevrilmiş halinin baş  
harfini büyük yapar
```

```
Select SQRT(81) -- karekökünü alır
```

```
Select SQUARE(7) -- karesini alır.
```

```
Select STR(564564.4546787,10,2) -- STR(sayı, toplam karakter, virgülden sonraki  
karakter)
```

```
SELECT RAND() -- 0-1 arasında rasgele sayı verir
```

```

SELECT RAND()*100 -- 0-100 arasında rasgele ondalıklı sayı verir
SELECT CONVERT(int,RAND()*100) -- 0-100 arasında rasgele tamsayı verir

SELECT GETDATE() -- günün tarih ve saatini verir

SELECT FORMAT(GETDATE(),'dd/MM/yyyy', 'en-US') -- tarihin gösterimini değiştir.
SELECT DATEADD(DAY, 3,GETDATE()) -- tarihe 3 gün ekler
SELECT DATEADD(MONTH, 2,GETDATE()) -- tarihe 2 ay ekler
SELECT DATEADD(YEAR, 5,GETDATE()) -- tarihe 5 yıl ekler

Select DATEDIFF(DAY, GETDATE(), '10.08.2023') as KalanGun -- iki tarih arasındaki
farkı verir

Select DAY(GETDATE()) as Gun,
MONTH(GETDATE()) as Ay,
YEAR(GETDATE()) as Yıl

SELECT ISDATE('05.05.1990')
SELECT ISDATE('05.1990')
SELECT ISDATE('may 1990')
SELECT ISDATE('mayıs 1990')
SELECT ISDATE('1990')
SELECT ISDATE('ewrewrew')

Select ISNUMERIC('fsdfs')
Select ISNUMERIC('55')
Select ISNUMERIC('5.5')
Select ISNUMERIC('5,5')
Select ISNUMERIC('05.05.1990')

Select ISNULL(KisiAdres,'Girilmemiş') from Kisi -- null değerlerin yerine girilmemiş
yazar

Select ROW_NUMBER() OVER(order by KisiAdi),KisiAdi,KisiSoyadi,KisiMaas from Kisi --
her satıra satır numarası verir
Select RANK() OVER(order by KisiAdi),KisiAdi,KisiSoyadi,KisiMaas from Kisi -- aynı
satırlara aynı satır numarası verir 1,2,3,3,5,6
Select DENSE_RANK() OVER(order by KisiAdi),KisiAdi,KisiSoyadi,KisiMaas from Kisi --
aynı satırlara aynı satır numarası verir 1,2,3,3,4,5

-- Kişi isminin 3 ,4,5 üncü karakteri (ilk karakteri büyük olsun) ile
-- soyadının 2 ,3 ve 4 karakterlerini (son karakteri büyük olsun)
-- yazdıran sorguyu yazınız. Ayhan CANSIZ için --> HaaN

Select KisiAdi,KisiSoyadi,
CONCAT(UPPER(LEFT(SUBSTRING(TRIM(KisiAdi),3,3),1)),
LOWER(RIGHT(SUBSTRING(TRIM(KisiAdi),3,3),LEN(SUBSTRING(TRIM(KisiAdi),3,3))-1)),
LOWER(LEFT(SUBSTRING(TRIM(KisiSoyadi),2,3),LEN(SUBSTRING(TRIM(KisiSoyadi),2,3))-1)),
UPPER(RIGHT(SUBSTRING(TRIM(KisiSoyadi),2,3),1))) as Sonuc

from Kisi

use PersonelDB;

select * from Kisi

----- UPDATE

```

```
Update Kisi Set KisiYoneticiciMi=1 where KisiMaas>=8000 -- Maaşı 8000'den büyük olanları yönetici yap.
```

```
Update Kisi Set KisiAdi=TRIM(KisiAdi), KisiSoyadi=TRIM(KisiSoyadi) -- adı ve soyadındaki boşlukları temizledik
```

```
--Maaşı 5000'den küçük olanlara 500 tl zam yapılacaktır.
```

```
Select KisiAdi, KisiSoyadi, (KisiMaas+500) as ZamliMaas from Kisi where KisiMaas<5000 -  
- tabloyu güncellemez
```

```
Select * from Kisi
```

```
Update Kisi Set KisiMaas = KisiMaas+500 where KisiMaas<5000
```

```
--Adresi Null olanlara girilmemiş yaz.
```

```
Select KisiAdi, KisiSoyadi, ISNULL(KisiAdres, 'Girilmemiş') from Kisi -- tabloyu güncellemez
```

```
Select * from Kisi
```

```
UPDATE Kisi SET KisiAdres = 'Girilmemiş' WHERE KisiAdres is null -- adresi null olanlara girilmemiş yaz
```

```
----- DELETE
```

```
Delete from Kisi Where KisiID=12
```

```
--Update Kisi Set KisiAdres = null where KisiID=10
```

```
Delete from Kisi Where KisiAdres is null
```

```
----- Nested query veya Subquery
```

```
-- Maaşı ortalama maaştan yüksek olan kişileri getir.
```

```
--AVG(KisiMaas) -- çalışmaz
```

```
Select AVG(KisiMaas) from Kisi -- ortalama maaş getirir.
```

```
Select KisiAdi, KisiSoyadi, KisiMaas from Kisi where KisiMaas>(Select AVG(KisiMaas)  
from Kisi)
```

```
Select KisiAdi, KisiSoyadi, (Select AVG(KisiMaas) from Kisi) from Kisi
```

```
Update Kisi Set KisiAdi = TRIM(KisiAdi)
```

```
Select * from Kisi
```

```
--KisiAdi, KisiSoyadi ve kişinin birim adını nested query ile getiriniz.
```

```
Select KisiAdi, KisiSoyadi, BirimAdi from Kisi, Birim where Kisi.KisiBirimID =  
Birim.BirimID
```

```
Select KisiAdi, KisiSoyadi, (Select BirimAdi from Birim where BirimID = KisiBirimID)  
as BirimAdi from Kisi
```

```
--2 numaralı şubedeki birimlerde çalışan kişileri getir. nested query ile
```

```
--Select * from Kisi Where KisiBirimID in (2, 3, 5)
```

```
Select * from Kisi Where KisiBirimID in (Select BirimID from Birim where BirimSube=2)
```

```
use EserSatisDB;
```

```
--Satışı yapılmayan eserin adını ve sanatçısının adını ve soyadını getiren sql sorgu
```

```
Select EserID, EserAdi, SanatciAdi, SanatciSoyadi  
from Eser, Sanatci
```

```
Where EserSanatciID = SanatciID and  
EserID not in (Select SatisEserID from Satis)
```

```
--Eser fiyatı ortalama eser fiyatından yüksek olan eserleri getir.
```

```
--(Select AVG(Fiyat) from Eser)
```

```
Select * from Eser Where Fiyat>(Select AVG(Fiyat) from Eser)
```

--Fiyatı Ortalama fiyattan yüksek olan eserleri satın alan müşterilerin adını soyadını getir.

```
Select (Select MusteriAdi from Musteri where SatisMusteriID=MusteriID),  
(Select MusteriSoyadi from Musteri where SatisMusteriID=MusteriID)  
from Satis where  
SatisEserID in (Select EserID from Eser Where Fiyat>(Select AVG(Fiyat) from Eser))
```