**Uygulama#3\_Cevaplar**

**(5p) Soru 1 : Tüm müşterileri listeleyen kodu yazınız.**

SELECT \* FROM BikeStores.dbo.customers

**(10p) Soru 2 : Personellerin adı, soyadı ve iletişim bilgilerini listeleyen kodu yazınız.**

SELECT s.first\_name,s.last\_name,s.phone,s.email FROM BikeStores.dbo.staffs as s

**(10p) Soru 3: Sipariş tarihi 2017-06-24 öncesi olan tüm siparişleri listeleyen kodu yazınız.**

SELECT \* from BikeStores.dbo.orders WHERE order\_date<'2017-06-24'

**(10p) Soru 4: Stok adeti-miktarı 7 den büyük 16 dan küçük olan stokları listeleyen kodu yazınız.**

SELECT count(\*) from BikeStores.dbo.stocks as s WHERE s.quantity>7 and s.quantity<16

SELECT \* from BikeStores.dbo.stocks as s WHERE s.quantity>7 and s.quantity<16

**(15p) Soru 5: Kategori adı "Electric Bikes" olan ürünlerin ürün adı, marka ID, kategori ID ve model bilgilerini listeleyen kodu yazınız.**

SELECT p.product\_name,p.brand\_id,c.category\_id,p.model\_year from BikeStores.dbo.categories as c, BikeStores.dbo.products as p

WHERE p.category\_id=c.category\_id and c.category\_name='Electric Bikes'

**(15p) Soru 6: Adı Jone olan müşteriye ait tüm siparişler için adı, soyadı, müşteri ID ve sipariş tarihini listeleyen kodu yazınız.**

SELECT c.first\_name,c.last\_name,c.customer\_id, o.order\_date FROM BikeStores.dbo.customers as c, BikeStores.dbo.orders as o

WHERE o.customer\_id=c.customer\_id and c.first\_name='Jone'

**(15p) Soru 7: Adı "Baldwin Bikes" olan mağazada çalışmayan personeller için mağaza adı, şehir, personel adı, personel soyadı, mağaza ID bilgilerini listeleyen kodu yazınız.**

SELECT s.store\_name,s.city,p.first\_name,p.last\_name,s.store\_id FROM BikeStores.dbo.stores as s, BikeStores.dbo.staffs p

WHERE s.store\_id=p.store\_id and s.store\_name!='Baldwin Bikes'

**(20p) Soru 8: Stok miktarı 19'dan az ve liste fiyatı 2500'den fazla olan ürünler için mağaza ID, ürün ID, ürün adı, adeti-miktarı ve liste fiyatını listeleyen kodu yazınız.**

SELECT st.store\_id, p.product\_id, p.product\_name, st.quantity, p.list\_price FROM BikeStores.dbo.stocks as st, BikeStores.dbo.products as p

WHERE st.product\_id=p.product\_id and st.quantity<19 and p.list\_price>2500

**Uygulama#4\_Cevaplar**

**(6p) Soru 1 : Müşterilere(customers) ait first\_name bilgisini tekrarlamayacak-yinelemeyecek şekilde listeleyen kodu yazınız. (sadece ilk 20 kaydı listeleyiniz)**

SELECT DISTINCT TOP(20) first\_name FROM BikeStores.dbo.customers

**(8p) Soru 2 : Siparişi(Orders) bulunan müşterilerin customer\_id, first\_name ve last\_name bilgilerini**

**customer\_id bilgisi tekrarlamayacak-yinelemeyecek şekilde listeleyen kodu yazınız.**

SELECT c.customer\_id,c.first\_name,c.last\_name FROM BikeStores.dbo.customers as c

WHERE c.customer\_id IN (SELECT DISTINCT o.customer\_id FROM BikeStores.dbo.orders as o)

**(8p) Soru 3: Manager\_ID 'si NULL olmayan personellerin(staffs) tüm bilgilerini store\_id'ye göre büyükten küçüğe doğru sıralı listeleyen kodu yazınız.**

SELECT \* FROM BikeStores.dbo.staffs WHERE manager\_id IS NOT NULL ORDER BY store\_id DESC

**(8p) Soru 4: Stok miktarı(quantity) 13 veya 24 olan ürün sayısını 'count\_of\_selected\_quantity' kolon adını kullanarak listeleyen kodu yazınız.**

SELECT COUNT(\*) AS count\_of\_selected\_quantity FROM BikeStores.dbo.stocks WHERE quantity=13 OR quantity=24

**(10p) Soru 5: Baldwin Bikes mağazasında(store) çalışan personel(staffs) sayısını 'count\_of\_baldwin\_staffs' kolon adını kullanarak listeleyen kodu yazınız.**

SELECT COUNT(\*) as count\_of\_baldwin\_staffs FROM BikeStores.dbo.staffs as s,BikeStores.dbo.stores as m WHERE s.store\_id=m.store\_id and m.store\_name='Baldwin Bikes'

**(10p) Soru 6: Ürünlerin(products) en küçük, en büyük ve ortalama liste fiyatını sırasıyla en\_kucuk, en\_buyuk ve ortalama kolon adlarıyla listeleyen kodu yazınız.**

SELECT MIN(p.list\_price) as en\_kucuk,MAX(p.list\_price) as en\_buyuk,AVG(p.list\_price) as ortalama FROM BikeStores.dbo.products as p

**(15p) Soru 7: Markası(brand) 'Trek' olan ürünlerin(product) toplam liste fiyatını(list\_price) sum\_of\_list\_prices kolon adıyla listeleyen kodu yazınız.**

SELECT SUM(p.list\_price) as sum\_of\_list\_prices

FROM BikeStores.dbo.brands as b, BikeStores.dbo.products as p WHERE b.brand\_id=p.brand\_id AND b.brand\_name='Trek'

**(15p) Soru 8: Rowlett Bikes mağazasında(store) satılan ve stok miktarı(quantity) 15'ten fazla olan ürün sayısını listeleyen kodu yazınız.**

SELECT count(\*) FROM BikeStores.dbo.stores as s,BikeStores.dbo.stocks as m

WHERE s.store\_id=m.store\_id and s.store\_name='Rowlett Bikes' and m.quantity>15

