# MARKET OTOMASYONU

ilk haftamızda, yapmak istediğimiz market veri tabanı projemiz için, olması gereken verileri topladık. Örnek bir market firması ile görüşüp ne tarzda tablolar çıkabilir onları inceledik. Daha sonra topladığımız verileri filtreleyerek bize uygun verilerimizi oluşturduk, daha sonra verileri uygun bir şekilde ilişkilendirip taslak bir tablolar oluşturduk.

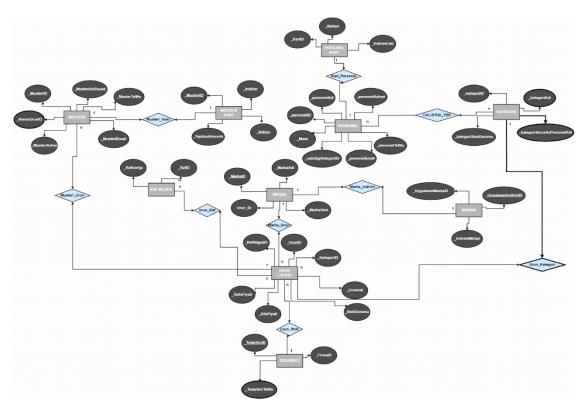
Bir müşteri gözünden projeye bakmak gerekirse; müşterinin indirim ve promosyonlar için kullanabileceği bir kartı olacak. Kullanıcı bizim arayüzümüz ile sisteme giriş yaptığında, tablolarımızda o müşterinin ID'sini sorgulayarak kart bilgilerini ve içeriğini görebilecek. Her alışveriş yaptığında ise yaptığı alışveriş ID'sine kayıt olacak. Böylece alışveriş geçmişini de görebilecek. Müşteriler tablosunun bağlı olduğu Müşteri Kart Tablosunda her bir müşteriye ait ID, Müşteriye özel indirimler, bakiyesi ve en son yaptığı alışveriş bilgileri bulunmaktadır. Eğer müşteriye özel indirim varsa bunun sorgusu yapılır ve uygulanır. Aynı şekilde personelinde kendine özel Personel Kartı bulunmaktadır. Personel Kartı ise firmanın personellere özel tanımladığı indirimler ve yemek ücreti gibi bilgileri tutar. Personel Kart tablosunda personelin ID'si, indirim bilgileri ve firmanın verdiği yemek ücret bakiyesi bulunmaktadır. Bu tablodaki ID ile Personel Tablosundaki ID ilişkilendirip bilgileri Personel tablosundan almaktadır.

Markette bulunan ürünlerimizi sorgulamak için Kategori – Ürün Listesinin bulunduğu bir tablo oluşturduk. Bu tabloda ürün ID'si , ürünün hangi kategoride bulunduğunu gösteren Kategori tablosundan aldığı kategori ID'si, ürün stok durumu, alış fiyatı, satış fiyatı, markette hangi rafta bulunduğunu gösteren Raf tablosundan aldığı raf ID bilgisi, marka tablosundan aldığı marka bilgisi, ve son stok güncelleme tarihi bulunmaktadır.

Ürünleri daha kolay ulaşabilmek için Kategori tablosu oluşturduk. Bu tablo sayesinde yapacağımız arayüzde kullanıcı ürüne daha kolay bulabilecek. Bu tabloda kategori ID'si, kategori adı, bu kategoriden sorumlu personel ID'si ve Kategori stok durumu yer almaktadır.

Personelin müşteriyi daha kolay yönlendirebilmesi için ürünleri raflara ayırdık. Raf tablosunda raf ID'si ve raf içeriği bulunmaktadır. Bu raf tablosunu kategori – ürün tablosu ile ilişkilendirdik. Aynı zamanda kategori – ürün tablosunu Tedarikçi tablosu ile ilişkilendirdik. Bu sayede ürün sorgularken stok durumunu da görebileceğiz. Tedarikçi tablosunda tedarikçi ID'si , Tedarik firma adı, Tedarikçinin iletişim bilgileri bulunmaktadır. Son olarak markette bulunan markaların tablosunu oluşturup her markaya ID atadık. Bu marka ID'sini diğer oluşturduğumuz tablolar ile ilişkilendirdik.

