**---MARKET PROJESİ VERİTABANI-1 KISMI (BÖLÜM 1)---**

**tbl\_Alis(Alis\_id,Urun\_id,AlimTarihi,Adet)**

**tbl\_AlisTurSebep(AlisTurSebep\_id,AlisTurSebepAd)**

**tbl\_Kategori(Kategori\_id,KategoriAdi)**

**tbl\_Marka(Marka\_id,MarkaAdi,UreticiFirma\_id,UrunTur\_id,AlisFiyat,SatisFiyat)**

**tbl\_Musteriİade(Musteriİade\_id,Musteri\_id,AlisTurSebep\_id,Urun\_id,Alis\_Tarihi,Aciklama,FirmayaGeriİadeDönüsTarihi)**

**tbl\_Musteri(Musteri\_id,MusteriAdi,MusteriSoyadi,Musteri\_tcNo,MusteriDogumTarihi, MusteriCinsiyet,MusteriTelNo,MusteriAdres)**

**tbl\_Personel(per\_id,per\_ad,per\_soyad,per\_tcNo,per\_DogumTarihi,per\_cinsiyet,per\_TelNo,**

**per\_adres,Maas,pertip\_id)**

**tbl\_PersonelTipi(pertip\_id,pertip\_adi)**

**tbl\_Satis(Satis\_id,Musteri\_id,SatisTarihi,per\_id)**

**tbl\_UreticiFirma(UreticiFirma\_id,UreticiFirmaAdi)**

**tbl\_UreticiFirmaGeriIade(Geriİade\_id,MusteriAlis\_id,Tarih)**

**tbl\_Urun(Urun\_id,UrunAd,Marka\_id,SonKullanmaTarihi,BarkodNo)**

**tbl\_UrunTur(UrunTur\_id,UrunTurAd,Kategori\_id)**

**--- VERİTABANI 2 KISMI (BÖLÜM 2) ---**

**a) VİEW**

**View'ler bir veya birden fazla tablodan istenilen verilerin bir arada sunulmasını sağlayan tanımlanmış sorgulardır**.

**1) Yazan:** **Ahmet Özdinçer**

**-- Alınmış Olan Ürünleri kimsenin görmemesi (sorguyu yazan dâhil) ve bilgilerin gizlenmesi için Encryption view i oluşturduk.(Bir VİEW'in Kaynak Kodlarını gizlemek için kullanılır.Böylece VİEW'in kaynak tablolaları ve kaynak kodları gizli duruma getirilir.)**

CREATE VIEW vw\_AlisUrunler

WITH ENCRYPTION

AS

SELECT A.adet,A.AlimTarihi,U.UrunAd,U.SonKullanmaTarihi,U.BarkodNo

FROM tbl\_Urun U INNER JOIN

tbl\_Alis A ON U.urun\_id = A.urun\_id

**2) Yazan: Ahmet Özdinçer**

**-- Ürünler ve üreticiFirma tablo bilgilerini marka\_id’ye göre birleştiren ve bilgilerini sorgulayan ancak değiştirilmemesini sağlamak için(ALTER yapılmasını) With Schemabındıng kullanılır. (VİEW'in referans aldığı kaynak alanlarında şema değişikliği yapılmasını engellemek için kullanılır. (ALTER Kullanılırsa Hata verir değişiklik yapıldığı için.))**

CREATE VIEW vw\_UrunUreticiFirma

WITH SCHEMABINDING

AS

SELECT U.UrunAd,U.SonKullanmaTarihi,U.BarkodNo,M.MarkaAdi,M.AlisFiyat,M.SatisFiyat FROM dbo.tbl\_Urun U INNER JOIN dbo.tbl\_Marka M ON U.Marka\_id=M.Marka\_id

**3) Yazan: Rıfat Koçak**

-- **Müşterinin İade ettiği Ürünleri kimsenin görmemesi (sorguyu yazan dâhil) ve bilgilerin gizlenmesi için Encryption view i oluşturduk.**

