



Mühendislik Fakültesi Bilgisayar Mühendisliği

BMÜ329 VERİ TABANI SİSTEMLERİ

BMÜ329 Veri Tabanı Sistemleri Dersi Dönem Projesi

Gereksinimleri ve E-R Diyagramı

Proje Başlığı: Sanat Eserleri ve Müzeler Yönetim Sistemi

Proje Ekibindeki kişiler: Muhammed Ali Ayas, Üzeyir Baysal

1- Proje Gereksinimleri

Sanat Eserleri ve Müzeler Veritabanı Projesi

Bu proje, müzelerde sergilenen sanat eserlerini, müze bilgilerini, ziyaretçilerin etkileşimlerini, etkinlikleri, koleksiyonları ve müze içi yönetimi detaylı bir şekilde yönetmek amacıyla oluşturulmuş bir veritabanı sistemidir. Proje, sanat eserleri ve koleksiyonlar ile müze içi görevler ve ziyaretçi etkileşimleri gibi birçok bileşeni kapsamaktadır.

İçindekiler

- Proje Özeti
- Veritabanı Yapısı
 - Varlıklar ve Nitelikleri
 - İlişkiler
 - Yorumlama ve Puanlama İlişkisi

Proje Özeti

Sanat Eserleri ve Müzeler Veritabanı, aşağıdaki kullanıcı türlerini ve işlevleri destekler:

- **Ziyaretçiler (Sanatseverler):** Sanat eserlerini görüntüleyebilir, yorum yapabilir, puan verebilir, müzeye ziyaretlerini kaydedebilir ve müze biletlerini satın alabilir.
- **Müze Yöneticileri:** Sanat eserleri, sanatçılar, koleksiyonlar, etkinlikler ve müze çalışanları hakkında bilgi ekleyebilir ve düzenleyebilir.
- **Sanatçılar:** Eser bilgilerini sisteme ekleyebilir ve güncelleyebilir.

Bu veritabanı modeli; müzeler, sanat eserleri, sanatçılar, ziyaretçiler, ziyaretler, koleksiyonlar, etkinlikler ve sponsorluklar gibi birçok varlığı kapsamaktadır.

Veritabanı Yapısı

Varlıklar ve Nitelikleri

1. **Müze**
 - **muze_id** (PK): Müzenin benzersiz kimlik numarası.

- **ad:** Müzenin adı.
- **adres:** Müzenin bulunduğu adres.
- **ziyaret_saatleri:** Müzenin ziyaret saatleri.
- **giris_ucreti:** Müze giriş ücreti.

2. Sanat Eseri

- **eser_id (PK):** Sanat eserin benzersiz kimlik numarası.
- **ad:** Eserin adı.
- **tur_id (FK):** Eserin türünü işaret eden yabancı anahtar.
- **yaratilis_tarihi:** Eserin yaratılış tarihi.
- **sanatci_id (FK):** Eserin sanatçısını işaret eden yabancı anahtar.
- **muze_id (FK):** Eserin sergilendiği müzeyi işaret eden yabancı anahtar.
- **koleksiyon_id (FK):** Eserin bir koleksiyona atanması durumunda koleksiyon kimliği.

3. Sanatçı

- **sanatci_id (PK):** Sanatçının benzersiz kimlik numarası.
- **ad:** Sanatçının adı.
- **dogum_tarihi:** Sanatçının doğum tarihi.
- **biyografi:** Sanatçı hakkında kısa bilgi.

4. Ziyaretçi

- **ziyaretci_id (PK):** Ziyaretçinin benzersiz kimlik numarası.
- **isim:** Ziyaretçinin adı.
- **email:** Ziyaretçinin e-posta adresi (tekil ve zorunludur).
- **parola:** Ziyaretçinin parola bilgisi (şifrelenmiş olarak saklanmalıdır).

5. Eser Türü

- **tur_id (PK):** Eser türünün benzersiz kimlik numarası.
- **ad:** Eser türünün adı (ör. Resim, Heykel, Fotoğraf, Dijital).

6. Personel

- **calisan_id (PK):** Çalışanın benzersiz kimlik numarası.
- **ad:** Çalışanın adı.
- **gorev:** Çalışanın görev tanımı.
- **muze_id (FK):** Çalışanın görev yaptığı müzeyi işaret eden yabancı anahtar.