Sonraki haftalarımızda projemizi şekillendirecek olan ER diyagramımızı ve İlişkisel şemamızı aşağıda göründüğü gibi olusturduk.



İlişkisel Şema aşağıdaki gibidir;

PERSONEL KART(\_KartID,\_Bakiye,\_IndirimCeki)

MUSTERI(\_MusteriAdSoyad,\_MusteriTelNo,\_MusteriEmail,\_MusteriAdres,\_AlinanUrunID)

MUSTERIKART(\_MusteriID,\_Indirim,\_Bakiye,\_YapilanAlisveris)

PERSONEL(\_personeID,\_personelAdi,\_personelAdresi,\_personelTelNo,\_personelEmail,\_calistigiKategoril D,\_Maas)

KATEGORI(\_kategoriID,\_kategoriAdi,\_kategoriSorumluPersonelAdi,\_kategoriStokDurumu)

RAFBILGISI(\_RafID,\_Raficerigi)

MARKA(\_MarkaID,\_MarkaAdi,\_MarkaStok,\_UrunID)

INDIRIM(\_UygulananIndirimID,\_UygulananMarkaID,\_IndirimMiktari)

URUNLISTESI(\_UrunID,\_KategoriID,\_UrunAdi,\_StokDurumu,\_AlisFiyati,\_SatisFiyati,\_RafBilgisiID)

TEDARIKCI(\_TedarikciID,\_FirmaID,\_TedarikciTeINo)

CALISTIGI\_YER(\_kategoriID,\_personelID)

URUN\_RAF(\_RafID,UrunID)

Bu proje, bir marketin günlük işlemlerini etkili bir şekilde yönetebilmek için geliştirilmiş bir veri tabanı yönetim sistemini içermektedir. Sistem, market personeli, müşteriler, ürünler, kategoriler, markalar, tedarikçiler ve indirimler gibi temel unsurları içerir ve bu unsurlar arasındaki ilişkileri yönetir.

## Tablolar ve Özellikleri:

- 1. PERSONEL KART Tablosu (\_KartID, \_Bakiye, \_IndirimCeki):
- Market personelinin kimlik bilgilerini ve kart özelliklerini içerir.
- \_KartID, her personel kartını benzersiz bir şekilde tanımlar.
- \_Bakiye, personelin kartındaki para miktarını tutar.
- \_IndirimCeki, personelin kullanabileceği indirim çeki miktarını gösterir.
- 2. MUSTERI Tablosu (\_MusteriID, \_MusteriAdSoyad, \_MusteriTelNo, \_MusteriEmail, \_MusteriAdres, \_AlinanUrunID):
- Müşteri bilgilerini ve satın alınan ürünleri içerir.
- \_MusterilD, her müşteriyi benzersiz bir şekilde tanımlar.
- \_AlinanUrunID, müşterinin satın aldığı ürünleri belirtir.
- 3. MUSTERIKART Tablosu (\_MusterilD, \_Indirim, \_Bakiye, \_YapilanAlisveris):
- Müşteri kart bilgilerini içerir.
- \_Indirim, müşterinin kartına tanımlanan indirim oranını gösterir.
- \_YapilanAlisveris, müşterinin toplam alışveriş miktarını tutar.
- 4. PERSONEL Tablosu (\_personeID, \_personelAdi, \_personelAdresi, \_personelTelNo, \_personelEmail, \_calistigiKategoriID, \_Maas):
- Market personelinin genel bilgilerini ve çalıştığı kategoriyi içerir.
- \_calistiqiKategoriID, personelin çalıştığı kategoriyi belirtir.
- Maas, personelin maaş bilgisini içerir.
- 5. KATEGORI Tablosu (\_kategoriID, \_kategoriAdi, \_kategoriSorumluPersonelAdi, \_kategoriStokDurumu):
- Ürün kategorilerini içerir.
- \_kategorilD, her kategoriyi benzersiz bir şekilde tanımlar.
- \_kategoriSorumluPersonelAdi, kategoriye sorumlu olan personelin adını belirtir.
- \_kategoriStokDurumu, kategorinin genel stok durumunu gösterir.
- 6. RAFBILGISI Tablosu (\_RafID, \_Raficerigi):

