

FIRAT ÜNİVERSİTESİ VERİTABANI DERSİ PROJE RAPORU

RESİM MÜZESİ MÜZAYEDE OTOMASYONU

HAZAN YÜCEL,ZEYNEP N. SİVRİ,HANIM İLAYDA ÖZGÜN

220260042

220260024

220260070

PROJENİN AMACI:

Projemiz müşterilerin katıldığı müzayedelerden satın aldığı resim veya resimlerin online sipariş oluşturularak müşteriye ulaştırılmasını sağlayan bir platform tasarlamayı hedeflemektedir. Platform da sadece müzayede de açık artırmaya çıkan resimler bulunmayacaktır. Stok miktarı birden fazla olan resimleri de içerir. Bu proje müzayede etkinliğinin daha geniş kitlelere tanıtılmasını sağlayarak koleksiyoncular ve sanatseverler arasında daha güçlü bir bağ kurulmasına olanak tanır. Müzayede etkinliğine katılan insanların sosyalleşmesini, resim sanatı hakkında konuşmalarını, resim zevklerini tartışarak yorum yapmalarını sağlar. Müzayededen alınan resimler 3 gün içinde eğer ödemesi yapılmazsa askıya alınma işlemi yapılır. Askıya alınan resim müzayede de tekrar açık artırmaya çıkarılır.

1.Müşteriler Tablosu: Bir müşteri müzayeden veya standart üretimle piyasaya sunulan resimlerden birden fazla alabilir.

İlişkiler:

- Müzayede tablosu ile N:N
- Resim tablosu ile N:N
- Teklif tablosu ile 1:N
- Ödeme tablosu ile 1:1

Kısıtlamalar:

- MusteriID: Her müşteriye özgü kimlik numarası (Primary Key).
- Ad: Müşteri adı, boş olamaz.
- Soyad: Müşteri soyadı, boş olamaz.
- Telefon: Müşteri telefon numarası, benzersiz olmalı ve boş olamaz.
- Email: E-posta adresi, benzersiz olmalı ve boş olamaz.
- Adres: Müşterinin adres bilgisi.
- Şifre: Müşterinin kullandığı şifre, benzersiz olmalı ve boş olamaz.

Müşteri Tablosu Oluşturma Kodu:

```
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)  
);
```

2.Sanatçı Tablosu:

İlişkiler:

- Resim tablosu ile 1:N

Kısıtlamalar:

- SanatciID: Her sanatçıya özgü kimlik numarası (Primary Key).
- OlumYili: Sanatçının öldüğü yıl.
- Ad: Sanatçının adı, boş olamaz.
- Soyad: Sanatçının soyadı, boş olamaz.

Sanatçı Tablosu Oluşturma Kodu:

```
CREATE TABLE SANATCI (  
    SanatciID INT PRIMARY KEY,  
    Ad NVARCHAR(50),  
    Soyad NVARCHAR(50),  
    OlumYili INT  
);
```

3.Resim Tablosu: Bir resim müzayede etkinliğinde açık artırmaya çıkarılmışsa sadece bir müşteri tarafından satın alınabilir. Eğer standart üretimle piyasaya sunulan bir resimse bir çok müşteri tarafında satın alınabilir.

İlişkiler:

- Müzayede tablosu ile N:N
- Müşteri tablosu ile N:N
- Sanatçı tablosu ile N:1
- Kategori tablosu ile N:1

Kısıtlamalar:

- ResimID: Her resme özgü kimlik numarası.(Primary Key)
- SanatciID: Her resim bir sanatçı kimlik numarasına sahip olmalı (Foreign Key).
- KategoriID: Her resim bir kategoriye aittir ve kimlik numarasına sahip olmalı (Foreign Key).
- ResimAdi: Resmin adı, boş olamaz.
- StokSayisi: Seri üretimde olan resimlerin stok sayısı, boş olamaz.
- SatisSayisi: Resimlerin fiyatı, boş olamaz.
- Yil: Resimlerin hangi yılda yapıldığı, boş olamaz.

Resim Tablosu Oluşturma Kodu:

```
CREATE TABLE RESIM2 (  
    ResimID INT PRIMARY KEY,  
    ResimAdi NVARCHAR(100),  
    StokSayisi INT,  
    SatisFiyati DECIMAL(10, 2),  
    Yil INT  
);
```

4.Müzayede Tablosu

İlişkiler:

- Müşteri tablosu ile N:N
- Resim tablosu ile N:N
- Teklif tablosu ile 1:N

Nitelikler:

- MuzayedeID: Her müzayede kimlik numarasına sahip olmalı (Primary Key).
- Tarih: Müzayedenin olduğu tarih, boş olamaz.
- BaslangicFiyat: Müzayede de açılışta söylenen fiyat, boş olamaz.
- Durum: Müzayedeye çıkarılan ürünün satılma durumu, boş olamaz.