**(20p) Soru 9: Sipariş tarihi(order\_date) 2017-07-22'den sonra olan siparişlerin(order)**

**personel adı(staff -> first\_name), mağaza adı(store\_name) ve sipariş tarihi(order\_date) bilgilerini**

**store\_name'e göre küçükten büyüğe sıralı şekilde listeleyen kodu yazınız.**

SELECT s.first\_name,m.store\_name,o.order\_date

FROM BikeStores.dbo.orders as o,BikeStores.dbo.staffs as s,BikeStores.dbo.stores as m

WHERE order\_date>'2017-07-22' AND o.store\_id=m.store\_id AND o.staff\_id=s.staff\_id ORDER BY m.store\_name ASC

**Uygulama#5\_Cevaplar**

**(10p) Soru 1 : Rowlett Bikes mağazasındaki(stores) stoklarda(stocks) bulunan ürünlerden(products) isminin ikinci harfi 'a' olan ürünleri listeleyiniz.**

**\* Mağaza adı, stok miktarı, ürün adı ve fiyatı sayısı çıktı olarak verilmelidir!**

SELECT store\_name, quantity, product\_name, list\_price FROM stores, stocks, products

WHERE stores.store\_id = stocks.store\_id AND stocks.product\_id = products.product\_id AND product\_name LIKE '\_a%' AND store\_name = 'Rowlett Bikes'

**(10p) Soru 2 : Hangi mağazadan(stores) toplamda kaç müşteri(customers) sipariş(orders) vermiş sayılarını listeleyiniz.**

**\* Mağaza adı ve toplam müşteri sipariş sayısı çıktı olarak verilmelidir!**

SELECT store\_name, COUNT(\*) FROM customers, orders, stores

WHERE customers.customer\_id = orders.customer\_id AND orders.store\_id = stores.store\_id

GROUP BY store\_name

**(15p) Soru 3 : Mağazalardaki(stores) yer alan ürünlerden(products), stok(stocks) miktarı 15'ten büyük olanların sayısını, maksimum fiyatını, minimum fiyatını listeleyiniz.**

**\* Mağaza adı, ürün sayısı, maksimum fiyatı, minimum fiyatı çıktı olarak verilmelidir.**

SELECT store\_name, COUNT(\*), MAX(list\_price), MIN(list\_price) FROM stores, stocks, products

WHERE stores.store\_id = stocks.store\_id AND stocks.product\_id = products.product\_id AND quantity > 15 GROUP BY store\_name

**(15p) Soru 4 : Yaşadığı bölgedeki mağazadan(stores), sipariş(orders) vermiş olan müşterilerin(customers) bilgilerini listeleyiniz.**

**\*\* IN kullanarak iç içe sorgu gerçekleştirilmelidir!**

SELECT \* FROM customers as C1 WHERE C1.first\_name IN (SELECT C2.first\_name FROM customers as C2, orders, stores WHERE C2.customer\_id = orders.customer\_id AND orders.store\_id = stores.store\_id AND C2.zip\_code = stores.zip\_code)

**(10p) Soru 5 : Stok(stocks) miktarı 25'ten büyük olan ürünleri(products) listeleyiniz.**

**\*\* ANY kullanarak iç içe sorgu gerçekleştirilmelidir!**

**\* Ürün adı ve stok miktarı çıktı olarak verilmelidir.**

SELECT product\_name FROM products WHERE products.product\_id = ANY (SELECT stocks.product\_id FROM stocks WHERE quantity > 25)

**(10p) Soru 6 : Stok miktarı(stocks) 25 olan tüm ürünleri(products) listeleyiniz.**

**\*\* ALL kullanarak iç içe sorgu gerçekleştirilmelidir!**

**\* Ürün adı çıktı olarak verilmelidir.**

SELECT product\_name FROM products WHERE products.product\_id = ALL (SELECT stocks.product\_id FROM stocks WHERE quantity = 25)

**(15p) Soru 7 : Stok(stocks) miktarı 20'den küçük olan ürünlerin(products), marka(brand) isimlerini listeleyiniz.**

**\*\* EXISTS kullanarak iç içe sorgu gerçekleştirilmelidir!**

**\* Marka adı çıktı olarak verilmelidir.**

SELECT brand\_name FROM brands WHERE EXISTS (SELECT product\_name FROM products, stocks WHERE products.brand\_id = brands.brand\_id AND products.product\_id =stocks.product\_id AND quantity < 20);

**(15p) Soru 8 : Yaşadığı bölgedeki mağazadan(stores), sipariş(orders) vermemiş olan müşteriler(customers) hariç tüm müşterileri listeleyiniz.**

**\*\* EXCEPT kullanarak iç içe sorgu gerçekleştirilmelidir!**

**\* Müşteri adı çıktı olarak verilmelidir.**

SELECT first\_name FROM customers as C1

EXCEPT

SELECT first\_name FROM customers AS c2, orders, stores WHERE c2.customer\_id = orders.customer\_id AND stores.store\_id = orders.store\_id AND C2.zip\_code != stores.zip\_code

**Uygulama#6\_Cevaplar**

**(10p) Soru 1 : Her bir siparişin kaç parça(item) içerdiğini aşağıdaki şekilde listeleyen kodu yazınız.**

**- sipariş ID ve adet bilgisini( as countOfItems) içermelidir**

**- sırasıyla adet bilgisi ve sipariş ID 'ye göre büyükten küçüğe doğru olmalıdır**

use BikeStores

SELECT oi.order\_id,COUNT(oi.item\_id) as countOfItems

FROM order\_items as oi

GROUP BY oi.order\_id

ORDER BY countOfItems,oi.order\_id DESC

**(10p) Soru 2: En çok satılan(order\_items) ilt 5 ürünü(product) aşağıdaki şekilde listeleyen kodu yazınız.**

**- ürün ID ve adet sayısını(as countOfOrder) içermelidir.**

**- adet sayısına göre büyükten küçüğe doğru sıralanmalıdır**

USE BikeStores

SELECT TOP(5) o.product\_id , count(\*) as countOfOrder

FROM order\_items as o

GROUP BY o.product\_id

ORDER BY countOfOrder DESC

**(10p) Soru 3 : Toplam indirim(discount) miktarı 2'den fazla olan ürünleri(product) aşağıdaki şekilde listeleyen kodu yazınız.**