- Rafların içeriğini ve hangi ürünlerin bulunduğunu içerir.
- \_RafID, her rafı benzersiz bir şekilde tanımlar.
- 7. MARKA Tablosu (\_MarkalD, \_MarkaAdi, \_MarkaStok, \_UrunID):
- Markette bulunan markaları içerir.
- \_MarkalD, her markayı benzersiz bir şekilde tanımlar.
- \_MarkaStok, markanın genel stok durumunu gösterir.
- 8. INDIRIM Tablosu (\_UygulananIndirimID, \_UygulananMarkaID, \_IndirimMiktari):
- Uygulanan indirimleri ve hangi markalara uygulandığını içerir.
- \_UygulananIndirimID, her indirimi benzersiz bir şekilde tanımlar.
- \_IndirimMiktari, uygulanan indirim miktarını gösterir.
- 9. URUNLISTESI Tablosu (\_UrunID, \_KategoriID, \_UrunAdi, \_StokDurumu, \_AlisFiyati, \_SatisFiyati, \_RafBilgisiID):
- Markette bulunan ürünleri ve bu ürünlerin detaylarını içerir.
- \_UrunID, her ürünü benzersiz bir şekilde tanımlar.
- \_StokDurumu, ürünün stok durumunu gösterir.
- \_RafBilgisiID, ürünün hangi rafta bulunduğunu belirtir.
- 10. TEDARIKCI Tablosu (\_TedarikciID, \_FirmaID, \_TedarikciTelNo):
- Markete ürün tedarik eden tedarikçilerin bilgilerini içerir.
- \_TedarikciID, her tedarikçiyi benzersiz bir şekilde tanımlar.
- 11. CALISTIGI\_YER Tablosu (\_kategoriID, \_personelID):
- Personellerin çalıştığı kategorileri belirten ilişkisel tablo.
- 12. URUN\_RAF Tablosu (\_RafID, UrunID):
- Ürünlerin hangi rafta bulunduğunu belirten ilişkisel tablo.

## İlişkiler:

- PERSONEL ve CALISTIGI\_YER tabloları arasında 1 n bir ilişki bulunmaktadır. Bir personel birden fazla kategoride çalışabilir, ancak her kategoride bir personel sorumlu olabilir.
- URUNLISTESI ve KATEGORI tabloları arasında bir 1 n ilişki vardır. Bir kategoriye ait birden fazla ürün olabilir, ancak her ürün bir kategoriye aittir.
- URUN\_RAF tablosu, URUNLISTESI ve RAFBILGISI tabloları arasında birleştirici bir tablo olarak kullanılarak, her ürünün hangi rafta bulunduğunu belirtir.

Bu veri tabanı, market yönetimine bir dizi araç sunarak müşteri memnuniyetini artırmayı, stok yönetimini optimize etmeyi ve personel performansını izlemeyi hedefler. İlişkisel tablolar ve veri modellemesi, veri tabanının etkili bir şekilde kullanılmasını sağlar ve iş süreçlerini daha anlamlı bir hale getirir.

