

--ÖNCE ANA TABLOLARI OLUŞTURUYORUZ:

```
CREATE TABLE MUSTERI (  
    MusteriID INT PRIMARY KEY,  
    Sifre NVARCHAR(50),  
    Ad NVARCHAR(50),  
    Soyad NVARCHAR(50),  
    Telefon NVARCHAR(15),  
    Adres NVARCHAR(255),  
    Email NVARCHAR(100)  
);  
  
CREATE TABLE MUZAYEDE (  
    MuzayedeID INT PRIMARY KEY,  
    Tarih DATE,  
    BaslangicFiyati DECIMAL(10, 2),  
    Durum NVARCHAR(50)  
);  
  
CREATE TABLE TEKLIF (  
    TeklifID INT PRIMARY KEY,  
    TeklifFiyati DECIMAL(10, 2),  
    MusteriID INT,  
    MuzayedeID INT,  
    TeklifDurumu NVARCHAR(50),  
    FOREIGN KEY (MusteriID) REFERENCES MUSTERI(MusteriID),  
    FOREIGN KEY (MuzayedeID) REFERENCES MUZAYEDE(MuzayedeID)  
);  
  
CREATE TABLE Odeme (  
    OdemeID INT PRIMARY KEY,  
    OdemeTutari DECIMAL(10, 2),  
    OdemeTarihi DATE,  
    OdemeDurumu NVARCHAR(50)  
);  
  
CREATE TABLE SANATCI (  
    SanatciID INT PRIMARY KEY,  
    Ad NVARCHAR(50),  
    Soyad NVARCHAR(50),  
    OlumYili INT  
);  
  
CREATE TABLE KATEGORI (  
    KategoriID INT PRIMARY KEY,  
    KategoriAdi NVARCHAR(50)  
);  
  
CREATE TABLE RESIM2 (  
    ResimID INT PRIMARY KEY,  
    ResimAdi NVARCHAR(100),  
    StokSayisi INT,  
    SatisFiyati DECIMAL(10, 2),  
    Yil INT  
);
```

--BCNF FORMATINA UYGUNLUK İÇİN YENİ TABLOLARIMIZI VE İLİŞKİLERİMİZİ DE OLUŞTURUYORUZ:

```
CREATE TABLE RESIM_KATEGORI (  
    ResimID INT,  
    KategoriID INT,  
    PRIMARY KEY (ResimID, KategoriID),  
    FOREIGN KEY (ResimID) REFERENCES RESIM2(ResimID),  
    FOREIGN KEY (KategoriID) REFERENCES KATEGORI(KategoriID)  
);
```

-- Resim-Sanatçı ilişkisi

```
CREATE TABLE RESIM_SANATCI (  
    ResimID INT,  
    SanatciID INT,  
    PRIMARY KEY (ResimID, SanatciID),  
    FOREIGN KEY (ResimID) REFERENCES RESIM2(ResimID),  
    FOREIGN KEY (SanatciID) REFERENCES SANATCI(SanatciID)  
);
```

-- Müşteri-Resim ilişkisi

```
CREATE TABLE Alir (  
    ResimID INT,  
    MusteriID INT,  
    PRIMARY KEY (ResimID, MusteriID),  
    FOREIGN KEY (ResimID) REFERENCES RESIM2(ResimID),  
    FOREIGN KEY (MusteriID) REFERENCES MUSTERI(MusteriID)  
);
```

-- Müzayede-Resim ilişkisi

```
CREATE TABLE Icerir (  
    ResimID INT,  
    MuzayedeID INT,  
    PRIMARY KEY (ResimID, MuzayedeID),  
    FOREIGN KEY (ResimID) REFERENCES RESIM2(ResimID),  
    FOREIGN KEY (MuzayedeID) REFERENCES MUZAYEDE(MuzayedeID)  
);
```

-- Müzayede-Müşteri ilişkisi

```
CREATE TABLE Katilma (  
    MuzayedeID INT,  
    MusteriID INT,  
    PRIMARY KEY (MusteriID, MuzayedeID),  
    FOREIGN KEY (MusteriID) REFERENCES MUSTERI(MusteriID),  
    FOREIGN KEY (MuzayedeID) REFERENCES MUZAYEDE(MuzayedeID)  
);
```

