

AÇIKLAMA: Karayolunun yanında, gerektiğinde deniz ve havayolu ile de taşıma yapabilecek şekilde uluslararası kargo ve yük taşımacılığı yapabilen bir şirket için bir veritabanı tasarlanacaktır. Firma müşterilerin tedarikçilerden verdiği farklı kategorideki ürünleri belli bir hizmet bedeliyle en uygun yolla taşımaktadır.

SORU 1: Verilen açıklamalara göre veri tabanını, veritabanının tablolarını, tablodaki alanları, her bir alanın veri tipini, birincil ve ikincil anahtarlarını veritabanı kurallarına göre **tasarlayınız** ve verilen açıklamalara göre bu veritabanını **SQL kodları** ile oluşturunuz ve veritabanını kodlarken kısıtlayıcılardan her birisini en az bir kere kullanınız. Ayrıca veritabanının varlık-ilişki modelini çıkarınız.

```
create database lojistik1
use lojistik1
create table tedarikçi(
tedarikci_id int not null identity(1,1) primary key,
tedarikci_adi varchar(50))

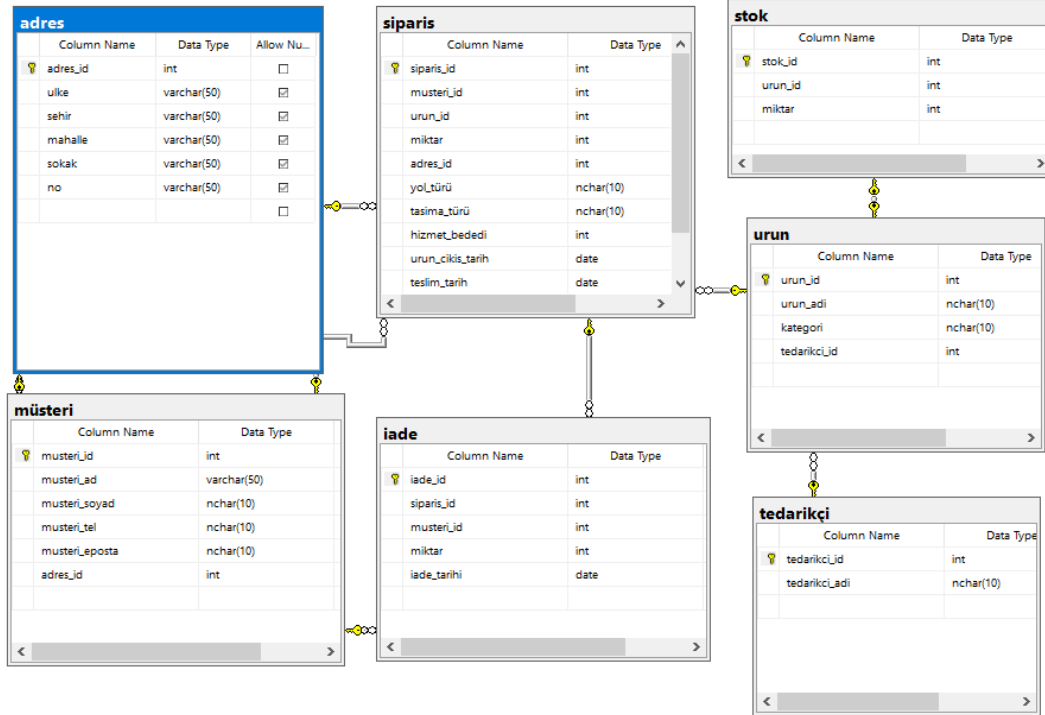
create table urun(
urun_id int not null identity(1,1) primary key,
urun_adi varchar(50),
kategori varchar(50),
tedarikci_id int foreign key references tedarikçi(tedarikci_id)
)

create table stok (
stok_id int not null identity(1,1) primary key,
urun_id int foreign key references urun(urun_id),
miktar int)

create table adres(
adres_id int not null identity(1,1) primary key,
ulke varchar(50),
sehir varchar(50),
mahalle varchar(50),
sokak varchar(50),
noo varchar(50)
)
create table müşteri(
musteri_id int not null identity(1,1) primary key,
musteri_ad varchar(50),
musteri_soyad varchar(50),
musteri_tel varchar(50),
musteri_eposta varchar(50),
adres_id int foreign key references adres(adres_id)
)
create table siparis (
siparis_id int not null identity(1,1) primary key,
musteri_id int foreign key references müşteri(musteri_id),
urun_id int foreign key references urun(urun_id),
miktar int,
adres_id int foreign key references adres(adres_id),
yol_türü varchar(50),
CONSTRAINT check_yol
check (yol_türü IN ('deniz', 'hava', 'kara')),
tasima_türü varchar(50),
CONSTRAINT check_tasima
check (tasima_türü IN ('kargo', 'yük')),
hizmet_bededı int,
urun_cikis_tarih date,
teslim_tarih date
```