Şimdi projemizde yaptığımız örnek sorgularımıza ve sql kodlarımıza diğer sayfalarda değineceğiz.
TRİGGERSLAR
PERSONEL KART ve PERSONEL Tabloları Arasındaki Tetikleyici:
CREATE TRIGGER trg_personel_kart_guncelleme
ON PERSONEL_KART
AFTER UPDATE
AS
BEGIN
UPDATE PERSONEL
SET Maas = i.Bakiye
FROM PERSONEL INNER JOIN inserted i ON PERSONEL.PersoneIID = i.KartID;
END;
ÜRÜNLİSTESİ ve MARKA Tabloları Arasındaki Tetikleyici:
CREATE TRIGGER trg_urun_ekleme
ON URUNLISTESI
AFTER INSERT
AS
BEGIN
IF NOT EXISTS (SELECT 1 FROM MARKA WHERE UrunID = (SELECT UrunID FROM inserted))
BEGIN
INSERT INTO MARKA (MarkaAdi, MarkaStok, UrunID)
VALUES ('Belirsiz', 0, (SELECT UrunID FROM inserted)):

```
END;
END:
-----
--ÜRÜN_RAF ve RAFBILGISI Tabloları Arasındaki Tetikleyici:
CREATE TRIGGER trg_urun_raf_guncelleme
ON URUN_RAF
AFTER INSERT, DELETE
AS
BEGIN
UPDATE RAFBILGISI
SET Raficerigi =
  CASE
   WHEN EXISTS (SELECT 1 FROM inserted) THEN
   CONCAT(Raficerigi, ', ', (SELECT UrunID FROM inserted))
   ELSE
   REPLACE(Raficerigi, CONCAT(', ', (SELECT UrunID FROM deleted)), ")
  END
WHERE RafID = (SELECT RafID FROM inserted) OR RafID = (SELECT RafID FROM deleted);
END;
--MÜŞTERIKART ve MÜŞTERİ Tabloları Arasındaki Tetikleyici:
CREATE TRIGGER trg_musteri_kart_guncelleme
ON MUSTERIKART
AFTER UPDATE
AS
BEGIN
UPDATE MUSTERI
SET MusteriAdSoyad = CONCAT('Müşteri ', i.MusteriID)
```

```
FROM MUSTERI INNER JOIN inserted i ON MUSTERI.MusteriID = i.MusteriID;
END;
-- indirim ve MARKA Tabloları Arasındaki Tetikleyici:
CREATE TRIGGER trg_indirim_guncelleme
ON INDIRIM
AFTER UPDATE
AS
BEGIN
UPDATE MARKA
SET MarkaStok = MarkaStok - i.IndirimMiktari
FROM MARKA INNER JOIN inserted i ON MARKA.MarkaID = i.UygulananMarkaID;
END;
_____
CREATE TRIGGER trg_UrunGuncelle
ON URUNLISTESI
AFTER UPDATE
AS
 BEGIN
 UPDATE m
  SET MarkaStok = u.StokDurumu
  FROM MARKA m
  INNER JOIN INSERTED i ON m.UrunID = i.UrunID
  INNER JOIN UPDATED u ON i.UrunID = u.UrunID;
END;
-- Tetikleyiciyi tekrar oluşturma
CREATE TRIGGER tr_urunlistesi_after_insert
```

```
ON MARKA
AFTER INSERT
AS
BEGIN
SET NOCOUNT ON;
UPDATE URUNLISTESI
SET MarkalD = i.MarkalD
FROM URUNLISTESI u
INNER JOIN INSERTED i ON u.UrunID = i.UrunID;
END;
-- URUNLISTESI tablosuna ürün eklendiğinde çalışacak trigger
CREATE TRIGGER trg_URUNLISTESI
ON URUNLISTESI
AFTER INSERT
AS
BEGIN
  DECLARE @RafID INT;
  -- Varsayılan bir raf seçimi, örneğin RafID = 1
  SET @RafID = 1;
  -- URUN_RAF tablosuna yeni ürünü ve rafı ekleyen SQL sorgusu
  INSERT INTO URUN_RAF (RafID, UrunID)
  SELECT @RafID, UrunID FROM INSERTED;
END;
---- kategori sorumlu personle verisini girme,
---- rastgele sorumlu ataması yapar
WITH RandomizedPersonel AS (
SELECT
KATEGORI.KategoriID,
```

```
PERSONEL.PersonelAdi,

ROW_NUMBER() OVER (PARTITION BY KATEGORI.KategorilD ORDER BY NEWID()) AS RowNum

FROM

KATEGORI

CROSS JOIN PERSONEL
)

UPDATE KATEGORI

SET KategoriSorumluPersonelAdi = RandomizedPersonel.PersonelAdi

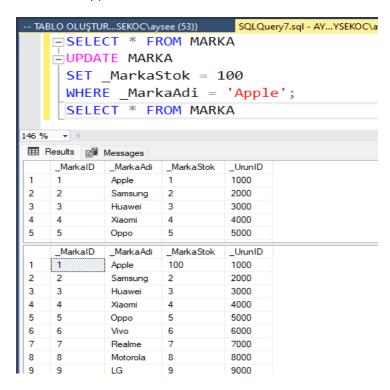
FROM KATEGORI

INNER JOIN RandomizedPersonel ON KATEGORI.KategorilD = RandomizedPersonel.KategorilD

WHERE RandomizedPersonel.RowNum = 1;
```