**- ürün ID ve toplam indirim bilgisini( as sumOfDiscount) içermelidir**

**- ürün ID'ye göre küçükten büyüğe doğru olmalıdır**

use BikeStores

SELECT oi.product\_id,SUM(oi.discount) as sumOfDiscount

FROM order\_items as oi

GROUP BY oi.product\_id

HAVING SUM(oi.discount)>2

ORDER BY oi.product\_id

**(15p) Soru 4: Aldığı sipariş(orders) sayısı 100'den fazla olan personelleri(staff) aşağıdaki şekilde listeleyen kodu yazınız.**

**- personel ID ve sipariş sayısı(as countOfOrder)**

**- sattığı toplam ürün sayısına göre büyükten küçüğe doğru sıralanmalıdır**

**- uygun JOIN yöntem-leri kullanılmalıdır !!**

use BikeStores

SELECT s.staff\_id,count(\*) as countOfOrder

FROM staffs as s

INNER JOIN orders as o

ON o.staff\_id=s.staff\_id

GROUP BY s.staff\_id

HAVING count(\*)>100

ORDER BY countOfOrder DESC

**(15p) Soru 5: Her bir mağaza için yapılan toplam indirimi(discount) aşağıdaki şekilde listeleyen kodu yazınız.**

**- mağaza ID ve toplam indirim bilgisini( as sumOfDiscount) içermelidir**

**- uygun JOIN yöntem-leri kullanılmalıdır !!**

USE BikeStores

SELECT s.store\_id,SUM(oi.discount) as sumOfDiscount

FROM stores as s

INNER JOIN orders as o

ON s.store\_id=o.store\_id

INNER JOIN order\_items as oi

ON o.order\_id=oi.order\_id

GROUP BY s.store\_id

**(20p) Soru 6: Her bir mağaza ve sipariş(order) için toplam parça(item) sayısını aşağıdaki şekilde listeleyen kodu yazınız.**

**- mağaza ID, sipariş ID ve toplam ürün sayısını( as countOfItems) içermelidir**

**- sırasıyla mağaza ID, sipariş ID ve toplam ürün sayısına göre küçükten büyüğe doğru sıralanmalıdır**

**- uygun JOIN yöntem-leri kullanılmalıdır !!**

USE BikeStores

SELECT s.store\_id,o.order\_id,COUNT(oi.item\_id) as countOfItems

FROM stores as s

INNER JOIN orders as o ON s.store\_id=o.store\_id

INNER JOIN order\_items as oi ON o.order\_id=oi.order\_id

GROUP BY s.store\_id,o.order\_id

ORDER BY s.store\_id,o.order\_id,countOfItems

**(20p) Soru 7: Toplam stok sayısı(quantity) 25'ten fazla olan ürünleri(product) aşağıdaki şekilde listeleyen kodu yazınız.**

**- ürün ID,ürün adı, marka ID, model yılı ve stok sayısı(as sumOfQuantity)**

**- ürün adına göre küçükten büyüğe doğru sıralı olmalısır**

**- uygun JOIN yöntem-leri ve SELECT yapıları kullanılmalıdır !!**

use BikeStores

SELECT p.product\_id,p.product\_name,p.brand\_id,p.model\_year,t.sumOfQuantity

FROM products as p

INNER JOIN ( SELECT st.product\_id,SUM(st.quantity) as sumOfQuantity FROM stocks as st GROUP BY st.product\_id HAVING SUM(st.quantity)>25 ) as t

ON p.product\_id=t.product\_id

ORDER BY p.product\_name,p.model\_year

**BONUS Soru: Her müşteri için alışveriş yapılmayan markaları listeleyen kodu aşağıdaki şekilde listeleyen kodu yazınız.**

**- müşteri ID ve marka ID bilgilerini içermelidir.**

**- müşteri ID ve marka ID bilgilerine göre küçükten büyüğe doğru sıralanmalıdır.**

**- istediğiniz sorgu yapısını kullanabilirsiniz**

use BikeStores

SELECT c.customer\_id,b.brand\_id from brands as b,customers as c

EXCEPT

SELECT c.customer\_id as cID,p.brand\_id as bID from customers as c

INNER JOIN orders as o on o.customer\_id=c.customer\_id

INNER JOIN order\_items as oi on o.order\_id=oi.order\_id

INNER JOIN products as p on p.product\_id=oi.product\_id

GROUP by c.customer\_id,p.brand\_id

ORDER BY c.customer\_id,b.brand\_id

**Uygulama#7\_Cevaplar**

**(10p) Soru 1 : Adı "b", "s" veya "p" ile başlamayan ürünleri(products) listeleyiniz.**

**\* Bütün sütünlar listelenebilir.**

SELECT \* FROM products WHERE product\_name LIKE '[!bsp]%'

**(10p) Soru 2 : İsimlerinin ilk harfi a-n arası son harfinin n-z arası ve soyisimlerinin ilk harfi n-z arası son harfinin a-n arazı olan müşterileri listeleyiniz.**

**\*\* Örneğin, İSİM: B......Y ve SOYİSİM: W......D**

**\* Bütün sütunlar listelenebilir.**

SELECT \* FROM customers WHERE customers.first\_name LIKE '[a-n]%[n-z]' AND customers.last\_name LIKE '[n-z]%[a-n]'

**(20p) Soru 3 : Yöneticileri(staffs.manager\_id) ve yöneticilerin yetkili olduğu mağazaları(stores) bulunuz.**

**\*\* Yönetici ID, Yönetici İsmi, Yönetici Soyismi, Mağaza İsmi ve Yöneticiye bağlı olan çalışan sayısı listelenmelidir.**

SELECT man.staff\_id,man.first\_name,man.last\_name, stores.store\_name, COUNT (emp.staff\_id) AS number\_of\_employees FROM ((staffs emp JOIN staffs man ON emp.manager\_id = man.staff\_id)