-- Resim-Satış Durumu Tablosu

```
CREATE TABLE RESIM_SATIS_DURUMU (  
    ResimID INT PRIMARY KEY,  
    SatisTuru NVARCHAR(50),  
    FOREIGN KEY (ResimID) REFERENCES RESIM2(ResimID)  
);
```

--BCNF FORMATINA UYGUN BİR ŞEKİLDE TABLOLARIMIZI VE İLİŞKİLERİMİZİ OLUŞTURDUK.

--PROSEDÜR VE TRIGGER OLUŞTURMA:

-- Stok Güncelleme Prosedürü

```
CREATE PROCEDURE StokGuncelleme
    @ResimID INT,
    @YeniStok INT
AS
BEGIN
    UPDATE RESIM2
    SET StokSayisi = @YeniStok
    WHERE ResimID = @ResimID;
END;
```

-- Ödeme Durumu Güncelleme Prosedürü

```
CREATE PROCEDURE OdemeDurumuGuncelle
    @OdemeID INT,
    @YeniDurum NVARCHAR(50)
AS
BEGIN
    UPDATE Odeme
    SET OdemeDurumu = @YeniDurum
    WHERE OdemeID = @OdemeID;
END;
```

-- Resim Satışını İzleyen Trigger

```
CREATE TRIGGER StokAzaltma
ON Alir
AFTER INSERT
AS
BEGIN
    DECLARE @ResimID INT;
    SELECT @ResimID = ResimID FROM INSERTED;
    UPDATE RESIM2
    SET StokSayisi = StokSayisi - 1 --Burada stok sayısını 1 azaltma işlemi gerçekleşiyor.
    WHERE ResimID = @ResimID;
END;
```

-- Ödeme Durumu Değişikliğini İzleyen Trigger

```
CREATE TRIGGER OdemeDurumuDegistir
ON Odeme
AFTER UPDATE
AS
BEGIN
    DECLARE @OdemeID INT;
    DECLARE @YeniDurum NVARCHAR(50);
    SELECT @OdemeID = OdemeID, @YeniDurum = OdemeDurumu FROM INSERTED;
    IF @YeniDurum = 'Başarılı'
    BEGIN
        PRINT 'Ödeme başarılı, işlem tamamlandı!';
    END;
END;
```

--TRANSACTION:

-- Transaction Yönetimi: Sipariş Verme Prosedürü

```
CREATE PROCEDURE SiparisVer
    @MusteriID INT,
    @ResimID INT,
    @OdemeTutari DECIMAL(10, 2)
AS
BEGIN
    BEGIN TRANSACTION;
    BEGIN TRY
        UPDATE RESIM2
        SET StokSayisi = StokSayisi - 1
        WHERE ResimID = @ResimID;

        IF (SELECT StokSayisi FROM RESIM2 WHERE ResimID = @ResimID) < 0
        BEGIN
            THROW 50001, 'Yetersiz stok!', 1;
        END

        INSERT INTO Odeme (OdemeTutari, OdemeTarihi, OdemeDurumu)
        VALUES (@OdemeTutari, GETDATE(), 'Bekliyor');

        INSERT INTO Alir (ResimID, MusteriID)
        VALUES (@ResimID, @MusteriID);

        COMMIT;
        PRINT 'Sipariş başarıyla tamamlandı.';
    END TRY
    BEGIN CATCH
        ROLLBACK;
        PRINT 'Hata oluştu, işlem tamamlanamadı: ' + ERROR_MESSAGE();
    END CATCH;
END;
```

-- Müşteri Siparişlerini Listeleyen Prosedür

```
CREATE PROCEDURE MusteriSiparisleri
    @MusteriID INT
AS
BEGIN
    SELECT A.ResimID, R.ResimAdi, R.SatisFiyati, O.OdemeDurumu, O.OdemeTarihi
    FROM Alir A
    JOIN RESIM2 R ON A.ResimID = R.ResimID
    JOIN Odeme O ON O.OdemeID = A.ResimID
    WHERE A.MusteriID = @MusteriID;
END;
```