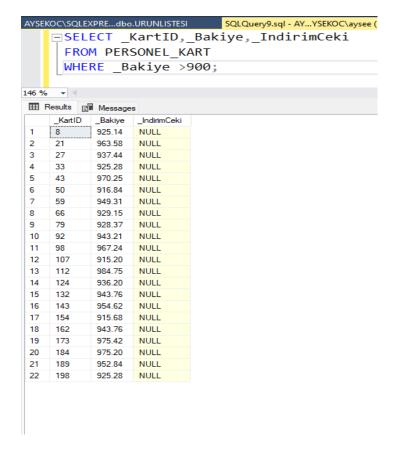
## **UPDATE ORNEK**

Marka adı 'apple' olan verinin marka stok durumunu 100 olarak güncelle



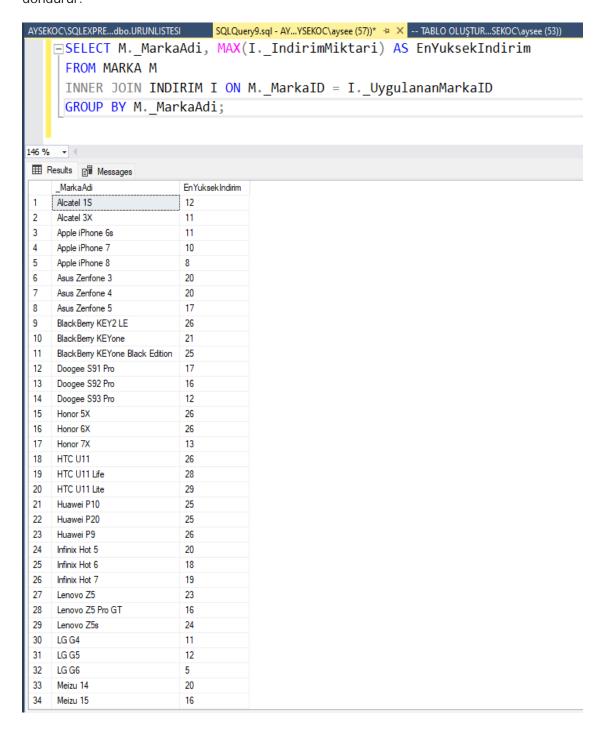
## **SELECT ORNEK**

PERSONEL\_KART tablosundaki \_Bakiye niteliği 900 den büyük olan personellerin listesi



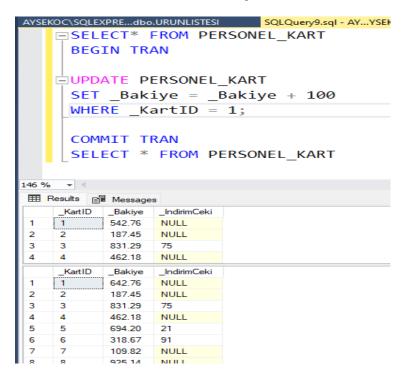
## SUBQUERY ORNEK

MARKA ve INDIRIM tablolarını iç içe birleştirerek, her marka için uygulanan en yüksek indirim miktarını döndürür.