CREATE VIEW vw\_UrunMusteriİade

WITH ENCRYPTION

AS

SELECT U.UrunAd,U.SonKullanmaTarihi,U.BarkodNo,Mİ.Aciklama,Mİ.Alis\_Tarihi,Mİ.FirmayaGeriİadeDönüsTarihi FROM dbo.tbl\_Urun U INNER JOIN

dbo.tbl\_Musteriİade Mİ ON U.Urun\_id = Mİ.Urun\_id

**4) Yazan: Rıfat Koçak**

**-- Musteriye Yapılan Satışların tablo bilgilerini Musteri\_id’ ye göre birleştiren ve bilgilerini sorgulayan ancak değiştirilmemesini sağlamak için(ALTER yapılmasını engellemek için)**

**With SchemaBındıng kullandık.**

CREATE VIEW vw\_SatisMusteri

WITH SCHEMABINDING

AS

SELECT M.MusteriAd,M.MusteriSoyad,M.Musteri\_tcNo,M.MusteriAdres,M.Mustericinsiyet,M.MusteriDogumTarihi,M.MusteriTelNo,S.SatisTarihi

FROM dbo.tbl\_Musteri M INNER JOIN

dbo.tbl\_Satis S ON M.Musteri\_id = S.Musteri\_id

**b) İNDEX**

**İndex tablodaki belirli sütunları kullanarak tablo içerisindeki verilere hızlı erişimi ve verilerin belirli sütunlara göre sıralanmasını sağlar.**

**1) Yazan: Ahmet Özdinçer**

**-- Musteri tablosundaki MusteriAd ve Musteri Soyad sütunlarına hızlı erişim sağlayan index.**

CREATE NONCLUSTERED index MusteriAdMusteriSoyad

ON tbl\_Musteri (MusteriAdi,MusteriSoyadi)

**2) Yazan: Ahmet Özdinçer**

**-- Personel tablosunda personellerin adını soyadını ve maaşını döndüren index.**

CREATE NONCLUSTERED index personeladpersonelsoyadMaas

ON tbl\_personel(per\_adi, per\_soyad,Maas)

**3) Yazan: Rıfat Koçak**

**-- Urun tablosunda Urunlerin adlarını döndüren index.**

CREATE NONCLUSTERED index UrunAd

on tbl\_Urun(UrunAd)

**4) Yazan: Rıfat Koçak**

**-- UreticiFirma tablosunda UreticiFirma adlarını döndüren index.**

CREATE NONCLUSTERED index UreticiFirmaAdi

ON tbl\_UreticiFirma (UreticiFirmaAdi)

**c) STORED PROCEDURE(Her tabloya Veri ekleme,Silme,Güncelleme Prosedürleri)**

**1) Yazan: Ahmet Özdinçer**

**--tbl\_Alis tablosuna veri ekleme,güncelleme,silme Stored Procedürü.**

CREATE PROCEDURE stp\_Alis(@durum int,@Alis\_id int,@Urun\_id int,@AlimTarihi smalldatetime,@Adet int)

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_Alis values(@Urun\_id,@AlimTarihi,@Adet)

PRINT 'Kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_Alis SET Urun\_id = @Urun\_id,AlimTarihi = @AlimTarihi,Adet = @Adet WHERE Alis\_id = @Alis\_id

PRINT 'Kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_Alis WHERE Alis\_id = @Alis\_id

PRINT 'Kayit silindi'

END

END

**2) Yazan: Ahmet Özdinçer**

**--** **tbl\_AlisTurSebep tablosuna veri ekleme,güncelleme,silme Stored Procedürü.**

CREATE PROCEDURE stp\_AlisTurSebep(@durum int,@AlisTurSebep\_id int,@AlisTurSebepAd nvarchar(50))

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_AlisTurSebep VALUES(@AlisTurSebepAd)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1  **--** **Güncelleme**

BEGIN

UPDATE tbl\_AlisTurSebep SET AlisTurSebepAd = @AlisTurSebepAd WHERE AlisTurSebep\_id = @AlisTurSebep\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_AlisTurSebep WHERE AlisTurSebep\_id = @AlisTurSebep\_id

