**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 v**erilen 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 müsteri(

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 müsteri(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\_bededi int,

urun\_cikis\_tarih date,

teslim\_tarih date

)

create table iade (

iade\_id int NOT NULL UNIQUE,

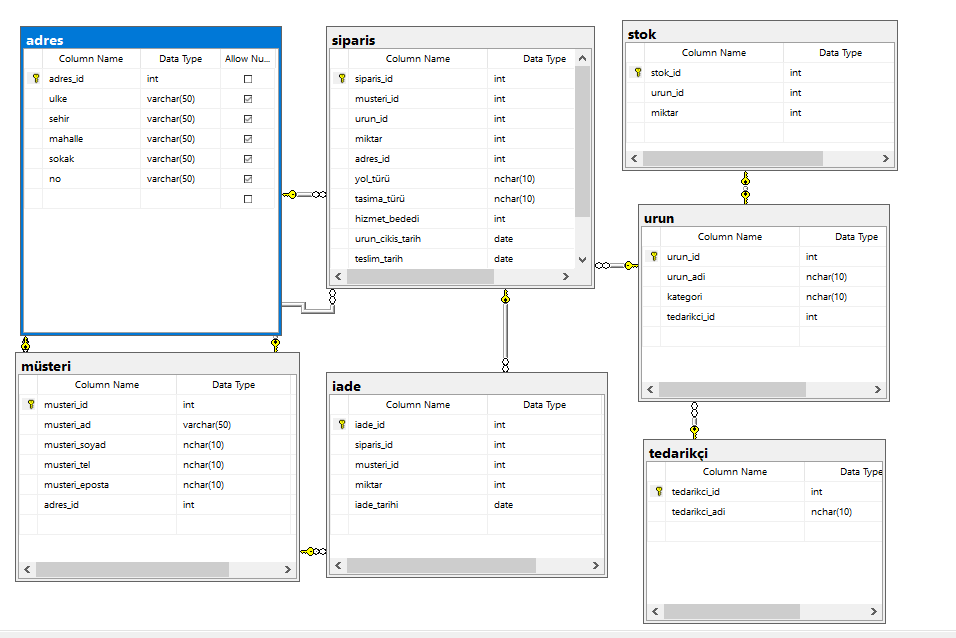
siparis\_id int foreign key references siparis(siparis\_id),

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 müsteri.musteri\_ad,müsteri.musteri\_soyad,adres.ulke,hizmet\_bededi,kategori,count(kategori) as kategorisayisi

from

siparis inner join müsteri on müsteri.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 müsteri.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

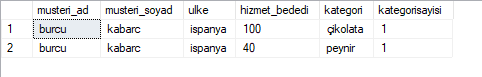
group by siparis.musteri\_id

order by a desc

)

)

group by urun.kategori, müsteri.musteri\_ad,müsteri.musteri\_soyad,adres.ulke,hizmet\_bededi



**SORU 3:**Kendisine gelentedarikçi 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 müsteri.musteri\_ad,müsteri.musteri\_soyad,siparis.miktar,siparis.yol\_türü,siparis.tasima\_türü

from müsteri,siparis,urun,tedarikçi

where tedarikçi.tedarikci\_id=urun.tedarikci\_id and urun.urun\_id=siparis.urun\_id and

müsteri.musteri\_id=siparis.musteri\_id and tedarikci\_adi not in

(

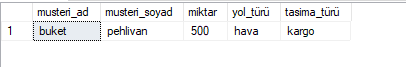
select tedarikci\_adi

from tedarikçi

where tedarikci\_adi=@tedarik

)

go

 execute siparis\_bul 'aykan'

**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 müsteri,siparis,urun

where müsteri.musteri\_id=siparis.musteri\_id and siparis.urun\_id=urun.urun\_id and

siparis.miktar=

(

select Min(siparis.miktar)

from müsteri,siparis,urun

where müsteri.musteri\_id=siparis.musteri\_id and

siparis.urun\_id=urun.urun\_id and siparis.miktar>

(

select SUM(siparis.miktar)/COUNT(siparis.urun\_id)

from müsteri,siparis,urun

where müsteri.musteri\_id=siparis.musteri\_id and siparis.urun\_id=urun.urun\_id

)

)

)

select \* from dbo.fonksiyonumuzun\_adı()



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

declare @test table(

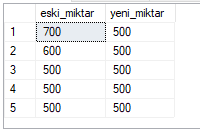
eski\_miktar varchar(50),

yeni\_miktar varchar(50))

update siparis set miktar=500

output deleted.miktar,inserted.miktar into @test

select \* from @test



**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

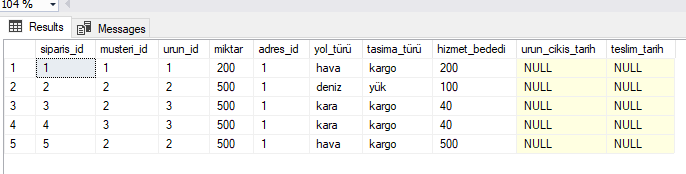
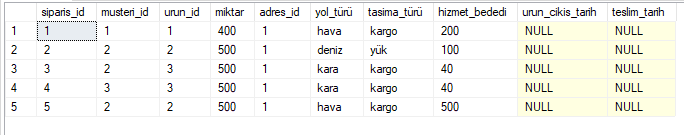
UPDATE stok SET miktar = miktar - @fark WHERE urun\_id = @urun\_id

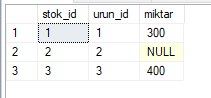
END

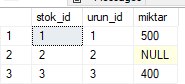
GO

UPDATE siparis SET miktar = 400 WHERE siparis\_id = 1

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







**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 müsteri

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 müsteri where musteri\_ad=@musteri\_ad and musteri\_soyad=@musteri\_soyad)

begin

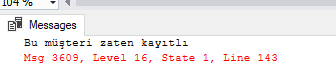
print 'Bu müşteri zaten kayıtlı'

rollback

end

INSERT INTO müsteri(musteri\_id, musteri\_ad, musteri\_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 müsteri.musteri\_ad ,müsteri.musteri\_soyad,müsteri.musteri\_id,count(iade.iade\_id) as

sayi,COUNT(siparis.urun\_id) as siparissayi

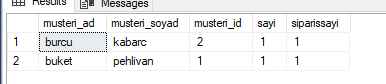
from müsteri,siparis,iade where müsteri.musteri\_id=siparis.siparis\_id

and siparis.siparis\_id=iade.siparis\_id and müsteri.musteri\_id=iade.musteri\_id

group by müsteri.musteri\_ad,müsteri.musteri\_soyad,müsteri.musteri\_id order by sayi desc

)

select \* from dbo.musteri\_liste()



**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\_türü,siparis.hizmet\_bededi ,DATEDIFF(DAY,teslim\_tarih,urun\_cikis\_tarih)

as "gün sayısı" from müsteri,adres,siparis

where adres.adres\_id=müsteri.adres\_id and

musteri\_ad not in

(select musteri\_ad from müsteri,adres,siparis

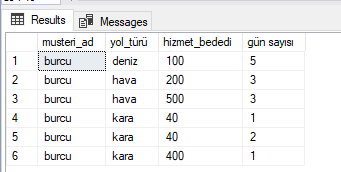
where adres.adres\_id=müsteri.adres\_id

and

ulke= (select ulke from adres where adres\_id in(select adres\_id from siparis)))

go

exec siparis\_bilgi'deneme'



**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,müsteri.musteri\_id

from siparis inner join müsteri on müsteri.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()