INNER JOIN stores ON man.store\_id = stores.store\_id) GROUP BY man.staff\_id, man.first\_name, man.last\_name, stores.store\_name

**(20p) Soru 4 : Müşterilerin(customers), sipariş(orders and order\_items) vermiş olduğu ürünlerin(products) fiyatı 1426'dan büyük olanların kategorilerini bulunuz.**

**\* Kategori adı ve sayıları listelenmelidir.**

SELECT categories.category\_name, COUNT(\*) AS Total FROM ((((customers

INNER JOIN orders ON customers.customer\_id = orders.customer\_id)

INNER JOIN order\_items ON orders.order\_id = order\_items.order\_id)

INNER JOIN products ON order\_items.product\_id = products.product\_id)

INNER JOIN categories ON products.category\_id = categories.category\_id) WHERE order\_items.list\_price > 1426 GROUP BY category\_name

**(20p) Soru 5 : Müşterilerin(customers), sipariş(orders and order\_items) vermiş olduğu ürünlerin(products) fiyatı 1426'dan büyük olanları ve model yılı 2018 olanların markalarını bulunuz.**

**\* Marka adı ve sayıları listelenmelidir.**

SELECT brands.brand\_name, COUNT(\*) AS Total FROM ((((customers

INNER JOIN orders ON customers.customer\_id = orders.customer\_id)

INNER JOIN order\_items ON orders.order\_id = order\_items.order\_id)

INNER JOIN products ON order\_items.product\_id = products.product\_id)

INNER JOIN brands ON products.brand\_id = brands.brand\_id) WHERE order\_items.list\_price > 1426 AND products.model\_year = 2018 GROUP BY brand\_name

**(20p) Soru 6 : Müşterilerin(customers), sipariş(orders and order\_items) vermiş olduğu ürünlerin(products) fiyatı ve siparişdeki tutarı aynı olanları listeleyiniz.**

**\* Mağaza adı, ürün sayısı, maksimum fiyatı, minimum fiyatı çıktı olarak verilmelidir.**

SELECT products.product\_name, COUNT(\*) AS Total FROM (((customers

INNER JOIN orders ON customers.customer\_id = orders.customer\_id)

INNER JOIN order\_items ON orders.order\_id = order\_items.order\_id)

INNER JOIN products ON order\_items.product\_id = products.product\_id) WHERE order\_items.list\_price = products.list\_price GROUP BY product\_name

**Uygulama#8\_Cevaplar**

**(5p) Soru 1 : "VtysLabDB" adıyla yeni bir veri tabanı oluşturunuz. Bundan sonraki aşamalarda bu tablo KULLANILACAKTIR.**

CREATE DATABASE VtysLabDB

**(10p) Soru 2 : Aşağıda verilen şekilde tabloyu oluşturunuz.**

**- Tablo adı : tblCustomer**

**- Kolonlar :**

**- customerID : Integer veri tipindedir. Primary KEY olarak kullanılacaktır. 100'den başlar ve otomatik olarak 1'er 1'er artış gösterir.**

**- first\_name : Varchar(255) veri tipindedir. NULL olmasına izin verilmez.**

**- last\_name : Varchar(255) veri tipindedir. DEFAULT değeri "NotSpecified" olarak kullanılaraktır.**

**- age : Integer veri tipindedir. NULL olmasına izin verilmez. 18'den büyük olmalıdır. Kullandığınız kısıta "CHC\_CustomerAge" adını veriniz !**

**use VtysLabDB**

CREATE TABLE tblCustomer

( customerID INT IDENTITY(100,1) PRIMARY KEY,

firs\_name VARCHAR(20) NOT NULL,

last\_name VARCHAR(20) DEFAULT 'NotSpecified',

age INT NOT NULL,

CONSTRAINT CHC\_CustomerAge CHECK (age>=18)

)

**SORU 2 İLE BAĞIMLIDIR. SORU 2 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.**

**(5p) Soru 3 : Aşağıda verilen kayıtları tblCustomer tablosuna ekleyiniz. Bu süreçte karşınıza çıkan hatalara uygun çözümler üreterek tabloya 3 kaydı da EKLEYİN.**

**Aldığınız hatayı, nedenini ve çözümünüzü MUTLAKA yorumda yazınız. Aksi halde puan verilmeyecektir.**

**kayıtlar : => 'Debra','Burks',24**

**=> 'Kasha','Todd',36**

**=> 'Robby','Sykes',12**

use VtysLabDB

INSERT INTO tblCustomer(firs\_name,last\_name,age)

VALUES

('Debra','Burks',24),

('Kasha','Todd',36),

('Robby','Sykes',22) /\* age kısıtı sağlamadığı için güncellendi. 18 den büyük eşit olması gerekiyor

**SORU 2 İLE BAĞIMLIDIR. SORU 2 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.**

**(5p) Soru 4 : Aşağıda verilen kaydı tblCustomer tablosuna ekleyiniz. Bu süreçte karşınıza çıkan hatalara uygun çözümler üreterek tabloya kaydı da EKLEYİN.**

**Aldığınız hatayı, nedenini ve çözümünüzü MUTLAKA yorumda yazınız. Aksi halde puan verilmeyecektir.**

**kayıt : => 'Melia',42**

use VtysLabDB

INSERT INTO tblCustomer(firs\_name,age)

VALUES

('Melia',42)

**(10p) Soru 5 : Aşağıda verilen şekilde tabloyu oluşturunuz.**

**- Tablo adı : tblProduct**

**- Kolonlar :**

**- productID : Integer veri tipindedir. Primary KEY olarak kullanılacaktır. 200'den başlar ve otomatik olarak 1'er 1'er artış gösterir.**

**- product\_name : Varchar(255) veri tipindedir. NULL olmasına izin verilmez.**

**- price : Float veri tipindedir. NULL olmasına izin verilmez.**

use VtysLabDB

CREATE TABLE tblProduct

(

productID INT IDENTITY(200,1) PRIMARY KEY,

product\_name VARCHAR(255) NOT NULL,

price FLOAT NOT NULL

)