)

```
create table iade (  
iade_id int NOT NULL UNIQUE,  
siparis_id int foreign key references siparis(siparis_id),  
musteri_id int foreign key references müsteri(musteri_id),  
miktar int,  
iade_tarihi date  
)
```



SORU 2:Bugüne kadar en fazla sipariş verilen tedarikçi firma için, bu tedarikçiden en fazla sipariş veren müşterinin ad, soyad, ülke, toplamda taşımaya ne kadar ücret ödediği ve kaç tane hangi kategoriden sipariş verdiğinin bilgisini listeleyecek **SQL sorgusunu** yazınız.

```
select
müşteri.musteri_ad,müşteri.musteri_soyad,adres.ulke,hizmet_beded,hizmet_bede,kategori,count(kateg
ori) as kategorisayisi
from
siparis inner join müşteri on müşteri.musteri_id=siparis.musteri_id
inner join urun on siparis.siparis_id=urun.urun_id
inner join tedarikçi on tedarikçi.tedarikci_id=urun.tedarikci_id
inner join adres on müşteri.adres_id=adres.adres_id
where tedarikçi.tedarikci_id=
(
select tedarikçi.tedarikci_id
from
siparis inner join urun on siparis.siparis_id=urun.urun_id
inner join tedarikçi on tedarikçi.tedarikci_id=urun.tedarikci_id
group by tedarikçi.tedarikci_id
having COUNT(tedarikçi.tedarikci_id)>=
(
select top 1 COUNT(*) as a
from
siparis inner join urun on siparis.siparis_id=urun.urun_id
inner join tedarikçi on tedarikçi.tedarikci_id=urun.tedarikci_id
group by tedarikçi.tedarikci_id
order by a desc
)
)
and
siparis.musteri_id=
(
select siparis.musteri_id
from
siparis inner join urun on siparis.siparis_id=urun.urun_id
inner join tedarikçi on tedarikçi.tedarikci_id=urun.tedarikci_id
group by siparis.musteri_id
having COUNT(siparis.musteri_id)>=
(
select top 1 COUNT(*) as a
from
siparis inner join urun on siparis.siparis_id=urun.urun_id
inner join tedarikçi on tedarikçi.tedarikci_id=urun.tedarikci_id
group by siparis.musteri_id
order by a desc
)
)
)
group by urun.kategori,
müşteri.musteri_ad,müşteri.musteri_soyad,adres.ulke,hizmet_bede
```

	musteri_ad	musteri_soyad	ulke	hizmet_bede	kategori	kategorisayisi
1	burcu	kabarc	ispanya	100	çikolata	1
2	burcu	kabarc	ispanya	40	peynir	1

SORU 3:Kendisine gelen tedarikçi ismine göre, bu tedarikçiden hiçbir sipariş vermeyen ve fakat başka tedarikçilerden sipariş veren müşterilerin ve ilgili siparişteki bilgileri ekrana listeleyecek bir **stored procedure** yazınız ve kullanınız.

```
create procedure siparis_bul(@tedarik varchar(10))
as
select
musteri.musteri_ad,musteri.musteri_soyad,siparis.miktar,siparis.yol_turu,siparis.tasima_turu
from musterisiparisurun,tedarikci
where tedarikci.tedarikci_id=urun.tedarikci_id and urun.urun_id=siparis.urun_id and
musteri.musteri_id=siparis.musteri_id and tedarikci_adi not in
(
select tedarikci_adi
from tedarikci
where tedarikci_adi=@tedarik
)
go
```

execute siparis_bul 'aykan'

	musteri_ad	musteri_soyad	miktar	yol_turu	tasima_turu
1	buket	pehlivan	500	hava	kargo