PRINT 'kayit silindi'

END

END

**3) Yazan: Ahmet Özdinçer**

**-- tbl\_Kategori tablosuna veri ekleme,güncelleme,silme Stored Procedurü.**

CREATE PROCEDURE stp\_Kategori(@durum int,@Kategori\_id int,@KategoriAdi nvarchar(50))

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_Kategori VALUES(@KategoriAdi)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_Kategori SET KategoriAdi = @KategoriAdi WHERE Kategori\_id = @Kategori\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_Kategori WHERE Kategori\_id = @Kategori\_id

PRINT 'kayit silindi'

END

END

**4) Yazan: Ahmet Özdinçer**

**-- tbl\_Marka tablosuna veri ekleme,güncelleme,silme işlemlerini yapan Stored Procedure.**

CREATE PROCEDURE stp\_Marka(@durum int,@Marka\_id int,@MarkaAdi nvarchar(50),@UreticiFirma\_id int,@UrunTur\_id int,@AlisFiyat money,@SatisFiyat money)

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_Marka VALUES(@MarkaAdi,@UreticiFirma\_id,@UrunTur\_id,@AlisFiyat,@SatisFiyat)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_Marka SET MarkaAdi = @MarkaAdi,UreticiFirma\_id = @UreticiFirma\_id,UrunTur\_id = @UrunTur\_id,AlisFiyat = @AlisFiyat,SatisFiyat = @SatisFiyat WHERE Marka\_id = @Marka\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_Marka WHERE Marka\_id = @Marka\_id

PRINT 'kayit silindi'

END

END

**5) Yazan: Ahmet Özdinçer**

**-- tbl\_Musteri tablosuna veri ekleme,güncelleme,silme Stored Procedurü.**

CREATE PROCEDURE stp\_Musteri(@durum int,@Musteri\_id int,@MusteriAd nvarchar(50),@MusteriSoyad nvarchar(50),@Musteri\_tcNo decimal,@MusteriDogumTarihi datetime,@Mustericinsiyet nvarchar(50),@MusteriTelNo nvarchar(50),@MusteriAdres nvarchar(500))

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_Musteri VALUES(@MusteriAd,@MusteriSoyad,@Musteri\_tcNo,@MusteriDogumTarihi,@Mustericinsiyet,@MusteriTelNo,@MusteriAdres)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_Musteri SET MusteriAd = @MusteriAd,MusteriSoyad = @MusteriSoyad,Musteri\_tcNo = @Musteri\_tcNo,MusteriDogumTarihi = @MusteriDogumTarihi,Mustericinsiyet = @Mustericinsiyet,MusteriTelNo = @MusteriTelNo,MusteriAdres = @MusteriAdres WHERE Musteri\_id = @Musteri\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_Musteri WHERE Musteri\_id = @Musteri\_id

PRINT 'kayit silindi'

END

END

**6) Yazan: Rıfat Koçak**

**-- tbl\_Musteriİade tablosuna veri ekleme,güncelleme,silme Stored Procedürü.**

CREATE PROCEDURE stp\_Müsteriİade(@durum int,@Musteriİade\_id int,@Musteri\_id int,@AlisTurSebep\_id int,@Urun\_id int,@Alis\_Tarihi smalldatetime,@Aciklama nvarchar(50),@FirmayaGeriİadeDönüsTarihi datetime)

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_Musteriİade values(@Musteri\_id,@AlisTurSebep\_id,@Urun\_id,@Alis\_Tarihi,@Aciklama,@FirmayaGeriİadeDönüsTarihi)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_Musteriİade SET Musteri\_id = @Musteri\_id,AlisTurSebep\_id = @AlisTurSebep\_id,Urun\_id = @Urun\_id,Alis\_Tarihi = @Alis\_Tarihi,Aciklama = @Aciklama,FirmayaGeriİadeDönüsTarihi = @FirmayaGeriİadeDönüsTarihi WHERE Musteriİade\_id = @Musteriİade\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2  **-- Silme**