-- Ödeme Süresi Kontrol Prosedürü

```
CREATE PROCEDURE OdemeSureKontrol
    @MusteriID INT,
    @ResimID INT
AS
BEGIN
    DECLARE @MuzayedeID INT;
    DECLARE @OdemeTarihi DATE;
```

```

DECLARE @SureFarki INT;

SELECT TOP 1 @MuzayedeID = MuzayedeID
FROM Icerir
WHERE ResimID = @ResimID;

SELECT TOP 1 @OdemeTarihi = OdemeTarihi
FROM Odeme
WHERE OdemeID IN (
    SELECT TeklifID
    FROM TEKLIF
    WHERE MusteriID = @MusteriID
    AND MuzayedeID = @MuzayedeID
);

SET @SureFarki = DATEDIFF(DAY, @OdemeTarihi, GETDATE());

IF @SureFarki <= 3 --EĞER ÖDEME SÜRESİ 3 GÜNÜ GEÇTİYSE İŞLEM ASKIYA ALINIR
BEGIN
    PRINT 'Ödeme zamanında yapılmış.';
END
ELSE
BEGIN
    PRINT 'Ödeme gecikti!';
END;
END;

--EN YÜKSEK TEKLİFİ BELİRLEYİP O MÜŞTERİYE SATIŞ YAPAN PROSEDÜR
CREATE PROCEDURE EnYuksekTeklifIsleme
    @MuzayedeID INT
AS
BEGIN
    BEGIN TRANSACTION;

    BEGIN TRY
        DECLARE @EnYuksekTeklif DECIMAL(10, 2);
        DECLARE @MusteriID INT;
        DECLARE @ResimID INT;

        SELECT TOP 1
            @EnYuksekTeklif = TeklifFiyati,
            @MusteriID = MusteriID,
            @ResimID = (SELECT TOP 1 ResimID FROM Icerir WHERE MuzayedeID = @MuzayedeID)
        FROM TEKLIF
        WHERE MuzayedeID = @MuzayedeID
        ORDER BY TeklifFiyati DESC;

        IF @EnYuksekTeklif IS NOT NULL AND @MusteriID IS NOT NULL AND @ResimID IS NOT NULL
        BEGIN
            INSERT INTO Alir (ResimID, MusteriID)
            VALUES (@ResimID, @MusteriID);

            UPDATE RESIM2
            SET StokSayisi = StokSayisi - 1
            WHERE ResimID = @ResimID;
        END
    END TRY
    BEGIN CATCH
        ROLLBACK;
    END CATCH
END

```

```

UPDATE MUZAYEDE
SET Durum = 'Tamamlandı'
WHERE MuzayedeID = @MuzayedeID;

PRINT 'En yüksek teklif başarıyla işleme alındı.';
END
ELSE
BEGIN
    PRINT 'Geçerli bir teklif bulunamadı veya işlem gerçekleştirilemedi.';
END

COMMIT;
END TRY
BEGIN CATCH
    ROLLBACK;
    PRINT 'Hata oluştu: ' + ERROR_MESSAGE();
END CATCH;
END;