SORU 4:Tüm taşıma işlemleri içerisinde bulunan ortalama ürün sayısından daha fazla ürün siparişi veren müşterilerden hangilerinin en az sayıda sipariş verdiği ürün kategorisini ve sipariş ettiği ürün sayısını listeleyecek bir **fonksiyon** ve **kullanınız**.

```
CREATE FUNCTION urun_bilgi1()
RETURNS TABLE
AS
RETURN
(
select urun.kategori,siparis.miktar
from musterisiparisurun
where musterisiparis.musteri_id=siparis.musteri_id and siparis.urun_id=urun.urun_id and
siparis.miktar=
(
select Min(siparis.miktar)
from musterisiparisurun
where musterisiparis.musteri_id=siparis.musteri_id and
siparis.urun_id=urun.urun_id and siparis.miktar>
(
select SUM(siparis.miktar)/COUNT(siparis.urun_id)
from musterisiparisurun
where musterisiparis.musteri_id=siparis.musteri_id and siparis.urun_id=urun.urun_id
)
)
)
select * from dbo.fonksiyonumuzun_adi()
```

	kategori	miktar
1	çikolata	600

SORU 5:Sipariş edilen bir ürünün müşteri tarafından miktarı değiştirildiğinde bunun direkt olarak ürünün stok sayısına yansımaları sırasında önceki ve yeni bilgileri kullanıcıya gösterecek bir **output** yazınız.

```
declare @test table(  
    eski_miktar varchar(50),  
    yeni_miktar varchar(50))  
update siparis set miktar=500  
output deleted.miktar,inserted.miktar into @test  
  
select * from @test
```

	eski_miktar	yeni_miktar
1	700	500
2	600	500
3	500	500
4	500	500
5	500	500

SORU 6:Gönderilecek ürünün miktarında yapılan **güncelleme** sonrasında sipariş ile ilgili gerekli düzenlemeleri yapan bir **trigger** yazınız.

```
CREATE TRIGGER SiparisDuzenleme  
ON Siparis  
AFTER UPDATE  
AS  
BEGIN  
    IF UPDATE(miktar)  
        DECLARE @eski_miktar int  
        DECLARE @yeni_miktar int  
        DECLARE @fark int  
        DECLARE @urun_id int  
  
        SELECT @eski_miktar=miktar FROM deleted  
        SELECT @yeni_miktar=miktar FROM inserted  
        SELECT @urun_id=urun_id FROM inserted  
  
        SET @fark = @yeni_miktar - @eski_miktar  
        -- 200 bunu 300'e yükseltirse fark +100  
        -- 200 numu 100'e düşürürse fark -100  
  
        UPDATE stok SET miktar = miktar - @fark WHERE urun_id =  
@urun_id  
END
```

GO

```
UPDATE siparis SET miktar = 400 WHERE siparis_id = 1
```

****1** numaralı sipariş id'sine sahip sipariş için miktar 400 yapıldığında önceki miktara göre 200 artış olmuştur. Bu stoktan sipariş için ürün çıktığı anlamına gelir ve aynı miktarda stoktan azalma olur.

104 %

	siparis_id	musteri_id	urun_id	miktar	adn
1	1	1	1	200	1
2	2	2	2	500	1
3	3	2	3	500	1
4	4	3	3	500	1
5	5	2	2	500	1

	siparis_id	musteri_id	urun_id	miktar	adn
1	1	1	1	400	1
2	2	2	2	500	1
3	3	2	3	500	1
4	4	3	3	500	1
5	5	2	2	500	1