BEGIN

DELETE FROM tbl\_Musteriİade WHERE Musteriİade\_id = @Musteriİade\_id

PRINT 'kayit silindi'

END

END

**7) Yazan: Rıfat Koçak Burda kaldık**

**-- tbl\_Personel tablosuna veri ekleme,güncelleme,silme yapan Stored Procedure.**

CREATE PROCEDURE stp\_Personel(@durum int,@per\_id int,@pertip\_id int,@per\_adi nvarchar(50),@per\_soyad nvarchar(50),@per\_tcNo decimal,@per\_DogumTarihi smalldatetime,@per\_cinsiyet nvarchar(50),@per\_TelNo nvarchar(50),@per\_adres nvarchar(200),@Maas money)

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_Personel VALUES(@pertip\_id,@per\_adi,@per\_soyad,@per\_tcNo,@per\_DogumTarihi,@per\_cinsiyet,@per\_TelNo,@per\_adres,@Maas)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_Personel SET pertip\_id = @pertip\_id,per\_adi = @per\_adi,per\_soyad = @per\_soyad,per\_tcNo = @per\_tcNo,per\_DogumTarihi = @per\_DogumTarihi,per\_cinsiyet = @per\_cinsiyet,per\_TelNo = @per\_TelNo,per\_adres = @per\_adres,Maas = @Maas WHERE per\_id = @per\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_Personel WHERE per\_id = @per\_id

PRINT 'kayit silindi'

END

END

**8) Yazan: Rıfat Koçak**

**-- tbl\_PersonelTipi tablosuna veri ekleme,güncelleme,silme yapan Stored Procedure.**

CREATE PROCEDURE stp\_PersonelTipi(@durum int,@pertip\_id int,@pertip\_adi nvarchar(50))

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_PersonelTipi VALUES(@pertip\_adi)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_PersonelTipi SET pertip\_adi = @pertip\_adi WHERE pertip\_id = @pertip\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_PersonelTipi WHERE pertip\_id = @pertip\_id

PRINT 'kayit silindi'

END

END

**9) Yazan: Rıfat Koçak**

**-- tbl\_Satis tablosuna veri ekleme,güncelleme,silme yapan Stored Procedure.**

CREATE PROCEDURE stp\_Satis(@durum int,@Satis\_id int,@Musteri\_id int,@SatisTarihi smalldatetime,@per\_id int)

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_Satis VALUES(@Musteri\_id,@SatisTarihi,@per\_id)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_Satis SET Musteri\_id = @Musteri\_id,SatisTarihi = @SatisTarihi,per\_id = @per\_id WHERE Satis\_id = @Satis\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_Satis WHERE Satis\_id = @Satis\_id

PRINT 'kayit silindi'

END

END

**10) Yazan: Rıfat Koçak**

**-- tbl\_Uretici Firma tablosuna veri ekleme,güncelleme,silme yapan Stored Procedure.**

CREATE PROCEDURE stp\_UreticiFirma(@durum int,@UreticiFirma\_id int,@UreticiFirmaAdi nvarchar(50))

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_UreticiFirma VALUES(@UreticiFirmaAdi)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_UreticiFirma SET UreticiFirmaAdi = @UreticiFirmaAdi WHERE UreticiFirma\_id = @UreticiFirma\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_UreticiFirma WHERE UreticiFirma\_id = @UreticiFirma\_id

PRINT 'kayit silindi'

END

END

**11) Yazan: Rıfat Koçak**

**-- tbl\_UreticiFirmaGeriİade tablosuna veri ekleme,güncelleme,silme Stored Procedure.**

CREATE PROCEDURE stp\_UreticiFirmaGeriİade(@durum int,@Geriİade\_id int,@Musteriİade\_id int,@Tarih date)

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_UreticiFirmaGeriİade VALUES(@Musteriİade\_id,@Tarih)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_UreticiFirmaGeriİade SET Musteriİade\_id = @Musteriİade\_id,Tarih = @Tarih WHERE Geriİade\_id = @Geriİade\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_UreticiFirmaGeriİade WHERE Geriİade\_id = @Geriİade\_id