```

--VERİLERİ TABLOLARA EKLEME:

```

INSERT INTO MUSTERI (MusteriID, Sifre, Ad, Soyad, Telefon, Adres, Email)
VALUES
(1, '123456', 'Ahmet', 'Yılmaz', '5551234567', 'Ankara', 'ahmet@gmail.com'),
(2, '234567', 'Mehmet', 'Kaya', '5552345678', 'İstanbul', 'mehmet@gmail.com'),
(3, '345678', 'Ayşe', 'Demir', '5553456789', 'İzmir', 'ayse@gmail.com'),
(4, '456789', 'Fatma', 'Şahin', '5554567890', 'Bursa', 'fatma@gmail.com'),
(5, '567890', 'Ali', 'Çelik', '5555678901', 'Antalya', 'ali@gmail.com'),
(6, '123454', 'Can', 'Aydın', '5556789012', 'Trabzon', 'can@gmail.com'),
(7, '234563', 'Emre', 'Uzun', '5557890123', 'Mersin', 'emre@gmail.com'),
(8, 'hello123', 'Zeynep', 'Kara', '5558901234', 'Adana', 'zeynep@gmail.com'),
(9, 'world123', 'Hakan', 'Koç', '5559012345', 'Gaziantep', 'hakan@gmail.com'),
(10, 'secure1', 'Merve', 'Yıldız', '5550123456', 'Konya', 'merve@gmail.com'),
(11, 'LGGFLdf', 'Burak', 'Polat', '5551239876', 'Eskişehir', 'burak@gmail.com'),
(12, 's56466', 'Deniz', 'Çakır', '5552348765', 'Kayseri', 'deniz@gmail.com'),
(13, '745676', 'Ece', 'Öztürk', '5553457654', 'Samsun', 'ece@gmail.com'),
(14, 'secure5', 'Gökhan', 'Ak', '5554566543', 'Malatya', 'gokhan@gmail.com'),
(15, 'secure6', 'Hande', 'Eren', '5555675432', 'Balıkesir', 'hande@gmail.com'),
(16, 'secure7', 'İsmail', 'Bulut', '5556784321', 'Manisa', 'ismail@gmail.com'),
(17, 'secure8', 'Leyla', 'Güneş', '5557893210', 'Denizli', 'leyla@gmail.com'),
(18, 'secure9', 'Okan', 'Arslan', '5558902109', 'Kocaeli', 'okan@gmail.com'),
(19, 'secure10', 'Pelin', 'Tan', '5559011098', 'Hatay', 'pelin@gmail.com'),
(20, 'secure11', 'Seda', 'Ece', '5550120987', 'Muğla', 'seda@gmail.com');

```

```

INSERT INTO SANATCI (SanatciID, Ad, Soyad, OlumYili)
VALUES
(1, 'Leonardo', 'da Vinci', 1519),
(2, 'Vincent', 'van Gogh', 1890),
(3, 'Pablo', 'Picasso', 1973),