	stok_id	urun_id	miktar
1	1	1	500
2	2	2	NULL
3	3	3	400

	stok_id	urun_id	miktar
1	1	1	300
2	2	2	NULL
3	3	3	400

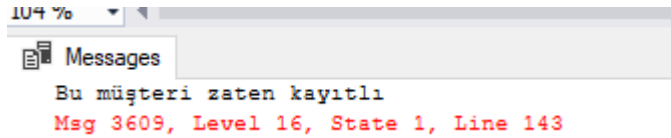


SORU 7:Müşteri web sitesine kayıt olurken eğer müşteri daha önceden kayıtlıysa, kaydının bulunduğu uyarısını veren, şayet ilk defa geliyorsa bu müşterinin kayıt işlemini yapan bir **trigger** yazınız.

```
create trigger k_kontrol
on müşteri
for insert
as
declare @musteri_ad varchar(50)
declare @musteri_soyad varchar(50)

select @musteri_ad=musteri_ad,@musteri_soyad=musteri_soyad from inserted
if exists(select * from müşteri where muster_i_ad=@musteri_ad and
musteri_soyad=@musteri_soyad)
begin
print 'Bu müşteri zaten kayıtlı'
rollback
end

INSERT INTO müşteri(musteri_id, muster_i_ad, muster_i_soyad)
VALUES (4, 'Buket', 'Pehlivan')
```



SORU 8:Sipariş ettiği ürünlerden, en fazla geri iade eden ilk üç müşterinin ID, ad, soyad ve kaç tane sipariş verdiği ve kaç tanesini geri iade ettiğini geriye döndürecek **bir fonksiyon** yazınız.

```
create function musteri_liste()  
RETURNS TABLE  
AS  
RETURN  
(  
select top 3 musteri.musteri_ad  
,musteri.musteri_soyad,musteri.musteri_id,count(iade.iade_id) as  
sayi,COUNT(siparis.urun_id) as siparissayi  
from musteri,siparis,iade where musteri.musteri_id=siparis.siparis_id  
and siparis.siparis_id=iade.siparis_id and musteri.musteri_id=iade.musteri_id  
group by musteri.musteri_ad,musteri.musteri_soyad,musteri.musteri_id order by sayi  
desc  
)  
select * from dbo.musteri_liste()
```

	musteri_ad	musteri_soyad	musteri_id	sayi	siparissayi
1	burcu	kabarc	2	1	1
2	buket	pehlivan	1	1	1

SORU 9:Bulunduğu ülke dışından sipariş veren müşterilerden her birisinin kargosunun/yükünün hangi yolla taşındığını, ne kadar ücret ödendiğini, teslimatın kaç gün sürdüğünü ve siparişin içeriğini listeleyecek **bir stored procedure** yazınız.

```
create procedure siparis_bilgi(@deneme varchar(10))  
as  
select distinct musteri_ad , siparis.yol_turu,siparis.hizmet_bededi  
,DATEDIFF(DAY,teslim_tarih,urun_cikis_tarih)  
as "gün sayısı" from musteri,adres,siparis  
where adres.adres_id=musteri.adres_id and  
musteri_ad not in  
(select musteri_ad from musteri,adres,siparis  
where adres.adres_id=musteri.adres_id  
and  
ulke= (select ulke from adres where adres_id in(select adres_id from siparis)))  
go  
exec siparis_bilgi'deneme'
```

	musteri_ad	yol_türü	hizmet_beded	gün sayısı
1	burcu	deniz	100	5
2	burcu	hava	200	3
3	burcu	hava	500	3
4	burcu	kara	40	1
5	burcu	kara	40	2
6	burcu	kara	400	1

SORU 10:Bugüne kadar 10’dan fazla sipariş veren her bir müşterinin en fazla sipariş verdiği ürün kategorilerinin hangilerinin denizyolu ile taşındığının bilgilerini listeleyecek **bir fonksiyon yazınız.**

```
create function bilgi1()
RETURNS TABLE
AS
RETURN
(
select kategori from (select kategori as kategori,
yol_türü,musteri_ad,müşteri.musteri_id
from siparis inner join müşteri on müşteri.musteri_id=siparis.siparis_id inner join
urun
on urun.urun_id=siparis.urun_id where (select COUNT(musteri_id) from siparis )>10
and yol_türü='deniz')y
)

select * from dbo.bilgi()
```

	kategori
1	çikolata