PRINT 'kayit silindi'

END

END

**12) Yazan:** **Rıfat Koçak bakılacak**

**-- tbl\_Urun tablosuna veri ekleme,güncelleme,silme Stored Procedurü.**

CREATE PROCEDURE stp\_Urun(@durum int,@Urun\_id int,@UrunAd nvarchar(50),@SonKullanmaTarihi smalldatetime,@Marka\_id int,@BarkodNo decimal)

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_Urun VALUES(@UrunAd,@SonKullanmaTarihi,@Marka\_id,@BarkodNo)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_Urun SET UrunAd = @UrunAd,SonKullanmaTarihi = @SonKullanmaTarihi,Marka\_id = @Marka\_id,BarkodNo = @BarkodNo WHERE Urun\_id = @Urun\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_Urun WHERE Urun\_id = @Urun\_id

PRINT 'kayit silindi'

END

END

**13) Yazan: Rıfat Koçak**

**--tbl\_UrunTur tablosuna veri ekleme,güncelleme,silme Stored Procedurü.**

CREATE PROCEDURE stp\_UrunTur(@durum int,@UrunTur\_id int,@UrunTurAd nvarchar(50),@Kategori\_id int)

AS

BEGIN

IF @durum = 0 **-- Ekleme**

BEGIN

INSERT INTO tbl\_UrunTur VALUES(@UrunTurAd,@Kategori\_id)

PRINT 'kayit eklendi'

END

ELSE IF @durum = 1 **-- Güncelleme**

BEGIN

UPDATE tbl\_UrunTur SET UrunTurAd = @UrunTurAd,Kategori\_id = @Kategori\_id WHERE UrunTur\_id = @UrunTur\_id

PRINT 'kayit guncellendi'

END

ELSE IF @durum = 2 **-- Silme**

BEGIN

DELETE FROM tbl\_UrunTur WHERE UrunTur\_id = @UrunTur\_id

PRINT 'kayit silindi'

END

END

**d) TRY-CATCH**

**TRY-CATCH(HATA KONTROLÜ)= Programlama dillerinde olduğu gibi t-sql de de try catch blokları oluşan hatayı yakalamamızı sağlar ve oluşan hata hakkında bilgi sahibi olmamıza imkân verir.**

**1) Yazan: Ahmet Özdinçer**

**-- tbl\_Musteri tablosuna ekleme işlemi yaparken yanlış bilgi girdiğimizde hatayı try catch sayesinde gösteren ve transaction … rollback sayesinde işlemi geri alan stored procedure.**

CREATE PROCEDURE stp\_Musteri\_giris\_guncelle\_delete

(

@durum int,

@Musteri\_id int,

@MusteriAd NVARCHAR(50),

@MusteriSoyad nvarchar(200),

@Musteri\_tcNo decimal(11, 0),

@MusteriDogumTarihi datetime,

@Mustericinsiyet nvarchar(50),

@MusteriTelNo nvarchar(50),

@MusteriAdres nvarchar(500)

)

AS

BEGIN

SET NOCOUNT ON

if @durum=1

BEGIN

BEGIN TRY

BEGIN TRANSACTION

INSERT INTO tbl\_Musteri

(

MusteriAd,

MusteriSoyad,

Musteri\_tcNo,

MusteriDogumTarihi,

Mustericinsiyet,

MusteriTelNo,

MusteriAdres

)

VALUES

(

@MusteriAd,

@MusteriSoyad,

@Musteri\_tcNo,

@MusteriDogumTarihi,

@Mustericinsiyet,

@MusteriTelNo,

@MusteriAdres

)

COMMIT

END TRY

BEGIN CATCH

PRINT 'tabloya deger eklerken yanlis formatta bilgi girmeyi denediniz.islem basarisiz'