7. Bilet

- **bilet_id (PK):** Biletin benzersiz kimlik numarası.
- **ziyaretci_id (FK):** Bileti alan ziyaretçiyi işaret eden yabancı anahtar.
- **muze_id (FK):** Biletin geçerli olduğu müzeyi işaret eden yabancı anahtar.
- **satın_alim_tarihi:** Biletin satın alındığı tarih.

8. Koleksiyon

- **koleksiyon_id (PK):** Koleksiyonun benzersiz kimlik numarası.

- **ad:** Koleksiyonun adı.
- **muze_id (FK):** Koleksiyonun ait olduğu müze.
- **aciklama:** Koleksiyon hakkında kısa bilgi.

9. **Etkinlik**

- **etkinlik_id (PK):** Etkinliğin benzersiz kimlik numarası.
- **ad:** Etkinliğin adı.
- **tarih:** Etkinliğin tarihi.
- **aciklama:** Etkinliğin açıklaması.
- **muze_id (FK):** Etkinliğin düzenlendiği müze.

10. **Puan ve Yorum**(İlişkisel Tablo)

- **puanlama_id:** Puanlamanın benzersiz kimlik numarası (PK).
- **eser_id:** Puanlanan sanat eserinin kimlik numarası (FK).
- **ziyaretci_id:** Puan veren ziyaretçinin kimlik numarası (FK).
- **puan:** Verilen puan (örneğin, 1-5 arasında bir değer).

İlişkiler

1. **Müze - Sanat Eseri:** Her müze birden fazla sanat eseri sergileyebilir (1-N).
2. **Sanatçı - Sanat Eseri:** Her sanatçı birden fazla eser yaratabilir (1-N).
3. **Sanat Eseri - Eser Türü:** Her sanat eseri bir türe aittir (1-N).
4. **Müze - Koleksiyon:** Her müze birden fazla koleksiyon içerebilir (1-N).
5. **Koleksiyon - Sanat Eseri:** Her koleksiyon birden fazla sanat eseri içerebilir (1-N).
6. **Müze - Etkinlik:** Her müze birden fazla etkinliğe ev sahipliği yapabilir (1-N).
7. **Etkinlik - Sanat Eseri:** Her etkinlik birden fazla sanat eserini sergileyebilir (1-N).
8. **Sanat Eseri - Puanlama:** Her sanat eseri birden fazla puan alabilir (1-N).
9. **Sanat Eseri - Yorum:** Her sanat eseri birden fazla yorum alabilir (1-N).
10. **Müze - Bilet:** Her müze için birden fazla bilet satılabilir (1-N).
11. **Müze - Sponsorluk:** Her müze birden fazla sponsorluk anlaşmasına sahip olabilir (N-M).
12. **Ziyaretçi - Sanat Eseri:** Her ziyaretçi birden fazla sanat eserine puan verebilir ve yorum yapabilir (N-M).

2. Proje Ayrıntıları:

2.1. Proje Amaçları:

Bu projenin temel amacı, sanat eserleri, müzeler ve koleksiyonlar ile ilgili verilerin sistematik bir şekilde yönetilmesini sağlayan bir veritabanı sistemi geliştirmektir. Proje kapsamında hedeflenen amaçlar şunlardır:

1. Sanat Eserlerinin Kapsamlı Yönetimi:

- Sanat eserlerine ait detaylı bilgilerin (örneğin, eser adı, türü, tarih, sanatçı bilgileri) kaydedilmesi ve bu bilgilere hızlı erişim sağlanması.
- Sanat eserlerinin sergilendiği müze bilgileriyle ilişkilendirilmesi.

2. Müzeler Arası Entegrasyon:

- Farklı müzelerde bulunan sanat eserleri arasında bağlantılar kurularak merkezi bir veri yönetim sistemi oluşturulması.
- Müze bilgileri (örneğin, müze adı, adresi, kurulum yılı) ve müzelerde sergilenen koleksiyonların kayıt altına alınması.

3. Sanatçılar ve Koleksiyon Yönetimi:

- Sanatçıların biyografik bilgileri ve eserleri arasındaki bağlantıların düzenlenmesi.
- Belirli bir sanatçıya ait eserlerin müzelerdeki dağılımının izlenebilmesi.