**/\* SORU 5 İLE BAĞIMLIDIR. SORU 5 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.**

**(5p) Soru 6 : Aşağıda verilen kayıtları tblProduct tablosuna ekleyiniz.**

**kayıtlar : => 'Surly Straggler',12.5**

**=> 'Electra Townie Original',10**

**=> 'Pure Cycles William',5**

**\*/**

use VtysLabDB

INSERT INTO tblProduct (product\_name,price)

VALUES

('Surly Straggler',12.5),

('Electra Townie Original',10),

('Pure Cycles William',5)

**SORU 5 İLE BAĞIMLIDIR. SORU 5 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.**

**(5p) Soru 7 : Aşağıda verilen şekilde tblProduct tablosunda güncelleme yapınız.**

**- "price" değerinin 0'dan büyük olmasını garanti edecek şekilde "CHC\_ProductPrice" adıyla bir CONSTRAINT 'i tabloya ekleyiniz.**

use VtysLabDB

ALTER TABLE tblProduct

ADD CONSTRAINT CHC\_ProductPrice CHECK (price>0)

**SORU 5 İLE BAĞIMLIDIR. SORU 5 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.**

**(5p) Soru 8 : Aşağıda verilen kaydı tblProduct tablosuna ekleyiniz. Bu süreçte karşınıza çıkan hatalara uygun çözümler üreterek tabloya kaydı da EKLEYİN.**

**Aldığınız hatayı, nedenini ve çözümünüzü MUTLAKA yorumda yazınız. Aksi halde puan verilmeyecektir.**

kayıt : => 'Pure Cycles William',0

use VtysLabDB

INSERT INTO tblProduct (product\_name,price)

VALUES

('Pure Cycles William',3) /\* price 0'dan büyük olmalıdır \*/

**(10p) Soru 9 : Aşağıda verilen şekilde tabloyu oluşturunuz.**

**- Tablo adı : tblOrder**

**- Kolonlar :**

**- orderID : Integer veri tipindedir. Primary KEY olarak kullanılacaktır. Otomatik olarak artış gösterir.**

**- productID : Integer veri tipindedir. "FK\_ProductOrder" adında tanımlı bir CONSTRAINT ile tblProduct(productID)'a FOREING KEY olarak bağlıdır.**

use VtysLabDB

CREATE TABLE tblOrder

( orderID INT IDENTITY PRIMARY KEY,

productID INT,

CONSTRAINT FK\_ProductOrder FOREIGN KEY (productID) REFERENCES tblProduct(productID)

)

**SORU 9 İLE BAĞIMLIDIR. SORU 9 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.**

**(10p) Soru 10 : Aşağıda verilen şekilde tblOrder tablosunda güncelleme yapınız. TEK BİR SORGU YAZINIZ !**

**- tabloya aşağıdaki özelliklerde "customerID" adında yeni bir kolon ekleyiniz.**

**- customerID : Integer veri tipindedir. "FK\_CustomertOrder" adında tanımlı bir CONSTRAINT ile tblCustomer(customerID)'a FOREING KEY olarak bağlıdır.**

use VtysLabDB

ALTER TABLE tblOrder

ADD customerID INT

CONSTRAINT FK\_CustomertOrder FOREIGN KEY (customerID) REFERENCES tblCustomer(customerID)

SORU 9 İLE BAĞIMLIDIR. SORU 9 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.

**(10p) Soru 11 : tblOrder tablosuna kendi belirlediğiniz 5 adet kayıt ekleyiniz. Bu süreçte karşınıza çıkan hatalara uygun çözümler üreterek tabloya en az 5 kaydı da EKLEYİN.**

**Aldığınız hatayı, nedenini ve çözümünüzü MUTLAKA yorumda yazınız. Aksi halde puan verilmeyecektir.**

use VtysLabDB

INSERT INTO tblOrder (customerID,productID)

VALUES

(106,202),

(106,201),

(106,200),

(103,202),

(103,201),

(104,200),

(104,202),

(105,200),

(105,201)

**(15p) Soru 12 : Aşağıda verilen şekilde bir VIEW tablosu oluşturunuz. Bu tabloda mevcut siparişlere ait müşteri ve ürün bilgileri bulunacaktır.**

**- Tablo adı : tblView**

**- Kolonlar :**

**- orderID :**

**- customerID :**

**- firs\_name :**

**- last\_name :**

**- age :**

**- productID :**

**- product\_name:**

**- price :**

CREATE VIEW tblView AS

SELECT o.orderID,c.customerID,c.firs\_name,c.last\_name,c.age,p.productID,p.product\_name,p.price

FROM tblCustomer as c

INNER JOIN tblOrder as o ON c.customerID=o.customerID

INNER JOIN tblProduct as p ON p.productID=o.productID

**SORU 12 İLE BAĞIMLIDIR. SORU 12 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.**

**(5p) Soru 13 : tblView adıyla oluşturduğunuz VIEW tablosundaki tüm kayıtları listeleyen kodu yazınız.**

USE VtysLabDB

SELECT \* FROM tblView

**Uygulama#9\_Cevaplar**

**Uygulama 8'de oluşturulan "VtysLabDB" veritabanı mevcutta veri tabanınınzda yoksa aşağıdaki gibi oluşturunuz.**

CREATE DATABASE VtysLabDB

**-- SORUYA HAZIRLIK --**

**VtysLabDB veri tabanında ilgili tabloları oluşturmak için kullanınız.**

USE VtysLabDB

CREATE TABLE tblCustomer

( customerID INT IDENTITY PRIMARY KEY,

Firs\_name VARCHAR(20) NOT NULL,

last\_name VARCHAR(20) DEFAULT 'NotSpecified',

age INT NOT NULL,

CONSTRAINT CHC\_CustomerAge CHECK (age>=18)

)

INSERT INTO tblCustomer(firs\_name,last\_name,age)

VALUES

('Debra','Burks',24),

('Kasha','Todd',36),

('Robby','Sykes',22)

CREATE TABLE tblProduct

( productID INT IDENTITY PRIMARY KEY,

product\_name VARCHAR(255) NOT NULL,

price FLOAT NOT NULL,

CONSTRAINT CHC\_ProductPrice CHECK (price>0)

)

INSERT INTO tblProduct (product\_name,price)