ROLLBACK

END CATCH

END

ELSE IF @durum=2

BEGIN

UPDATE tbl\_Musteri

SET MusteriAd=@MusteriAd,

MusteriSoyad=@MusteriSoyad,

Musteri\_tcNo=@Musteri\_tcNo,

MusteriDogumTarihi=@MusteriDogumTarihi,

Mustericinsiyet=@Mustericinsiyet,

MusteriTelNo=@MusteriTelNo,

MusteriAdres=@MusteriAdres

WHERE Musteri\_id=@Musteri\_id

END

ELSE IF @durum=3

DELETE FROM tbl\_Musteri WHERE Musteri\_id=@Musteri\_id

END

**2) Yazan: Rıfat Koçak**

**--**

**e) FONKSİYONLAR**

**Skaler Fonksiyonlar =** Skaler değerli fonksiyonlar tek değer döndüren fonksiyonlardır.(Tek Kayıt Döndüren) **1) Yazan: Ahmet Özdinçer**

**-- Hangi personel id’yi girmişsek karşılığındaki değeri döndüren skaler fonksiyon.**

CREATE FUNCTION fnc\_personel (@per\_id int)

RETURNS nvarchar

AS

BEGIN

RETURN

(

SELECT CASE @per\_id

WHEN 20 THEN 'Emre'

WHEN 21 THEN 'Emre'

WHEN 22 THEN 'Gökhan'

WHEN 23 THEN 'Bilgehan'

WHEN 24 THEN 'Abdullah'

ELSE 'Tanimli Degil'

END

)

End

**2) Yazan: Ahmet Özdinçer**

**-- Hangi musteri id’yi girmişsek karşılığındaki değeri döndüren skaler fonksiyon** .

CREATE FUNCTION fnc\_musteri (@Musteri\_id int)

RETURNS nvarchar(20)

AS

BEGIN

RETURN

(

SELECT CASE @Musteri\_id

WHEN 1 THEN 'Ahmet'

WHEN 4 THEN 'Rıfat'

WHEN 5 THEN 'Mesut'

WHEN 6 THEN 'Hamid'

WHEN 9 THEN 'Ali'

WHEN 10 THEN 'İlker'

WHEN 11 THEN 'Mustafa'

WHEN 13 THEN 'Gökçe'

WHEN 14 THEN 'Aliye'

ELSE 'Tanimli Degil'

END

)

END

**1) Yazan: Rıfat Koçak**

**-- Bu fonksiyonda urun\_id sütunundaki değere göre karşılığındaki ismi bize geri döndüren bir fonksiyon.**

CREATE FUNCTION fnc\_urun (@Urun\_id int)

RETURNS nvarchar(20)

AS

BEGIN

RETURN

(

SELECT CASE @Urun\_id

WHEN 8 THEN 'Fanta Portakal'

WHEN 9 THEN 'Fanta Zero'

WHEN 10 THEN 'Powerade Zero'

WHEN 12 THEN 'İkram'

WHEN 13 THEN 'Albeni'

ELSE 'Tanimli Degil'

END

)

END

**2) Yazan: Rıfat Koçak bak**

**-- Bu fonksiyonda dönen değere göre karşılığındaki pertipadını bize geri döndüren fonksiyon.**

CREATE FUNCTION fnc\_personelTipi (@pertip\_id int)

RETURNS nvarchar(20)

AS

BEGIN

RETURN

(

SELECT CASE @pertip\_id

WHEN 1 THEN 'Meşrubat Reyon Görevlisi'

WHEN 2 THEN 'Çikolata Reyonu Görevlisi'

WHEN 3 THEN 'Bisküvi Reyonu Görevlisi'

WHEN 4 THEN 'Süt Ürünleri Reyonu Görevlisi'

WHEN 5 THEN 'Et Ürünleri Reyonu Görevlisi'

ELSE 'Tanimli Degil'

END

)