4. Ziyaretçi ve Etkinlik Takibi:

- Müze ziyaretçileri için düzenlenen etkinliklerin kayıt altına alınması.
- Ziyaretçi istatistiklerinin analiz edilerek müze yönetimine bilgi sağlanması.

5. Verimlilik ve Hız:

- Veritabanı sayesinde sanat eserleri ve müzelerle ilgili sorgulamaların daha hızlı ve güvenilir bir şekilde yapılması.
- Manuel veri yönetimi süreçlerinden kaynaklanan hata oranlarının azaltılması.

6. Gelecekteki Geliştirmelere Uygun Altyapı Sağlanması:

- Sistematik bir veri modelinin oluşturularak veritabanının ileride yapılacak entegrasyon veya geliştirme çalışmaları için hazır hale getirilmesi.

2.2. Proje Süreci:

Proje Süreci

1. **Planlama:** Proje kapsamında sanat eserleri, müzeler, sanatçılar ve etkinlikler için bir veritabanı oluşturulması hedeflendi. Kullanıcı gereksinimleri analiz edilerek temel işlevler belirlendi.
2. **Kavramsal Tasarım:** ER diyagramı oluşturularak varlıklar (Sanat_Eseri, Müze, Sanatçı, vb.) ve aralarındaki ilişkiler tasarlandı.
3. **Mantıksal Tasarım:** Veritabanı normalizasyonu yapılarak tablolar (ör. Sanat_Eseri, Müze) ve alanlar belirlendi.
4. **Fiziksel Tasarım:** SQL Server üzerinde tablolar, ilişkiler ve veri kısıtlamaları oluşturuldu.
5. **Uygulama ve Test:** Veriler eklendi, SQL sorguları test edildi ve sistem başarıyla çalıştırıldı.

3. Proje Sonuçları:

1. Veritabanı Sistemi Başarıyla Oluşturuldu:

- Sanat eserleri, müzeler, sanatçılar ve etkinlikler için ilişkisel bir veritabanı tasarlandı ve normalizasyon kurallarına uygun olarak oluşturuldu.
- Tüm tablolar ve ilişkiler başarıyla tanımlandı ve örnek veriler sisteme eklendi.

2. Temel İşlevler Başarıyla Gerçekleştirildi:

- SQL sorguları kullanılarak sanat eserlerinin ve müzelerin bilgilerine erişim sağlandı.
- Ziyaretçi ve etkinlik verilerinin yönetimi için gerekli işlemler başarıyla uygulandı.

3. Veri Doğruluğu ve Güvenilirlik Sağlandı:

- Tetikleyiciler ve kısıtlamalar ile veri bütünlüğü sağlandı.

- Örnek test senaryolarında doğru sonuçlar elde edildi.
- 4. **Kullanıcı Gereksinimlerine Uygunluk:**
 - Proje kullanıcı gereksinimlerine uygun bir şekilde tamamlandı ve sistemin gelecekte genişletilebilir bir yapıya sahip olması sağlandı.