```

```
(4, 'Salvador', 'Dali', 1989),
(5, 'Claude', 'Monet', 1926),
(6, 'Edvard', 'Munch', 1944),
(7, 'Frida', 'Kahlo', 1954),
(8, 'Andy', 'Warhol', 1987),
(9, 'Jackson', 'Pollock', 1956),
(10, 'Henri', 'Matisse', 1954),
(11, 'Paul', 'Cezanne', 1906),
(12, 'Gustav', 'Klimt', 1918),
(13, 'Joan', 'Miro', 1983),
(14, 'Georges', 'Seurat', 1891),
(15, 'Wassily', 'Kandinsky', 1944),
(16, 'Marc', 'Chagall', 1985),
(17, 'Caravaggio', 'Merisi', 1610),
(18, 'Raphael', 'Sanzio', 1520),
(19, 'Titian', 'Vecelli', 1576),
(20, 'Rembrandt', 'van Rijn', 1669);
```

```
INSERT INTO KATEGORI (KategoriID, KategoriAdi)
VALUES
```

```
(1, 'Rönesans'),
(2, 'Barok'),
(3, 'Empresyonizm'),
(4, 'Ekspresyonizm'),
(5, 'Modernizm'),
(6, 'Kübizm'),
(7, 'Sürrealizm'),
(8, 'Pop Art'),
(9, 'Abstract Art'),
(10, 'Romantizm'),
(11, 'Neo-Klasik'),
(12, 'Fütürizm'),
(13, 'Dadaizm'),
(14, 'Minimalizm'),
(15, 'Post-Modernizm'),
(16, 'Realizm'),
(17, 'Simgecilik'),
(18, 'Gotik'),
(19, 'Ortaçağ'),
(20, 'Neo-Gotik');
```

```
INSERT INTO RESIM2 (ResimID, ResimAdi, StokSayisi, SatisFiyati, Yil)
VALUES
```

```
(1, 'Yıldızlı Gece', 5, 100000.00, 1889),
(2, 'Mona Lisa', 2, 200000.00, 1503),
(3, 'Guernica', 4, 150000.00, 1937),
(4, 'The Persistence of Memory', 3, 120000.00, 1931),
(5, 'The Scream', 1, 80000.00, 1893),
(6, 'The Kiss', 5, 250000.00, 1907),
(7, 'Girl with a Pearl Earring', 4, 95000.00, 1665),
(8, 'The Night Watch', 2, 180000.00, 1642),
(9, 'Las Meninas', 3, 170000.00, 1656),
(10, 'The Creation of Adam', 1, 300000.00, 1512),
(11, 'Çığlık', 3, 110000.00, 1893),
(12, 'Kadın Portresi', 4, 130000.00, 1910),
(13, 'Yemek Masası', 5, 160000.00, 1925),
```

```

(14, 'İçsel Yansıma', 2, 140000.00, 1930),
(15, 'Büyük Dalga', 1, 90000.00, 1850),
(16, 'Hayalet', 3, 135000.00, 1945),
(17, 'Bahçe Yolu', 4, 125000.00, 1918),
(18, 'Kırmızı Gözler', 2, 115000.00, 2000),
(19, 'Gizli İmaj', 5, 250000.00, 1967),
(20, 'Renkli Ev', 3, 145000.00, 1980);

-- Test İçin Gerekli Veriler

-- Stokta sadece 1 adet resim olduğunu varsayalım
UPDATE RESIM2 SET StokSayisi = 1 WHERE ResimID = 1;

-- Başarılı İşlem: Yeterli stok var
EXEC SiparisVer @MusteriID = 1, @ResimID = 1, @OdemeTutari = 100.50;

-- Başarılı işlem sonucu kontrol
SELECT * FROM RESIM2 WHERE ResimID = 1; -- StokSayisi 0 olmalı
SELECT * FROM Alir WHERE MusteriID = 1; -- Yeni sipariş kaydı olmalı
SELECT * FROM Odeme WHERE OdemeTutari = 100.50; -- Ödeme kaydı olmalı

-- Hatalı İşlem: Yetersiz stok durumu
-- Stokta 0 olan bir ürün için sipariş vermeyi deneyelim
EXEC SiparisVer @MusteriID = 2, @ResimID = 1, @OdemeTutari = 200.00;

-- Hatalı işlem sonucu kontrol
SELECT * FROM RESIM2 WHERE ResimID = 1; -- StokSayisi değişmemeli (0)
SELECT * FROM Alir WHERE MusteriID = 2; -- Yeni sipariş kaydı olmamalı
SELECT * FROM Odeme WHERE OdemeTutari = 200.00; -- Yeni ödeme kaydı olmamalı

-- Simüle edilen hata: Stok yetersiz olduğunda hata verdirmek
UPDATE RESIM2 SET StokSayisi = 0 WHERE ResimID = 2;

-- Prosedürü çalıştır
EXEC SiparisVer @MusteriID = 3, @ResimID = 2, @OdemeTutari = 150.00;

-- Hata sonucu tabloları kontrol et
SELECT * FROM RESIM2 WHERE ResimID = 2; -- StokSayisi değişmemeli
SELECT * FROM Alir WHERE MusteriID = 3; -- Sipariş kaydı olmamalı
SELECT * FROM Odeme WHERE OdemeTutari = 150.00; -- Ödeme kaydı olmamalı

EXEC StokGuncelleme @ResimID = 2, @YeniStok = 10;
EXEC OdemeDurumuGuncelle @OdemeID = 1, @YeniDurum = 'Başarılı'; -- ÖdemeID=1 olan ödemeyi
'Başarılı' olarak günceller.
EXEC OdemeSureKontrol @MusteriID = 10, @ResimID = 1; -- MüşteriID=101 ve ResimID=1 olan
resim için ödeme süresini kontrol eder.
SELECT *
FROM sys.triggers;
SELECT *
FROM sys.objects
WHERE type = 'P'; -- Prosedürleri listeler

```