VALUES

('Surly Straggler',12.5),

('Electra Townie Original',10),

('Pure Cycles William',5)

CREATE TABLE tblOrder

( orderID INT IDENTITY PRIMARY KEY,

productID INT,

customerID INT,

CONSTRAINT FK\_ProductOrder FOREIGN KEY (productID) REFERENCES tblProduct(productID),

CONSTRAINT FK\_CustomertOrder FOREIGN KEY (customerID) REFERENCES tblCustomer(customerID)

)

**-- SORUYA HAZIRLIK --**

**"tblOrder" tablosuna aşağıdaki 5 adet kayıt ekleyiniz.**

**use VtysLabDB**

INSERT INTO tblOrder (customerID,productID)

VALUES

(1,1),

(1,2),

(1,3),

(2,1),

(2,2),

(2,3),

(3,1),

(3,2),

(3,3)

**(15p) Soru 1 : "TRG\_CustomerCount" adında bir trigger oluşturunuz. Bu trigger "tblcustomer" tablosuna kayıt eklendikten sonra toplam müşteri(customer) sayısı "countOfCustomer" adıyla ekrana yazdırsın.**

CREATE TRIGGER TRG\_CustomerCount ON tblCustomer

AFTER INSERT

AS

BEGIN

SELECT count(\*) as countOfCustomer FROM tblCustomer

END

**SORU 1 İLE BAĞIMLIDIR. SORU 1 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.**

**(5p) Soru 2 : Aşağıda verilen kaydı "tblCustomer" tablosuna ekleyiniz. Aldığınız sonucu MUTLAKA açıklayınız**

use VtysLabDB

INSERT INTO tblCustomer(firs\_name,last\_name,age)

VALUES

('Melia','Korly',42)

**(15p) Soru 3 : "TRG\_DeletedProduct" adında bir trigger oluşturunuz. Bu trigger "tblProduct" tablosundan bir kayıt silindikten sonra silinen kaydı ekrana yazdırsın.**

CREATE TRIGGER TRG\_DeletedProduct ON tblProduct

AFTER DELETE

AS

BEGIN

SELECT \* FROM DELETED

END

**SORU 3 İLE BAĞIMLIDIR. SORU 3 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.(25p) Soru 4 : "tblProduct" tablosundan sizin belirlediğiniz bir kaydı siliniz. Bu süreçte karşınıza çıkan hatalara uygun çözümler üreterek tablodan kaydı silin.**

**Aldığınız hatayı, nedenini ve çözümünüzü MUTLAKA yorumda yazınız. Aksi halde puan verilmeyecektir.**

ALTER TABLE tblOrder

DROP CONSTRAINT FK\_ProductOrder

ALTER TABLE tblOrder

ADD CONSTRAINT FK\_ProductOrder FOREIGN KEY (productID) REFERENCES tblProduct(productID) ON DELETE CASCADE

DELETE FROM tblProduct WHERE productID=2;

**-- SORUYA HAZIRLIK --**

**Aşağıda verilen şekilde tabloyu oluşturunuz. Bu tabloyu bir sonraki soruda "silinen siparişleri(order) saklamak için kullanacağız."**

**- Tablo adı : tblLoss**

**- Kolonlar :**

**- lossID : Integer veri tipindedir. Primary KEY olarak kullanılacaktır. Otomatik olarak artış gösterir.**

**- orderID : Integer veri tipindedir.**

**- productID : Integer veri tipindedir.**

**- product\_name : Varchar(255) veri tipindedir. NULL olmasına izin verilmez.**

**- price : Float veri tipindedir. NULL olmasına izin verilmez.**

CREATE TABLE tblLoss

(

lossID INT IDENTITY PRIMARY KEY,

orderID INT,

productID INT,

product\_name VARCHAR(255) NOT NULL,

price FLOAT NOT NULL

)

**(35p) Soru 5 : "TRG\_DeletedOrder" adında bir trigger oluşturunuz. Bu trigger "tblOrder" tablosundan bir kayıt silindikten sonra aşağıda belirtilen görevleri yerine getirmelidir.**

**- "tblOrder" tablosundan silinen kaydı "tblLoss" tablosuna kaydedecektir. "tblLoss" tablosundaki kolonlara dikkat ediniz.**

**- "tblLoss" tablosundaki kayıtların ürün bazında toplam fiyatını(price) "totalPriceLoss" adıyla ekrana yazdıracaktır. (Hangi üründen kaç liralık sipariş silinmiştir? )**

CREATE TRIGGER TRG\_DeletedOrder ON tblOrder

AFTER DELETE

AS

BEGIN

DECLARE @oID INT

DECLARE @pID INT

DECLARE @pName VARCHAR(255)

DECLARE @price FLOAT

SELECT @oID=orderID, @pID=productID FROM DELETED

SELECT @pName=product\_name, @price=price FROM tblProduct WHERE productID=@pID

INSERT INTO tblLoss(orderID,productID,product\_name,price)

VALUES (@oID,@pID,@pName,@price)

SELECT SUM(price) as totalPriceLoss FROM tblLoss GROUP BY productID

END

**SORU 5 İLE BAĞIMLIDIR. SORU 5 YAPILMADIĞI TAKTİRDE PUANLAMA OLMAYACAKTIR.**

**(5p) Soru 6 : "tblOrder" tablosundan sizin belirlediğiniz en az 2 farklı siparişi siliniz. Bu siparişler farklı ürünler için olmalıdır. Sonucu yorumda açıklayınız.**

DELETE FROM tblOrder WHERE orderID=1;

**Uygulama#10\_Cevaplar**

**(10p) Soru 1: Fiyatı en yüksek 26 ürünü(products) aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Ürün adı ve ürün fiyatı çıktı olarak verilmelidir.**

**- Ürün fiyatına göre büyükten küçüğe sıralanmalıdır.**

**- Parametre değeri olarak 26 kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getTopMostExpensiveProducts

@count INTEGER

AS

SELECT TOP(@count) products.product\_name, products.list\_price

FROM products ORDER BY list\_price DESC

GO

EXEC getTopMostExpensiveProducts 26

**(10p) Soru 2: Toplam indirim(discount) miktarı, 2'den fazla olan ürünleri(product) aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Ürün ID ve toplam indirim bilgisi (as sumOfDiscount) çıktı olarak verilmelidir.**