4. SQL Sorguları:

4.1 Create Table:

```
CREATE TABLE Muze (  
    muze_id INT PRIMARY KEY,  
    ad NVARCHAR(255) NOT NULL,  
    adres NVARCHAR(255) NOT NULL,  
    ziyaret_saatleri NVARCHAR(255) NOT NULL,  
    giris_ucreti DECIMAL(10, 2) NOT NULL  
);
```

```
CREATE TABLE EserTuru (  
    tur_id INT PRIMARY KEY,  
    ad NVARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE Sanatci (  
    sanatci_id INT PRIMARY KEY,  
    ad NVARCHAR(255) NOT NULL,  
    dogum_tarihi DATE NOT NULL,  
    biyografi NVARCHAR(MAX) NOT NULL  
);
```

```
CREATE TABLE Koleksiyon (  
    koleksiyon_id INT PRIMARY KEY,  
    ad NVARCHAR(255) NOT NULL,  
    muze_id INT NOT NULL,  
    aciklama NVARCHAR(MAX) NOT NULL,  
    FOREIGN KEY (muze_id) REFERENCES Muze(muze_id)  
);
```

```
CREATE TABLE SanatEseri (  
    eser_id INT PRIMARY KEY,  
    ad NVARCHAR(255) NOT NULL,  
    tur_id INT NOT NULL,  
    yaratilis_tarihi DATE NOT NULL,  
    sanatci_id INT NOT NULL,  
    muze_id INT NOT NULL,  
    koleksiyon_id INT NULL,  
    FOREIGN KEY (tur_id) REFERENCES EserTuru(tur_id),  
    FOREIGN KEY (sanatci_id) REFERENCES Sanatci(sanatci_id),  
    FOREIGN KEY (muze_id) REFERENCES Muze(muze_id),  
    FOREIGN KEY (koleksiyon_id) REFERENCES Koleksiyon(koleksiyon_id)  
);
```

```

CREATE TABLE Ziyaretci (
    ziyaretci_id INT PRIMARY KEY,
    isim NVARCHAR(255) NOT NULL,
    email NVARCHAR(255) UNIQUE NOT NULL,
    parola NVARCHAR(255) NOT NULL
);

CREATE TABLE MuzeCalisani (
    calisan_id INT PRIMARY KEY,
    ad NVARCHAR(255) NOT NULL,
    gorev NVARCHAR(255) NOT NULL,
    muze_id INT NOT NULL,
    FOREIGN KEY (muze_id) REFERENCES Muze(muze_id)
);

CREATE TABLE Bilet (
    bilet_id INT PRIMARY KEY,
    ziyaretci_id INT NOT NULL,
    muze_id INT NOT NULL,
    satin_alim_tarihi DATE NOT NULL,
    FOREIGN KEY (ziyaretci_id) REFERENCES Ziyaretci(ziyaretci_id),
    FOREIGN KEY (muze_id) REFERENCES Muze(muze_id)
);

CREATE TABLE Etkinlik (
    etkinlik_id INT PRIMARY KEY,
    ad NVARCHAR(255) NOT NULL,
    tarih DATE NOT NULL,
    aciklama NVARCHAR(MAX) NOT NULL,
    muze_id INT NOT NULL,
    FOREIGN KEY (muze_id) REFERENCES Muze(muze_id)
);

CREATE TABLE PuanYorum (
    puanlama_id INT PRIMARY KEY,
    eser_id INT NOT NULL,
    ziyaretci_id INT NOT NULL,
    puan INT CHECK (puan BETWEEN 1 AND 5),
    FOREIGN KEY (eser_id) REFERENCES SanatEseri(eser_id),
    FOREIGN KEY (ziyaretci_id) REFERENCES Ziyaretci(ziyaretci_id)
);

```

4.2 Insert Into:

```

INSERT INTO Muze (muze_id, ad, adres, ziyaret_saatleri, giris_ucreti)
VALUES
(1, 'Modern Sanatlar Müzesi', 'İstanbul, Türkiye', '09:00 - 18:00', 50.00),
(2, 'Klasik Sanatlar Müzesi', 'Ankara, Türkiye', '10:00 - 17:00', 40.00);

INSERT INTO EserTuru (tur_id, ad)
VALUES
(1, 'Resim'),
(2, 'Heykel'),