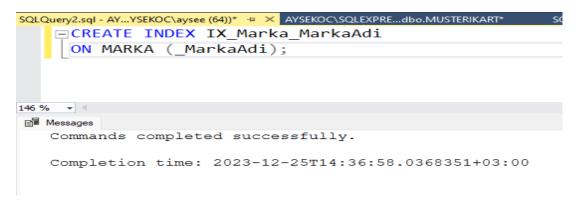


## TRANSACTION ORNEK

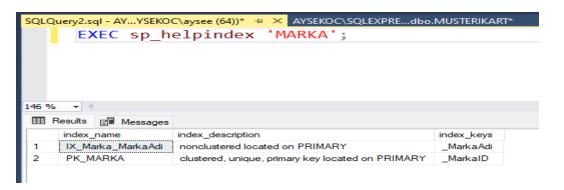
PERSONEL\_KART tablosundaki \_KartID'si '1' olan personelin \_Bakiye durumunu 100 arttırdık. Tablonun önceki ve sonraki durumları gösterdik.



## **INDEX ORNEK**



#### INDEX SORGU ORNEK



#### STORED PROCEDURE ORNEK

belirli bir müşteriye ait alışveriş miktarını günceller, indirim uygular ve bakiyeyi ayarlar

```
SQLQuery2.sql - AY...YSEKOC\aysee (64))* → × AYSEKOC\SQLEXPRE...dbo.MUSTERIKART* SQLQuery1.sql - AY...YSEKOC\aysee (70))*
      -- Örnek Stored Procedure
    □CREATE PROCEDURE SP UPDATE MUSTERIKART
         @MusteriID INT,
          @IndirimMiktari DECIMAL(5, 2),
         @AlisverisMiktari INT
     AS
    BEGIN
          -- Alışveriş miktarını güncelle
         UPDATE MUSTERIKART
         SET YapilanAlisveris = YapilanAlisveris + @AlisverisMiktari;
          -- Indirim uygula
         UPDATE MUSTERIKART
          SET _Indirim = @IndirimMiktari
          WHERE _MusteriID = @MusteriID;
          -- Bakiyeyi güncelle
         UPDATE MUSTERIKART
         SET _Bakiye = _Bakiye - @IndirimMiktari
         WHERE _MusteriID = @MusteriID;
         PRINT 'MUSTERIKART Bilgileri Güncellendi.';
     FND:
146 % 🕶 🖣
   Commands completed successfully.
   Completion time: 2023-12-25T14:06:54.4672428+03:00
```

## STORED PROCEDURE KODU CALISMA SORGUSU

```
SQLQuery2.sql - AY...YSEKOC\aysee (64))* + × AYSEKOC\SQLEXPRE...dbo.MUSTERIKART* SQLQuery1
            -- Bakiyeyi güncelle
           UPDATE MUSTERIKART
           SET _Bakiye = _Bakiye - @IndirimMiktari
           WHERE _MusteriID = @MusteriID;
           PRINT 'MUSTERIKART Bilgileri Güncellendi.';
     --END;
     DECLARE @MusteriID INT = 1;
     DECLARE @IndirimMiktari DECIMAL(5, 2) = 10.00;
     DECLARE @AlisverisMiktari INT = 3;
     EXEC SP_UPDATE_MUSTERIKART
         @MusteriID.
         @IndirimMiktari.
         @AlisverisMiktari;
   (200 rows affected)
   (1 row affected)
   (1 row affected)
   MUSTERIKART Bilgileri Güncellendi.
   Completion time: 2023-12-25T14:10:30.8187377+03:00
```

#### **FUNCTION ORNEK**

Bu SQL fonksiyonu, MusterilD'si 1 olan müşteriye ait müşteri kartındaki bakiyeyi hesaplamak için kullanılır.