**- Ürün ID'ye göre küçükten büyüğe sıralanmalıdır.**

**- Parametre değeri olarak 2 kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getDiscountedProducts

@discountValue INTEGER

AS

SELECT oi.product\_id, SUM(oi.discount) as sumOfDiscount FROM order\_items as oi GROUP BY oi.product\_id HAVING SUM(oi.discount) > @discountValue ORDER BY oi.product\_id

GO

EXEC getDiscountedProducts 2

**(10p) Soru 3: Aldığı sipariş(orders) sayısı, 100'den fazla olan personelleri(staff) aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Personel ID ve sipariş sayısı(as countOfOrder) çıktı olarak verilmelidir.**

**- Satılan toplam ürün sayısına göre büyükten küçüğe sıralanmalıdır.**

**- Uygun JOIN yöntemleri kullanılmalıdır.**

**- Parametre değeri olarak 100 kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getOrderCountofStaff

@orderCount INTEGER

AS

SELECT s.staff\_id, count(\*) as countOfOrder FROM staffs as s INNER JOIN orders as o ON o.staff\_id=s.staff\_id GROUP BY s.staff\_id HAVING count(\*) > @orderCount ORDER BY countOfOrder DESC

GO

EXEC getOrderCountofStaff 100

**(10p) Soru 4: Toplam stok sayısı(quantity), 25'ten fazla olan ürünleri(product) aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Ürün ID, ürün adı, marka ID, model yılı ve stok sayısı(as sumOfQuantity) çıktı olarak verilmelidir.**

**- Ürün adına göre küçükten büyüğe sıralanmalıdır.**

**- Uygun JOIN yöntemleri kullanılmalıdır.**

**- Parametre değeri olarak 25 kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getQuantityofProducts

@quantityValue INTEGER

AS

SELECT p.product\_id, p.product\_name, p.brand\_id, p.model\_year, t.sumOfQuantity FROM products as p INNER JOIN ( SELECT st.product\_id, SUM(st.quantity) as sumOfQuantity

FROM stocks as st GROUP BY st.product\_id HAVING SUM(st.quantity) > @quantityValue ) as t ON p.product\_id=t.product\_id ORDER BY p.product\_name, p.model\_year

GO

EXEC getQuantityofProducts 25

**(10p) Soru 5: Rowlett Bikes mağazasındaki(stores) stoklarda(stocks) bulunan ürünlerden(products) isminin ikinci harfi 'a' olan ürünleri aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Mağaza adı, stok miktarı, ürün adı ve fiyatı sayısı çıktı olarak verilmelidir.**

**- Parametre değeri olarak Rowlett Bikes kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getProductsinStore

@storeNameValue NVARCHAR(50)

AS

SELECT store\_name, quantity, product\_name, list\_price FROM stores, stocks, products WHERE stores.store\_id = stocks.store\_id AND stocks.product\_id = products.product\_id AND product\_name

LIKE '\_a%' AND store\_name = @storeNameValue

GO

EXEC getProductsinStore 'Rowlett Bikes'

**(10p) Soru 6: Mağazalardaki(stores) bulunan ürünlerden(products), stok(stocks) miktarı 15'ten büyük olanların stok sayısını, maksimum fiyatını ve minimum fiyatını aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Mağaza adı, toplam ürün sayısı as (totalCount), maksimum fiyat değeri as (maxPrice) ve minimum fiyat as (minPrice) değeri çıktı olarak verilmelidir.**

**- Parametre değeri olarak 15 kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getProductDetailsinStores

@quantityValue INTEGER

AS

SELECT store\_name, COUNT(\*), MAX(list\_price), MIN(list\_price) FROM stores, stocks, products WHERE stores.store\_id = stocks.store\_id AND stocks.product\_id = products.product\_id

AND quantity > @quantityValue GROUP BY store\_name

GO

EXEC getProductDetailsinStores 15

**(10p) Soru 7: Baldwin Bikes mağazasında(store) çalışan personel(staffs) sayısını aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Çalışan sayısı as (countofStaffs) çıktı olarak verilmelidir.**

**- Parametre değeri olarak Baldwin Bikes kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getCountofStaffs

@storeNameValue NVARCHAR(50)

AS

SELECT COUNT(\*) as count\_of\_baldwin\_staffs FROM BikeStores.dbo.staffs as s, BikeStores.dbo.stores as m WHERE s.store\_id=m.store\_id and m.store\_name = @storeNameValue

GO

EXEC getCountofStaffs 'Baldwin Bikes'

**(10p) Soru 8: Markası(brand), Trek olan ürünlerin(product) toplam liste fiyatını(list\_price) aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Liste fiyatının toplamı as (sumofListPrices) çıktı olarak verilmelidir.**

**- Parametre değeri olarak Trek kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getSumofPrices

@brandNameValue NVARCHAR(50)

AS

SELECT SUM(p.list\_price) as sumofListPrices FROM BikeStores.dbo.brands as b, BikeStores.dbo.products as p WHERE b.brand\_id=p.brand\_id AND b.brand\_name = @brandNameValue

GO

EXEC getSumofPrices 'Trek'

**(10p) Soru 9: Adı Jone olan müşteriye(customer) ait tüm siparişler için sipariş(orders) tarihini aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Müşteri Adı, müşteri soyadı ve sipariş tarihi çıktı olarak verilmelidir.**

**- Parametre değeri olarak Jone kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getCustomerOrdersDetails

@customerNameValue NVARCHAR(50)

AS

SELECT c.first\_name,c.last\_name,c.customer\_id, o.order\_date FROM BikeStores.dbo.customers as c, BikeStores.dbo.orders as o WHERE o.customer\_id=c.customer\_id and c.first\_name = @customerNameValue

GO

EXEC getCustomerOrdersDetails 'Jone'

**(10p) Soru 10: Stok(stocks) miktarı 19'dan az ve liste fiyatı 2500'den fazla olan ürünleri(products) aşağıdaki şekilde listeleyen saklı yordam kodunu yazınız.**