END

**Tablo Döndüren Fonksiyonlar(İnline) Parametreli=** Bir tek select ifadesi içerir. Bu ifadenin sonucunu gösterir.

**1) Yazan: Ahmet Özdinçer**

**--Müşteri Tablosundaki girilen değere göre müşteri ad ve müşteri soyadı döndüren fonksiyon.**

CREATE FUNCTION **fn\_**Musteri**(@**Musteri**\_id int)**

RETURNS TABLE

AS RETURN

SELECT musteriAd,musteriSoyad FROM tbl\_musteri WHERE musteri\_id = @Musteri\_id

**2)** **Yazan: Ahmet Özdinçer**

**-- Personel Tablosundaki girilen değere göre personel adını ve personel soyadını döndüren fonksiyon.**

CREATE FUNCTION fn\_Personel(@Personel\_id int)

RETURNS TABLE

AS RETURN

SELECT per\_adi,per\_soyad FROM tbl\_Personel WHERE per\_id = @Personel\_id

**1) Yazan: Rıfat Koçak**

**-- PersonelTipi tablosundaki girilen değere göre personel tipini döndüren fonksiyon.**

CREATE FUNCTION fn\_PersonelTipi(@PersonelTipi\_id int)

RETURNS TABLE

AS RETURN

SELECT pertip\_adi FROM tbl\_PersonelTipi WHERE pertip\_id = @PersonelTipi\_id

**2) Yazan: Rıfat Koçak**

**-- Urun Tablosundaki girilen değere göre Urunadını, SonKullanmaTarihi,BarkodNo yu döndüren fonksiyon.**

CREATE FUNCTION fn\_Urun(@Urun\_id int)

RETURNS TABLE

AS RETURN

SELECT UrunAd,SonKullanmaTarihi,BarkodNo FROM tbl\_Urun WHERE urun\_id = @Urun\_id

**Tablo Döndüren Parametresiz Fonksiyonlar**:

**1) Yazan: Ahmet Özdinçer**

**--**

CREATE FUNCTION fn\_PersonelSatis()

RETURNS @personelsatis TABLE (

per\_adi nvarchar(50),

per\_soyad nvarchar(50),

per\_tcNo nvarchar(50),

per\_DogumTarihi smalldatetime,

per\_cinsiyet nvarchar(50),

per\_TelNo nvarchar(50),

per\_adres nvarchar(50),

Maas money,

SatisTarihi smalldatetime)

AS

BEGIN

INSERT INTO @personelsatis SELECT p.per\_adi,p.per\_soyad,p.per\_tcNo,p.per\_DogumTarihi,p.per\_cinsiyet,p.per\_TelNo,p.per\_adres,p.Maas,s.SatisTarihi

FROM tbl\_Personel p JOIN tbl\_Satis s ON p.per\_id = s.per\_id

RETURN

END

**2) Yazan: Ahmet Özdinçer**

**--**

**1) Yazan: Rıfat Koçak**

**--**

CREATE FUNCTION fn\_UreticiFirmaMarka()

RETURNS @v TABLE (

UreticiFirmaAdi NVARCHAR (50),

MarkaAdi NVARCHAR(50),

AlisFiyat MONEY,

SatisFiyat MONEY

)

AS

BEGIN

INSERT INTO @v SELECT u.UreticiFirmaAdi, m.MarkaAdi, m.AlisFiyat, m.SatisFiyat

FROM tbl\_UreticiFirma u JOIN tbl\_Marka m ON m.UreticiFirma\_id = u.UreticiFirma\_id

RETURN

END

**2) Yazan: Rıfat Koçak bak**

**--**

CREATE FUNCTION fn\_SatisBilgileri()

RETURNS @v TABLE (

alim\_tarih Smalldatetime,

MusteriAd nvarchar(50),

MusteriSoyad nvarchar(50),

per\_adi nvarchar(50),

per\_soyad nvarchar(50),

satis\_tarihi smalldatetime

)

AS

BEGIN

INSERT INTO @v SELECT a.AlimTarihi, m.MusteriAd, m.MusteriSoyad, p.per\_adi,p.per\_soyad, s.SatisTarihi

FROM tbl\_Alis a JOIN tbl\_Satis s ON a.Alis\_id = s.Alis\_id

JOIN tbl\_Musteri m ON m.Musteri\_id = s.Musteri\_id

JOIN tbl\_Personel p ON s.per\_id = p.per\_id

RETURN

END

GO

**f) TRIGGER(Tetikleyici)**

**MAİL TRİGGER=** Herhangi bir tablo üzerinde yapılan insert, update, delete işlemi öncesinde veya sonrasında otomatik olarak aynı veya başka bir tabloda veri ekleme, veri değiştirme veya veri silme işlemi yapılmak istendiğinde kullanılır.