## TRIGGER ORNEK

#### TRIGGER OLUSTURMA

```
SQLQuery2.sql - AY...YSEKOC\aysee (64))* → × AYSEKOC\SQLEXPRE...dbo.MUSTERIKART* SQLQuery1.sql - AY...YSEKOC\aysee (70))* -- TABLO OLUŞTUR...SEKOC\aysee
     -- Örnek Trigger
   □CREATE TRIGGER TR_AfterInsertUpdate_Marka
     ON MARKA
     AFTER INSERT, UPDATE
     AS
   ⊟BEGIN
         -- Eklenen veya güncellenen kayıtları kontrol etmek için "inserted" kullanılır
         IF (SELECT COUNT(*) FROM inserted WHERE _MarkaStok IS NOT NULL) > 0
         BEGIN
             -- Eğer MarkaStok güncellenirse, belirli işlemleri gerçekleştir
             PRINT 'Marka Stok Güncellendi. İşlem Yapılıyor...';
             -- Örneğin, bir log tablosuna ekleme yapabiliriz
             INSERT INTO MarkaLog (MarkaID, IslemTipi, IslemTarihi)
             SELECT m._MarkaID, 'UPDATE', GETDATE()
             FROM MARKA m
             INNER JOIN inserted i ON m._MarkaID = i._MarkaID;
         END
         ELSE
         BEGIN
              -- Eğer başka bir alan güncellenirse, burada gerekli işlemleri yapabilirsiniz
             PRINT 'Marka Bilgileri Güncellendi. İşlem Yapılıyor...';
         END
     END;
146 % +
   Commands completed successfully.
   Completion time: 2023-12-25T13:48:28.2503873+03:00
```

## TRIGGER SORGULAMA KODU

```
SQLQuery2.sql - AY...YSEKOC\aysee (64))* ⇒ × AYSEKOC\SQLEXPRE...dbo.MUSTERIKART* SQLQuery1.sql - AY...YSEKOC\aysee (70))*
                                                                                                              -- TABLO OLUŞTUR...SEKOC\aysee (52))
     =SELECT *
      FROM sys.triggers
       WHERE name = 'TR AfterInsertUpdate Marka';
146 % 🕶 🕙
Results Messages
                           object_id
                                    parent_class parent_class_desc
                                                                 parent_id type type_desc
                                                                                              create_date
                                                                                                                  modify_date
                                                                                                                                     is_ms_shipped is_disabled is_not_for_replication is_instead_of_trigger
    TR_AfterinsertUpdate_Marka 1845581613 1 OBJECT_OR_COLUMN 1557580587 TR SQL_TRIGGER 2023-12-25 13-48-28-247 2023-12-25 13-48-28-247 0
                                                                                                                                               0
                                                                                                                                                       0
```

## **VIEW ORNEK**

#### **VIEW SORGULAMA ORNEK**

```
SQLQuery2.sql - AY...YSEKOC\aysee (64))* * X AYSEKOC\SQLEXPRE...dbo.MUSTERIKART*

--CREATE VIEW VW_MUSTERIKART_BILGILERI
                                                                                                                         SQLQuery1.sql - AY...YSEKOC\aysee (70))*
                                                                                                                                                                                       -- TABLO OLUŞTUR...SEKOC\aysee (52))
          --AS
          --SELECT
                       _MusteriID,
                        _Indirim,
                        _Bakiye,
                       ____YapilanAlisveris,
'Musteri_' + CAST(_MusteriID AS VARCHAR) AS MusteriAdi -- Örnek bir türetilmiş sütun
          --FROM
                       MUSTERIKART;
          SELECT * FROM VW MUSTERIKART BILGILERI;
Results Messages
         _MusteriID _Indirim
                                                _Yapilan Alisveris
503
                                                                      MusteriAdi
                                                                      Musteri_1
Musteri_2
Musteri_3
Musteri_4
                        10
                                    300
                                                1503
                                    400
                                                2003
                                                                      Musteri_4
Musteri_5
Musteri_6
Musteri_7
Musteri_8
Musteri_9
Musteri_10
                                   500
600
                                    700
                                                3503
                                   800
900
1000
                                                4003
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
                                                5003
                                                                      Musteri_11
Musteri_12
Musteri_13
Musteri_14
                                    1100
                                                5503
        11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
                                    1200
1300
1400
                                                6003
6503
                                                7003
                                                                      Musteri_15
Musteri_16
Musteri_17
Musteri_18
                                    1500
                                                7503
                                    1600
1700
1800
                                                8003
8503
                                                9003
                                   1900
2000
2100
                                                9503
10003
10503
                                                                      Musteri_19
Musteri_20
Musteri_21
                                                                      Musteri_22
Musteri_23
Musteri_24
                                    2200
                                                11003
                                   2300
2400
                                                11503
12003
                                    2500
                                                12503
                                                                      Musteri_25
```