**- Mağaza ID, ürün ID, ürün adı, stok miktarı ve liste fiyatı çıktı olarak verilmelidir.**

**- Parametre değeri olarak 19 ve 2500 kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE PROCEDURE getProducts

@quantityValue INTEGER,

@priceValue INTEGER

AS

SELECT st.store\_id, p.product\_id, p.product\_name, st.quantity, p.list\_price FROM BikeStores.dbo.stocks as st, BikeStores.dbo.products as p

WHERE st.product\_id=p.product\_id and st.quantity < @quantityValue and p.list\_price > @priceValue

GO

EXEC getProducts 19, 2500

**Uygulama#11\_Cevaplar**

**(25p) Soru 1: Belirli bir mağazadaki(stores) çalışanlara(staffs) email gönderilmek istenmektedir. Aşağıdaki koşullara göre fonksiyon kodunu yazınız.**

**- Mağazanın adı parametre olarak fonksiyona verildiğinde o mağazadaki çalışanların email adresleri listelenmelidir.**

**- Hiç kayıt bulunamıyorsa 'Value cannot be found!' uyarısını listelemelidir.**

**- Parametre değeri olarak 'Rowlett Bikes' kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE FUNCTION storeMailGroup (@store\_name nvarchar(50))

RETURNS @staffsMail TABLE (email nvarchar(50))

AS

BEGIN

INSERT INTO @staffsMail

SELECT staffs.email FROM stores INNER JOIN staffs ON stores.store\_id = staffs.store\_id AND store\_name LIKE @store\_name

DECLARE @count INTEGER

SELECT @count = COUNT(\*) FROM @staffsMail

IF @count = 0

INSERT INTO @staffsMail VALUES ('Value cannot be found!')

RETURN

END

GO

SELECT \* FROM storeMailGroup('Rowlett Bikes')

**(25p) Soru 2: Belirli bir müşterinin(customers), sipariş(orders) vermiş olduğu order\_date listelenmek istenmektedir. Aşağıdaki koşullara göre fonksiyon kodunu yazınız.**

**- Müşterinin adı parametre olarak fonksiyona verildiğinde o müşteriye ait siparişlerin tarihleri listelenmelidir.**

**- Hiç kayıt bulunamıyorsa 'Value cannot be found!' uyarısını listelemelidir.**

**- Parametre değeri olarak 'Debra Burks' kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

**NOT: Parametre değerinin iki ayrı sütunu (first\_name, last\_name) temsil ettiğini unutmayınız!**

CREATE FUNCTION customerOrderDate (@customers\_name nvarchar(50), @customers\_surname nvarchar(50))

RETURNS @customerOrdersDate TABLE (order\_date DATE)

AS

BEGIN

INSERT INTO @customerOrdersDate

SELECT orders.order\_date FROM customers INNER JOIN orders ON customers.customer\_id = orders.customer\_id AND (customers.first\_name LIKE @customers\_name AND customers.last\_name LIKE @customers\_surname)

DECLARE @count INTEGER

SELECT @count = COUNT(\*) FROM @customerOrdersDate

IF @count = 0

INSERT INTO @customerOrdersDate VALUES ('Value cannot be found!')

RETURN

END

GO

SELECT \* FROM customerOrderDate('Debra', 'Burks')

**(25p) Soru 3: Belirli bir mağazadaki(stores) stoklarda(stocks) bulunan ürünlerin(products) stok adet miktarı görüntülenmek istenmektedir. Aşağıdaki koşullara göre fonksiyon kodunu yazınız.**

**- Mağazanın adı parametre olarak fonksiyona verildiğinde o mağazadaki bulunan ürünlerin adı ve stok miktarı listelenmelidir.**

**- Hiç kayıt bulunamıyorsa 'Value cannot be found!' uyarısını listelemelidir.**

**- Parametre değeri olarak 'Santa Cruz Bikes' kullanılacaktır. (Parametre değeri çalışmayan uygulamalar değerlendirilmeyecektir!)**

CREATE FUNCTION storeProductStocks8 (@store\_name nvarchar(50))

RETURNS @productStocks TABLE (product\_name nvarchar(255), quantity INTEGER)

AS

BEGIN

INSERT INTO @productStocks

SELECT products.product\_name, stocks.quantity FROM ((stocks INNER JOIN products ON stocks.product\_id = products.product\_id) INNER JOIN stores ON stocks.store\_id = stores.store\_id AND store\_name LIKE @store\_name)

DECLARE @count INTEGER

SELECT @count = COUNT(\*) FROM @productStocks

IF @count = 0

INSERT INTO @productStocks VALUES ('Value cannot be found!', 0)

RETURN

END

GO

SELECT \* FROM storeProductStocks8('Santa Cruz Bikes')

**(25p) Soru 4: Marka(brands) adına ve kategori(categories) adına göre gruplandırılma yapıldığında ürünlerin(products) toplam sayıları görüntülenmek istenmektedir. Aşağıdaki koşullara göre fonksiyon kodunu yazınız.**

**- Fonksiyon çalıştırıldığında marka adı, kategori adı ve toplam ürün sayısı listelenmelidir.**

**- Hiç kayıt bulunamıyorsa 'Value cannot be found!' uyarısını listelemelidir.**

**- Parametre değeri bulunmamaktadır.**

CREATE FUNCTION sumofProductsbyBrandandCategory ()

RETURNS @productStocks TABLE (brand\_name nvarchar(255), category\_name nvarchar(255), sumofProducts INTEGER)

AS

BEGIN

INSERT INTO @productStocks

SELECT brands.brand\_name, categories.category\_name, COUNT(\*) FROM ((products INNER JOIN brands ON products.brand\_id = brands.brand\_id) INNER JOIN categories ON products.category\_id = categories.category\_id) GROUP BY brand\_name, category\_name

DECLARE @count INTEGER

SELECT @count = COUNT(\*) FROM @productStocks

IF @count = 0

INSERT INTO @productStocks VALUES ('Value cannot be found!', 'Value cannot be found!', 0)

RETURN

END

GO

SELECT \* FROM sumofProductsbyBrandandCategory()