**1) Yazan: Ahmet Özdinçer**

**-- Personel Tablosuna yeni bir kayıt eklendiğinde mail yollayan trigger.**

CREATE TRIGGER trg\_mailPersonel

ON dbo.tbl\_Personel

AFTER INSERT

AS

DECLARE @posta nvarchar(150),@adi nvarchar(100) = 'Personel Eklendi.'

set @posta='ahmtozdncr@gmail.com'

EXEC msdb.dbo.sp\_send\_dbmail

@profile\_name='AHMET',

@recipients=@posta,

@subject=@adi,

@body='Personeliniz ise baslamistir bilginize.',

@body\_format='HTML'

**2) Yazan: Rıfat Koçak**

**-- Müşteri Tablosuna yeni bir kayıt eklendiğinde mail yollayan trigger.**

CREATE TRIGGER trg\_mailMusteri

ON dbo.tbl\_Musteri

AFTER INSERT

AS

DECLARE @posta nvarchar(150),@adi nvarchar(100) = 'Müsteri Eklendi.'

set @posta='rfatkocak@gmail.com'

EXEC msdb.dbo.sp\_send\_dbmail

@profile\_name='RIFAT',

@recipients=@posta,

@subject=@adi,

@body='Müsteri eklenmistir.',

@body\_format='HTML'

**e) INSTEAD OF TRIGGER**

**1) Yazan: Ahmet Özdinçer**

**-- tbl\_Musteri tablomuzdan herhangi bir Müşterinin hiçbir şekilde silinmemesini sağlar.**

CREATE TRIGGER trg\_musteriSil

ON tbl\_Musteri

INSTEAD OF DELETE

AS

DECLARE @Musteri\_id INT

SELECT @Musteri\_id=Musteri\_id

FROM DELETED

PRINT 'musteri silinemez'

ROLLBACK

GO

**2) Yazan: Rıfat Koçak**

**-- tbl\_Urun tablomuzdan bir ürünü silmeye çalıştığımızda eğerki trigger enable ise tablodan ürünü silmemize engel olacaktır.**

CREATE TRIGGER trg\_Urun

ON tbl\_Urun

INSTEAD OF DELETE

AS

DECLARE @Urun\_id INT

SELECT @Urun\_id = Urun\_id

FROM DELETED

PRINT 'Urun Silinemez'

ROLLBACK

GO

**AFTER TRİGGER**

**1) Yazan: Ahmet Özdinçer**

**-- Marka tablosunda alis fiyatı satış fiyatından düşük olmak zorunda olduğundan eğer satis fiyatı alis fiyatından yüksekse Kayıt eklendi diyecek ve kayıdı tbl\_marka tablosuna ekliycek**

**Eğer satis fiyati alis fiyatından düşük ise Print diyip Satis Fiyati Alis Fiyatina eşit veya kücük olamaz yazıcak. Ve Rollback diyip geri alıcak.**

CREATE TRIGGER trg\_satisFiyatKontrol

ON tbl\_Marka

AFTER INSERT

AS

BEGIN

SET NOCOUNT OFF

DECLARE @satis money

DECLARE @alis money

SELECT @satis = SatisFiyat FROM INSERTED

SELECT @alis = AlisFiyat FROM INSERTED

IF @satis <= @alis

BEGIN

PRINT ‘Satis Fiyati Alis Fiyatina esit veya kucuk olamaz.’

ROLLBACK

END

ELSE

PRINT ‘Kayit eklendi’

SET NOCOUNT ON

END

**2) Yazan: Rıfat Koçak**

**--**