Müzayede Tablosu Oluşturma Kodu:

```
CREATE TABLE MUZAYEDE (  
    MuzayedeID INT PRIMARY KEY,  
    Tarih DATE,  
    BaslangicFiyati DECIMAL(10, 2),  
    Durum NVARCHAR(50)  
);
```

5.Teklif Tablosu: Her müzayede etkinliğinde açık artırmaya çıkarılan resme birden fazla teklif gelebilir.

İlişkiler:

- Müzayede tablosu ile 1:N
- Müşteri tablosu ile N:1

Nitelikler:

- TeklifID: Her teklifin kimlik numarası olmalı (Primary Key).
- MuzayedeID: Her teklifin müzayede kimlik numarası olmalı (Foreign Key).
- MüşteriID: Her teklifin müşteri kimlik numarası olmalı (Foreign Key).
- TeklifFiyati: Müzayedeye çıkan ürün için teklif fiyatı.

Teklif Tablosu Oluşturma Kodu:

```
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)  
);
```

6. Ödeme Tablosu

İlişkiler:

- Müşteri tablosu ile 1:1

Nitelikler:

- OdemeID: Her ödemenin kimlik numarası olmalı (Primary Key).
- OdemeTutari: Alınan ürün veya ürünlerin toplam fiyatı.
- OdemeTarihi: Ödeme yapılacak tarih.
- OdemeDurumu: Ödemenin yapılma durumu.

Ödeme Tablosu Oluşturma Kodu:

```
CREATE TABLE Odeme (  
    OdemeID INT PRIMARY KEY,  
    OdemeTutari DECIMAL(10, 2),  
    OdemeTarihi DATE,  
    OdemeDurumu NVARCHAR(50)  
);
```





Varlık 1	Varlık 2	İlişki Türü
Müşteri	Müzayede	Katılma (N:N)
Müşteri	Ödeme	Alım İşlemleri (1:1)
Müşteri	Resim	Alır (N:N)
Müşteri	Teklif	Sunar (1:N)
Resim	Müzayede	İçerir (N:N)
Resim	Kategori	Sınıflandırma (N:1)
Resim	Sanatçı	Çizer (N:1)
Müzayede	Teklif	Açık Artırma (N:1)

--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)  
);
```

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

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

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

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

```
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:

-- Resim satışı olduğunda stok azaltma ya da yeni stok geldiğinde stok arttırma yapan prosedür.

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

-- Satışı yapılan resmin ödenip ödenmeme durumunu gerçekleştiren prosedür.

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

-- Resim satın alındığında stok miktarını birer azaltan 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
    gerçekleştiriyor.
    WHERE ResimID = @ResimID;
END;
```

-- Ödeme yapıldığında başarılı olduysa 'Ödeme başarılı, işlem tamamlandı!' bilgisini veren 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 Listeleme Prosedürü

```
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! Sipariş askıya alındı!';
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;

```

```

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, 'Çılgılık', 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 SENARYOLARI:
```

```
EXEC StokGuncelleme @ResimID = 1, @YeniStok = 4;
```

```
SELECT ResimID, StokSayisi
```

```
FROM RESIM2
WHERE ResimID = 1;
```

	ResimID	ResimAdi	StokSayisi	SatisFiyati	Yil
▶	1	Yıldızlı Gece	4	100000,00	1889

Results		Messages	
	ResimID	StokSayisi	
1	1	4	

```
INSERT INTO Odeme (OdemeID, OdemeTutari, OdemeTarihi, OdemeDurumu)
VALUES (1, 1000.00, '2025-01-12', 'Bekliyor');
```

```
EXEC OdemeDurumuGuncelle @OdemeID = 1, @YeniDurum = 'Başarılı';
```

```
SELECT OdemeDurumu
FROM Odeme
WHERE OdemeID = 1;
```

	OdemeID	OdemeTut...	OdemeTarihi	OdemeDur...
▶	1	1000,00	2025-01-12	Başarılı

Ödeme başarılı, işlem tamamlandı

```
INSERT INTO Alir (ResimID, MusteriID)
VALUES (2, 2);
```

```
SELECT ResimID, StokSayisi
FROM RESIM2
WHERE ResimID = 2;
```

	ResimID	StokSayisi
1	2	9

```
EXEC SiparisVer @MusteriID = 1, @ResimID = 2, @OdemeTutari = 5000.00;
```

```
SELECT StokSayisi
FROM RESIM2
WHERE ResimID = 2;
```

```
SELECT *
FROM Odeme;
```

```
SELECT *
FROM Alir;
```

Results Messages

	StokSayisi
1	9

	OdemeID	OdemeTutari	OdemeTarihi	OdemeDurumu
1	1	1000.00	2025-01-12	Başarılı

	ResimID	MusteriID
1	1	1
2	2	2
3	4	1