```

```
(3, 'Fotoğraf'),
(4, 'Dijital Sanat');
```

```
INSERT INTO Sanatci (sanatci_id, ad, dogum_tarihi, biyografi)
VALUES
(1, 'Leonardo da Vinci', '1452-04-15', 'Rönesans dönemi ünlü İtalyan sanatçı.'),
(2, 'Pablo Picasso', '1881-10-25', '20. yüzyılın en etkili sanatçılarından.'),
(3, 'Ara Güler', '1928-08-16', 'Türk fotoğraf sanatçısı.');
```

```
INSERT INTO Koleksiyon (koleksiyon_id, ad, muze_id, aciklama)
VALUES
(1, 'Rönesans Koleksiyonu', 1, 'Rönesans dönemine ait önemli eserler.'),
(2, 'Modern Fotoğrafçılık', 2, '20. yüzyıl fotoğraf sanatçılarına ait eserler.');
```

```
INSERT INTO SanatEseri (eser_id, ad, tur_id, yaratilis_tarihi, sanatci_id, muze_id,
koleksiyon_id)
VALUES
(1, 'Mona Lisa', 1, '1503-01-01', 1, 1, 1),
(2, 'Guernica', 1, '1937-01-01', 2, 2, NULL),
(3, 'İstanbul Sokakları', 3, '1950-01-01', 3, 2, 2);
```

```
INSERT INTO Ziyaretci (ziyaretci_id, isim, email, parola)
VALUES
(1, 'Ahmet Yılmaz', 'ahmet.yilmaz@example.com', 'hashed_password1'),
(2, 'Ayşe Kara', 'ayse.kara@example.com', 'hashed_password2'),
(3, 'Mehmet Öz', 'mehmet.oz@example.com', 'hashed_password3');
```

```
INSERT INTO MuzeCalisani (calisan_id, ad, gorev, muze_id)
VALUES
(1, 'Fatma Çalışkan', 'Rehber', 1),
(2, 'Ali Duran', 'Sanat Tarihçisi', 2);
```

```
INSERT INTO Bilet (bilet_id, ziyaretci_id, muze_id, satin_alim_tarihi)
VALUES
(1, 1, 1, '2023-12-01'),
(2, 2, 2, '2023-12-02'),
(3, 3, 1, '2023-12-03');
```

```
INSERT INTO Etkinlik (etkinlik_id, ad, tarih, aciklama, muze_id)
VALUES
(1, 'Rönesans Sanatı Paneli', '2024-01-15', 'Sanat tarihçileri ile özel bir panel.',
1),
(2, 'Fotoğrafçılık Atölyesi', '2024-02-20', 'Ara Güler’in eserlerinden ilham alınarak
bir atölye.', 2);
```

```
INSERT INTO PuanYorum (puanlama_id, eser_id, ziyaretci_id, puan)
VALUES
(1, 1, 1, 5),
(2, 2, 2, 4),
(3, 3, 3, 5);
```

4.3 Trigger

```
CREATE TRIGGER TekrarPuanEngelle
ON PuanYorum
AFTER INSERT
AS
BEGIN
    DELETE FROM PuanYorum
    WHERE puanlama_id IN (
        SELECT i.puanlama_id
        FROM inserted AS i
        JOIN PuanYorum AS p
        ON i.eser_id = p.eser_id AND i.ziyaretci_id = p.ziyaretci_id
        WHERE i.puanlama_id > p.puanlama_id
    );
END;
```

```
CREATE TRIGGER PuanKontrol
ON PuanYorum
AFTER INSERT, UPDATE
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM inserted
        WHERE puan < 1 OR puan > 5
    )
    BEGIN
        RAISERROR ('Puan 1 ile 5 arasında olmalıdır.', 16, 1);
        ROLLBACK TRANSACTION;
    END
END;
```

```
CREATE TRIGGER KoleksiyonTekrarlamaEngel
ON Koleksiyon
AFTER INSERT
AS
BEGIN
    IF EXISTS (
        SELECT 1
        FROM Koleksiyon k
        JOIN inserted i ON k.ad = i.ad
    )
    BEGIN
        ROLLBACK TRANSACTION;
    END
END;
```

4.4 Transaction:

```
BEGIN TRANSACTION;

BEGIN TRY
    INSERT INTO Muze (muze_id, ad, adres, ziyaret_saatleri, giris_ucreti)
```



```

VALUES (3, 'Antik Sanatlar Müzesi', 'Antalya, Türkiye', '10:00 - 18:00', 60.00);

INSERT INTO SanatEseri (eser_id, ad, tur_id, yaratilis_tarihi, sanatci_id,
muze_id, koleksiyon_id)
VALUES (4, 'Venüs de Milo', 1, '100-01-01', 1, 3, NULL);

COMMIT TRANSACTION;
PRINT 'İşlemler başarıyla tamamlandı.';
END TRY

BEGIN CATCH

    ROLLBACK TRANSACTION;
    PRINT 'Bir hata oluştu, işlemler geri alındı.';
    PRINT ERROR_MESSAGE();
END CATCH;

```

4.5 Stored Procedure:

```

CREATE PROCEDURE GetSanatEseriSayisi
    @MuzeID INT,
    @SanatEseriSayisi INT OUTPUT
AS
BEGIN
    SELECT @SanatEseriSayisi = COUNT(*)
    FROM Sanat_Eseri
    WHERE Muze_ID = @MuzeID;
END;

```

6 ER Diyagramları:



